

CS162 - Algorithm Visualizer

Generated by Doxygen 1.9.6



---

<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List . . . . .	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 File Index</b>	<b>7</b>
4.1 File List . . . . .	7
<b>5 Namespace Documentation</b>	<b>11</b>
5.1 Algorithm Namespace Reference . . . . .	11
5.2 Animation Namespace Reference . . . . .	11
5.3 AnimationFactory Namespace Reference . . . . .	11
5.3.1 Function Documentation . . . . .	12
5.3.1.1 BlendColor() . . . . .	12
5.3.1.2 BounceOut() . . . . .	12
5.3.1.3 Dist() . . . . .	12
5.3.1.4 DrawActiveArrow() . . . . .	12
5.3.1.5 DrawCircularArrow() . . . . .	12
5.3.1.6 DrawDirectionalArrow() . . . . .	13
5.3.1.7 DrawDoubleActiveArrow() . . . . .	13
5.3.1.8 DrawDoubleDirectionalArrow() . . . . .	13
5.3.1.9 ElasticOut() . . . . .	13
5.3.1.10 InverseVector() . . . . .	13
5.3.1.11 MoveNode() . . . . .	14
5.3.1.12 ReCalculateEnds() . . . . .	14
5.4 Category Namespace Reference . . . . .	14
5.4.1 Enumeration Type Documentation . . . . .	14
5.4.1.1 ID . . . . .	14
5.5 ColorTheme Namespace Reference . . . . .	14
5.5.1 Enumeration Type Documentation . . . . .	15
5.5.1.1 ID . . . . .	15
5.6 Core Namespace Reference . . . . .	16
5.7 DataStructures Namespace Reference . . . . .	17
5.7.1 Enumeration Type Documentation . . . . .	17
5.7.1.1 ID . . . . .	17
5.8 Fonts Namespace Reference . . . . .	17
5.8.1 Enumeration Type Documentation . . . . .	17
5.8.1.1 ID . . . . .	17
5.9 global Namespace Reference . . . . .	18
5.9.1 Variable Documentation . . . . .	18

5.9.1.1 defaultColorPath . . . . .	18
5.9.1.2 favicon . . . . .	18
5.9.1.3 kTitle . . . . .	18
5.9.1.4 SCREEN_HEIGHT . . . . .	19
5.9.1.5 SCREEN_WIDTH . . . . .	19
5.10 GUI Namespace Reference . . . . .	19
5.11 States Namespace Reference . . . . .	19
5.11.1 Enumeration Type Documentation . . . . .	19
5.11.1.1 ID . . . . .	19
5.12 Textures Namespace Reference . . . . .	20
5.12.1 Enumeration Type Documentation . . . . .	20
5.12.1.1 ID . . . . .	20
5.13 Utils Namespace Reference . . . . .	21
5.13.1 Function Documentation . . . . .	21
5.13.1.1 DrawIcon() . . . . .	21
5.13.1.2 DrawTextBoxed() . . . . .	21
5.13.1.3 DrawTextBoxedSelectable() . . . . .	22
5.13.1.4 OpenFileDialog() . . . . .	22
5.13.1.5 Rand() . . . . .	22
5.13.1.6 ReadInputFromFile() . . . . .	22
<b>6 Class Documentation</b> . . . . .	<b>23</b>
6.1 Algorithm::Algorithm< GUIAlgorithm, AnimationState > Class Template Reference . . . . .	23
6.1.1 Constructor & Destructor Documentation . . . . .	24
6.1.1.1 Algorithm() [1/2] . . . . .	24
6.1.1.2 Algorithm() [2/2] . . . . .	24
6.1.1.3 ~Algorithm() . . . . .	24
6.1.2 Member Function Documentation . . . . .	24
6.1.2.1 ApplyInput() . . . . .	24
6.1.2.2 Empty() . . . . .	24
6.1.2.3 EmptyGenerator() . . . . .	25
6.1.2.4 GenerateAnimation() . . . . .	25
6.1.2.5 GenerateRelayoutAnimation() . . . . .	25
6.1.2.6 InitAction() . . . . .	25
6.1.2.7 Random() . . . . .	25
6.1.2.8 RandomFixedSize() . . . . .	25
6.1.2.9 RandomFixedSizeGenerator() . . . . .	26
6.1.2.10 RandomGenerator() . . . . .	26
6.1.2.11 ReadFromExternalFile() . . . . .	26
6.1.2.12 ReadFromFileGenerator() . . . . .	26
6.1.2.13 UserDefined() . . . . .	26
6.1.2.14 UserDefinedGenerator() . . . . .	26

---

6.1.3 Member Data Documentation . . . . .	27
6.1.3.1 animController . . . . .	27
6.1.3.2 codeHighlighter . . . . .	27
6.1.3.3 visualizer . . . . .	27
6.2 Animation::AnimationController< T > Class Template Reference . . . . .	27
6.2.1 Member Typedef Documentation . . . . .	28
6.2.1.1 Ptr . . . . .	29
6.2.2 Constructor & Destructor Documentation . . . . .	29
6.2.2.1 AnimationController() . . . . .	29
6.2.2.2 ~AnimationController() . . . . .	29
6.2.3 Member Function Documentation . . . . .	29
6.2.3.1 AddAnimation() . . . . .	29
6.2.3.2 Clear() . . . . .	29
6.2.3.3 Continue() . . . . .	29
6.2.3.4 CurrentAnimationIndex() . . . . .	30
6.2.3.5 Done() . . . . .	30
6.2.3.6 GetAnimateFrame() . . . . .	30
6.2.3.7 GetAnimation() . . . . .	30
6.2.3.8 GetAnimationDuration() . . . . .	30
6.2.3.9 GetAnimationIndex() . . . . .	30
6.2.3.10 GetNumAnimation() . . . . .	30
6.2.3.11 GetSpeed() . . . . .	31
6.2.3.12 GetStopDuration() . . . . .	31
6.2.3.13 InteractionAllow() . . . . .	31
6.2.3.14 InteractionLock() . . . . .	31
6.2.3.15 IsInteractionAllow() . . . . .	31
6.2.3.16 IsPlaying() . . . . .	31
6.2.3.17 Pause() . . . . .	31
6.2.3.18 PopAnimation() . . . . .	32
6.2.3.19 Reset() . . . . .	32
6.2.3.20 ResetCurrent() . . . . .	32
6.2.3.21 RunAll() . . . . .	32
6.2.3.22 SetAnimation() . . . . .	32
6.2.3.23 SetSpeed() . . . . .	32
6.2.3.24 StepBackward() . . . . .	32
6.2.3.25 StepForward() . . . . .	33
6.2.3.26 Update() . . . . .	33
6.2.4 Member Data Documentation . . . . .	33
6.2.4.1 animationGroup . . . . .	33
6.2.4.2 currStopDuration . . . . .	33
6.2.4.3 defaultSpeed . . . . .	33
6.2.4.4 interactionLock . . . . .	33

6.2.4.5 mCurrentAnimationIndex . . . . .	34
6.2.4.6 mSpeed . . . . .	34
6.2.4.7 Playing . . . . .	34
6.2.4.8 stopDuration . . . . .	34
6.3 Animation::AnimationState< T > Class Template Reference . . . . .	34
6.3.1 Member Typedef Documentation . . . . .	35
6.3.1.1 Ptr . . . . .	35
6.3.2 Constructor & Destructor Documentation . . . . .	35
6.3.2.1 AnimationState() . . . . .	35
6.3.2.2 ~AnimationState() . . . . .	36
6.3.3 Member Function Documentation . . . . .	36
6.3.3.1 Done() . . . . .	36
6.3.3.2 Draw() . . . . .	36
6.3.3.3 GetActionDescription() . . . . .	36
6.3.3.4 GetCurrentPlayingAt() . . . . .	36
6.3.3.5 GetDataStructure() . . . . .	36
6.3.3.6 GetDuration() . . . . .	37
6.3.3.7 GetHighlightedLine() . . . . .	37
6.3.3.8 PlayingAt() . . . . .	37
6.3.3.9 Reset() . . . . .	37
6.3.3.10 SetActionDescription() . . . . .	37
6.3.3.11 SetAnimation() . . . . .	37
6.3.3.12 SetDuration() . . . . .	38
6.3.3.13 SetHighlightLine() . . . . .	38
6.3.3.14 SetSourceDataStructure() . . . . .	38
6.3.3.15 Update() . . . . .	38
6.3.4 Member Data Documentation . . . . .	38
6.3.4.1 actionDescription . . . . .	38
6.3.4.2 mAnimation . . . . .	38
6.3.4.3 mCurrentPlayingAt . . . . .	39
6.3.4.4 mDataStructureBefore . . . . .	39
6.3.4.5 mDuration . . . . .	39
6.3.4.6 mHighlightedLine . . . . .	39
6.4 Application Class Reference . . . . .	39
6.4.1 Constructor & Destructor Documentation . . . . .	40
6.4.1.1 Application() . . . . .	40
6.4.1.2 ~Application() . . . . .	40
6.4.2 Member Function Documentation . . . . .	40
6.4.2.1 Close() . . . . .	40
6.4.2.2 Init() . . . . .	40
6.4.2.3 LoadResources() . . . . .	41
6.4.2.4 RegisterStates() . . . . .	41

---

6.4.2.5 Render()	41
6.4.2.6 Run()	41
6.4.2.7 Update()	41
6.4.2.8 WindowClosed()	41
6.4.3 Member Data Documentation	41
6.4.3.1 categories	41
6.4.3.2 closed	42
6.4.3.3 dsInfo	42
6.4.3.4 fonts	42
6.4.3.5 images	42
6.4.3.6 mStack	42
6.5 ArrayState< T > Class Template Reference	43
6.5.1 Member Typedef Documentation	45
6.5.1.1 Ptr	45
6.5.2 Constructor & Destructor Documentation	45
6.5.2.1 ArrayState()	45
6.5.2.2 ~ArrayState()	45
6.5.3 Member Function Documentation	45
6.5.3.1 AddAccessOperation()	45
6.5.3.2 AddDeleteOperation()	46
6.5.3.3 AddInitializeOperation()	46
6.5.3.4 AddInsertOperation()	46
6.5.3.5 AddIntFieldOperationOption()	46
6.5.3.6 AddNoFieldOperationOption()	46
6.5.3.7 AddOperations()	47
6.5.3.8 AddSearchOperation()	47
6.5.3.9 AddStringFieldOption()	47
6.5.3.10 AddUpdateOperation()	47
6.5.3.11 ClearAction()	47
6.5.3.12 ClearError()	47
6.5.3.13 Draw()	48
6.5.3.14 DrawCurrentActionText()	48
6.5.3.15 DrawCurrentErrorText()	48
6.5.3.16 GetContext()	48
6.5.3.17 InitNavigationBar()	48
6.5.3.18 RequestStackClear()	48
6.5.3.19 RequestStackPop()	48
6.5.3.20 RequestStackPush()	49
6.5.3.21 SetCurrentAction()	49
6.5.3.22 SetCurrentError()	49
6.5.3.23 Success()	49
6.5.3.24 Update()	49

6.5.4 Member Data Documentation . . . . .	49
6.5.4.1 activeDS . . . . .	49
6.5.4.2 animController . . . . .	50
6.5.4.3 codeHighlighter . . . . .	50
6.5.4.4 footer . . . . .	50
6.5.4.5 mContext . . . . .	50
6.5.4.6 mCurrentAction . . . . .	50
6.5.4.7 mCurrentError . . . . .	50
6.5.4.8 mStack . . . . .	50
6.5.4.9 navigation . . . . .	51
6.5.4.10 operationList . . . . .	51
6.6 GUI::Button Class Reference . . . . .	51
6.6.1 Member Typedef Documentation . . . . .	54
6.6.1.1 Ptr . . . . .	54
6.6.2 Member Enumeration Documentation . . . . .	54
6.6.2.1 TextAlign . . . . .	54
6.6.3 Constructor & Destructor Documentation . . . . .	54
6.6.3.1 Button() [1/2] . . . . .	55
6.6.3.2 Button() [2/2] . . . . .	55
6.6.3.3 ~Button() . . . . .	55
6.6.4 Member Function Documentation . . . . .	55
6.6.4.1 deselect() . . . . .	55
6.6.4.2 DisableFitContent() . . . . .	55
6.6.4.3 Draw() . . . . .	55
6.6.4.4 DrawButtonText() . . . . .	56
6.6.4.5 EnableFitContent() . . . . .	56
6.6.4.6 FitContent() . . . . .	56
6.6.4.7 GetContentPos() . . . . .	56
6.6.4.8 GetContentSize() . . . . .	56
6.6.4.9 GetFontSize() . . . . .	56
6.6.4.10 GetHoverStatus() [1/2] . . . . .	56
6.6.4.11 GetHoverStatus() [2/2] . . . . .	57
6.6.4.12 GetPosition() . . . . .	57
6.6.4.13 GetSize() . . . . .	57
6.6.4.14 GetVisible() . . . . .	57
6.6.4.15 IsClicked() . . . . .	57
6.6.4.16 IsSelectable() . . . . .	57
6.6.4.17 isSelected() . . . . .	57
6.6.4.18 select() . . . . .	58
6.6.4.19 SetAction() . . . . .	58
6.6.4.20 SetActionOnHover() . . . . .	58
6.6.4.21 SetButtonColor() . . . . .	58

---

6.6.4.22 SetButtonHoverColor() . . . . .	58
6.6.4.23 SetFontSize() . . . . .	58
6.6.4.24 SetPosition() [1/2] . . . . .	58
6.6.4.25 SetPosition() [2/2] . . . . .	59
6.6.4.26 SetSize() . . . . .	59
6.6.4.27SetText() . . . . .	59
6.6.4.28 SetTextAlignment() . . . . .	59
6.6.4.29 SetTextColor() . . . . .	59
6.6.4.30 SetVisible() . . . . .	59
6.6.4.31 ToggleVisible() . . . . .	60
6.6.4.32 UpdateMouseCursorWhenHover() [1/2] . . . . .	60
6.6.4.33 UpdateMouseCursorWhenHover() [2/2] . . . . .	60
6.6.5 Member Data Documentation . . . . .	60
6.6.5.1 action . . . . .	60
6.6.5.2 alignment . . . . .	60
6.6.5.3 bound . . . . .	60
6.6.5.4 buttonColor . . . . .	61
6.6.5.5 buttonHoverColor . . . . .	61
6.6.5.6 content . . . . .	61
6.6.5.7 fitContent . . . . .	61
6.6.5.8 fonts . . . . .	61
6.6.5.9 fontSize . . . . .	61
6.6.5.10 isHover . . . . .	61
6.6.5.11 mActionOnHover . . . . .	61
6.6.5.12 mIsSelected . . . . .	62
6.6.5.13 mPosition . . . . .	62
6.6.5.14 mVisible . . . . .	62
6.6.5.15 textColor . . . . .	62
6.7 GUI::Card Class Reference . . . . .	62
6.7.1 Member Typedef Documentation . . . . .	64
6.7.1.1 Ptr . . . . .	65
6.7.2 Constructor & Destructor Documentation . . . . .	65
6.7.2.1 Card() [1/2] . . . . .	65
6.7.2.2 Card() [2/2] . . . . .	65
6.7.2.3 ~Card() . . . . .	65
6.7.3 Member Function Documentation . . . . .	65
6.7.3.1 deselect() . . . . .	65
6.7.3.2 Draw() . . . . .	65
6.7.3.3 DrawImage() . . . . .	66
6.7.3.4 DrawTitle() . . . . .	66
6.7.3.5 GetHoverStatus() [1/2] . . . . .	66
6.7.3.6 GetHoverStatus() [2/2] . . . . .	66

6.7.3.7 GetPosition()	66
6.7.3.8 GetSize()	66
6.7.3.9 GetVisible()	67
6.7.3.10 IsSelectable()	67
6.7.3.11 IsSelected()	67
6.7.3.12 select()	67
6.7.3.13 SetLink()	67
6.7.3.14 SetPosition() [1/2]	67
6.7.3.15 SetPosition() [2/2]	67
6.7.3.16 SetStateID()	68
6.7.3.17 SetText()	68
6.7.3.18 SetVisible()	68
6.7.3.19 ToggleVisible()	68
6.7.3.20 UpdateMouseCursorWhenHover() [1/2]	68
6.7.3.21 UpdateMouseCursorWhenHover() [2/2]	68
6.7.4 Member Data Documentation	68
6.7.4.1 fonts	69
6.7.4.2 hoverBounds	69
6.7.4.3 isHover	69
6.7.4.4 mIsSelected	69
6.7.4.5 mPosition	69
6.7.4.6 mVisible	69
6.7.4.7 stateID	69
6.7.4.8 thumbnail	69
6.7.4.9 title	70
6.7.4.10 toLink	70
6.8 CategoryInfo Class Reference	70
6.8.1 Member Function Documentation	71
6.8.1.1 Get()	71
6.8.1.2 Register()	71
6.8.2 Member Data Documentation	71
6.8.2.1 mFactories	71
6.9 Algorithm::CircularLinkedList Class Reference	72
6.9.1 Constructor & Destructor Documentation	74
6.9.1.1 CircularLinkedList() [1/2]	74
6.9.1.2 CircularLinkedList() [2/2]	74
6.9.1.3 ~CircularLinkedList()	74
6.9.2 Member Function Documentation	74
6.9.2.1 ApplyInput()	74
6.9.2.2 DeleteHead()	74
6.9.2.3 DeleteMiddle()	74
6.9.2.4 DeleteTail()	75

---

6.9.2.5 Empty()	75
6.9.2.6 EmptyGenerator()	75
6.9.2.7 GenerateAnimation()	75
6.9.2.8 GenerateRelayoutAnimation()	75
6.9.2.9 HighlightArrowFromCur()	75
6.9.2.10 HighlightCircularArrow()	76
6.9.2.11 InitAction()	76
6.9.2.12 InsertAfterTail()	76
6.9.2.13 InsertHead()	76
6.9.2.14 InsertMiddle()	76
6.9.2.15 Random()	76
6.9.2.16 RandomFixedSize()	76
6.9.2.17 RandomFixedSizeGenerator()	77
6.9.2.18 RandomGenerator()	77
6.9.2.19 ReadFromExternalFile()	77
6.9.2.20 ReadFromFileGenerator()	77
6.9.2.21 ResetVisualizer()	77
6.9.2.22 Search()	77
6.9.2.23 size()	77
6.9.2.24 Update()	78
6.9.2.25 UserDefined()	78
6.9.2.26 UserDefinedGenerator()	78
6.9.3 Member Data Documentation	78
6.9.3.1 animController	78
6.9.3.2 codeHighlighter	78
6.9.3.3 maxN	78
6.9.3.4 visualizer	79
6.10 GUI::CircularLinkedList Class Reference	79
6.10.1 Member Typedef Documentation	82
6.10.1.1 Ptr	82
6.10.2 Member Enumeration Documentation	82
6.10.2.1 ArrowType	82
6.10.2.2 Orientation	83
6.10.3 Constructor & Destructor Documentation	83
6.10.3.1 CircularLinkedList() [1/2]	83
6.10.3.2 CircularLinkedList() [2/2]	83
6.10.3.3 ~CircularLinkedList()	83
6.10.4 Member Function Documentation	83
6.10.4.1 DeleteNode()	83
6.10.4.2 deselect()	84
6.10.4.3 Draw() [1/2]	84
6.10.4.4 Draw() [2/2]	84

---

6.10.4.5 DrawArrow()	84
6.10.4.6 DrawCurrent()	84
6.10.4.7 GenerateNode()	84
6.10.4.8 GetArrowType()	85
6.10.4.9 GetChildren()	85
6.10.4.10 GetCircularArrowType()	85
6.10.4.11 GetCircularEnds()	85
6.10.4.12 GetHoverStatus() [1/2]	85
6.10.4.13 GetHoverStatus() [2/2]	85
6.10.4.14 GetList()	85
6.10.4.15 GetNodeDefaultPosition()	86
6.10.4.16 GetPosition()	86
6.10.4.17 GetShape()	86
6.10.4.18 GetSize()	86
6.10.4.19 GetVisible()	86
6.10.4.20 Import()	86
6.10.4.21 InsertNode()	87
6.10.4.22 isSelectable()	87
6.10.4.23 isSelected()	87
6.10.4.24 pack()	87
6.10.4.25 Relayout()	87
6.10.4.26 ResetArrow()	87
6.10.4.27 select()	87
6.10.4.28 SetArrowType()	88
6.10.4.29 SetCircularArrowType()	88
6.10.4.30 SetCircularEnds()	88
6.10.4.31 SetOrientation()	88
6.10.4.32 SetPosition() [1/2]	88
6.10.4.33 SetPosition() [2/2]	88
6.10.4.34 SetShape()	89
6.10.4.35 SetShowHeadAndTail()	89
6.10.4.36 SetVisible()	89
6.10.4.37 size()	89
6.10.4.38 ToggleVisible()	89
6.10.4.39 UnpackAll()	89
6.10.4.40 UpdateMouseCursorWhenHover() [1/2]	89
6.10.4.41 UpdateMouseCursorWhenHover() [2/2]	90
6.10.5 Member Data Documentation	90
6.10.5.1 arrowState	90
6.10.5.2 circularArrowState	90
6.10.5.3 fonts	90
6.10.5.4 list	90

---

6.10.5.5 mChildren . . . . .	90
6.10.5.6 mCircularEnds . . . . .	90
6.10.5.7 mDisplayHeadAndTail . . . . .	91
6.10.5.8 mIsSelected . . . . .	91
6.10.5.9 mNodeDistance . . . . .	91
6.10.5.10 mOrientation . . . . .	91
6.10.5.11 mPosition . . . . .	91
6.10.5.12 mShape . . . . .	91
6.10.5.13 mVisible . . . . .	91
6.11 CLLState Class Reference . . . . .	92
6.11.1 Member Typedef Documentation . . . . .	94
6.11.1.1 Ptr . . . . .	94
6.11.2 Constructor & Destructor Documentation . . . . .	94
6.11.2.1 CLLState() . . . . .	94
6.11.2.2 ~CLLState() . . . . .	94
6.11.3 Member Function Documentation . . . . .	94
6.11.3.1 AddDeleteOperation() . . . . .	94
6.11.3.2 AddInitializeOperation() . . . . .	94
6.11.3.3 AddInsertOperation() . . . . .	95
6.11.3.4 AddIntFieldOperationOption() . . . . .	95
6.11.3.5 AddNoFieldOperationOption() . . . . .	95
6.11.3.6 AddOperations() . . . . .	95
6.11.3.7 AddSearchOperation() . . . . .	95
6.11.3.8 AddStringFieldOption() . . . . .	95
6.11.3.9 AddUpdateOperation() . . . . .	96
6.11.3.10 ClearAction() . . . . .	96
6.11.3.11 ClearError() . . . . .	96
6.11.3.12 Draw() . . . . .	96
6.11.3.13 DrawCurrentActionText() . . . . .	96
6.11.3.14 DrawCurrentErrorText() . . . . .	96
6.11.3.15 GetContext() . . . . .	96
6.11.3.16 InitNavigationBar() . . . . .	97
6.11.3.17 RequestStackClear() . . . . .	97
6.11.3.18 RequestStackPop() . . . . .	97
6.11.3.19 RequestStackPush() . . . . .	97
6.11.3.20 SetCurrentAction() . . . . .	97
6.11.3.21 SetCurrentError() . . . . .	97
6.11.3.22 Success() . . . . .	97
6.11.3.23 Update() . . . . .	98
6.11.4 Member Data Documentation . . . . .	98
6.11.4.1 activeDS . . . . .	98
6.11.4.2 animController . . . . .	98

6.11.4.3 CLL	98
6.11.4.4 codeHighlighter	98
6.11.4.5 footer	98
6.11.4.6 mContext	99
6.11.4.7 mCurrentAction	99
6.11.4.8 mCurrentError	99
6.11.4.9 mStack	99
6.11.4.10 navigation	99
6.11.4.11 operationList	99
6.12 GUI::CodeHighlighter Class Reference	100
6.12.1 Member Typedef Documentation	102
6.12.1.1 Ptr	102
6.12.2 Constructor & Destructor Documentation	102
6.12.2.1 CodeHighlighter()	102
6.12.2.2 ~CodeHighlighter()	102
6.12.3 Member Function Documentation	102
6.12.3.1 AddActionDescription()	102
6.12.3.2 AddCode()	102
6.12.3.3 deselect()	103
6.12.3.4 Draw()	103
6.12.3.5 DrawActionDescription()	103
6.12.3.6 DrawCodeHighlighter()	103
6.12.3.7 GetHoverStatus() [1/2]	103
6.12.3.8 GetHoverStatus() [2/2]	103
6.12.3.9 GetPosition()	104
6.12.3.10 GetSize()	104
6.12.3.11 GetVisible()	104
6.12.3.12 Highlight()	104
6.12.3.13 InitButtons()	104
6.12.3.14 isSelectable()	104
6.12.3.15 isSelected()	104
6.12.3.16 select()	105
6.12.3.17 SetPosition() [1/2]	105
6.12.3.18 SetPosition() [2/2]	105
6.12.3.19 SetShowAction()	105
6.12.3.20 SetShowCode()	105
6.12.3.21 SetVisible()	105
6.12.3.22 ToggleShowAction()	105
6.12.3.23 ToggleShowCode()	106
6.12.3.24 ToggleVisible()	106
6.12.3.25 UpdateMouseCursorWhenHover() [1/2]	106
6.12.3.26 UpdateMouseCursorWhenHover() [2/2]	106

---

6.12.4 Member Data Documentation . . . . .	106
6.12.4.1 fonts . . . . .	106
6.12.4.2 mActionDescription . . . . .	106
6.12.4.3 mButtonShowAction . . . . .	107
6.12.4.4 mButtonShowCode . . . . .	107
6.12.4.5 mCode . . . . .	107
6.12.4.6 mHighlightedLine . . . . .	107
6.12.4.7 mIsSelected . . . . .	107
6.12.4.8 mPosition . . . . .	107
6.12.4.9 mShowActionDescription . . . . .	107
6.12.4.10 mShowCode . . . . .	107
6.12.4.11 mVisible . . . . .	108
6.13 GUI::Component Class Reference . . . . .	108
6.13.1 Member Typedef Documentation . . . . .	109
6.13.1.1 Ptr . . . . .	109
6.13.2 Constructor & Destructor Documentation . . . . .	109
6.13.2.1 Component() . . . . .	110
6.13.2.2 ~Component() . . . . .	110
6.13.3 Member Function Documentation . . . . .	110
6.13.3.1 deselect() . . . . .	110
6.13.3.2 Draw() . . . . .	110
6.13.3.3 GetHoverStatus() [1/2] . . . . .	110
6.13.3.4 GetHoverStatus() [2/2] . . . . .	110
6.13.3.5 GetPosition() . . . . .	111
6.13.3.6 GetSize() . . . . .	111
6.13.3.7 GetVisible() . . . . .	111
6.13.3.8 isSelectable() . . . . .	111
6.13.3.9 isSelected() . . . . .	111
6.13.3.10 select() . . . . .	111
6.13.3.11 SetPosition() [1/2] . . . . .	111
6.13.3.12 SetPosition() [2/2] . . . . .	112
6.13.3.13 SetVisible() . . . . .	112
6.13.3.14 ToggleVisible() . . . . .	112
6.13.3.15 UpdateMouseCursorWhenHover() [1/2] . . . . .	112
6.13.3.16 UpdateMouseCursorWhenHover() [2/2] . . . . .	112
6.13.4 Member Data Documentation . . . . .	112
6.13.4.1 mIsSelected . . . . .	112
6.13.4.2 mPosition . . . . .	113
6.13.4.3 mVisible . . . . .	113
6.14 Core::List< T >::const_iterator Class Reference . . . . .	113
6.14.1 Member Typedef Documentation . . . . .	114
6.14.1.1 difference_type . . . . .	114

6.14.1.2 iterator_category . . . . .	114
6.14.1.3 pointer . . . . .	114
6.14.1.4 reference . . . . .	114
6.14.1.5 value_type . . . . .	114
6.14.2 Constructor & Destructor Documentation . . . . .	115
6.14.2.1 const_iterator() [1/4] . . . . .	115
6.14.2.2 const_iterator() [2/4] . . . . .	115
6.14.2.3 const_iterator() [3/4] . . . . .	115
6.14.2.4 const_iterator() [4/4] . . . . .	115
6.14.3 Member Function Documentation . . . . .	115
6.14.3.1 operator"!="() . . . . .	115
6.14.3.2 operator*() . . . . .	116
6.14.3.3 operator+() . . . . .	116
6.14.3.4 operator++() [1/2] . . . . .	116
6.14.3.5 operator++() [2/2] . . . . .	116
6.14.3.6 operator-() . . . . .	116
6.14.3.7 operator--() [1/2] . . . . .	116
6.14.3.8 operator--() [2/2] . . . . .	117
6.14.3.9 operator->() . . . . .	117
6.14.3.10 operator=() [1/2] . . . . .	117
6.14.3.11 operator=() [2/2] . . . . .	117
6.14.3.12 operator==() . . . . .	117
6.14.3.13 swap() . . . . .	117
6.14.4 Friends And Related Function Documentation . . . . .	117
6.14.4.1 List . . . . .	118
6.14.5 Member Data Documentation . . . . .	118
6.14.5.1 ptr . . . . .	118
6.15 GUI::Container Class Reference . . . . .	118
6.15.1 Member Typedef Documentation . . . . .	119
6.15.1.1 Ptr . . . . .	119
6.15.2 Constructor & Destructor Documentation . . . . .	120
6.15.2.1 Container() . . . . .	120
6.15.3 Member Function Documentation . . . . .	120
6.15.3.1 deselect() . . . . .	120
6.15.3.2 Draw() . . . . .	120
6.15.3.3 DrawCurrent() . . . . .	120
6.15.3.4 GetChildren() . . . . .	120
6.15.3.5 GetHoverStatus() [1/2] . . . . .	121
6.15.3.6 GetHoverStatus() [2/2] . . . . .	121
6.15.3.7 GetPosition() . . . . .	121
6.15.3.8 GetSize() . . . . .	121
6.15.3.9 GetVisible() . . . . .	121

---

6.15.3.10 isSelectable() . . . . .	121
6.15.3.11 isSelected() . . . . .	122
6.15.3.12 pack() . . . . .	122
6.15.3.13 select() . . . . .	122
6.15.3.14 SetPosition() [1/2] . . . . .	122
6.15.3.15 SetPosition() [2/2] . . . . .	122
6.15.3.16 SetVisible() . . . . .	122
6.15.3.17 ToggleVisible() . . . . .	122
6.15.3.18 UnpackAll() . . . . .	123
6.15.3.19 UpdateMouseCursorWhenHover() [1/2] . . . . .	123
6.15.3.20 UpdateMouseCursorWhenHover() [2/2] . . . . .	123
6.15.4 Member Data Documentation . . . . .	123
6.15.4.1 mChildren . . . . .	123
6.15.4.2 mIsSelected . . . . .	123
6.15.4.3 mPosition . . . . .	123
6.15.4.4 mVisible . . . . .	124
6.16 State::Context Struct Reference . . . . .	124
6.16.1 Constructor & Destructor Documentation . . . . .	124
6.16.1.1 Context() [1/2] . . . . .	124
6.16.1.2 Context() [2/2] . . . . .	124
6.16.2 Member Data Documentation . . . . .	125
6.16.2.1 categories . . . . .	125
6.16.2.2 dsInfo . . . . .	125
6.16.2.3 fonts . . . . .	125
6.16.2.4 textures . . . . .	125
6.17 GUI::DataStructure Class Reference . . . . .	126
6.17.1 Member Typedef Documentation . . . . .	127
6.17.1.1 Ptr . . . . .	127
6.17.2 Constructor & Destructor Documentation . . . . .	128
6.17.2.1 DataStructure() . . . . .	128
6.17.2.2 ~DataStructure() . . . . .	128
6.17.3 Member Function Documentation . . . . .	128
6.17.3.1 deselect() . . . . .	128
6.17.3.2 Draw() . . . . .	128
6.17.3.3 DrawCurrent() . . . . .	128
6.17.3.4 GetChildren() . . . . .	128
6.17.3.5 GetHoverStatus() [1/2] . . . . .	129
6.17.3.6 GetHoverStatus() [2/2] . . . . .	129
6.17.3.7 GetPosition() . . . . .	129
6.17.3.8 GetSize() . . . . .	129
6.17.3.9 GetVisible() . . . . .	129
6.17.3.10 isSelectable() . . . . .	129

6.17.3.11 isSelected() . . . . .	130
6.17.3.12 pack() . . . . .	130
6.17.3.13 select() . . . . .	130
6.17.3.14 SetPosition() [1/2] . . . . .	130
6.17.3.15 SetPosition() [2/2] . . . . .	130
6.17.3.16 SetVisible() . . . . .	130
6.17.3.17 ToggleVisible() . . . . .	130
6.17.3.18 UnpackAll() . . . . .	131
6.17.3.19 UpdateMouseCursorWhenHover() [1/2] . . . . .	131
6.17.3.20 UpdateMouseCursorWhenHover() [2/2] . . . . .	131
6.17.4 Member Data Documentation . . . . .	131
6.17.4.1 mChildren . . . . .	131
6.17.4.2 mIsSelected . . . . .	131
6.17.4.3 mPosition . . . . .	131
6.17.4.4 mVisible . . . . .	132
6.18 DLLState Class Reference . . . . .	132
6.18.1 Member Typedef Documentation . . . . .	134
6.18.1.1 Ptr . . . . .	135
6.18.2 Constructor & Destructor Documentation . . . . .	135
6.18.2.1 DLLState() . . . . .	135
6.18.2.2 ~DLLState() . . . . .	135
6.18.3 Member Function Documentation . . . . .	135
6.18.3.1 AddDeleteOperation() . . . . .	135
6.18.3.2 AddInitializeOperation() . . . . .	135
6.18.3.3 AddInsertOperation() . . . . .	135
6.18.3.4 AddIntFieldOperationOption() . . . . .	136
6.18.3.5 AddNoFieldOperationOption() . . . . .	136
6.18.3.6 AddOperations() . . . . .	136
6.18.3.7 AddSearchOperation() . . . . .	136
6.18.3.8 AddStringFieldOption() . . . . .	136
6.18.3.9 AddUpdateOperation() . . . . .	136
6.18.3.10 ClearAction() . . . . .	137
6.18.3.11 ClearError() . . . . .	137
6.18.3.12 Draw() . . . . .	137
6.18.3.13 DrawCurrentActionText() . . . . .	137
6.18.3.14 DrawCurrentErrorText() . . . . .	137
6.18.3.15 GetContext() . . . . .	137
6.18.3.16 InitNavigationBar() . . . . .	137
6.18.3.17 RequestStackClear() . . . . .	138
6.18.3.18 RequestStackPop() . . . . .	138
6.18.3.19 RequestStackPush() . . . . .	138
6.18.3.20 SetCurrentAction() . . . . .	138

---

6.18.3.21 SetCurrentError()	138
6.18.3.22 Success()	138
6.18.3.23 Update()	138
6.18.4 Member Data Documentation	139
6.18.4.1 activeDS	139
6.18.4.2 animController	139
6.18.4.3 codeHighlighter	139
6.18.4.4 footer	139
6.18.4.5 mContext	139
6.18.4.6 mCurrentAction	139
6.18.4.7 mCurrentError	139
6.18.4.8 mDLL	140
6.18.4.9 mStack	140
6.18.4.10 navigation	140
6.18.4.11 operationList	140
6.19 Algorithm::DoublyLinkedList Class Reference	140
6.19.1 Constructor & Destructor Documentation	142
6.19.1.1 DoublyLinkedList() [1/2]	143
6.19.1.2 DoublyLinkedList() [2/2]	143
6.19.1.3 ~DoublyLinkedList()	143
6.19.2 Member Function Documentation	143
6.19.2.1 ApplyInput()	143
6.19.2.2 DeleteHead()	143
6.19.2.3 DeleteMiddle()	143
6.19.2.4 DeleteTail()	144
6.19.2.5 Empty()	144
6.19.2.6 EmptyGenerator()	144
6.19.2.7 GenerateAnimation()	144
6.19.2.8 GenerateRelayoutAnimation()	144
6.19.2.9 HighlightArrowBoth()	144
6.19.2.10 HighlightArrowNext()	145
6.19.2.11 HighlightArrowPrev()	145
6.19.2.12 InitAction()	145
6.19.2.13 InsertAfterTail()	145
6.19.2.14 InsertHead()	145
6.19.2.15 InsertMiddle()	145
6.19.2.16 Random()	146
6.19.2.17 RandomFixedSize()	146
6.19.2.18 RandomFixedSizeGenerator()	146
6.19.2.19 RandomGenerator()	146
6.19.2.20 ReadFromExternalFile()	146
6.19.2.21 ReadFromFileGenerator()	146

6.19.2.22 ResetVisualizer()	146
6.19.2.23 Search()	147
6.19.2.24 size()	147
6.19.2.25 Update()	147
6.19.2.26 UserDefined()	147
6.19.2.27 UserDefinedGenerator()	147
6.19.3 Member Data Documentation	147
6.19.3.1 animController	147
6.19.3.2 codeHighlighter	148
6.19.3.3 maxN	148
6.19.3.4 visualizer	148
6.20 GUI::DoublyLinkedList Class Reference	148
6.20.1 Member Typedef Documentation	151
6.20.1.1 Ptr	151
6.20.2 Member Enumeration Documentation	151
6.20.2.1 ArrowType	151
6.20.2.2 Orientation	152
6.20.3 Constructor & Destructor Documentation	152
6.20.3.1 DoublyLinkedList() [1/2]	152
6.20.3.2 DoublyLinkedList() [2/2]	152
6.20.3.3 ~DoublyLinkedList()	152
6.20.4 Member Function Documentation	152
6.20.4.1 DeleteNode()	152
6.20.4.2 deselect()	153
6.20.4.3 Draw() [1/2]	153
6.20.4.4 Draw() [2/2]	153
6.20.4.5 DrawArrow()	153
6.20.4.6 DrawCurrent()	153
6.20.4.7 GenerateNode()	153
6.20.4.8 GetArrowTypeNext()	154
6.20.4.9 GetArrowTypePrev()	154
6.20.4.10 GetChildren()	154
6.20.4.11 GetHoverStatus() [1/2]	154
6.20.4.12 GetHoverStatus() [2/2]	154
6.20.4.13 GetList()	154
6.20.4.14 GetNodeDefaultPosition()	154
6.20.4.15 GetPosition()	155
6.20.4.16 GetShape()	155
6.20.4.17 GetSize()	155
6.20.4.18 GetVisible()	155
6.20.4.19 Import()	155
6.20.4.20 InsertNode()	155

---

6.20.4.21 isSelectable() . . . . .	156
6.20.4.22 isSelected() . . . . .	156
6.20.4.23 pack() . . . . .	156
6.20.4.24 Relayout() . . . . .	156
6.20.4.25 ResetArrow() . . . . .	156
6.20.4.26 select() . . . . .	156
6.20.4.27 SetArrowTypeNext() . . . . .	156
6.20.4.28 SetArrowTypePrev() . . . . .	157
6.20.4.29 SetOrientation() . . . . .	157
6.20.4.30 SetPosition() [1/2] . . . . .	157
6.20.4.31 SetPosition() [2/2] . . . . .	157
6.20.4.32 SetShape() . . . . .	157
6.20.4.33 SetShowHeadAndTail() . . . . .	157
6.20.4.34 SetVisible() . . . . .	158
6.20.4.35 size() . . . . .	158
6.20.4.36 ToggleVisible() . . . . .	158
6.20.4.37 UnpackAll() . . . . .	158
6.20.4.38 UpdateMouseCursorWhenHover() [1/2] . . . . .	158
6.20.4.39 UpdateMouseCursorWhenHover() [2/2] . . . . .	158
6.20.5 Member Data Documentation . . . . .	158
6.20.5.1 arrowStateNext . . . . .	159
6.20.5.2 arrowStatePrev . . . . .	159
6.20.5.3 fonts . . . . .	159
6.20.5.4 list . . . . .	159
6.20.5.5 mChildren . . . . .	159
6.20.5.6 mDisplayHeadAndTail . . . . .	159
6.20.5.7 mIsSelected . . . . .	159
6.20.5.8 mNodeDistance . . . . .	159
6.20.5.9 mOrientation . . . . .	160
6.20.5.10 mPosition . . . . .	160
6.20.5.11 mShape . . . . .	160
6.20.5.12 mVisible . . . . .	160
6.21 DSInfo Class Reference . . . . .	160
6.21.1 Member Function Documentation . . . . .	161
6.21.1.1 Get() . . . . .	161
6.21.1.2 Register() . . . . .	161
6.21.2 Member Data Documentation . . . . .	161
6.21.2.1 mFactories . . . . .	161
6.22 Algorithm::DynamicArray Class Reference . . . . .	162
6.22.1 Constructor & Destructor Documentation . . . . .	163
6.22.1.1 DynamicArray() [1/2] . . . . .	163
6.22.1.2 DynamicArray() [2/2] . . . . .	164

---

6.22.1.3 ~DynamicArray() . . . . .	164
6.22.2 Member Function Documentation . . . . .	164
6.22.2.1 Access() . . . . .	164
6.22.2.2 ApplyInput() . . . . .	164
6.22.2.3 Empty() . . . . .	164
6.22.2.4 EmptyGenerator() . . . . .	164
6.22.2.5 GenerateAnimation() . . . . .	165
6.22.2.6 GenerateRelayoutAnimation() . . . . .	165
6.22.2.7 InitAction() . . . . .	165
6.22.2.8 PopBack() . . . . .	165
6.22.2.9 PushBack() . . . . .	165
6.22.2.10 Random() . . . . .	165
6.22.2.11 RandomFixedSize() . . . . .	165
6.22.2.12 RandomFixedSizeGenerator() . . . . .	166
6.22.2.13 RandomGenerator() . . . . .	166
6.22.2.14 ReadFromExternalFile() . . . . .	166
6.22.2.15 ReadFromFileGenerator() . . . . .	166
6.22.2.16 ResetVisualizer() . . . . .	166
6.22.2.17 Search() . . . . .	166
6.22.2.18 size() . . . . .	166
6.22.2.19 Update() . . . . .	167
6.22.2.20 UserDefined() . . . . .	167
6.22.2.21 UserDefinedGenerator() . . . . .	167
6.22.3 Member Data Documentation . . . . .	167
6.22.3.1 animController . . . . .	167
6.22.3.2 codeHighlighter . . . . .	167
6.22.3.3 maxN . . . . .	167
6.22.3.4 visualizer . . . . .	168
6.23 GUI::DynamicArray Class Reference . . . . .	168
6.23.1 Member Typedef Documentation . . . . .	171
6.23.1.1 Ptr . . . . .	171
6.23.2 Constructor & Destructor Documentation . . . . .	171
6.23.2.1 DynamicArray() [1/2] . . . . .	171
6.23.2.2 DynamicArray() [2/2] . . . . .	171
6.23.2.3 ~DynamicArray() . . . . .	171
6.23.3 Member Function Documentation . . . . .	171
6.23.3.1 Clear() . . . . .	172
6.23.3.2 DeleteNode() . . . . .	172
6.23.3.3 deselect() . . . . .	172
6.23.3.4 Draw() [1/2] . . . . .	172
6.23.3.5 Draw() [2/2] . . . . .	172
6.23.3.6 DrawCurrent() . . . . .	172

---

6.23.3.7 GenerateNode()	173
6.23.3.8 GetCapacity()	173
6.23.3.9 GetCapacityFromLength()	173
6.23.3.10 GetChildren()	173
6.23.3.11 GetHoverStatus() [1/2]	173
6.23.3.12 GetHoverStatus() [2/2]	173
6.23.3.13 GetLength()	173
6.23.3.14 GetList()	174
6.23.3.15 GetNodeDefaultPosition()	174
6.23.3.16 GetPosition()	174
6.23.3.17 GetShape()	174
6.23.3.18 GetSize()	174
6.23.3.19 GetVisible()	174
6.23.3.20 Import()	174
6.23.3.21 InsertNode()	175
6.23.3.22 isSelectable()	175
6.23.3.23 isSelected()	175
6.23.3.24 operator[]() [1/2]	175
6.23.3.25 operator[]() [2/2]	175
6.23.3.26 pack()	175
6.23.3.27 Relayout()	175
6.23.3.28 Reserve()	176
6.23.3.29 Resize()	176
6.23.3.30 select()	176
6.23.3.31 SetPosition() [1/2]	176
6.23.3.32 SetPosition() [2/2]	176
6.23.3.33 SetShape()	176
6.23.3.34 SetVisible()	176
6.23.3.35 ToggleVisible()	177
6.23.3.36 UnpackAll()	177
6.23.3.37 UpdateMouseCursorWhenHover() [1/2]	177
6.23.3.38 UpdateMouseCursorWhenHover() [2/2]	177
6.23.4 Member Data Documentation	177
6.23.4.1 capacity	177
6.23.4.2 fonts	177
6.23.4.3 length	178
6.23.4.4 list	178
6.23.4.5 mChildren	178
6.23.4.6 mIsSelected	178
6.23.4.7 mNodeDistance	178
6.23.4.8 mPosition	178
6.23.4.9 mShape	178

6.23.4.10 mVisible . . . . .	179
6.24 DynamicArrayState Class Reference . . . . .	179
6.24.1 Member Typedef Documentation . . . . .	182
6.24.1.1 Ptr . . . . .	182
6.24.2 Constructor & Destructor Documentation . . . . .	182
6.24.2.1 DynamicArrayState() . . . . .	182
6.24.2.2 ~DynamicArrayState() . . . . .	182
6.24.3 Member Function Documentation . . . . .	182
6.24.3.1 AddAccessOperation() . . . . .	182
6.24.3.2 AddDeleteOperation() . . . . .	182
6.24.3.3 AddInitializeOperation() . . . . .	183
6.24.3.4 AddInsertOperation() . . . . .	183
6.24.3.5 AddIntFieldOperationOption() . . . . .	183
6.24.3.6 AddNoFieldOperationOption() . . . . .	183
6.24.3.7 AddOperations() . . . . .	183
6.24.3.8 AddSearchOperation() . . . . .	183
6.24.3.9 AddStringFieldOption() . . . . .	184
6.24.3.10 AddUpdateOperation() . . . . .	184
6.24.3.11 ClearAction() . . . . .	184
6.24.3.12 ClearError() . . . . .	184
6.24.3.13 Draw() . . . . .	184
6.24.3.14 DrawCurrentActionText() . . . . .	184
6.24.3.15 DrawCurrentErrorText() . . . . .	185
6.24.3.16 GetContext() . . . . .	185
6.24.3.17 InitNavigationBar() . . . . .	185
6.24.3.18 RequestStackClear() . . . . .	185
6.24.3.19 RequestStackPop() . . . . .	185
6.24.3.20 RequestStackPush() . . . . .	185
6.24.3.21 SetCurrentAction() . . . . .	185
6.24.3.22 SetCurrentError() . . . . .	186
6.24.3.23 Success() . . . . .	186
6.24.3.24 Update() . . . . .	186
6.24.4 Member Data Documentation . . . . .	186
6.24.4.1 activeDS . . . . .	186
6.24.4.2 animController . . . . .	186
6.24.4.3 codeHighlighter . . . . .	186
6.24.4.4 footer . . . . .	187
6.24.4.5 mContext . . . . .	187
6.24.4.6 mCurrentAction . . . . .	187
6.24.4.7 mCurrentError . . . . .	187
6.24.4.8 mDynamicArray . . . . .	187
6.24.4.9 mStack . . . . .	187

---

6.24.4.10 navigation . . . . .	187
6.24.4.11 operationList . . . . .	188
6.25 FontHolder Class Reference . . . . .	188
6.25.1 Constructor & Destructor Documentation . . . . .	189
6.25.1.1 FontHolder() . . . . .	189
6.25.1.2 ~FontHolder() . . . . .	189
6.25.2 Member Function Documentation . . . . .	189
6.25.2.1 Get() [1/2] . . . . .	189
6.25.2.2 Get() [2/2] . . . . .	189
6.25.2.3 InsertResource() . . . . .	189
6.25.2.4 Load() . . . . .	190
6.25.3 Member Data Documentation . . . . .	190
6.25.3.1 mFontMap . . . . .	190
6.26 GUI::Footer< T > Class Template Reference . . . . .	190
6.26.1 Member Typedef Documentation . . . . .	192
6.26.1.1 Ptr . . . . .	192
6.26.2 Constructor & Destructor Documentation . . . . .	192
6.26.2.1 Footer() . . . . .	193
6.26.2.2 ~Footer() . . . . .	193
6.26.3 Member Function Documentation . . . . .	193
6.26.3.1 deselect() . . . . .	193
6.26.3.2 Draw() [1/2] . . . . .	193
6.26.3.3 Draw() [2/2] . . . . .	193
6.26.3.4 DrawCurrent() . . . . .	193
6.26.3.5 GetChildren() . . . . .	194
6.26.3.6 GetHoverStatus() [1/2] . . . . .	194
6.26.3.7 GetHoverStatus() [2/2] . . . . .	194
6.26.3.8 GetPosition() . . . . .	194
6.26.3.9 GetSize() . . . . .	194
6.26.3.10 GetVisible() . . . . .	194
6.26.3.11 isSelectable() . . . . .	195
6.26.3.12 isSelected() . . . . .	195
6.26.3.13 pack() . . . . .	195
6.26.3.14 select() . . . . .	195
6.26.3.15 SetPosition() [1/2] . . . . .	195
6.26.3.16 SetPosition() [2/2] . . . . .	195
6.26.3.17 SetVisible() . . . . .	196
6.26.3.18 ToggleVisible() . . . . .	196
6.26.3.19 UnpackAll() . . . . .	196
6.26.3.20 UpdateMouseCursorWhenHover() [1/2] . . . . .	196
6.26.3.21 UpdateMouseCursorWhenHover() [2/2] . . . . .	196
6.26.4 Member Data Documentation . . . . .	196

---

6.26.4.1 mChildren . . . . .	196
6.26.4.2 mIsSelected . . . . .	197
6.26.4.3 mPosition . . . . .	197
6.26.4.4 mVisible . . . . .	197
6.27 HomepageState Class Reference . . . . .	197
6.27.1 Member Typedef Documentation . . . . .	199
6.27.1.1 Ptr . . . . .	199
6.27.2 Constructor & Destructor Documentation . . . . .	199
6.27.2.1 HomepageState() . . . . .	200
6.27.2.2 ~HomepageState() . . . . .	200
6.27.3 Member Function Documentation . . . . .	200
6.27.3.1 CreateCard() . . . . .	200
6.27.3.2 Draw() . . . . .	200
6.27.3.3 DrawIntroduction() . . . . .	200
6.27.3.4 GetContext() . . . . .	200
6.27.3.5 InitCards() . . . . .	201
6.27.3.6 InitNavigationBar() . . . . .	201
6.27.3.7 RequestStackClear() . . . . .	201
6.27.3.8 RequestStackPop() . . . . .	201
6.27.3.9 RequestStackPush() . . . . .	201
6.27.3.10 Update() . . . . .	201
6.27.4 Member Data Documentation . . . . .	201
6.27.4.1 hasInitializeCard . . . . .	201
6.27.4.2 mCards . . . . .	202
6.27.4.3 mContext . . . . .	202
6.27.4.4 mStack . . . . .	202
6.27.4.5 navigation . . . . .	202
6.28 CategoryInfo::Info Struct Reference . . . . .	202
6.28.1 Constructor & Destructor Documentation . . . . .	202
6.28.1.1 Info() . . . . .	203
6.28.2 Member Data Documentation . . . . .	203
6.28.2.1 categoryID . . . . .	203
6.28.2.2 categoryName . . . . .	203
6.28.2.3 mDS . . . . .	203
6.29 DSInfo::Info Struct Reference . . . . .	203
6.29.1 Constructor & Destructor Documentation . . . . .	204
6.29.1.1 Info() . . . . .	204
6.29.2 Member Data Documentation . . . . .	204
6.29.2.1 abbr . . . . .	204
6.29.2.2 categoryID . . . . .	204
6.29.2.3 ID . . . . .	204
6.29.2.4 name . . . . .	204

6.29.2.5 stateID . . . . .	204
6.29.2.6 thumbnail . . . . .	205
6.30 GUI::InputField Class Reference . . . . .	205
6.30.1 Member Typedef Documentation . . . . .	208
6.30.1.1 Ptr . . . . .	208
6.30.2 Constructor & Destructor Documentation . . . . .	208
6.30.2.1 InputField() . . . . .	208
6.30.2.2 ~InputField() . . . . .	208
6.30.3 Member Function Documentation . . . . .	208
6.30.3.1 AllFieldDisableEdit() . . . . .	208
6.30.3.2 deselect() . . . . .	208
6.30.3.3 Draw() . . . . .	208
6.30.3.4 DrawField() . . . . .	209
6.30.3.5 ExtractValue() . . . . .	209
6.30.3.6 GetEditMode() . . . . .	209
6.30.3.7 GetHoverStatus() [1/2] . . . . .	209
6.30.3.8 GetHoverStatus() [2/2] . . . . .	209
6.30.3.9 GetLabel() . . . . .	209
6.30.3.10 GetPosition() . . . . .	210
6.30.3.11 GetSize() . . . . .	210
6.30.3.12 GetVisible() . . . . .	210
6.30.3.13 IsClicked() . . . . .	210
6.30.3.14 isSelectable() . . . . .	210
6.30.3.15 isSelected() . . . . .	210
6.30.3.16 Randomize() . . . . .	210
6.30.3.17 select() . . . . .	211
6.30.3.18 SetEditMode() . . . . .	211
6.30.3.19 SetInputFieldSize() . . . . .	211
6.30.3.20 SetLabel() . . . . .	211
6.30.3.21 SetLabelSize() . . . . .	211
6.30.3.22 SetPosition() [1/2] . . . . .	211
6.30.3.23 SetPosition() [2/2] . . . . .	211
6.30.3.24 SetVisible() . . . . .	212
6.30.3.25 ToggleVisible() . . . . .	212
6.30.3.26 UpdateMouseCursorWhenHover() [1/2] . . . . .	212
6.30.3.27 UpdateMouseCursorWhenHover() [2/2] . . . . .	212
6.30.4 Member Data Documentation . . . . .	212
6.30.4.1 editMode . . . . .	212
6.30.4.2 extractedValue . . . . .	212
6.30.4.3 fields . . . . .	213
6.30.4.4 fonts . . . . .	213
6.30.4.5 inputFieldSize . . . . .	213

---

6.30.4.6 label . . . . .	213
6.30.4.7 labelFontSize . . . . .	213
6.30.4.8 mFieldIndex . . . . .	213
6.30.4.9 mIsSelected . . . . .	213
6.30.4.10 mPosition . . . . .	213
6.30.4.11 mVisible . . . . .	214
6.31 ArrayState< T >::IntegerInput Struct Reference . . . . .	214
6.31.1 Member Data Documentation . . . . .	214
6.31.1.1 label . . . . .	214
6.31.1.2 maxValue . . . . .	214
6.31.1.3 minValue . . . . .	214
6.31.1.4 width . . . . .	215
6.32 LLState< T >::IntegerInput Struct Reference . . . . .	215
6.32.1 Member Data Documentation . . . . .	215
6.32.1.1 label . . . . .	215
6.32.1.2 maxValue . . . . .	215
6.32.1.3 minValue . . . . .	215
6.32.1.4 width . . . . .	216
6.33 GUI::IntegerField Class Reference . . . . .	216
6.33.1 Member Typedef Documentation . . . . .	219
6.33.1.1 Ptr . . . . .	219
6.33.2 Constructor & Destructor Documentation . . . . .	219
6.33.2.1 IntegerInputField() . . . . .	219
6.33.2.2 ~IntegerField() . . . . .	219
6.33.3 Member Function Documentation . . . . .	219
6.33.3.1 AllFieldDisableEdit() . . . . .	219
6.33.3.2 deselect() . . . . .	219
6.33.3.3 Draw() . . . . .	220
6.33.3.4 DrawField() . . . . .	220
6.33.3.5 ExtractValue() . . . . .	220
6.33.3.6 GetEditMode() . . . . .	220
6.33.3.7 GetHoverStatus() [1/2] . . . . .	220
6.33.3.8 GetHoverStatus() [2/2] . . . . .	220
6.33.3.9 GetLabel() . . . . .	221
6.33.3.10 GetPosition() . . . . .	221
6.33.3.11 GetSize() . . . . .	221
6.33.3.12 GetVisible() . . . . .	221
6.33.3.13 IsClicked() . . . . .	221
6.33.3.14 isSelectable() . . . . .	221
6.33.3.15 isSelected() . . . . .	221
6.33.3.16 Randomize() . . . . .	222
6.33.3.17 select() . . . . .	222

---

6.33.3.18 SetConstraint() . . . . .	222
6.33.3.19 SetEditMode() . . . . .	222
6.33.3.20 SetInputFieldSize() . . . . .	222
6.33.3.21 SetLabel() . . . . .	222
6.33.3.22 SetLabelSize() . . . . .	222
6.33.3.23 SetPosition() [1/2] . . . . .	223
6.33.3.24 SetPosition() [2/2] . . . . .	223
6.33.3.25 SetVisible() . . . . .	223
6.33.3.26 ToggleVisible() . . . . .	223
6.33.3.27 UpdateMouseCursorWhenHover() [1/2] . . . . .	223
6.33.3.28 UpdateMouseCursorWhenHover() [2/2] . . . . .	223
6.33.4 Member Data Documentation . . . . .	224
6.33.4.1 editMode . . . . .	224
6.33.4.2 extractedValue . . . . .	224
6.33.4.3 fields . . . . .	224
6.33.4.4 fonts . . . . .	224
6.33.4.5 input . . . . .	224
6.33.4.6 inputFieldSize . . . . .	224
6.33.4.7 label . . . . .	224
6.33.4.8 labelFontSize . . . . .	225
6.33.4.9 mFieldIndex . . . . .	225
6.33.4.10 mIsSelected . . . . .	225
6.33.4.11 mMaxValue . . . . .	225
6.33.4.12 mMinValue . . . . .	225
6.33.4.13 mPosition . . . . .	225
6.33.4.14 mVisible . . . . .	225
6.34 Core::List< T >::iterator Class Reference . . . . .	226
6.34.1 Member Typedef Documentation . . . . .	226
6.34.1.1 difference_type . . . . .	226
6.34.1.2 iterator_category . . . . .	227
6.34.1.3 pointer . . . . .	227
6.34.1.4 reference . . . . .	227
6.34.1.5 value_type . . . . .	227
6.34.2 Constructor & Destructor Documentation . . . . .	227
6.34.2.1 iterator() [1/2] . . . . .	227
6.34.2.2 iterator() [2/2] . . . . .	227
6.34.3 Member Function Documentation . . . . .	227
6.34.3.1 operator"!="() . . . . .	228
6.34.3.2 operator*() . . . . .	228
6.34.3.3 operator+() . . . . .	228
6.34.3.4 operator++() [1/2] . . . . .	228
6.34.3.5 operator++() [2/2] . . . . .	228

6.34.3.6 operator-() . . . . .	228
6.34.3.7 operator--() [1/2] . . . . .	229
6.34.3.8 operator--() [2/2] . . . . .	229
6.34.3.9 operator->() . . . . .	229
6.34.3.10 operator=() . . . . .	229
6.34.3.11 operator==() . . . . .	229
6.34.3.12 swap() . . . . .	229
6.34.4 Friends And Related Function Documentation . . . . .	229
6.34.4.1 List . . . . .	230
6.34.5 Member Data Documentation . . . . .	230
6.34.5.1 ptr . . . . .	230
6.35 GUI::LinkedList Class Reference . . . . .	230
6.35.1 Member Typedef Documentation . . . . .	233
6.35.1.1 Ptr . . . . .	233
6.35.2 Member Enumeration Documentation . . . . .	233
6.35.2.1 ArrowType . . . . .	233
6.35.2.2 Orientation . . . . .	233
6.35.3 Constructor & Destructor Documentation . . . . .	234
6.35.3.1 LinkedList() [1/2] . . . . .	234
6.35.3.2 LinkedList() [2/2] . . . . .	234
6.35.3.3 ~LinkedList() . . . . .	234
6.35.4 Member Function Documentation . . . . .	234
6.35.4.1 DeleteNode() . . . . .	234
6.35.4.2 deselect() . . . . .	234
6.35.4.3 Draw() [1/2] . . . . .	235
6.35.4.4 Draw() [2/2] . . . . .	235
6.35.4.5 DrawCurrent() . . . . .	235
6.35.4.6 GenerateNode() . . . . .	235
6.35.4.7 GetChildren() . . . . .	235
6.35.4.8 GetHoverStatus() [1/2] . . . . .	235
6.35.4.9 GetHoverStatus() [2/2] . . . . .	236
6.35.4.10 GetList() . . . . .	236
6.35.4.11 GetNodeDefaultPosition() . . . . .	236
6.35.4.12 GetPosition() . . . . .	236
6.35.4.13 GetShape() . . . . .	236
6.35.4.14 GetSize() . . . . .	236
6.35.4.15 GetVisible() . . . . .	236
6.35.4.16 Import() . . . . .	237
6.35.4.17 InsertNode() . . . . .	237
6.35.4.18 isSelectable() . . . . .	237
6.35.4.19 isSelected() . . . . .	237
6.35.4.20 pack() . . . . .	237

---

6.35.4.21 Relayout()	237
6.35.4.22 select()	238
6.35.4.23 SetOrientation()	238
6.35.4.24 SetPosition() [1/2]	238
6.35.4.25 SetPosition() [2/2]	238
6.35.4.26 SetShape()	238
6.35.4.27 SetShowHeadAndTail()	238
6.35.4.28 SetVisible()	238
6.35.4.29 size()	239
6.35.4.30 ToggleVisible()	239
6.35.4.31 UnpackAll()	239
6.35.4.32 UpdateMouseCursorWhenHover() [1/2]	239
6.35.4.33 UpdateMouseCursorWhenHover() [2/2]	239
6.35.5 Member Data Documentation	239
6.35.5.1 fonts	239
6.35.5.2 list	240
6.35.5.3 mChildren	240
6.35.5.4 mDisplayHeadAndTail	240
6.35.5.5 mIsSelected	240
6.35.5.6 mNodeDistance	240
6.35.5.7 mOrientation	240
6.35.5.8 mPosition	240
6.35.5.9 mShape	240
6.35.5.10 mVisible	241
6.36 Core::List< T > Class Template Reference	241
6.36.1 Constructor & Destructor Documentation	242
6.36.1.1 List() [1/4]	242
6.36.1.2 List() [2/4]	242
6.36.1.3 List() [3/4]	242
6.36.1.4 List() [4/4]	242
6.36.2 Member Function Documentation	243
6.36.2.1 at() [1/2]	243
6.36.2.2 at() [2/2]	243
6.36.2.3 back() [1/2]	243
6.36.2.4 back() [2/2]	243
6.36.2.5 begin() [1/2]	243
6.36.2.6 begin() [2/2]	243
6.36.2.7 clear()	244
6.36.2.8 empty()	244
6.36.2.9 end() [1/2]	244
6.36.2.10 end() [2/2]	244
6.36.2.11 front() [1/2]	244

6.36.2.12 front() [2/2] . . . . .	244
6.36.2.13 get_iterator() . . . . .	244
6.36.2.14 insert_previous() . . . . .	245
6.36.2.15 move_previous() . . . . .	245
6.36.2.16 operator=() . . . . .	245
6.36.2.17 operator[]() [1/2] . . . . .	245
6.36.2.18 operator[]() [2/2] . . . . .	245
6.36.2.19 pop_back() . . . . .	245
6.36.2.20 pop_front() . . . . .	246
6.36.2.21 push_back() . . . . .	246
6.36.2.22 push_front() . . . . .	246
6.36.2.23 remove() [1/3] . . . . .	246
6.36.2.24 remove() [2/3] . . . . .	246
6.36.2.25 remove() [3/3] . . . . .	246
6.36.2.26 remove_if() [1/2] . . . . .	247
6.36.2.27 remove_if() [2/2] . . . . .	247
6.36.2.28 reset() . . . . .	247
6.36.2.29 size() . . . . .	247
6.36.3 Member Data Documentation . . . . .	247
6.36.3.1 mBegin . . . . .	247
6.36.3.2 mEnd . . . . .	247
6.36.3.3 mSize . . . . .	248
6.37 LLState< T > Class Template Reference . . . . .	248
6.37.1 Member Typedef Documentation . . . . .	251
6.37.1.1 Ptr . . . . .	251
6.37.2 Constructor & Destructor Documentation . . . . .	251
6.37.2.1 LLState() . . . . .	251
6.37.2.2 ~LLState() . . . . .	251
6.37.3 Member Function Documentation . . . . .	251
6.37.3.1 AddDeleteOperation() . . . . .	251
6.37.3.2 AddInitializeOperation() . . . . .	251
6.37.3.3 AddInsertOperation() . . . . .	252
6.37.3.4 AddIntFieldOperationOption() . . . . .	252
6.37.3.5 AddNoFieldOperationOption() . . . . .	252
6.37.3.6 AddOperations() . . . . .	252
6.37.3.7 AddSearchOperation() . . . . .	252
6.37.3.8 AddStringFieldOption() . . . . .	253
6.37.3.9 AddUpdateOperation() . . . . .	253
6.37.3.10 ClearAction() . . . . .	253
6.37.3.11 ClearError() . . . . .	253
6.37.3.12 Draw() . . . . .	253
6.37.3.13 DrawCurrentActionText() . . . . .	253

---

6.37.3.14 DrawCurrentErrorText()	254
6.37.3.15 GetContext()	254
6.37.3.16 InitNavigationBar()	254
6.37.3.17 RequestStackClear()	254
6.37.3.18 RequestStackPop()	254
6.37.3.19 RequestStackPush()	254
6.37.3.20 SetCurrentAction()	254
6.37.3.21 SetCurrentError()	255
6.37.3.22 Success()	255
6.37.3.23 Update()	255
6.37.4 Member Data Documentation	255
6.37.4.1 activeDS	255
6.37.4.2 animController	255
6.37.4.3 codeHighlighter	255
6.37.4.4 footer	256
6.37.4.5 mContext	256
6.37.4.6 mCurrentAction	256
6.37.4.7 mCurrentError	256
6.37.4.8 mStack	256
6.37.4.9 navigation	256
6.37.4.10 operationList	256
6.38 GUI::NavigationBar Class Reference	257
6.38.1 Member Typedef Documentation	259
6.38.1.1 Ptr	259
6.38.2 Constructor & Destructor Documentation	259
6.38.2.1 NavigationBar() [1/2]	259
6.38.2.2 NavigationBar() [2/2]	259
6.38.2.3 ~NavigationBar()	259
6.38.3 Member Function Documentation	260
6.38.3.1 AtSettings()	260
6.38.3.2 ClearTitle()	260
6.38.3.3 deselect()	260
6.38.3.4 Draw()	260
6.38.3.5 DrawLogo()	260
6.38.3.6 DrawSettings()	260
6.38.3.7 DrawTitles()	261
6.38.3.8 GetHoverStatus() [1/2]	261
6.38.3.9 GetHoverStatus() [2/2]	261
6.38.3.10 GetPosition()	261
6.38.3.11 GetSize()	261
6.38.3.12 GetVisible()	261
6.38.3.13 InsertTitle()	262

---

6.38.3.14 isSelectable() . . . . .	262
6.38.3.15 isSelected() . . . . .	262
6.38.3.16 select() . . . . .	262
6.38.3.17 SetActiveTitle() . . . . .	262
6.38.3.18 SetBackToPreviousLink() . . . . .	262
6.38.3.19 SetCategory() . . . . .	263
6.38.3.20 SetDirectLink() . . . . .	263
6.38.3.21 SetHomepageID() . . . . .	263
6.38.3.22 SetPosition() [1/2] . . . . .	263
6.38.3.23 SetPosition() [2/2] . . . . .	263
6.38.3.24 SetSettingsID() . . . . .	263
6.38.3.25 SetVisibleTitle() . . . . .	263
6.38.3.26 setVisible() . . . . .	264
6.38.3.27 ToggleVisible() . . . . .	264
6.38.3.28 UpdateMouseCursorWhenHover() [1/2] . . . . .	264
6.38.3.29 UpdateMouseCursorWhenHover() [2/2] . . . . .	264
6.38.4 Member Data Documentation . . . . .	264
6.38.4.1 activeTitle . . . . .	264
6.38.4.2 atSettings . . . . .	264
6.38.4.3 backToPrvState . . . . .	265
6.38.4.4 currentCategory . . . . .	265
6.38.4.5 fonts . . . . .	265
6.38.4.6 hasTitle . . . . .	265
6.38.4.7 homepageID . . . . .	265
6.38.4.8 hoverBounds . . . . .	265
6.38.4.9 isHover . . . . .	265
6.38.4.10 mIsSelected . . . . .	265
6.38.4.11 mPosition . . . . .	266
6.38.4.12 mTitles . . . . .	266
6.38.4.13 mVisible . . . . .	266
6.38.4.14 settingsID . . . . .	266
6.38.4.15 toLink . . . . .	266
6.39 Core::Node< T > Class Template Reference . . . . .	266
6.39.1 Constructor & Destructor Documentation . . . . .	267
6.39.1.1 Node() [1/4] . . . . .	267
6.39.1.2 Node() [2/4] . . . . .	267
6.39.1.3 Node() [3/4] . . . . .	267
6.39.1.4 Node() [4/4] . . . . .	267
6.39.2 Member Data Documentation . . . . .	267
6.39.2.1 mNext . . . . .	268
6.39.2.2 mPrev . . . . .	268
6.39.2.3 mValue . . . . .	268

---

6.40 GUI::Node Class Reference . . . . .	268
6.40.1 Member Typedef Documentation . . . . .	271
6.40.1.1 Ptr . . . . .	271
6.40.2 Member Enumeration Documentation . . . . .	271
6.40.2.1 Shape . . . . .	271
6.40.2.2 State . . . . .	272
6.40.3 Constructor & Destructor Documentation . . . . .	272
6.40.3.1 Node() [1/2] . . . . .	272
6.40.3.2 Node() [2/2] . . . . .	272
6.40.3.3 ~Node() . . . . .	272
6.40.4 Member Function Documentation . . . . .	273
6.40.4.1 AddColor() . . . . .	273
6.40.4.2 AnimationOnNode() . . . . .	273
6.40.4.3 ClearLabel() . . . . .	273
6.40.4.4 deselect() . . . . .	273
6.40.4.5 Draw() [1/2] . . . . .	273
6.40.4.6 Draw() [2/2] . . . . .	273
6.40.4.7 DrawLabel() . . . . .	274
6.40.4.8 DrawNode() . . . . .	274
6.40.4.9 GetBackgroundColor() . . . . .	274
6.40.4.10 GetHoverStatus() [1/2] . . . . .	274
6.40.4.11 GetHoverStatus() [2/2] . . . . .	274
6.40.4.12 GetNodeState() . . . . .	274
6.40.4.13 GetOutlineColor() . . . . .	275
6.40.4.14 GetPosition() . . . . .	275
6.40.4.15 GetReachable() . . . . .	275
6.40.4.16 GetShape() . . . . .	275
6.40.4.17 GetSize() . . . . .	275
6.40.4.18 GetTextColor() . . . . .	275
6.40.4.19 GetValue() . . . . .	275
6.40.4.20 GetVisible() . . . . .	276
6.40.4.21 IsActive() . . . . .	276
6.40.4.22 isSelectable() . . . . .	276
6.40.4.23 isSelected() . . . . .	276
6.40.4.24 select() . . . . .	276
6.40.4.25 setActive() . . . . .	276
6.40.4.26 SetLabel() . . . . .	276
6.40.4.27 SetLabelFontSize() . . . . .	277
6.40.4.28 SetNodeState() . . . . .	277
6.40.4.29 SetPosition() [1/2] . . . . .	277
6.40.4.30 SetPosition() [2/2] . . . . .	277
6.40.4.31 SetRadius() . . . . .	277

---

6.40.4.32 SetReachable()	277
6.40.4.33 SetShape()	277
6.40.4.34 SetValue()	278
6.40.4.35 SetValueFontSize()	278
6.40.4.36 SetVisible()	278
6.40.4.37 ToggleVisible()	278
6.40.4.38 UpdateMouseCursorWhenHover() [1/2]	278
6.40.4.39 UpdateMouseCursorWhenHover() [2/2]	278
6.40.5 Member Data Documentation	278
6.40.5.1 animateNode	279
6.40.5.2 fonts	279
6.40.5.3 labelFontSize	279
6.40.5.4 mActive	279
6.40.5.5 mActiveColor	279
6.40.5.6 mBackgroundColor	279
6.40.5.7 mBorderColor	279
6.40.5.8 mDefaultColor	279
6.40.5.9 mIsSelected	280
6.40.5.10 mLabel	280
6.40.5.11 mNodeState	280
6.40.5.12 mOutlineColor	280
6.40.5.13 mPosition	280
6.40.5.14 mRadius	280
6.40.5.15 mReachable	280
6.40.5.16 mShape	280
6.40.5.17 mTextColor	281
6.40.5.18 mValue	281
6.40.5.19 mVisible	281
6.40.5.20 valueFontSize	281
6.41 NonCopyable< T > Class Template Reference	281
6.41.1 Constructor & Destructor Documentation	281
6.41.1.1 NonCopyable() [1/2]	282
6.41.1.2 NonCopyable() [2/2]	282
6.41.1.3 ~NonCopyable()	282
6.41.2 Member Function Documentation	282
6.41.2.1 operator=()	282
6.42 GUI::OperationContainer Class Reference	283
6.42.1 Member Typedef Documentation	284
6.42.1.1 Ptr	285
6.42.2 Constructor & Destructor Documentation	285
6.42.2.1 OperationContainer()	285
6.42.2.2 ~OperationContainer()	285

---

6.42.3 Member Function Documentation . . . . .	285
6.42.3.1 deselect() . . . . .	285
6.42.3.2 Draw() . . . . .	285
6.42.3.3 DrawCurrent() . . . . .	285
6.42.3.4 GetChildren() . . . . .	286
6.42.3.5 GetHoverStatus() [1/2] . . . . .	286
6.42.3.6 GetHoverStatus() [2/2] . . . . .	286
6.42.3.7 GetPosition() . . . . .	286
6.42.3.8 GetSize() . . . . .	286
6.42.3.9 GetVisible() . . . . .	286
6.42.3.10 isSelectable() . . . . .	287
6.42.3.11 isSelected() . . . . .	287
6.42.3.12 pack() . . . . .	287
6.42.3.13 select() . . . . .	287
6.42.3.14 SetPosition() [1/2] . . . . .	287
6.42.3.15 SetPosition() [2/2] . . . . .	287
6.42.3.16 SetVisible() . . . . .	288
6.42.3.17 ToggleVisible() . . . . .	288
6.42.3.18 UnpackAll() . . . . .	288
6.42.3.19 UpdateMouseCursorWhenHover() [1/2] . . . . .	288
6.42.3.20 UpdateMouseCursorWhenHover() [2/2] . . . . .	288
6.42.3.21 UpdatePosition() . . . . .	288
6.42.4 Member Data Documentation . . . . .	288
6.42.4.1 mChildren . . . . .	289
6.42.4.2 mIsSelected . . . . .	289
6.42.4.3 mPosition . . . . .	289
6.42.4.4 mVisible . . . . .	289
6.43 GUI::OperationList Class Reference . . . . .	289
6.43.1 Member Typedef Documentation . . . . .	291
6.43.1.1 Ptr . . . . .	292
6.43.2 Constructor & Destructor Documentation . . . . .	292
6.43.2.1 OperationList() [1/2] . . . . .	292
6.43.2.2 OperationList() [2/2] . . . . .	292
6.43.2.3 ~OperationList() . . . . .	292
6.43.3 Member Function Documentation . . . . .	292
6.43.3.1 AddOperation() . . . . .	292
6.43.3.2 deselect() . . . . .	292
6.43.3.3 Draw() . . . . .	293
6.43.3.4 DrawCurrent() . . . . .	293
6.43.3.5 GetChildren() . . . . .	293
6.43.3.6 GetHoverStatus() [1/2] . . . . .	293
6.43.3.7 GetHoverStatus() [2/2] . . . . .	293

---

6.43.3.8 GetPosition()	293
6.43.3.9 GetSize()	294
6.43.3.10 GetVisible()	294
6.43.3.11 HideAllOptions()	294
6.43.3.12 InitActionBar()	294
6.43.3.13 IsSelectable()	294
6.43.3.14 IsSelected()	294
6.43.3.15 pack()	294
6.43.3.16 select()	295
6.43.3.17 SetPosition() [1/2]	295
6.43.3.18 SetPosition() [2/2]	295
6.43.3.19 SetVisible()	295
6.43.3.20 ShowOptions()	295
6.43.3.21 ToggleOperations()	295
6.43.3.22 ToggleVisible()	295
6.43.3.23 UnpackAll()	296
6.43.3.24 UpdateMouseCursorWhenHover() [1/2]	296
6.43.3.25 UpdateMouseCursorWhenHover() [2/2]	296
6.43.4 Member Data Documentation	296
6.43.4.1 buttons	296
6.43.4.2 isHide	296
6.43.4.3 mChildren	296
6.43.4.4 mIsSelected	297
6.43.4.5 mPosition	297
6.43.4.6 mVisible	297
6.43.4.7 optionContainers	297
6.43.4.8 toggleButton	297
6.44 GUI::OptionInputField Class Reference	298
6.44.1 Member Typedef Documentation	300
6.44.1.1 Ptr	300
6.44.2 Constructor & Destructor Documentation	300
6.44.2.1 OptionInputField()	300
6.44.2.2 ~OptionInputField()	300
6.44.3 Member Function Documentation	300
6.44.3.1 AddInputField()	300
6.44.3.2 AddSubmitField()	300
6.44.3.3 deselect()	301
6.44.3.4 Draw()	301
6.44.3.5 DrawCurrent()	301
6.44.3.6 ExtractInput()	301
6.44.3.7 GetChildren()	301
6.44.3.8 GetHoverStatus() [1/2]	301

---

6.44.3.9 GetHoverStatus() [2/2] . . . . .	302
6.44.3.10 GetPosition() . . . . .	302
6.44.3.11 GetSize() . . . . .	302
6.44.3.12 GetVisible() . . . . .	302
6.44.3.13 HasInputField() . . . . .	302
6.44.3.14 IsSelectable() . . . . .	302
6.44.3.15 IsSelected() . . . . .	303
6.44.3.16 pack() . . . . .	303
6.44.3.17 select() . . . . .	303
6.44.3.18 SetNoFieldOption() . . . . .	303
6.44.3.19 SetOption() . . . . .	303
6.44.3.20 SetPosition() [1/2] . . . . .	303
6.44.3.21 SetPosition() [2/2] . . . . .	303
6.44.3.22 SetVisible() . . . . .	304
6.44.3.23 ToggleInputFields() . . . . .	304
6.44.3.24 ToggleVisible() . . . . .	304
6.44.3.25 UnpackAll() . . . . .	304
6.44.3.26 UpdateMouseCursorWhenHover() [1/2] . . . . .	304
6.44.3.27 UpdateMouseCursorWhenHover() [2/2] . . . . .	304
6.44.4 Member Data Documentation . . . . .	304
6.44.4.1 fonts . . . . .	305
6.44.4.2 hasInputField . . . . .	305
6.44.4.3 mChildren . . . . .	305
6.44.4.4 mInput . . . . .	305
6.44.4.5 mInputField . . . . .	305
6.44.4.6 mIsSelected . . . . .	305
6.44.4.7 mPosition . . . . .	305
6.44.4.8 mVisible . . . . .	306
6.45 StateStack::PendingChange Struct Reference . . . . .	306
6.45.1 Constructor & Destructor Documentation . . . . .	306
6.45.1.1 PendingChange() . . . . .	306
6.45.2 Member Data Documentation . . . . .	306
6.45.2.1 action . . . . .	306
6.45.2.2 stateID . . . . .	307
6.46 Algorithm::Queue Class Reference . . . . .	307
6.46.1 Constructor & Destructor Documentation . . . . .	309
6.46.1.1 Queue() [1/2] . . . . .	309
6.46.1.2 Queue() [2/2] . . . . .	310
6.46.1.3 ~Queue() . . . . .	310
6.46.2 Member Function Documentation . . . . .	310
6.46.2.1 ApplyInput() . . . . .	310
6.46.2.2 Dequeue() . . . . .	310

6.46.2.3 Empty() . . . . .	310
6.46.2.4 EmptyGenerator() . . . . .	310
6.46.2.5 Enqueue() . . . . .	310
6.46.2.6 EnqueueEmpty() . . . . .	311
6.46.2.7 GenerateAnimation() . . . . .	311
6.46.2.8 GenerateRelayoutAnimation() . . . . .	311
6.46.2.9 HighlightArrowFromCur() . . . . .	311
6.46.2.10 InitAction() . . . . .	311
6.46.2.11 PeekBack() . . . . .	311
6.46.2.12 PeekFront() . . . . .	312
6.46.2.13 Random() . . . . .	312
6.46.2.14 RandomFixedSize() . . . . .	312
6.46.2.15 RandomFixedSizeGenerator() . . . . .	312
6.46.2.16 RandomGenerator() . . . . .	312
6.46.2.17 ReadFromExternalFile() . . . . .	312
6.46.2.18 ReadFromFileGenerator() . . . . .	312
6.46.2.19 size() . . . . .	313
6.46.2.20 UserDefined() . . . . .	313
6.46.2.21 UserDefinedGenerator() . . . . .	313
6.46.3 Member Data Documentation . . . . .	313
6.46.3.1 animController . . . . .	313
6.46.3.2 codeHighlighter . . . . .	313
6.46.3.3 maxN . . . . .	313
6.46.3.4 visualizer . . . . .	314
6.47 QueueState Class Reference . . . . .	314
6.47.1 Member Typedef Documentation . . . . .	316
6.47.1.1 Ptr . . . . .	317
6.47.2 Constructor & Destructor Documentation . . . . .	317
6.47.2.1 QueueState() . . . . .	317
6.47.2.2 ~QueueState() . . . . .	317
6.47.3 Member Function Documentation . . . . .	317
6.47.3.1 AddDeleteOperation() . . . . .	317
6.47.3.2 AddInitializeOperation() . . . . .	317
6.47.3.3 AddInsertOperation() . . . . .	317
6.47.3.4 AddIntFieldOperationOption() . . . . .	318
6.47.3.5 AddNoFieldOperationOption() . . . . .	318
6.47.3.6 AddOperations() . . . . .	318
6.47.3.7 AddSearchOperation() . . . . .	318
6.47.3.8 AddStringFieldOption() . . . . .	318
6.47.3.9 AddUpdateOperation() . . . . .	318
6.47.3.10 ClearAction() . . . . .	319
6.47.3.11 ClearError() . . . . .	319

---

6.47.3.12 Draw()	319
6.47.3.13 DrawCurrentActionText()	319
6.47.3.14 DrawCurrentErrorText()	319
6.47.3.15 GetContext()	319
6.47.3.16 InitNavigationBar()	319
6.47.3.17 RequestStackClear()	320
6.47.3.18 RequestStackPop()	320
6.47.3.19 RequestStackPush()	320
6.47.3.20 SetCurrentAction()	320
6.47.3.21 SetCurrentError()	320
6.47.3.22 Success()	320
6.47.3.23 Update()	320
6.47.4 Member Data Documentation	321
6.47.4.1 activeDS	321
6.47.4.2 animController	321
6.47.4.3 codeHighlighter	321
6.47.4.4 footer	321
6.47.4.5 mContext	321
6.47.4.6 mCurrentAction	321
6.47.4.7 mCurrentError	321
6.47.4.8 mStack	322
6.47.4.9 navigation	322
6.47.4.10 operationList	322
6.47.4.11 queue	322
6.48 SceneNode Class Reference	322
6.48.1 Member Typedef Documentation	323
6.48.1.1 Ptr	323
6.48.2 Constructor & Destructor Documentation	323
6.48.2.1 SceneNode()	323
6.48.2.2 ~SceneNode()	323
6.48.3 Member Function Documentation	323
6.48.3.1 AttachChild()	323
6.48.3.2 DetachChild()	323
6.48.3.3 Draw()	324
6.48.3.4 DrawCurrent()	324
6.48.4 Member Data Documentation	324
6.48.4.1 mChildren	324
6.48.4.2 mParent	324
6.49 Settings Class Reference	324
6.49.1 Constructor & Destructor Documentation	326
6.49.1.1 Settings() [1/2]	326
6.49.1.2 Settings() [2/2]	326

---

6.49.1.3 ~Settings()	326
6.49.2 Member Function Documentation	326
6.49.2.1 getColor() [1/2]	326
6.49.2.2 getColor() [2/2]	326
6.49.2.3 getInstance()	327
6.49.2.4 LoadDefaultColors()	327
6.49.2.5 LoadFromFile()	327
6.49.2.6 operator=()	327
6.49.2.7 SaveToFile()	327
6.49.3 Member Data Documentation	327
6.49.3.1 mColors	327
6.50 SettingsState Class Reference	328
6.50.1 Member Typedef Documentation	329
6.50.1.1 Ptr	329
6.50.2 Constructor & Destructor Documentation	329
6.50.2.1 SettingsState()	329
6.50.2.2 ~SettingsState()	330
6.50.3 Member Function Documentation	330
6.50.3.1 Draw()	330
6.50.3.2 GetContext()	330
6.50.3.3 InitNavigationBar()	330
6.50.3.4 RequestStackClear()	330
6.50.3.5 RequestStackPop()	330
6.50.3.6 RequestStackPush()	330
6.50.3.7 Update()	331
6.50.4 Member Data Documentation	331
6.50.4.1 mContext	331
6.50.4.2 mStack	331
6.50.4.3 navigation	331
6.51 Algorithm::SinglyLinkedList Class Reference	332
6.51.1 Constructor & Destructor Documentation	333
6.51.1.1 SinglyLinkedList() [1/2]	334
6.51.1.2 SinglyLinkedList() [2/2]	334
6.51.1.3 ~SinglyLinkedList()	334
6.51.2 Member Function Documentation	334
6.51.2.1 ApplyInput()	334
6.51.2.2 DeleteHead()	334
6.51.2.3 DeleteMiddle()	334
6.51.2.4 DeleteTail()	335
6.51.2.5 Empty()	335
6.51.2.6 EmptyGenerator()	335
6.51.2.7 GenerateAnimation()	335

---

6.51.2.8 GenerateRelayoutAnimation()	335
6.51.2.9 HighlightArrowFromCur()	335
6.51.2.10 InitAction()	336
6.51.2.11 InsertAfterTail()	336
6.51.2.12 InsertHead()	336
6.51.2.13 InsertMiddle()	336
6.51.2.14 Random()	336
6.51.2.15 RandomFixedSize()	336
6.51.2.16 RandomFixedSizeGenerator()	336
6.51.2.17 RandomGenerator()	337
6.51.2.18 ReadFromExternalFile()	337
6.51.2.19 ReadFromFileGenerator()	337
6.51.2.20 ResetVisualizer()	337
6.51.2.21 Search()	337
6.51.2.22 size()	337
6.51.2.23 Update()	337
6.51.2.24 UserDefined()	338
6.51.2.25 UserDefinedGenerator()	338
6.51.3 Member Data Documentation	338
6.51.3.1 animController	338
6.51.3.2 codeHighlighter	338
6.51.3.3 maxN	338
6.51.3.4 visualizer	338
6.52 GUI::SinglyLinkedList Class Reference	339
6.52.1 Member Typedef Documentation	341
6.52.1.1 Ptr	341
6.52.2 Member Enumeration Documentation	341
6.52.2.1 ArrowType	341
6.52.2.2 Orientation	342
6.52.3 Constructor & Destructor Documentation	342
6.52.3.1 SinglyLinkedList() [1/2]	342
6.52.3.2 SinglyLinkedList() [2/2]	342
6.52.3.3 ~SinglyLinkedList()	342
6.52.4 Member Function Documentation	342
6.52.4.1 DeleteNode()	342
6.52.4.2 deselect()	343
6.52.4.3 Draw() [1/2]	343
6.52.4.4 Draw() [2/2]	343
6.52.4.5 DrawArrow()	343
6.52.4.6 DrawCurrent()	343
6.52.4.7 GenerateNode()	343
6.52.4.8 GetArrowType()	344

---

6.52.4.9 GetChildren()	344
6.52.4.10 GetHoverStatus() [1/2]	344
6.52.4.11 GetHoverStatus() [2/2]	344
6.52.4.12 GetList()	344
6.52.4.13 GetNodeDefaultPosition()	344
6.52.4.14 GetPosition()	344
6.52.4.15 GetShape()	345
6.52.4.16 GetSize()	345
6.52.4.17 GetVisible()	345
6.52.4.18 Import()	345
6.52.4.19 InsertNode()	345
6.52.4.20 isSelectable()	345
6.52.4.21 isSelected()	346
6.52.4.22 pack()	346
6.52.4.23 Relayout()	346
6.52.4.24 ResetArrow()	346
6.52.4.25 select()	346
6.52.4.26 SetArrowType()	346
6.52.4.27 SetOrientation()	346
6.52.4.28 SetPosition() [1/2]	347
6.52.4.29 SetPosition() [2/2]	347
6.52.4.30 SetShape()	347
6.52.4.31 SetShowHeadAndTail()	347
6.52.4.32 SetVisible()	347
6.52.4.33 size()	347
6.52.4.34 ToggleVisible()	347
6.52.4.35 UnpackAll()	348
6.52.4.36 UpdateMouseCursorWhenHover() [1/2]	348
6.52.4.37 UpdateMouseCursorWhenHover() [2/2]	348
6.52.5 Member Data Documentation	348
6.52.5.1 arrowState	348
6.52.5.2 fonts	348
6.52.5.3 list	348
6.52.5.4 mChildren	349
6.52.5.5 mDisplayHeadAndTail	349
6.52.5.6 mIsSelected	349
6.52.5.7 mNodeDistance	349
6.52.5.8 mOrientation	349
6.52.5.9 mPosition	349
6.52.5.10 mShape	349
6.52.5.11 mVisible	350
6.53 SLLState Class Reference	350

---

6.53.1 Member Typedef Documentation . . . . .	353
6.53.1.1 Ptr . . . . .	353
6.53.2 Constructor & Destructor Documentation . . . . .	353
6.53.2.1 SLLState() . . . . .	353
6.53.2.2 ~SLLState() . . . . .	353
6.53.3 Member Function Documentation . . . . .	353
6.53.3.1 AddDeleteOperation() . . . . .	353
6.53.3.2 AddInitializeOperation() . . . . .	353
6.53.3.3 AddInsertOperation() . . . . .	354
6.53.3.4 AddIntFieldOperationOption() . . . . .	354
6.53.3.5 AddNoFieldOperationOption() . . . . .	354
6.53.3.6 AddOperations() . . . . .	354
6.53.3.7 AddSearchOperation() . . . . .	354
6.53.3.8 AddStringFieldOption() . . . . .	354
6.53.3.9 AddUpdateOperation() . . . . .	355
6.53.3.10 ClearAction() . . . . .	355
6.53.3.11 ClearError() . . . . .	355
6.53.3.12 Draw() . . . . .	355
6.53.3.13 DrawCurrentActionText() . . . . .	355
6.53.3.14 DrawCurrentErrorText() . . . . .	355
6.53.3.15 GetContext() . . . . .	355
6.53.3.16 InitNavigationBar() . . . . .	356
6.53.3.17 RequestStackClear() . . . . .	356
6.53.3.18 RequestStackPop() . . . . .	356
6.53.3.19 RequestStackPush() . . . . .	356
6.53.3.20 SetCurrentAction() . . . . .	356
6.53.3.21 SetCurrentError() . . . . .	356
6.53.3.22 Success() . . . . .	356
6.53.3.23 Update() . . . . .	357
6.53.4 Member Data Documentation . . . . .	357
6.53.4.1 activeDS . . . . .	357
6.53.4.2 animController . . . . .	357
6.53.4.3 codeHighlighter . . . . .	357
6.53.4.4 footer . . . . .	357
6.53.4.5 mContext . . . . .	357
6.53.4.6 mCurrentAction . . . . .	358
6.53.4.7 mCurrentError . . . . .	358
6.53.4.8 mStack . . . . .	358
6.53.4.9 navigation . . . . .	358
6.53.4.10 operationList . . . . .	358
6.53.4.11 SLL . . . . .	358
6.54 Algorithm::Stack Class Reference . . . . .	359

---

6.54.1 Constructor & Destructor Documentation . . . . .	360
6.54.1.1 Stack() [1/2] . . . . .	361
6.54.1.2 Stack() [2/2] . . . . .	361
6.54.1.3 ~Stack() . . . . .	361
6.54.2 Member Function Documentation . . . . .	361
6.54.2.1 ApplyInput() . . . . .	361
6.54.2.2 Empty() . . . . .	361
6.54.2.3 EmptyGenerator() . . . . .	361
6.54.2.4 GenerateAnimation() . . . . .	362
6.54.2.5 GenerateRelayoutAnimation() . . . . .	362
6.54.2.6 HighlightArrowFromCur() . . . . .	362
6.54.2.7 InitAction() . . . . .	362
6.54.2.8 Peek() . . . . .	362
6.54.2.9 Pop() . . . . .	362
6.54.2.10 Push() . . . . .	363
6.54.2.11 Random() . . . . .	363
6.54.2.12 RandomFixedSize() . . . . .	363
6.54.2.13 RandomFixedSizeGenerator() . . . . .	363
6.54.2.14 RandomGenerator() . . . . .	363
6.54.2.15 ReadFromExternalFile() . . . . .	363
6.54.2.16 ReadFromFileGenerator() . . . . .	363
6.54.2.17 size() . . . . .	364
6.54.2.18 UserDefined() . . . . .	364
6.54.2.19 UserDefinedGenerator() . . . . .	364
6.54.3 Member Data Documentation . . . . .	364
6.54.3.1 animController . . . . .	364
6.54.3.2 codeHighlighter . . . . .	364
6.54.3.3 maxN . . . . .	364
6.54.3.4 mStackOrientation . . . . .	365
6.54.3.5 visualizer . . . . .	365
6.55 StackState Class Reference . . . . .	365
6.55.1 Member Typedef Documentation . . . . .	367
6.55.1.1 Ptr . . . . .	368
6.55.2 Constructor & Destructor Documentation . . . . .	368
6.55.2.1 StackState() . . . . .	368
6.55.2.2 ~StackState() . . . . .	368
6.55.3 Member Function Documentation . . . . .	368
6.55.3.1 AddDeleteOperation() . . . . .	368
6.55.3.2 AddInitializeOperation() . . . . .	368
6.55.3.3 AddInsertOperation() . . . . .	368
6.55.3.4 AddIntFieldOperationOption() . . . . .	369
6.55.3.5 AddNoFieldOperationOption() . . . . .	369

---

6.55.3.6 AddOperations() . . . . .	369
6.55.3.7 AddSearchOperation() . . . . .	369
6.55.3.8 AddStringFieldOption() . . . . .	369
6.55.3.9 AddUpdateOperation() . . . . .	369
6.55.3.10 ClearAction() . . . . .	370
6.55.3.11 ClearError() . . . . .	370
6.55.3.12 Draw() . . . . .	370
6.55.3.13 DrawCurrentActionText() . . . . .	370
6.55.3.14 DrawCurrentErrorText() . . . . .	370
6.55.3.15 GetContext() . . . . .	370
6.55.3.16 InitNavigationBar() . . . . .	370
6.55.3.17 RequestStackClear() . . . . .	371
6.55.3.18 RequestStackPop() . . . . .	371
6.55.3.19 RequestStackPush() . . . . .	371
6.55.3.20 SetCurrentAction() . . . . .	371
6.55.3.21 SetCurrentError() . . . . .	371
6.55.3.22 Success() . . . . .	371
6.55.3.23 Update() . . . . .	371
6.55.4 Member Data Documentation . . . . .	372
6.55.4.1 activeDS . . . . .	372
6.55.4.2 animController . . . . .	372
6.55.4.3 codeHighlighter . . . . .	372
6.55.4.4 footer . . . . .	372
6.55.4.5 mContext . . . . .	372
6.55.4.6 mCurrentAction . . . . .	372
6.55.4.7 mCurrentError . . . . .	372
6.55.4.8 mStack . . . . .	373
6.55.4.9 mStackAlgorithm . . . . .	373
6.55.4.10 navigation . . . . .	373
6.55.4.11 operationList . . . . .	373
6.56 State Class Reference . . . . .	373
6.56.1 Member Typedef Documentation . . . . .	374
6.56.1.1 Ptr . . . . .	374
6.56.2 Constructor & Destructor Documentation . . . . .	374
6.56.2.1 State() . . . . .	375
6.56.2.2 ~State() . . . . .	375
6.56.3 Member Function Documentation . . . . .	375
6.56.3.1 Draw() . . . . .	375
6.56.3.2 GetContext() . . . . .	375
6.56.3.3 InitNavigationBar() . . . . .	375
6.56.3.4 RequestStackClear() . . . . .	375
6.56.3.5 RequestStackPop() . . . . .	376

---

6.56.3.6 RequestStackPush()	376
6.56.3.7 Update()	376
6.56.4 Member Data Documentation	376
6.56.4.1 mContext	376
6.56.4.2 mStack	376
6.56.4.3 navigation	376
6.57 StateStack Class Reference	377
6.57.1 Member Enumeration Documentation	378
6.57.1.1 Action	378
6.57.2 Constructor & Destructor Documentation	378
6.57.2.1 StateStack()	378
6.57.3 Member Function Documentation	379
6.57.3.1 ApplyPendingChanges()	379
6.57.3.2 ClearStates()	379
6.57.3.3 createState()	379
6.57.3.4 Draw()	379
6.57.3.5 IsEmpty()	379
6.57.3.6 PopState()	379
6.57.3.7 PushState()	379
6.57.3.8 RegisterState()	380
6.57.3.9 Update()	380
6.57.4 Member Data Documentation	380
6.57.4.1 mContext	380
6.57.4.2 mFactories	380
6.57.4.3 mPendingList	380
6.57.4.4 mStack	380
6.58 Algorithm::StaticArray Class Reference	381
6.58.1 Constructor & Destructor Documentation	382
6.58.1.1 StaticArray() [1/2]	382
6.58.1.2 StaticArray() [2/2]	383
6.58.1.3 ~StaticArray()	383
6.58.2 Member Function Documentation	383
6.58.2.1 Access()	383
6.58.2.2 ApplyInput()	383
6.58.2.3 Empty()	383
6.58.2.4 EmptyGenerator()	383
6.58.2.5 GenerateAnimation()	384
6.58.2.6 GenerateRelayoutAnimation()	384
6.58.2.7 InitAction()	384
6.58.2.8 Random()	384
6.58.2.9 RandomFixedSize()	384
6.58.2.10 RandomFixedSizeGenerator()	384

---

6.58.2.11 RandomGenerator() . . . . .	385
6.58.2.12 ReadFromExternalFile() . . . . .	385
6.58.2.13 ReadFromFileGenerator() . . . . .	385
6.58.2.14 ResetVisualizer() . . . . .	385
6.58.2.15 Search() . . . . .	385
6.58.2.16 size() . . . . .	385
6.58.2.17 Update() . . . . .	385
6.58.2.18 UserDefined() . . . . .	386
6.58.2.19 UserDefinedGenerator() . . . . .	386
6.58.3 Member Data Documentation . . . . .	386
6.58.3.1 animController . . . . .	386
6.58.3.2 codeHighlighter . . . . .	386
6.58.3.3 maxN . . . . .	386
6.58.3.4 visualizer . . . . .	386
6.59 StaticArrayState Class Reference . . . . .	387
6.59.1 Member Typedef Documentation . . . . .	389
6.59.1.1 Ptr . . . . .	389
6.59.2 Constructor & Destructor Documentation . . . . .	389
6.59.2.1 StaticArrayState() . . . . .	389
6.59.2.2 ~StaticArrayState() . . . . .	389
6.59.3 Member Function Documentation . . . . .	389
6.59.3.1 AddAccessOperation() . . . . .	389
6.59.3.2 AddDeleteOperation() . . . . .	390
6.59.3.3 AddInitializeOperation() . . . . .	390
6.59.3.4 AddInsertOperation() . . . . .	390
6.59.3.5 AddIntFieldOperationOption() . . . . .	390
6.59.3.6 AddNoFieldOperationOption() . . . . .	390
6.59.3.7 AddOperations() . . . . .	390
6.59.3.8 AddSearchOperation() . . . . .	391
6.59.3.9 AddStringFieldOption() . . . . .	391
6.59.3.10 AddUpdateOperation() . . . . .	391
6.59.3.11 ClearAction() . . . . .	391
6.59.3.12 ClearError() . . . . .	391
6.59.3.13 Draw() . . . . .	391
6.59.3.14 DrawCurrentActionText() . . . . .	392
6.59.3.15 DrawCurrentErrorText() . . . . .	392
6.59.3.16 GetContext() . . . . .	392
6.59.3.17 InitNavigationBar() . . . . .	392
6.59.3.18 RequestStackClear() . . . . .	392
6.59.3.19 RequestStackPop() . . . . .	392
6.59.3.20 RequestStackPush() . . . . .	392
6.59.3.21 SetCurrentAction() . . . . .	393

6.59.3.22 SetCurrentError()	393
6.59.3.23 Success()	393
6.59.3.24 Update()	393
6.59.4 Member Data Documentation	393
6.59.4.1 activeDS	393
6.59.4.2 animController	393
6.59.4.3 codeHighlighter	394
6.59.4.4 footer	394
6.59.4.5 mContext	394
6.59.4.6 mCurrentAction	394
6.59.4.7 mCurrentError	394
6.59.4.8 mStack	394
6.59.4.9 mStaticArray	394
6.59.4.10 navigation	395
6.59.4.11 operationList	395
6.60 GUI::StringInputField Class Reference	395
6.60.1 Member Typedef Documentation	398
6.60.1.1 Ptr	398
6.60.2 Constructor & Destructor Documentation	398
6.60.2.1 StringInputField()	398
6.60.2.2 ~StringInputField()	398
6.60.3 Member Function Documentation	398
6.60.3.1 AllFieldDisableEdit()	398
6.60.3.2 deselect()	398
6.60.3.3 Draw()	399
6.60.3.4 DrawField()	399
6.60.3.5 ExtractValue()	399
6.60.3.6 GetEditMode()	399
6.60.3.7 GetHoverStatus() [1/2]	399
6.60.3.8 GetHoverStatus() [2/2]	399
6.60.3.9 GetLabel()	400
6.60.3.10 GetPosition()	400
6.60.3.11 GetSize()	400
6.60.3.12 GetVisible()	400
6.60.3.13 IsClicked()	400
6.60.3.14 isSelectable()	400
6.60.3.15 isSelected()	400
6.60.3.16 Randomize()	401
6.60.3.17 select()	401
6.60.3.18 SetEditMode()	401
6.60.3.19 SetInputFieldSize()	401
6.60.3.20 SetLabel()	401

---

6.60.3.21 SetLabelSize() . . . . .	401
6.60.3.22 SetPosition() [1/2] . . . . .	401
6.60.3.23 SetPosition() [2/2] . . . . .	402
6.60.3.24 SetVisible() . . . . .	402
6.60.3.25 ToggleVisible() . . . . .	402
6.60.3.26 UpdateMouseCursorWhenHover() [1/2] . . . . .	402
6.60.3.27 UpdateMouseCursorWhenHover() [2/2] . . . . .	402
6.60.4 Member Data Documentation . . . . .	402
6.60.4.1 content . . . . .	402
6.60.4.2 editMode . . . . .	403
6.60.4.3 extractedValue . . . . .	403
6.60.4.4 fields . . . . .	403
6.60.4.5 fonts . . . . .	403
6.60.4.6 inputFieldSize . . . . .	403
6.60.4.7 label . . . . .	403
6.60.4.8 labelFontSize . . . . .	403
6.60.4.9 mFieldIndex . . . . .	403
6.60.4.10 mIsSelected . . . . .	404
6.60.4.11 mMaxLength . . . . .	404
6.60.4.12 mPosition . . . . .	404
6.60.4.13 mVisible . . . . .	404
6.61 TextureHolder Class Reference . . . . .	404
6.61.1 Constructor & Destructor Documentation . . . . .	406
6.61.1.1 TextureHolder() . . . . .	406
6.61.1.2 ~TextureHolder() . . . . .	406
6.61.2 Member Function Documentation . . . . .	406
6.61.2.1 Get() [1/2] . . . . .	406
6.61.2.2 Get() [2/2] . . . . .	406
6.61.2.3 InsertResource() . . . . .	406
6.61.2.4 Load() [1/2] . . . . .	406
6.61.2.5 Load() [2/2] . . . . .	407
6.61.2.6 LoadFromImage() [1/2] . . . . .	407
6.61.2.7 LoadFromImage() [2/2] . . . . .	407
6.61.3 Member Data Documentation . . . . .	407
6.61.3.1 mTextureMap . . . . .	407
6.62 GUI::NavigationBar::TitleInfo Struct Reference . . . . .	407
6.62.1 Member Data Documentation . . . . .	408
6.62.1.1 abbrTitle . . . . .	408
6.62.1.2 stateID . . . . .	408
6.62.1.3 titleName . . . . .	408
<b>7 File Documentation</b> . . . . .	<b>409</b>

---

7.1 src/Algorithms/Algorithm.hpp File Reference . . . . .	409
7.2 Algorithm.hpp . . . . .	410
7.3 src/Algorithms/Array/DynamicArray.cpp File Reference . . . . .	413
7.4 src/Components/Visualization/DynamicArray.cpp File Reference . . . . .	413
7.5 src/Algorithms/Array/DynamicArray.hpp File Reference . . . . .	414
7.6 DynamicArray.hpp . . . . .	415
7.7 src/Components/Visualization/DynamicArray.hpp File Reference . . . . .	415
7.8 DynamicArray.hpp . . . . .	416
7.9 src/Algorithms/Array/StaticArray.cpp File Reference . . . . .	417
7.10 src/Algorithms/Array/StaticArray.hpp File Reference . . . . .	417
7.11 StaticArray.hpp . . . . .	418
7.12 src/Algorithms/LinkedList/CircularLinkedList.cpp File Reference . . . . .	419
7.12.1 Typedef Documentation . . . . .	419
7.12.1.1 ArrowType . . . . .	419
7.13 src/Components/Visualization/CircularLinkedList.cpp File Reference . . . . .	420
7.14 src/Algorithms/LinkedList/CircularLinkedList.hpp File Reference . . . . .	420
7.15 CircularLinkedList.hpp . . . . .	421
7.16 src/Components/Visualization/CircularLinkedList.hpp File Reference . . . . .	422
7.17 CircularLinkedList.hpp . . . . .	423
7.18 src/Algorithms/LinkedList/DoublyLinkedList.cpp File Reference . . . . .	423
7.18.1 Typedef Documentation . . . . .	424
7.18.1.1 ArrowType . . . . .	424
7.19 src/Components/Visualization/DoublyLinkedList.cpp File Reference . . . . .	424
7.20 src/Algorithms/LinkedList/DoublyLinkedList.hpp File Reference . . . . .	425
7.21 DoublyLinkedList.hpp . . . . .	426
7.22 src/Components/Visualization/DoublyLinkedList.hpp File Reference . . . . .	427
7.23 DoublyLinkedList.hpp . . . . .	427
7.24 src/Algorithms/LinkedList/Queue.cpp File Reference . . . . .	428
7.24.1 Typedef Documentation . . . . .	428
7.24.1.1 ArrowType . . . . .	429
7.25 src/Algorithms/LinkedList/Queue.hpp File Reference . . . . .	429
7.26 Queue.hpp . . . . .	430
7.27 src/Algorithms/LinkedList/SinglyLinkedList.cpp File Reference . . . . .	430
7.27.1 Typedef Documentation . . . . .	431
7.27.1.1 ArrowType . . . . .	431
7.28 src/Components/Visualization/SinglyLinkedList.cpp File Reference . . . . .	431
7.29 src/Algorithms/LinkedList/SinglyLinkedList.hpp File Reference . . . . .	431
7.30 SinglyLinkedList.hpp . . . . .	432
7.31 src/Components/Visualization/SinglyLinkedList.hpp File Reference . . . . .	433
7.32 SinglyLinkedList.hpp . . . . .	434
7.33 src/Algorithms/LinkedList/Stack.cpp File Reference . . . . .	434
7.34 src/Algorithms/LinkedList/Stack.hpp File Reference . . . . .	435

---

---

7.34.1 Typedef Documentation . . . . .	436
7.34.1.1 ArrowType . . . . .	436
7.34.1.2 Orientation . . . . .	436
7.35 Stack.hpp . . . . .	436
7.36 src/Animation/AnimationController.hpp File Reference . . . . .	437
7.36.1 Typedef Documentation . . . . .	438
7.36.1.1 CLLAnimationController . . . . .	438
7.36.1.2 DArrayAnimationController . . . . .	438
7.36.1.3 DLLAnimationController . . . . .	438
7.36.1.4 SLLAnimationController . . . . .	438
7.37 AnimationController.hpp . . . . .	438
7.38 src/Animation/AnimationFactory.cpp File Reference . . . . .	441
7.39 src/Animation/AnimationFactory.hpp File Reference . . . . .	442
7.40 AnimationFactory.hpp . . . . .	443
7.41 src/Animation/AnimationState.hpp File Reference . . . . .	443
7.41.1 Typedef Documentation . . . . .	444
7.41.1.1 CLLAnimation . . . . .	445
7.41.1.2 DArrayAnimation . . . . .	445
7.41.1.3 DLLAnimation . . . . .	445
7.41.1.4 SLLAnimation . . . . .	445
7.42 AnimationState.hpp . . . . .	445
7.43 src/Application.cpp File Reference . . . . .	447
7.43.1 Macro Definition Documentation . . . . .	448
7.43.1.1 RAYGUI_IMPLEMENTATION . . . . .	448
7.44 src/Application.hpp File Reference . . . . .	448
7.45 Application.hpp . . . . .	449
7.46 src/Component.cpp File Reference . . . . .	449
7.47 src/Component.hpp File Reference . . . . .	449
7.48 Component.hpp . . . . .	450
7.49 src/Components/Common/Button.cpp File Reference . . . . .	451
7.50 src/Components/Common/Button.hpp File Reference . . . . .	451
7.51 Button.hpp . . . . .	452
7.52 src/Components/Common/Card.cpp File Reference . . . . .	453
7.53 src/Components/Common/Card.hpp File Reference . . . . .	453
7.54 Card.hpp . . . . .	454
7.55 src/Components/Common/CodeHighlighter.cpp File Reference . . . . .	455
7.56 src/Components/Common/CodeHighlighter.hpp File Reference . . . . .	455
7.57 CodeHighlighter.hpp . . . . .	456
7.58 src/Components/Common/Footer.cpp File Reference . . . . .	457
7.59 src/Components/Common/Footer.hpp File Reference . . . . .	457
7.60 Footer.hpp . . . . .	458
7.61 src/Components/Common/InputField.cpp File Reference . . . . .	459

---

7.62 src/Components/Common/InputField.hpp File Reference . . . . .	459
7.63 InputField.hpp . . . . .	460
7.64 src/Components/Common/IntegerField.cpp File Reference . . . . .	461
7.65 src/Components/Common/IntegerField.hpp File Reference . . . . .	462
7.66 IntegerInputField.hpp . . . . .	462
7.67 src/Components/Common/NavigationBar.cpp File Reference . . . . .	463
7.68 src/Components/Common/NavigationBar.hpp File Reference . . . . .	463
7.69 NavigationBar.hpp . . . . .	464
7.70 src/Components/Common/OperationContainer.cpp File Reference . . . . .	465
7.71 src/Components/Common/OperationContainer.hpp File Reference . . . . .	465
7.72 OperationContainer.hpp . . . . .	466
7.73 src/Components/Common/OperationList.cpp File Reference . . . . .	466
7.74 src/Components/Common/OperationList.hpp File Reference . . . . .	467
7.75 OperationList.hpp . . . . .	467
7.76 src/Components/Common/OptionInputField.cpp File Reference . . . . .	468
7.77 src/Components/Common/OptionInputField.hpp File Reference . . . . .	468
7.78 OptionInputField.hpp . . . . .	469
7.79 src/Components/Common/StringInputField.cpp File Reference . . . . .	470
7.80 src/Components/Common/StringInputField.hpp File Reference . . . . .	470
7.81 StringInputField.hpp . . . . .	471
7.82 src/Components/Visualization/DataStructure.cpp File Reference . . . . .	471
7.83 src/Components/Visualization/DataStructure.hpp File Reference . . . . .	471
7.84 DataStructure.hpp . . . . .	472
7.85 src/Components/Visualization/LinkedList.cpp File Reference . . . . .	473
7.86 src/Components/Visualization/LinkedList.hpp File Reference . . . . .	473
7.87 LinkedList.hpp . . . . .	474
7.88 src/Components/Visualization/Node.cpp File Reference . . . . .	475
7.89 src/Components/Visualization/Node.hpp File Reference . . . . .	475
7.90 Node.hpp . . . . .	476
7.91 src/Core/Node.hpp File Reference . . . . .	477
7.92 Node.hpp . . . . .	478
7.93 src/Container.cpp File Reference . . . . .	478
7.94 src/Container.hpp File Reference . . . . .	478
7.95 Container.hpp . . . . .	479
7.96 src/Core/List.hpp File Reference . . . . .	480
7.97 List.hpp . . . . .	481
7.98 src/FontHolder.cpp File Reference . . . . .	486
7.99 src/FontHolder.hpp File Reference . . . . .	486
7.100 FontHolder.hpp . . . . .	487
7.101 src/Global.hpp File Reference . . . . .	487
7.102 Global.hpp . . . . .	488
7.103 src/Identifiers/CategoryIdentifiers.hpp File Reference . . . . .	488

---

7.104 CategoryIdentifiers.hpp . . . . .	488
7.105 src/Identifiers/CategoryInfo.cpp File Reference . . . . .	489
7.106 src/Identifiers/CategoryInfo.hpp File Reference . . . . .	489
7.107 CategoryInfo.hpp . . . . .	490
7.108 src/Identifiers/ColorThemelIdentifiers.hpp File Reference . . . . .	490
7.109 ColorThemelIdentifiers.hpp . . . . .	491
7.110 src/Identifiers/DSIdentifiers.hpp File Reference . . . . .	492
7.111 DSIdentifiers.hpp . . . . .	493
7.112 src/Identifiers/DSInfo.cpp File Reference . . . . .	493
7.113 src/Identifiers/DSInfo.hpp File Reference . . . . .	494
7.114 DSInfo.hpp . . . . .	494
7.115 src/Identifiers/ResourcelIdentifiers.hpp File Reference . . . . .	495
7.116 ResourcelIdentifiers.hpp . . . . .	495
7.117 src/Identifiers/StatelIdentifiers.hpp File Reference . . . . .	496
7.118 StatelIdentifiers.hpp . . . . .	496
7.119 src/Main.cpp File Reference . . . . .	497
7.119.1 Function Documentation . . . . .	497
7.119.1.1 main() . . . . .	497
7.120 src/NonCopyable.hpp File Reference . . . . .	497
7.121 NonCopyable.hpp . . . . .	498
7.122 src/SceneNode.cpp File Reference . . . . .	498
7.123 src/SceneNode.hpp File Reference . . . . .	498
7.124 SceneNode.hpp . . . . .	499
7.125 src/Settings.cpp File Reference . . . . .	500
7.126 src/Settings.hpp File Reference . . . . .	500
7.127 Settings.hpp . . . . .	501
7.128 src/State.cpp File Reference . . . . .	501
7.129 src/State.hpp File Reference . . . . .	502
7.130 State.hpp . . . . .	503
7.131 src/States/Array/ArrayTypeState.hpp File Reference . . . . .	503
7.132 ArrayTypeState.hpp . . . . .	504
7.133 src/States/Array/DynamicArrayTypeState.cpp File Reference . . . . .	507
7.134 src/States/Array/DynamicArrayTypeState.hpp File Reference . . . . .	508
7.135 DynamicArrayTypeState.hpp . . . . .	509
7.136 src/States/Array/StaticArrayTypeState.cpp File Reference . . . . .	509
7.137 src/States/Array/StaticArrayTypeState.hpp File Reference . . . . .	509
7.138 StaticArrayTypeState.hpp . . . . .	510
7.139 src/States/HomepageState.cpp File Reference . . . . .	511
7.140 src/States/HomepageState.hpp File Reference . . . . .	511
7.141 HomepageState.hpp . . . . .	512
7.142 src/States/LinkedList/CLLState.cpp File Reference . . . . .	512
7.143 src/States/LinkedList/CLLState.hpp File Reference . . . . .	512

---

7.144 CLLState.hpp . . . . .	513
7.145 src/States/LinkedList/DLLState.cpp File Reference . . . . .	514
7.146 src/States/LinkedList/DLLState.hpp File Reference . . . . .	514
7.147 DLLState.hpp . . . . .	515
7.148 src/States/LinkedList/LLState.hpp File Reference . . . . .	515
7.149 LLState.hpp . . . . .	516
7.150 src/States/LinkedList/QueueState.cpp File Reference . . . . .	519
7.151 src/States/LinkedList/QueueState.hpp File Reference . . . . .	519
7.152 QueueState.hpp . . . . .	520
7.153 src/States/LinkedList/SLLState.cpp File Reference . . . . .	521
7.154 src/States/LinkedList/SLLState.hpp File Reference . . . . .	521
7.155 SLLState.hpp . . . . .	522
7.156 src/States/LinkedList/StackState.cpp File Reference . . . . .	522
7.157 src/States/LinkedList/StackState.hpp File Reference . . . . .	523
7.158 StackState.hpp . . . . .	524
7.159 src/States/SettingsState.cpp File Reference . . . . .	524
7.160 src/States/SettingsState.hpp File Reference . . . . .	524
7.161 SettingsState.hpp . . . . .	525
7.162 src/StateStack.cpp File Reference . . . . .	525
7.163 src/StateStack.hpp File Reference . . . . .	526
7.164 StateStack.hpp . . . . .	526
7.165 src/TextureHolder.cpp File Reference . . . . .	527
7.166 src/TextureHolder.hpp File Reference . . . . .	527
7.167 TextureHolder.hpp . . . . .	528
7.168 src/Utils/Utils.cpp File Reference . . . . .	529
7.169 src/Utils/Utils.hpp File Reference . . . . .	529
7.170 Utils.hpp . . . . .	530
<b>Index</b>	<b>531</b>

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Algorithm	11
Animation	11
AnimationFactory	11
Category	14
ColorTheme	14
Core	16
DataStructures	17
Fonts	17
global	18
GUI	19
States	19
Textures	20
Utils	21



# Chapter 2

## Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Algorithm::Algorithm< GUIAlgorithm, AnimationState > . . . . .	23
Algorithm::Algorithm< GUI::CircularLinkedList, CLLAnimation > . . . . .	23
Algorithm::CircularLinkedList . . . . .	72
Algorithm::Algorithm< GUI::DoublyLinkedList, DLLAnimation > . . . . .	23
Algorithm::DoublyLinkedList . . . . .	140
Algorithm::Algorithm< GUI::DynamicArray, DArrayAnimation > . . . . .	23
Algorithm::DynamicArray . . . . .	162
Algorithm::StaticArray . . . . .	381
Algorithm::Algorithm< GUI::SinglyLinkedList, SLLAnimation > . . . . .	23
Algorithm::Queue . . . . .	307
Algorithm::SinglyLinkedList . . . . .	332
Algorithm::Stack . . . . .	359
Animation::AnimationController< T > . . . . .	27
Animation::AnimationController< AnimationState > . . . . .	27
Animation::AnimationController< CLLAnimation > . . . . .	27
Animation::AnimationController< DArrayAnimation > . . . . .	27
Animation::AnimationController< DLLAnimation > . . . . .	27
Animation::AnimationController< SLLAnimation > . . . . .	27
Animation::AnimationState< T > . . . . .	34
Application . . . . .	39
GUI::Component . . . . .	108
GUI::Button . . . . .	51
GUI::Card . . . . .	62
GUI::CodeHighlighter . . . . .	100
GUI::Container . . . . .	118
GUI::Footer< CLLAnimationController > . . . . .	190
GUI::Footer< DLLAnimationController > . . . . .	190
GUI::Footer< DArrayAnimationController > . . . . .	190
GUI::Footer< SLLAnimationController > . . . . .	190
GUI::DataStructure . . . . .	126
GUI::DynamicArray . . . . .	168
GUI::LinkedList . . . . .	230
GUI::CircularLinkedList . . . . .	79
GUI::DoublyLinkedList . . . . .	148

GUI::SinglyLinkedList . . . . .	339
GUI::Footer< T > . . . . .	190
GUI::OperationContainer . . . . .	283
GUI::OperationList . . . . .	289
GUI::OptionInputField . . . . .	298
GUI::InputField . . . . .	205
GUI::IntegerInputField . . . . .	216
GUI::StringInputField . . . . .	395
GUI::NavigationBar . . . . .	257
GUI::Node . . . . .	268
Core::List< T >::const_iterator . . . . .	113
State::Context . . . . .	124
CategoryInfo::Info . . . . .	202
DSInfo::Info . . . . .	203
ArrayState< T >::IntegerInput . . . . .	214
LLState< T >::IntegerInput . . . . .	215
Core::List< T >::iterator . . . . .	226
Core::List< T > . . . . .	241
Core::Node< T > . . . . .	266
NonCopyable< T > . . . . .	281
NonCopyable< CategoryInfo > . . . . .	281
CategoryInfo . . . . .	70
NonCopyable< DSInfo > . . . . .	281
DSInfo . . . . .	160
NonCopyable< FontHolder > . . . . .	281
FontHolder . . . . .	188
NonCopyable< Settings > . . . . .	281
Settings . . . . .	324
NonCopyable< State > . . . . .	281
State . . . . .	373
ArrayType< DArrayAnimationController > . . . . .	43
DynamicArrayType . . . . .	179
StaticArrayType . . . . .	387
LLState< CLLAnimationController > . . . . .	248
CLLState . . . . .	92
LLState< DLLAnimationController > . . . . .	248
DLLState . . . . .	132
LLState< SLLAnimationController > . . . . .	248
QueueState . . . . .	314
SLLState . . . . .	350
StackState . . . . .	365
ArrayType< T > . . . . .	43
HomepageState . . . . .	197
LLState< T > . . . . .	248
SettingsState . . . . .	328
NonCopyable< StateStack > . . . . .	281
StateStack . . . . .	377
NonCopyable< TextureHolder > . . . . .	281
TextureHolder . . . . .	404
StateStack::PendingChange . . . . .	306
SceneNode . . . . .	322
GUI::NavigationBar::TitleInfo . . . . .	407

# Chapter 3

## Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Algorithm::Algorithm< GUIAlgorithm, AnimationState >	23
Animation::AnimationController< T >	27
Animation::AnimationState< T >	34
Application	39
ArrayState< T >	43
GUI::Button	51
GUI::Card	62
CategoryInfo	70
Algorithm::CircularLinkedList	72
GUI::CircularLinkedList	79
CLLState	92
GUI::CodeHighlighter	100
GUI::Component	108
Core::List< T >::const_iterator	113
GUI::Container	118
State::Context	124
GUI::DataStructure	126
DLLState	132
Algorithm::DoublyLinkedList	140
GUI::DoublyLinkedList	148
DSInfo	160
Algorithm::DynamicArray	162
GUI::DynamicArray	168
DynamicArrayState	179
FontHolder	188
GUI::Footer< T >	190
HomepageState	197
CategoryInfo::Info	202
DSInfo::Info	203
GUI::InputField	205
ArrayState< T >::IntegerInput	214
LLState< T >::IntegerInput	215
GUI::IntegerInputField	216
Core::List< T >::iterator	226
GUI::LinkedList	230

Core::List< T >	241
LLState< T >	248
GUI::NavigationBar	257
Core::Node< T >	266
GUI::Node	268
NonCopyable< T >	281
GUI::OperationContainer	283
GUI::OperationList	289
GUI::OptionInputField	298
StateStack::PendingChange	306
Algorithm::Queue	307
QueueState	314
SceneNode	322
Settings	324
SettingsState	328
Algorithm::SinglyLinkedList	332
GUI::SinglyLinkedList	339
SLLState	350
Algorithm::Stack	359
StackState	365
State	373
StateStack	377
Algorithm::StaticArray	381
StaticArrayState	387
GUI::StringInputField	395
TextureHolder	404
GUI::NavigationBar::TitleInfo	407

# Chapter 4

## File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

src/Application.cpp	447
src/Application.hpp	448
src/Component.cpp	449
src/Component.hpp	449
src/Container.cpp	478
src/Container.hpp	478
src/FontHolder.cpp	486
src/FontHolder.hpp	486
src/Global.hpp	487
src/Main.cpp	497
src/NonCopyable.hpp	497
src/SceneNode.cpp	498
src/SceneNode.hpp	498
src/Settings.cpp	500
src/Settings.hpp	500
src/State.cpp	501
src/State.hpp	502
src/StateStack.cpp	525
src/StateStack.hpp	526
src/TextureHolder.cpp	527
src/TextureHolder.hpp	527
src/Algorithms/Algorithm.hpp	409
src/Algorithms/Array/DynamicArray.cpp	413
src/Algorithms/Array/DynamicArray.hpp	414
src/Algorithms/Array/StaticArray.cpp	417
src/Algorithms/Array/StaticArray.hpp	417
src/Algorithms/LinkedList/CircularLinkedList.cpp	419
src/Algorithms/LinkedList/CircularLinkedList.hpp	420
src/Algorithms/LinkedList/DoublyLinkedList.cpp	423
src/Algorithms/LinkedList/DoublyLinkedList.hpp	425
src/Algorithms/LinkedList/Queue.cpp	428
src/Algorithms/LinkedList/Queue.hpp	429
src/Algorithms/LinkedList/SinglyLinkedList.cpp	430
src/Algorithms/LinkedList/SinglyLinkedList.hpp	431
src/Algorithms/LinkedList/Stack.cpp	434

src/Algorithms/LinkedList/ <a href="#">Stack.hpp</a>	435
src/Animation/ <a href="#">AnimationController.hpp</a>	437
src/Animation/ <a href="#">AnimationFactory.cpp</a>	441
src/Animation/ <a href="#">AnimationFactory.hpp</a>	442
src/Animation/ <a href="#">AnimationState.hpp</a>	443
src/Components/Common/ <a href="#">Button.hpp</a>	451
src/Components/Common/ <a href="#">Button.cpp</a>	451
src/Components/Common/ <a href="#">Card.hpp</a>	453
src/Components/Common/ <a href="#">Card.cpp</a>	453
src/Components/Common/ <a href="#">CodeHighlighter.hpp</a>	455
src/Components/Common/ <a href="#">CodeHighlighter.cpp</a>	455
src/Components/Common/ <a href="#">Footer.hpp</a>	457
src/Components/Common/ <a href="#">Footer.cpp</a>	457
src/Components/Common/ <a href="#">InputField.hpp</a>	459
src/Components/Common/ <a href="#">InputField.cpp</a>	459
src/Components/Common/ <a href="#">IntegerField.hpp</a>	461
src/Components/Common/ <a href="#">IntegerField.cpp</a>	462
src/Components/Common/ <a href="#">NavigationBar.hpp</a>	463
src/Components/Common/ <a href="#">NavigationBar.cpp</a>	463
src/Components/Common/ <a href="#">OperationContainer.hpp</a>	465
src/Components/Common/ <a href="#">OperationContainer.cpp</a>	465
src/Components/Common/ <a href="#">OperationList.hpp</a>	466
src/Components/Common/ <a href="#">OperationList.cpp</a>	467
src/Components/Common/ <a href="#">OptionInputField.hpp</a>	468
src/Components/Common/ <a href="#">OptionInputField.cpp</a>	468
src/Components/Common/ <a href="#">StringInputField.hpp</a>	470
src/Components/Common/ <a href="#">StringInputField.cpp</a>	470
src/Components/Visualization/ <a href="#">CircularLinkedList.hpp</a>	420
src/Components/Visualization/ <a href="#">CircularLinkedList.cpp</a>	422
src/Components/Visualization/ <a href="#">DataStructure.hpp</a>	471
src/Components/Visualization/ <a href="#">DataStructure.cpp</a>	471
src/Components/Visualization/ <a href="#">DoublyLinkedList.hpp</a>	424
src/Components/Visualization/ <a href="#">DoublyLinkedList.cpp</a>	427
src/Components/Visualization/ <a href="#">DynamicArray.hpp</a>	413
src/Components/Visualization/ <a href="#">DynamicArray.cpp</a>	415
src/Components/Visualization/ <a href="#">LinkedList.hpp</a>	473
src/Components/Visualization/ <a href="#">LinkedList.cpp</a>	473
src/Components/Visualization/ <a href="#">Node.hpp</a>	475
src/Components/Visualization/ <a href="#">Node.cpp</a>	475
src/Components/Visualization/ <a href="#">SinglyLinkedList.hpp</a>	431
src/Components/Visualization/ <a href="#">SinglyLinkedList.cpp</a>	433
src/Core/ <a href="#">List.hpp</a>	480
src/Core/ <a href="#">Node.hpp</a>	477
src/Identifiers/ <a href="#">CategoryIdentifiers.hpp</a>	488
src/Identifiers/ <a href="#">CategoryInfo.cpp</a>	489
src/Identifiers/ <a href="#">CategoryInfo.hpp</a>	489
src/Identifiers/ <a href="#">ColorThemIdentifiers.hpp</a>	490
src/Identifiers/ <a href="#">DSIdentifiers.hpp</a>	492
src/Identifiers/ <a href="#">DSInfo.cpp</a>	493
src/Identifiers/ <a href="#">DSInfo.hpp</a>	494
src/Identifiers/ <a href="#">ResourceIdentifiers.hpp</a>	495
src/Identifiers/ <a href="#">StatelIdentifiers.hpp</a>	496
src/States/HomepageState.cpp	511
src/States/HomepageState.hpp	511
src/States/SettingsState.cpp	524
src/States/SettingsState.hpp	524
src/States/Array/ <a href="#">ArrayState.hpp</a>	503
src/States/Array/ <a href="#">DynamicArrayListState.cpp</a>	507

src/States/Array/ <a href="#">DynamicArrayState.hpp</a>	508
src/States/Array/ <a href="#">StaticArrayState.cpp</a>	509
src/States/Array/ <a href="#">StaticArrayState.hpp</a>	509
src/States/LinkedList/ <a href="#">CLLState.cpp</a>	512
src/States/LinkedList/ <a href="#">CLLState.hpp</a>	512
src/States/LinkedList/ <a href="#">DLLState.cpp</a>	514
src/States/LinkedList/ <a href="#">DLLState.hpp</a>	514
src/States/LinkedList/ <a href="#">LLState.hpp</a>	515
src/States/LinkedList/ <a href="#">QueueState.cpp</a>	519
src/States/LinkedList/ <a href="#">QueueState.hpp</a>	519
src/States/LinkedList/ <a href="#">SLLState.cpp</a>	521
src/States/LinkedList/ <a href="#">SLLState.hpp</a>	521
src/States/LinkedList/ <a href="#">StackState.cpp</a>	522
src/States/LinkedList/ <a href="#">StackState.hpp</a>	523
src/Utils/ <a href="#">Utils.cpp</a>	529
src/Utils/ <a href="#">Utils.hpp</a>	529



# Chapter 5

## Namespace Documentation

### 5.1 Algorithm Namespace Reference

#### Classes

- class [Algorithm](#)
- class [CircularLinkedList](#)
- class [DoublyLinkedList](#)
- class [DynamicArray](#)
- class [Queue](#)
- class [SinglyLinkedList](#)
- class [Stack](#)
- class [StaticArray](#)

### 5.2 Animation Namespace Reference

#### Classes

- class [AnimationController](#)
- class [AnimationState](#)

### 5.3 AnimationFactory Namespace Reference

#### Functions

- float [BounceOut](#) (float t)
- float [ElasticOut](#) (float t)
- void [DrawDirectionalArrow](#) (Vector2 start, Vector2 end, bool active, float t)
- void [DrawActiveArrow](#) (Vector2 start, Vector2 end, float t)
- void [DrawDoubleDirectionalArrow](#) (Vector2 start, Vector2 end, bool activeStart, bool activeEnd, float tStart, float tEnd)
- void [DrawDoubleActiveArrow](#) (Vector2 start, Vector2 end, float tStart, float tEnd)
- void [DrawCircularArrow](#) (Vector2 start, Vector2 end, bool active, float t)
- float [Dist](#) (Vector2 p1, Vector2 p2)
- Vector2 [InverseVector](#) (Vector2 vector)
- Vector2 [MoveNode](#) (Vector2 src, Vector2 dst, float t)
- Color [BlendColor](#) (Color src, Color dst, float t)
- void [ReCalculateEnds](#) (Vector2 &start, Vector2 &end, float radius, bool applyX=true, bool applyY=true)

### 5.3.1 Function Documentation

#### 5.3.1.1 BlendColor()

```
Color AnimationFactory::BlendColor (
    Color src,
    Color dst,
    float t )
```

#### 5.3.1.2 BounceOut()

```
float AnimationFactory::BounceOut (
    float t )
```

#### 5.3.1.3 Dist()

```
float AnimationFactory::Dist (
    Vector2 p1,
    Vector2 p2 )
```

#### 5.3.1.4 DrawActiveArrow()

```
void AnimationFactory::DrawActiveArrow (
    Vector2 start,
    Vector2 end,
    float t )
```

#### 5.3.1.5 DrawCircularArrow()

```
void AnimationFactory::DrawCircularArrow (
    Vector2 start,
    Vector2 end,
    bool active,
    float t )
```

**5.3.1.6 DrawDirectionalArrow()**

```
void AnimationFactory::DrawDirectionalArrow (
    Vector2 start,
    Vector2 end,
    bool active,
    float t )
```

**5.3.1.7 DrawDoubleActiveArrow()**

```
void AnimationFactory::DrawDoubleActiveArrow (
    Vector2 start,
    Vector2 end,
    float tStart,
    float tEnd )
```

**5.3.1.8 DrawDoubleDirectionalArrow()**

```
void AnimationFactory::DrawDoubleDirectionalArrow (
    Vector2 start,
    Vector2 end,
    bool activeStart,
    bool activeEnd,
    float tStart,
    float tEnd )
```

**5.3.1.9 ElasticOut()**

```
float AnimationFactory::ElasticOut (
    float t )
```

**5.3.1.10 InverseVector()**

```
Vector2 AnimationFactory::InverseVector (
    Vector2 vector )
```

### 5.3.1.11 MoveNode()

```
Vector2 AnimationFactory::MoveNode (
    Vector2 src,
    Vector2 dst,
    float t )
```

### 5.3.1.12 ReCalculateEnds()

```
void AnimationFactory::ReCalculateEnds (
    Vector2 & start,
    Vector2 & end,
    float radius,
    bool applyX = true,
    bool applyY = true )
```

## 5.4 Category Namespace Reference

### Enumerations

- enum [ID](#) { [None](#) , [Array](#) , [LinkedList](#) , [Count](#) }

### 5.4.1 Enumeration Type Documentation

#### 5.4.1.1 ID

[enum Category::ID](#)

##### Enumerator

<a href="#">None</a>	
<a href="#">Array</a>	
<a href="#">LinkedList</a>	
<a href="#">Count</a>	

## 5.5 ColorTheme Namespace Reference

### Enumerations

- enum [ID](#) {
 [Background](#) , [Logo1FirstPart](#) , [Logo1SecondPart](#) , [Logo2FirstPart](#) ,

```

Logo2SecondPart , NavigationBar_SelectedTitle , NavigationBar_UnselectedTitle , NavigationBar_Background
,
Card_Background , Card_Text , ActionList_Text , ActionList_Background ,
ActionList_HoverBackground , CodeHighlighter_Background , ActionDescription_Background , Visualizer_Label
,
Visualizer_Node_Default_Outline1 , Visualizer_Node_Default_Outline2 , Visualizer_Node_Default_Background1
, Visualizer_Node_Default_Background2 ,
Visualizer_Node_Default_Text1 , Visualizer_Node_Default_Text2 , Visualizer_Node_Active_Outline1 ,
Visualizer_Node_Active_Outline2 ,
Visualizer_Node_Active_Background1 , Visualizer_Node_Active_Background2 , Visualizer_Node_Active_Text1
, Visualizer_Node_Active_Text2 ,
Visualizer_Node_ActiveBlue_Outline1 , Visualizer_Node_ActiveBlue_Outline2 , Visualizer_Node_ActiveBlue_Background1
, Visualizer_Node_ActiveBlue_Background2 ,
Visualizer_Node_ActiveBlue_Text1 , Visualizer_Node_ActiveBlue_Text2 , Visualizer_Node_ActiveGreen_Outline1
, Visualizer_Node_ActiveGreen_Outline2 ,
Visualizer_Node_ActiveGreen_Background1 , Visualizer_Node_ActiveGreen_Background2 , Visualizer_Node_ActiveGreen_Text1
, Visualizer_Node_ActiveGreen_Text2 ,
Visualizer_Node_ActiveRed_Outline1 , Visualizer_Node_ActiveRed_Outline2 , Visualizer_Node_ActiveRed_Background1
, Visualizer_Node_ActiveRed_Background2 ,
Visualizer_Node_ActiveRed_Text1 , Visualizer_Node_ActiveRed_Text2 , Visualizer_Node_Iterated_Outline1
, Visualizer_Node_Iterated_Outline2 ,
Visualizer_Node_Iterated_Background1 , Visualizer_Node_Iterated_Background2 , Visualizer_Node_Iterated_Text1
, Visualizer_Node_Iterated_Text2 ,
Visualizer_Arrow_Default , Visualizer_Arrow_Active , Count }

```

## 5.5.1 Enumeration Type Documentation

### 5.5.1.1 ID

```
enum ColorTheme::ID
```

#### Enumerator

Background	
Logo1FirstPart	
Logo1SecondPart	
Logo2FirstPart	
Logo2SecondPart	
NavigationBar_SelectedTitle	
NavigationBar_UnselectedTitle	
NavigationBar_Background	
Card_Background	
Card_Text	
ActionList_Text	
ActionList_Background	
ActionList_HoverBackground	
CodeHighlighter_Background	
ActionDescription_Background	
Visualizer_Label	
Visualizer_Node_Default_Outline1	

## Enumerator

Visualizer_Node_Default_Outline2	
Visualizer_Node_Default_Background1	
Visualizer_Node_Default_Background2	
Visualizer_Node_Default_Text1	
Visualizer_Node_Default_Text2	
Visualizer_Node_Active_Outline1	
Visualizer_Node_Active_Outline2	
Visualizer_Node_Active_Background1	
Visualizer_Node_Active_Background2	
Visualizer_Node_Active_Text1	
Visualizer_Node_Active_Text2	
Visualizer_Node_ActiveBlue_Outline1	
Visualizer_Node_ActiveBlue_Outline2	
Visualizer_Node_ActiveBlue_Background1	
Visualizer_Node_ActiveBlue_Background2	
Visualizer_Node_ActiveBlue_Text1	
Visualizer_Node_ActiveBlue_Text2	
Visualizer_Node_ActiveGreen_Outline1	
Visualizer_Node_ActiveGreen_Outline2	
Visualizer_Node_ActiveGreen_Background1	
Visualizer_Node_ActiveGreen_Background2	
Visualizer_Node_ActiveGreen_Text1	
Visualizer_Node_ActiveGreen_Text2	
Visualizer_Node_ActiveRed_Outline1	
Visualizer_Node_ActiveRed_Outline2	
Visualizer_Node_ActiveRed_Background1	
Visualizer_Node_ActiveRed_Background2	
Visualizer_Node_ActiveRed_Text1	
Visualizer_Node_ActiveRed_Text2	
Visualizer_Node_Iterated_Outline1	
Visualizer_Node_Iterated_Outline2	
Visualizer_Node_Iterated_Background1	
Visualizer_Node_Iterated_Background2	
Visualizer_Node_Iterated_Text1	
Visualizer_Node_Iterated_Text2	
Visualizer_Arrow_Default	
Visualizer_Arrow_Active	
Count	

## 5.6 Core Namespace Reference

### Classes

- class [List](#)
- class [Node](#)

## 5.7 DataStructures Namespace Reference

### Enumerations

- enum ID {  
    None , StaticArray , DynamicArray , SinglyLinkedList ,  
    DoublyLinkedList , CircularLinkedList , Stack , Queue ,  
    Count }

#### 5.7.1 Enumeration Type Documentation

##### 5.7.1.1 ID

```
enum DataStructures::ID
```

###### Enumerator

None	
StaticArray	
DynamicArray	
SinglyLinkedList	
DoublyLinkedList	
CircularLinkedList	
Stack	
Queue	
Count	

## 5.8 Fonts Namespace Reference

### Enumerations

- enum ID {  
    Default , Default\_Italic , Default\_Bold , Silkscreen ,  
    Consolas , Courier , Courier\_Bold , Count }

#### 5.8.1 Enumeration Type Documentation

##### 5.8.1.1 ID

```
enum Fonts::ID
```

**Enumerator**

Default	
Default_Italic	
Default_Bold	
Silkscreen	
Consolas	
Courier	
Courier_Bold	
Count	

## 5.9 global Namespace Reference

### Variables

- `constexpr int SCREEN_WIDTH = 1300`
- `constexpr int SCREEN_HEIGHT = 800`
- `const std::string kTitle = "CS162 - VisuAlgo Clone"`
- `const std::string favicon = "assets/images/favicon.png"`
- `const std::string defaultColorPath = "assets/default-theme.dat"`

#### 5.9.1 Variable Documentation

##### 5.9.1.1 defaultColorPath

```
const std::string global::defaultColorPath = "assets/default-theme.dat"
```

##### 5.9.1.2 favicon

```
const std::string global::favicon = "assets/images/favicon.png"
```

##### 5.9.1.3 kTitle

```
const std::string global::kTitle = "CS162 - VisuAlgo Clone"
```

### 5.9.1.4 SCREEN\_HEIGHT

```
constexpr int global::SCREEN_HEIGHT = 800 [constexpr]
```

### 5.9.1.5 SCREEN\_WIDTH

```
constexpr int global::SCREEN_WIDTH = 1300 [constexpr]
```

## 5.10 GUI Namespace Reference

### Classes

- class [Button](#)
- class [Card](#)
- class [CircularLinkedList](#)
- class [CodeHighlighter](#)
- class [Component](#)
- class [Container](#)
- class [DataStructure](#)
- class [DoublyLinkedList](#)
- class [DynamicArray](#)
- class [Footer](#)
- class [InputField](#)
- class [IntegerInputField](#)
- class [LinkedList](#)
- class [NavigationBar](#)
- class [Node](#)
- class [OperationContainer](#)
- class [OperationList](#)
- class [OptionInputField](#)
- class [SinglyLinkedList](#)
- class [StringInputField](#)

## 5.11 States Namespace Reference

### Enumerations

- enum [ID](#) {  
    [None](#) , [Homepage](#) , [Settings](#) , [StaticArray](#) ,  
    [DynamicArray](#) , [SinglyLinkedList](#) , [DoublyLinkedList](#) , [CircularLinkedList](#) ,  
    [Stack](#) , [Queue](#) , [Count](#) }

### 5.11.1 Enumeration Type Documentation

#### 5.11.1.1 ID

```
enum States::ID
```

**Enumerator**

None	
Homepage	
Settings	
StaticArray	
DynamicArray	
SinglyLinkedList	
DoublyLinkedList	
CircularLinkedList	
Stack	
Queue	
Count	

## 5.12 Textures Namespace Reference

### Enumerations

- enum `ID` {
 `Blank` , `StaticArray` , `DynamicArray` , `SinglyLinkedList` ,
 `DoublyLinkedList` , `CircularLinkedList` , `Stack` , `Queue` ,
 `Favicon` , `Count` }

#### 5.12.1 Enumeration Type Documentation

##### 5.12.1.1 ID

```
enum Textures::ID
```

**Enumerator**

Blank	
StaticArray	
DynamicArray	
SinglyLinkedList	
DoublyLinkedList	
CircularLinkedList	
Stack	
Queue	
Favicon	
Count	

## 5.13 Utils Namespace Reference

### Functions

- std::string [OpenFileDialog](#) (std::string title, std::string description, std::vector< std::string > filters, std::string defaultPath, bool allowMultipleSelect)
- std::string [ReadInputFromFile](#) (std::string path)
- int [Rand](#) (int lower, int upper)
- void [DrawTextBoxed](#) (Font font, const char \*text, Rectangle rec, float fontSize, float spacing, bool wordWrap, Color tint)
- void [DrawTextBoxedSelectable](#) (Font font, const char \*text, Rectangle rec, float fontSize, float spacing, bool wordWrap, Color tint, int selectStart, int selectLength, Color selectTint, Color selectBackTint)
- bool [DrawIcon](#) (int iconID, int x, int y, int pixelSize, Color color, Color hoverColor)

#### 5.13.1 Function Documentation

##### 5.13.1.1 DrawIcon()

```
bool Utils::DrawIcon (
    int iconID,
    int x,
    int y,
    int pixelSize,
    Color color,
    Color hoverColor )
```

##### 5.13.1.2 DrawTextBoxed()

```
void Utils::DrawTextBoxed (
    Font font,
    const char * text,
    Rectangle rec,
    float fontSize,
    float spacing,
    bool wordWrap,
    Color tint )
```

### 5.13.1.3 DrawTextBboxedSelectable()

```
void Utils::DrawTextBboxedSelectable (
    Font font,
    const char * text,
    Rectangle rec,
    float fontSize,
    float spacing,
    bool wordWrap,
    Color tint,
    int selectStart,
    int selectLength,
    Color selectTint,
    Color selectBackTint )
```

### 5.13.1.4 OpenFileDialog()

```
std::string Utils::OpenFileDialog (
    std::string title,
    std::string description,
    std::vector< std::string > filters,
    std::string defaultPath,
    bool allowMultipleSelect )
```

### 5.13.1.5 Rand()

```
int Utils::Rand (
    int lower,
    int upper )
```

### 5.13.1.6 ReadInputFromFile()

```
std::string Utils::ReadInputFromFile (
    std::string path )
```

# Chapter 6

## Class Documentation

### 6.1 Algorithm::Algorithm< GUIAlgorithm, AnimationState > Class Template Reference

```
#include <Algorithm.hpp>
```

#### Public Member Functions

- `Algorithm (GUI::CodeHighlighter::Ptr codeHighlighter, typename Animation::AnimationController< AnimationState >::Ptr animController, FontHolder *fonts)`
- `Algorithm ()`
- `~Algorithm ()`
- `virtual void Empty ()`
- `virtual void Random ()`
- `virtual void RandomFixedSize (int N)`
- `virtual void UserDefined (std::string input)`
- `virtual void ReadFromExternalFile (std::string path)`
- `virtual void ApplyInput (std::vector< int > input, std::size_t nMaxSize=10)`

#### Protected Member Functions

- `std::vector< int > EmptyGenerator ()`
- `std::vector< int > RandomGenerator ()`
- `std::vector< int > RandomFixedSizeGenerator (int nSize)`
- `std::vector< int > UserDefinedGenerator (std::string input)`
- `std::vector< int > ReadFromFileGenerator (std::string inputFile)`
- `virtual void GenerateRelayoutAnimation (Vector2 newPosition)`
- `virtual AnimationState GenerateAnimation (float duration, int highlightLine, std::string actionDescription)`
- `virtual void InitAction (std::vector< std::string > code)`

#### Protected Attributes

- `GUIAlgorithm visualizer`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `Animation::AnimationController< AnimationState >::Ptr animController`

## 6.1.1 Constructor & Destructor Documentation

### 6.1.1.1 Algorithm() [1/2]

```
template<typename GUIAlgorithm , typename AnimationState >
Algorithm::Algorithm< GUIAlgorithm, AnimationState >::Algorithm (
    GUI::CodeHighlighter::Ptr codeHighlighter,
    typename Animation::AnimationController< AnimationState >::Ptr animController,
    FontHolder * fonts ) [inline]
```

### 6.1.1.2 Algorithm() [2/2]

```
template<typename GUIAlgorithm , typename AnimationState >
Algorithm::Algorithm< GUIAlgorithm, AnimationState >::Algorithm
```

### 6.1.1.3 ~Algorithm()

```
template<typename GUIAlgorithm , typename AnimationState >
Algorithm::Algorithm< GUIAlgorithm, AnimationState >::~Algorithm
```

## 6.1.2 Member Function Documentation

### 6.1.2.1 ApplyInput()

```
template<typename GUIAlgorithm , typename AnimationState >
void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::ApplyInput (
    std::vector< int > input,
    std::size_t nMaxSize = 10 ) [virtual]
```

Reimplemented in [Algorithm::StaticArray](#).

### 6.1.2.2 Empty()

```
template<typename GUIAlgorithm , typename AnimationState >
void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::Empty [virtual]
```

### 6.1.2.3 EmptyGenerator()

```
template<typename GUIAlgorithm , typename AnimationState >
std::vector< int > Algorithm::Algorithm< GUIAlgorithm, AnimationState >::EmptyGenerator [protected]
```

### 6.1.2.4 GenerateAnimation()

```
template<typename GUIAlgorithm , typename AnimationState >
AnimationState Algorithm::Algorithm< GUIAlgorithm, AnimationState >::GenerateAnimation (
    float duration,
    int highlightLine,
    std::string actionDescription ) [protected], [virtual]
```

### 6.1.2.5 GenerateRelayoutAnimation()

```
template<typename GUIAlgorithm , typename AnimationState >
void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::GenerateRelayoutAnimation (
    Vector2 newPosition ) [inline], [protected], [virtual]
```

### 6.1.2.6 InitAction()

```
template<typename GUIAlgorithm , typename AnimationState >
void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::InitAction (
    std::vector< std::string > code ) [protected], [virtual]
```

### 6.1.2.7 Random()

```
template<typename GUIAlgorithm , typename AnimationState >
void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::Random [virtual]
```

### 6.1.2.8 RandomFixedSize()

```
template<typename GUIAlgorithm , typename AnimationState >
void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::RandomFixedSize (
    int N ) [virtual]
```

### 6.1.2.9 RandomFixedSizeGenerator()

```
template<typename GUIAlgorithm , typename AnimationState >
std::vector< int > Algorithm::Algorithm< GUIAlgorithm, AnimationState >::RandomFixedSizeGenerator (
    int nSize ) [protected]
```

### 6.1.2.10 RandomGenerator()

```
template<typename GUIAlgorithm , typename AnimationState >
std::vector< int > Algorithm::Algorithm< GUIAlgorithm, AnimationState >::RandomGenerator
[protected]
```

### 6.1.2.11 ReadFromExternalFile()

```
template<typename GUIAlgorithm , typename AnimationState >
void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::ReadFromExternalFile (
    std::string path ) [virtual]
```

### 6.1.2.12 ReadFromFileGenerator()

```
template<typename GUIAlgorithm , typename AnimationState >
std::vector< int > Algorithm::Algorithm< GUIAlgorithm, AnimationState >::ReadFromFileGenerator (
    std::string inputFile ) [protected]
```

### 6.1.2.13 UserDefined()

```
template<typename GUIAlgorithm , typename AnimationState >
void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::UserDefined (
    std::string input ) [virtual]
```

### 6.1.2.14 UserDefinedGenerator()

```
template<typename GUIAlgorithm , typename AnimationState >
std::vector< int > Algorithm::Algorithm< GUIAlgorithm, AnimationState >::UserDefinedGenerator (
    std::string input ) [protected]
```

### 6.1.3 Member Data Documentation

#### 6.1.3.1 animController

```
template<typename GUIAlgorithm , typename AnimationState >
Animation::AnimationController<AnimationState>::Ptr Algorithm::Algorithm< GUIAlgorithm, AnimationState >::animController [protected]
```

#### 6.1.3.2 codeHighlighter

```
template<typename GUIAlgorithm , typename AnimationState >
GUI::CodeHighlighter::Ptr Algorithm::Algorithm< GUIAlgorithm, AnimationState >::codeHighlighter [protected]
```

#### 6.1.3.3 visualizer

```
template<typename GUIAlgorithm , typename AnimationState >
GUIAlgorithm Algorithm::Algorithm< GUIAlgorithm, AnimationState >::visualizer [protected]
```

The documentation for this class was generated from the following file:

- src/Algorithms/Algorithm.hpp

## 6.2 Animation::AnimationController< T > Class Template Reference

```
#include <AnimationController.hpp>
```

### Public Types

- `typedef std::shared_ptr< AnimationController > Ptr`

## Public Member Functions

- `AnimationController ()`
- `~AnimationController ()`
- `void RunAll ()`
- `void Reset ()`
- `void ResetCurrent ()`
- `void SetAnimation (std::size_t animationIndex)`
- `std::size_t CurrentAnimationIndex () const`
- `void AddAnimation (T animation)`
- `void PopAnimation ()`
- `void Clear ()`
- `float GetAnimationDuration ()`
- `float GetAnimateFrame (float dt) const`
- `std::size_t GetNumAnimation () const`
- `std::size_t GetAnimationIndex () const`
- `bool Done () const`
- `bool IsPlaying () const`
- `void StepForward ()`
- `void StepBackward ()`
- `void Pause ()`
- `void Continue ()`
- `void InteractionLock ()`
- `void InteractionAllow ()`
- `bool IsInteractionAllow () const`
- `void Update (float dt)`
- `void SetSpeed (float speed)`
- `float GetSpeed () const`
- `T GetAnimation ()`

## Private Member Functions

- `float GetStopDuration ()`

## Private Attributes

- `std::vector< T > animationGroup`
- `std::size_t mCurrentAnimationIndex`
- `float mSpeed`
- `bool Playing`
- `bool interactionLock`
- `float currStopDuration`

## Static Private Attributes

- `static constexpr float defaultSpeed = 1`
- `static constexpr float stopDuration = 0.25`

### 6.2.1 Member Typedef Documentation

### 6.2.1.1 Ptr

```
template<typename T = SLLAnimation>
typedef std::shared_ptr< AnimationController > Animation::AnimationController< T >::Ptr
```

## 6.2.2 Constructor & Destructor Documentation

### 6.2.2.1 AnimationController()

```
template<typename T >
Animation::AnimationController< T >::AnimationController
```

### 6.2.2.2 ~AnimationController()

```
template<typename T >
Animation::AnimationController< T >::~AnimationController
```

## 6.2.3 Member Function Documentation

### 6.2.3.1 AddAnimation()

```
template<typename T >
void Animation::AnimationController< T >::AddAnimation (
    T animation )
```

### 6.2.3.2 Clear()

```
template<typename T >
void Animation::AnimationController< T >::Clear [inline]
```

### 6.2.3.3 Continue()

```
template<typename T >
void Animation::AnimationController< T >::Continue
```

#### 6.2.3.4 CurrentAnimationIndex()

```
template<typename T >
std::size_t Animation::AnimationController< T >::CurrentAnimationIndex
```

#### 6.2.3.5 Done()

```
template<typename T >
bool Animation::AnimationController< T >::Done
```

#### 6.2.3.6 GetAnimateFrame()

```
template<typename T >
float Animation::AnimationController< T >::GetAnimateFrame (
    float dt ) const
```

#### 6.2.3.7 GetAnimation()

```
template<typename T >
T Animation::AnimationController< T >::GetAnimation
```

#### 6.2.3.8 GetAnimationDuration()

```
template<typename T >
float Animation::AnimationController< T >::GetAnimationDuration
```

#### 6.2.3.9 GetAnimationIndex()

```
template<typename T >
std::size_t Animation::AnimationController< T >::GetAnimationIndex
```

#### 6.2.3.10 GetNumAnimation()

```
template<typename T >
std::size_t Animation::AnimationController< T >::GetNumAnimation
```

### 6.2.3.11 GetSpeed()

```
template<typename T >
float Animation::AnimationController< T >::GetSpeed
```

### 6.2.3.12 GetStopDuration()

```
template<typename T >
float Animation::AnimationController< T >::GetStopDuration [inline], [private]
```

### 6.2.3.13 InteractionAllow()

```
template<typename T >
void Animation::AnimationController< T >::InteractionAllow [inline]
```

### 6.2.3.14 InteractionLock()

```
template<typename T >
void Animation::AnimationController< T >::InteractionLock [inline]
```

### 6.2.3.15 IsInteractionAllow()

```
template<typename T >
bool Animation::AnimationController< T >::IsInteractionAllow [inline]
```

### 6.2.3.16 IsPlaying()

```
template<typename T >
bool Animation::AnimationController< T >::IsPlaying
```

### 6.2.3.17 Pause()

```
template<typename T >
void Animation::AnimationController< T >::Pause
```

### 6.2.3.18 PopAnimation()

```
template<typename T >
void Animation::AnimationController< T >::PopAnimation
```

### 6.2.3.19 Reset()

```
template<typename T >
void Animation::AnimationController< T >::Reset
```

### 6.2.3.20 ResetCurrent()

```
template<typename T >
void Animation::AnimationController< T >::ResetCurrent
```

### 6.2.3.21 RunAll()

```
template<typename T >
void Animation::AnimationController< T >::RunAll
```

### 6.2.3.22 SetAnimation()

```
template<typename T >
void Animation::AnimationController< T >::SetAnimation (
    std::size_t animationIndex )
```

### 6.2.3.23 SetSpeed()

```
template<typename T >
void Animation::AnimationController< T >::SetSpeed (
    float speed )
```

### 6.2.3.24 StepBackward()

```
template<typename T >
void Animation::AnimationController< T >::StepBackward
```

### 6.2.3.25 StepForward()

```
template<typename T >
void Animation::AnimationController< T >::StepForward
```

### 6.2.3.26 Update()

```
template<typename T >
void Animation::AnimationController< T >::Update (
    float dt )
```

## 6.2.4 Member Data Documentation

### 6.2.4.1 animationGroup

```
template<typename T = SLLAnimation>
std::vector< T > Animation::AnimationController< T >::animationGroup [private]
```

### 6.2.4.2 currStopDuration

```
template<typename T = SLLAnimation>
float Animation::AnimationController< T >::currStopDuration [private]
```

### 6.2.4.3 defaultSpeed

```
template<typename T = SLLAnimation>
constexpr float Animation::AnimationController< T >::defaultSpeed = 1 [static], [constexpr],
[private]
```

### 6.2.4.4 interactionLock

```
template<typename T = SLLAnimation>
bool Animation::AnimationController< T >::interactionLock [private]
```

#### 6.2.4.5 mCurrentAnimationIndex

```
template<typename T = SLLAnimation>
std::size_t Animation::AnimationController< T >::mCurrentAnimationIndex [private]
```

#### 6.2.4.6 mSpeed

```
template<typename T = SLLAnimation>
float Animation::AnimationController< T >::mSpeed [private]
```

#### 6.2.4.7 Playing

```
template<typename T = SLLAnimation>
bool Animation::AnimationController< T >::Playing [private]
```

#### 6.2.4.8 stopDuration

```
template<typename T = SLLAnimation>
constexpr float Animation::AnimationController< T >::stopDuration = 0.25 [static], [constexpr],
[private]
```

The documentation for this class was generated from the following file:

- src/Animation/AnimationController.hpp

## 6.3 Animation::AnimationState< T > Class Template Reference

```
#include <AnimationState.hpp>
```

### Public Types

- `typedef std::shared_ptr< AnimationState< T > > Ptr`

## Public Member Functions

- `AnimationState ()`
- `~AnimationState ()`
- `void PlayingAt (float playingAt)`
- `float GetCurrentPlayingAt () const`
- `void Draw (Vector2 base=(Vector2){0, 0})`
- `void Update (float dt)`
- `void Reset ()`
- `void SetDuration (float duration)`
- `float GetDuration () const`
- `void SetAnimation (std::function< T(T, float, Vector2) > animation)`
- `void SetSourceDataStructure (T dataStructure)`
- `T GetDataStructure (float progress, Vector2 base=(Vector2){0, 0})`
- `bool Done () const`
- `void SetHighlightLine (int line)`
- `int GetHighlightedLine () const`
- `void SetActionDescription (std::string description)`
- `std::string GetActionDescription () const`

## Private Attributes

- `float mDuration`
- `float mCurrentPlayingAt`
- `int mHighlightedLine`
- `std::string actionDescription`
- `T mDataStructureBefore`
- `std::function< T(T, float, Vector2) > mAnimation`

### 6.3.1 Member Typedef Documentation

#### 6.3.1.1 Ptr

```
template<typename T >
typedef std::shared_ptr< AnimationState< T > > Animation::AnimationState< T >::Ptr
```

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 AnimationState()

```
template<typename T >
Animation::AnimationState< T >::AnimationState
```

### 6.3.2.2 ~AnimationState()

```
template<typename T >
Animation::AnimationState< T >::~AnimationState
```

## 6.3.3 Member Function Documentation

### 6.3.3.1 Done()

```
template<typename T >
bool Animation::AnimationState< T >::Done
```

### 6.3.3.2 Draw()

```
template<class T >
void Animation::AnimationState< T >::Draw (
    Vector2 base = (Vector2){0, 0} )
```

### 6.3.3.3 GetActionDescription()

```
template<typename T >
std::string Animation::AnimationState< T >::GetActionDescription [inline]
```

### 6.3.3.4 GetCurrentPlayingAt()

```
template<typename T >
float Animation::AnimationState< T >::GetCurrentPlayingAt
```

### 6.3.3.5 GetDataStructure()

```
template<typename T >
T Animation::AnimationState< T >::GetDataStructure (
    float progress,
    Vector2 base = (Vector2){0, 0} )
```

### 6.3.3.6 GetDuration()

```
template<typename T >
float Animation::AnimationState< T >::GetDuration
```

### 6.3.3.7 GetHighlightedLine()

```
template<typename T >
int Animation::AnimationState< T >::GetHighlightedLine
```

### 6.3.3.8 PlayingAt()

```
template<typename T >
void Animation::AnimationState< T >::PlayingAt (
    float playingAt )
```

### 6.3.3.9 Reset()

```
template<typename T >
void Animation::AnimationState< T >::Reset
```

### 6.3.3.10 SetActionDescription()

```
template<typename T >
void Animation::AnimationState< T >::SetActionDescription (
    std::string description ) [inline]
```

### 6.3.3.11 SetAnimation()

```
template<typename T >
void Animation::AnimationState< T >::SetAnimation (
    std::function< T(T, float, Vector2) > animation )
```

### 6.3.3.12 SetDuration()

```
template<typename T >
void Animation::AnimationState< T >::SetDuration (
    float duration )
```

### 6.3.3.13 SetHighlightLine()

```
template<typename T >
void Animation::AnimationState< T >::SetHighlightLine (
    int line )
```

### 6.3.3.14 SetSourceDataStructure()

```
template<typename T >
void Animation::AnimationState< T >::SetSourceDataStructure (
    T dataStructure )
```

### 6.3.3.15 Update()

```
template<typename T >
void Animation::AnimationState< T >::Update (
    float dt )
```

## 6.3.4 Member Data Documentation

### 6.3.4.1 actionDescription

```
template<typename T >
std::string Animation::AnimationState< T >::actionDescription [private]
```

### 6.3.4.2 mAnimation

```
template<typename T >
std::function< T(T, float, Vector2) > Animation::AnimationState< T >::mAnimation [private]
```

### 6.3.4.3 mCurrentPlayingAt

```
template<typename T >
float Animation::AnimationState< T >::mCurrentPlayingAt [private]
```

### 6.3.4.4 mDataStructureBefore

```
template<typename T >
T Animation::AnimationState< T >::mDataStructureBefore [private]
```

### 6.3.4.5 mDuration

```
template<typename T >
float Animation::AnimationState< T >::mDuration [private]
```

### 6.3.4.6 mHighlightedLine

```
template<typename T >
int Animation::AnimationState< T >::mHighlightedLine [private]
```

The documentation for this class was generated from the following file:

- src/Animation/AnimationState.hpp

## 6.4 Application Class Reference

```
#include <Application.hpp>
```

### Public Member Functions

- [Application \(\)](#)
- [~Application \(\)](#)
- [void Run \(\)](#)
- [void Close \(\)](#)
- [void Init \(\)](#)
- [bool WindowClosed \(\)](#)

## Private Member Functions

- void `Render ()`
- void `RegisterStates ()`
- void `LoadResources ()`
- void `Update (float dt)`

## Private Attributes

- bool `closed` = false
- `StateStack mStack`
- `FontHolder * fonts`
- `TextureHolder * images`
- `CategoryInfo * categories`
- `DSInfo * dsInfo`

### 6.4.1 Constructor & Destructor Documentation

#### 6.4.1.1 Application()

```
Application::Application ( )
```

#### 6.4.1.2 ~Application()

```
Application::~Application ( )
```

### 6.4.2 Member Function Documentation

#### 6.4.2.1 Close()

```
void Application::Close ( )
```

#### 6.4.2.2 Init()

```
void Application::Init ( )
```

**6.4.2.3 LoadResources()**

```
void Application::LoadResources ( ) [private]
```

**6.4.2.4 RegisterStates()**

```
void Application::RegisterStates ( ) [private]
```

**6.4.2.5 Render()**

```
void Application::Render ( ) [private]
```

**6.4.2.6 Run()**

```
void Application::Run ( )
```

**6.4.2.7 Update()**

```
void Application::Update ( float dt ) [private]
```

**6.4.2.8 WindowClosed()**

```
bool Application::WindowClosed ( )
```

**6.4.3 Member Data Documentation****6.4.3.1 categories**

[CategoryInfo\\*](#) Application::categories [private]

#### 6.4.3.2 closed

```
bool Application::closed = false [private]
```

#### 6.4.3.3 dsInfo

```
DSInfo* Application::dsInfo [private]
```

#### 6.4.3.4 fonts

```
FontHolder* Application::fonts [private]
```

#### 6.4.3.5 images

```
TextureHolder* Application::images [private]
```

#### 6.4.3.6 mStack

```
StateStack Application::mStack [private]
```

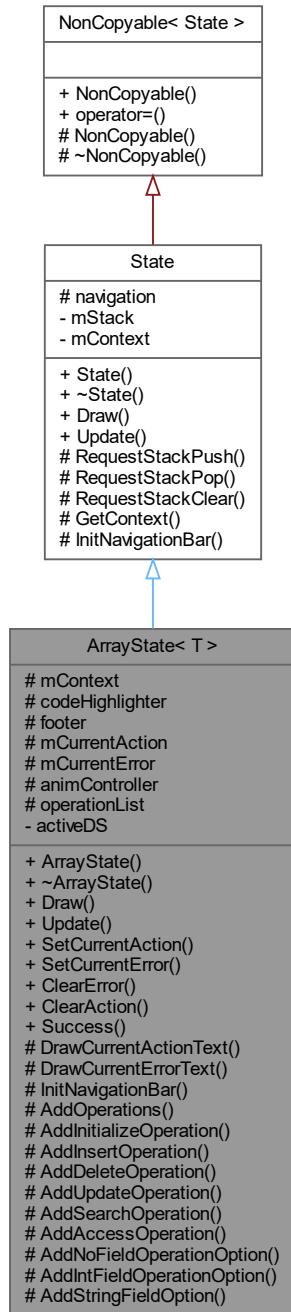
The documentation for this class was generated from the following files:

- src/[Application.hpp](#)
- src/[Application.cpp](#)

## 6.5 ArrayState< T > Class Template Reference

```
#include <ArrayState.hpp>
```

Inheritance diagram for ArrayState< T >:



## Classes

- struct [IntegerInput](#)

## Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `ArrayState (StateStack &stack, Context context, DataStructures::ID activeDS)`
- `~ArrayState ()`
- `virtual void Draw ()=0`
- `virtual bool Update (float dt)`
- `virtual void SetCurrentAction (std::string action)`
- `virtual void SetCurrentError (std::string error)`
- `virtual void ClearError ()`
- `virtual void ClearAction ()`
- `virtual void Success ()`

## Protected Member Functions

- `virtual void DrawCurrentActionText ()`
- `virtual void DrawCurrentErrorText ()`
- `void InitNavigationBar ()`
- `virtual void AddOperations ()`
- `virtual void AddInitializeOperation ()`
- `virtual void AddInsertOperation ()`
- `virtual void AddDeleteOperation ()`
- `virtual void AddUpdateOperation ()`
- `virtual void AddSearchOperation ()`
- `virtual void AddAccessOperation ()`
- `virtual void AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void() > action)`
- `virtual void AddIntegerFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >) > action)`
- `virtual void AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >) > action)`
- `void RequestStackPush (States::ID stateID)`
- `void RequestStackPop ()`
- `void RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< T > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Attributes

- `DataStructures::ID activeDS`
- `StateStack * mStack`

### 6.5.1 Member Typedef Documentation

#### 6.5.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

### 6.5.2 Constructor & Destructor Documentation

#### 6.5.2.1 `ArrayType()`

```
template<typename T >
ArrayType< T >::ArrayType (
    StateStack & stack,
    Context context,
    DataStructures::ID activeDS )
```

#### 6.5.2.2 `~ArrayType()`

```
template<typename T >
ArrayType< T >::~ArrayType
```

### 6.5.3 Member Function Documentation

#### 6.5.3.1 `AddAccessOperation()`

```
template<typename T >
void ArrayType< T >::AddAccessOperation [inline], [protected], [virtual]
```

Reimplemented in [DynamicArrayList](#), and [StaticArrayList](#).

### 6.5.3.2 AddDeleteOperation()

```
template<typename T >
void ArrayState< T >::AddDeleteOperation [protected], [virtual]
```

Reimplemented in [DynamicArrayList](#), [StaticArrayList](#).

### 6.5.3.3 AddInitializeOperation()

```
template<typename T >
void ArrayState< T >::AddInitializeOperation [protected], [virtual]
```

Reimplemented in [DynamicArrayList](#), and [StaticArrayList](#).

### 6.5.3.4 AddInsertOperation()

```
template<typename T >
void ArrayState< T >::AddInsertOperation [protected], [virtual]
```

Reimplemented in [DynamicArrayList](#).

### 6.5.3.5 AddIntFieldOperationOption()

```
template<typename T >
void ArrayState< T >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual]
```

### 6.5.3.6 AddNoFieldOperationOption()

```
template<typename T >
void ArrayState< T >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual]
```

### 6.5.3.7 `AddOperations()`

```
template<typename T >
void ArrayType< T >::AddOperations [protected], [virtual]
```

### 6.5.3.8 `AddSearchOperation()`

```
template<typename T >
void ArrayType< T >::AddSearchOperation [protected], [virtual]
```

Reimplemented in [DynamicArrayList](#), and [StaticArrayList](#).

### 6.5.3.9 `AddStringFieldOption()`

```
template<typename T >
void ArrayType< T >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual]
```

### 6.5.3.10 `AddUpdateOperation()`

```
template<typename T >
void ArrayType< T >::AddUpdateOperation [protected], [virtual]
```

Reimplemented in [DynamicArrayList](#), and [StaticArrayList](#).

### 6.5.3.11 `ClearAction()`

```
template<typename T >
void ArrayType< T >::ClearAction [inline], [virtual]
```

### 6.5.3.12 `ClearError()`

```
template<typename T >
void ArrayType< T >::ClearError [inline], [virtual]
```

### 6.5.3.13 Draw()

```
template<typename T >
virtual void ArrayState< T >::Draw ( ) [pure virtual]
```

Implements [State](#).

Implemented in [DynamicArrayList](#), and [StaticArrayList](#).

### 6.5.3.14 DrawCurrentActionText()

```
template<typename T >
void ArrayState< T >::DrawCurrentActionText [inline], [protected], [virtual]
```

### 6.5.3.15 DrawCurrentErrorText()

```
template<typename T >
void ArrayState< T >::DrawCurrentErrorText [inline], [protected], [virtual]
```

### 6.5.3.16 GetContext()

```
State::Context State::GetContext ( ) const [protected], [inherited]
```

### 6.5.3.17 InitNavigationBar()

```
template<typename T >
void ArrayState< T >::InitNavigationBar [protected]
```

### 6.5.3.18 RequestStackClear()

```
void State::RequestStackClear ( ) [protected], [inherited]
```

### 6.5.3.19 RequestStackPop()

```
void State::RequestStackPop ( ) [protected], [inherited]
```

### 6.5.3.20 `RequestStackPush()`

```
void State::RequestStackPush (
    States::ID stateID ) [protected], [inherited]
```

### 6.5.3.21 `SetCurrentAction()`

```
template<typename T >
void ArrayType< T >::SetCurrentAction (
    std::string action ) [inline], [virtual]
```

### 6.5.3.22 `SetCurrentError()`

```
template<typename T >
void ArrayType< T >::SetCurrentError (
    std::string error ) [inline], [virtual]
```

### 6.5.3.23 `Success()`

```
template<typename T >
void ArrayType< T >::Success [inline], [virtual]
```

### 6.5.3.24 `Update()`

```
template<typename T >
bool ArrayType< T >::Update (
    float dt ) [virtual]
```

Implements `State`.

## 6.5.4 Member Data Documentation

### 6.5.4.1 `activeDS`

```
template<typename T >
DataStructures::ID ArrayType< T >::activeDS [private]
```

#### 6.5.4.2 animController

```
template<typename T >
T::Ptr ArrayState< T >::animController [protected]
```

#### 6.5.4.3 codeHighlighter

```
template<typename T >
GUI::CodeHighlighter::Ptr ArrayState< T >::codeHighlighter [protected]
```

#### 6.5.4.4 footer

```
template<typename T >
GUI::Footer< T > ArrayState< T >::footer [protected]
```

#### 6.5.4.5 mContext

```
template<typename T >
Context ArrayState< T >::mContext [protected]
```

#### 6.5.4.6 mCurrentAction

```
template<typename T >
std::string ArrayState< T >::mCurrentAction [protected]
```

#### 6.5.4.7 mCurrentError

```
template<typename T >
std::string ArrayState< T >::mCurrentError [protected]
```

#### 6.5.4.8 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.5.4.9 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

#### 6.5.4.10 operationList

```
template<typename T >
GUI::OperationList ArrayState< T >::operationList [protected]
```

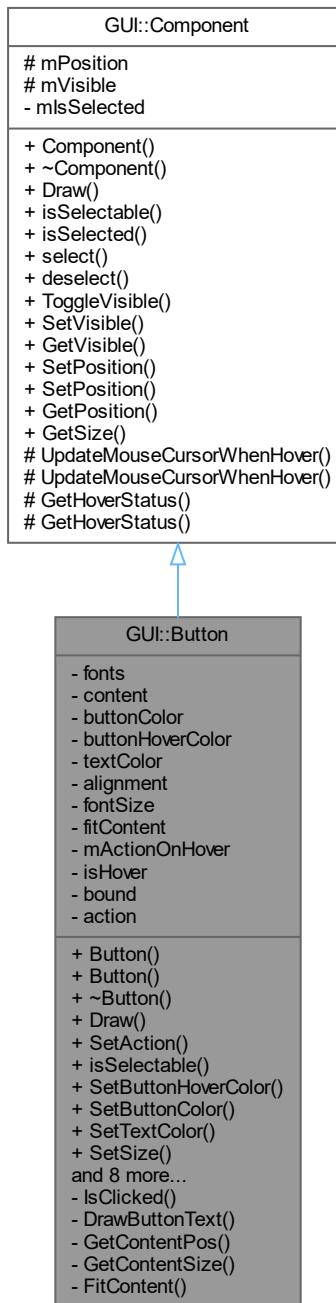
The documentation for this class was generated from the following file:

- src/States/Array/[ArrayState.hpp](#)

## 6.6 GUI::Button Class Reference

```
#include <Button.hpp>
```

Inheritance diagram for GUI::Button:



## Public Types

- enum `TextAlignment` { `Left` , `Center` , `Right` , `AlignmentCount` }
- typedef `std::shared_ptr<Button>` `Ptr`

## Public Member Functions

- `Button (std::string text, FontHolder *fonts)`

- `Button ()`
- `~Button ()`
- `void Draw (Vector2 base=(Vector2){0, 0})`
- `void SetAction (std::function< void() > clickedAction)`
- `bool IsSelectable () const`
- `void SetButtonHoverColor (Color color)`
- `void SetButtonColor (Color color)`
- `void SetTextColor (Color color)`
- `void SetSize (float width, float height)`
- `voidSetText (std::string text)`
- `void SetFontSize (float textFontSize)`
- `float GetFontSize () const`
- `void SetTextAlignment (TextAlignment textAlignment)`
- `void EnableFitContent ()`
- `void DisableFitContent ()`
- `Vector2 GetSize ()`
- `void SetActionOnHover (bool actionOnHover)`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2GetPosition ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `bool IsClicked ()`
- `void DrawButtonText ()`
- `Vector2 GetContentPos ()`
- `Vector2 GetContentSize ()`
- `void FitContent ()`

## Private Attributes

- `FontHolder * fonts`
- `std::string content`
- `Color buttonColor`
- `Color buttonHoverColor`
- `Color textColor`
- `TextAlignment alignment`
- `float fontSize`
- `bool fitContent`
- `bool mActionOnHover`
- `bool isHover`
- `Rectangle bound`
- `std::function< void() > action`
- `bool mIsSelected`

### 6.6.1 Member Typedef Documentation

#### 6.6.1.1 Ptr

```
typedef std::shared_ptr< Button > GUI::Button::Ptr
```

### 6.6.2 Member Enumeration Documentation

#### 6.6.2.1 TextAlignment

```
enum GUI::Button::TextAlignment
```

Enumerator

Left	
Center	
Right	
AlignmentCount	

### 6.6.3 Constructor & Destructor Documentation

### 6.6.3.1 Button() [1/2]

```
GUI::Button::Button (
    std::string text,
    FontHolder * fonts )
```

### 6.6.3.2 Button() [2/2]

```
GUI::Button::Button ( )
```

### 6.6.3.3 ~Button()

```
GUI::Button::~Button ( )
```

## 6.6.4 Member Function Documentation

### 6.6.4.1 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.6.4.2 DisableFitContent()

```
void GUI::Button::DisableFitContent ( )
```

### 6.6.4.3 Draw()

```
void GUI::Button::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Component](#).

#### 6.6.4.4 DrawButtonText()

```
void GUI::Button::DrawButtonText ( ) [private]
```

#### 6.6.4.5 EnableFitContent()

```
void GUI::Button::EnableFitContent ( )
```

#### 6.6.4.6 FitContent()

```
void GUI::Button::FitContent ( ) [private]
```

#### 6.6.4.7 GetContentPos()

```
Vector2 GUI::Button::GetContentPos ( ) [private]
```

#### 6.6.4.8 GetContentSize()

```
Vector2 GUI::Button::GetContentSize ( ) [private]
```

#### 6.6.4.9 GetFontSize()

```
float GUI::Button::GetFontSize ( ) const
```

#### 6.6.4.10 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

**6.6.4.11 GetHoverStatus() [2/2]**

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

**6.6.4.12 GetPosition()**

```
Vector2 GUI::Component::GetPosition () [inherited]
```

**6.6.4.13 GetSize()**

```
Vector2 GUI::Button::GetSize () [virtual]
```

Reimplemented from [GUI::Component](#).

**6.6.4.14 GetVisible()**

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

**6.6.4.15 IsClicked()**

```
bool GUI::Button::IsClicked () [private]
```

**6.6.4.16 isSelectable()**

```
bool GUI::Button::isSelectable () const [virtual]
```

Implements [GUI::Component](#).

**6.6.4.17 isSelected()**

```
bool GUI::Component::isSelected () const [inherited]
```

#### 6.6.4.18 **select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

#### 6.6.4.19 **SetAction()**

```
void GUI::Button::SetAction (
    std::function< void() > clickedAction )
```

#### 6.6.4.20 **SetActionOnHover()**

```
void GUI::Button::SetActionOnHover (
    bool actionOnHover )
```

#### 6.6.4.21 **SetButtonColor()**

```
void GUI::Button::SetButtonColor (
    Color color )
```

#### 6.6.4.22 **SetButtonHoverColor()**

```
void GUI::Button::SetButtonHoverColor (
    Color color )
```

#### 6.6.4.23 **SetFontSiz()**

```
void GUI::Button::SetFontSiz (
    float textFontSize )
```

#### 6.6.4.24 **SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.6.4.25 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position )  [inherited]
```

**6.6.4.26 SetSize()**

```
void GUI::Button::SetSize (
    float width,
    float height )
```

**6.6.4.27 SetText()**

```
void GUI::Button::SetText (
    std::string text )
```

**6.6.4.28 SetTextAlignment()**

```
void GUI::Button::SetTextAlignment (
    TextAlignment textAlignment )
```

**6.6.4.29 SetTextColor()**

```
void GUI::Button::SetTextColor (
    Color color )
```

**6.6.4.30 SetVisible()**

```
void GUI::Component::SetVisible (
    bool visible )  [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.6.4.31 **ToggleVisible()**

```
void GUI::Component::ToggleVisible () [virtual], [inherited]
```

#### 6.6.4.32 **UpdateMouseCursorWhenHover() [1/2]**

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.6.4.33 **UpdateMouseCursorWhenHover() [2/2]**

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.6.5 Member Data Documentation

#### 6.6.5.1 **action**

```
std::function< void() > GUI::Button::action [private]
```

#### 6.6.5.2 **alignment**

```
TextAlignment GUI::Button::alignment [private]
```

#### 6.6.5.3 **bound**

```
Rectangle GUI::Button::bound [private]
```

#### 6.6.5.4 buttonColor

```
Color GUI::Button::buttonColor [private]
```

#### 6.6.5.5 buttonHoverColor

```
Color GUI::Button::buttonHoverColor [private]
```

#### 6.6.5.6 content

```
std::string GUI::Button::content [private]
```

#### 6.6.5.7 fitContent

```
bool GUI::Button::fitContent [private]
```

#### 6.6.5.8 fonts

```
FontHolder* GUI::Button::fonts [private]
```

#### 6.6.5.9 fontSize

```
float GUI::Button::fontSize [private]
```

#### 6.6.5.10 isHover

```
bool GUI::Button::isHover [private]
```

#### 6.6.5.11 mActionOnHover

```
bool GUI::Button::mActionOnHover [private]
```

#### 6.6.5.12 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.6.5.13 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.6.5.14 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

#### 6.6.5.15 textColor

```
Color GUI::Button::textColor [private]
```

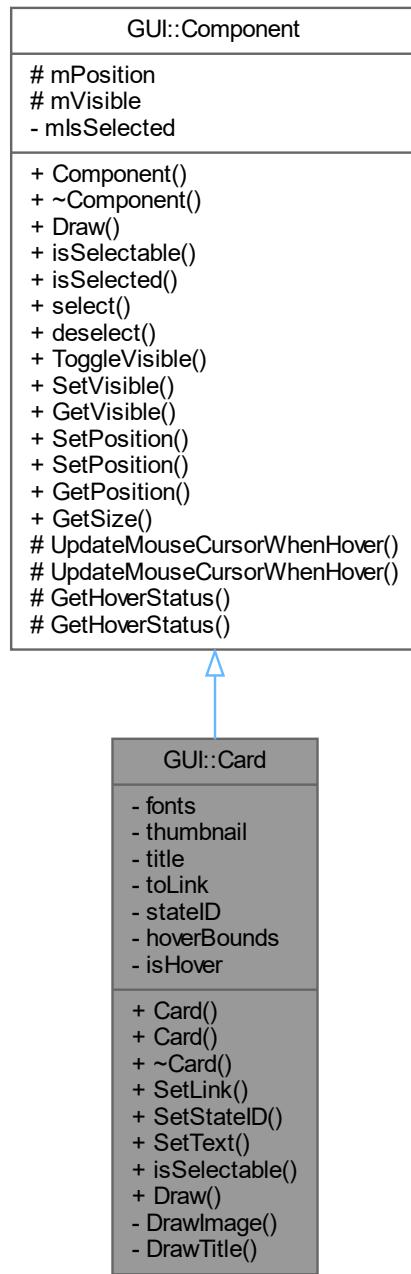
The documentation for this class was generated from the following files:

- src/Components/Common/[Button.hpp](#)
- src/Components/Common/[Button.cpp](#)

## 6.7 GUI::Card Class Reference

```
#include <Card.hpp>
```

Inheritance diagram for GUI::Card:



## Public Types

- `typedef std::shared_ptr< Component > Ptr`

## Public Member Functions

- `Card (std::string text, Texture thumbnail, FontHolder *fonts)`

- `Card ()`
- `~Card ()`
- `void SetLink (std::function< void(States::ID) > link)`
- `void SetStateID (States::ID stateID)`
- `void SetText (std::string text)`
- `bool IsSelectable () const`
- `void Draw (Vector2 base=(Vector2){0, 0})`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`
- `virtual Vector2 GetSize ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `bool DrawImage (Vector2 base)`
- `bool DrawTitle (Vector2 base)`

## Private Attributes

- `FontHolder * fonts`
- `Texture thumbnail`
- `std::string title`
- `std::function< void(States::ID) > toLink`
- `States::ID stateID`
- `std::map< std::string, Rectangle > hoverBounds`
- `bool isHover`
- `bool mIsSelected`

### 6.7.1 Member Typedef Documentation

### 6.7.1.1 Ptr

```
typedef std::shared_ptr< Component > GUI::Component::Ptr [inherited]
```

## 6.7.2 Constructor & Destructor Documentation

### 6.7.2.1 Card() [1/2]

```
GUI::Card::Card (
    std::string text,
    Texture thumbnail,
    FontHolder * fonts )
```

### 6.7.2.2 Card() [2/2]

```
GUI::Card::Card ( )
```

### 6.7.2.3 ~Card()

```
GUI::Card::~Card ( )
```

## 6.7.3 Member Function Documentation

### 6.7.3.1 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.7.3.2 Draw()

```
void GUI::Card::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Component](#).

### 6.7.3.3 DrawImage()

```
bool GUI::Card::DrawImage (
    Vector2 base ) [private]
```

### 6.7.3.4 DrawTitle()

```
bool GUI::Card::DrawTitle (
    Vector2 base ) [private]
```

### 6.7.3.5 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.7.3.6 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.7.3.7 GetPosition()

```
Vector2 GUI::Component::GetPosition ( ) [inherited]
```

### 6.7.3.8 GetSize()

```
Vector2 GUI::Component::GetSize ( ) [virtual], [inherited]
```

Reimplemented in [GUI::Button](#), [GUI::InputField](#), [GUI::OperationList](#), [GUI::OptionInputField](#), and [GUI::Container](#).

**6.7.3.9 GetVisible()**

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

**6.7.3.10 isSelectable()**

```
bool GUI::Card::isSelectable ( ) const [virtual]
```

Implements [GUI::Component](#).

**6.7.3.11 isSelected()**

```
bool GUI::Component::isSelected ( ) const [inherited]
```

**6.7.3.12 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.7.3.13 SetLink()**

```
void GUI::Card::SetLink (
    std::function< void(States::ID) > link )
```

**6.7.3.14 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.7.3.15 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

### 6.7.3.16 SetStateID()

```
void GUI::Card::SetStateID (
    States::ID stateID )
```

### 6.7.3.17 SetText()

```
void GUI::Card::SetText (
    std::string text )
```

### 6.7.3.18 SetVisible()

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.7.3.19 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

### 6.7.3.20 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.7.3.21 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.7.4 Member Data Documentation

#### 6.7.4.1 fonts

```
FontHolder* GUI::Card::fonts [private]
```

#### 6.7.4.2 hoverBounds

```
std::map< std::string, Rectangle > GUI::Card::hoverBounds [private]
```

#### 6.7.4.3 isHover

```
bool GUI::Card::isHover [private]
```

#### 6.7.4.4 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.7.4.5 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.7.4.6 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

#### 6.7.4.7 stateID

```
States::ID GUI::Card::stateID [private]
```

#### 6.7.4.8 thumbnail

```
Texture GUI::Card::thumbnail [private]
```

#### 6.7.4.9 title

```
std::string GUI::Card::title [private]
```

#### 6.7.4.10 toLink

```
std::function< void(States::ID) > GUI::Card::toLink [private]
```

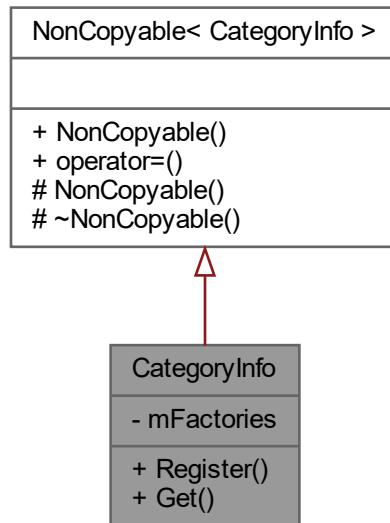
The documentation for this class was generated from the following files:

- src/Components/Common/[Card.hpp](#)
- src/Components/Common/[Card.cpp](#)

## 6.8 CategoryInfo Class Reference

```
#include <CategoryInfo.hpp>
```

Inheritance diagram for CategoryInfo:



## Classes

- struct [Info](#)

## Public Member Functions

- void [Register \(Category::ID, Info info\)](#)
- const [Info & Get \(Category::ID id\) const](#)

## Private Attributes

- std::map< [Category::ID, Info](#) > mFactories

### 6.8.1 Member Function Documentation

#### 6.8.1.1 Get()

```
const CategoryInfo::Info & CategoryInfo::Get (
    Category::ID id ) const
```

#### 6.8.1.2 Register()

```
void CategoryInfo::Register (
    Category::ID id,
    Info info )
```

### 6.8.2 Member Data Documentation

#### 6.8.2.1 mFactories

```
std::map< Category::ID, Info > CategoryInfo::mFactories [private]
```

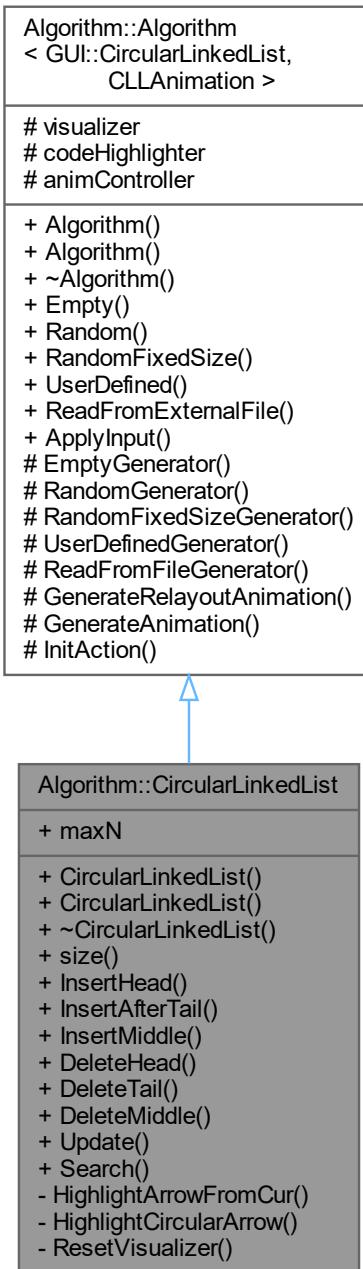
The documentation for this class was generated from the following files:

- src/Identifiers/[CategoryInfo.hpp](#)
- src/Identifiers/[CategoryInfo.cpp](#)

## 6.9 Algorithm::CircularLinkedList Class Reference

```
#include <CircularLinkedList.hpp>
```

Inheritance diagram for Algorithm::CircularLinkedList:



### Public Member Functions

- [CircularLinkedList \(\)](#)

- `CircularLinkedList (GUI::CodeHighlighter::Ptr codeHighlighter, CLLAnimationController::Ptr animController, FontHolder *fonts)`
- `~CircularLinkedList ()`
- `std::size_t size () const`
- `void InsertHead (int value)`
- `void InsertAfterTail (int value)`
- `void InsertMiddle (int index, int value)`
- `void DeleteHead ()`
- `void DeleteTail ()`
- `void DeleteMiddle (int index)`
- `void Update (int index, int value)`
- `void Search (int value)`
- `virtual void Empty ()`
- `virtual void Random ()`
- `virtual void RandomFixedSize (int N)`
- `virtual void UserDefined (std::string input)`
- `virtual void ReadFromExternalFile (std::string path)`
- `virtual void ApplyInput (std::vector< int > input, std::size_t nMaxSize=10)`

## Static Public Attributes

- `static constexpr int maxN = 10`

## Protected Member Functions

- `std::vector< int > EmptyGenerator ()`
- `std::vector< int > RandomGenerator ()`
- `std::vector< int > RandomFixedSizeGenerator (int nSize)`
- `std::vector< int > UserDefinedGenerator (std::string input)`
- `std::vector< int > ReadFromFileGenerator (std::string inputFile)`
- `virtual void GenerateRelayoutAnimation (Vector2 newPosition)`
- `virtual CLLAnimation GenerateAnimation (float duration, int highlightLine, std::string actionDescription)`
- `virtual void InitAction (std::vector< std::string > code)`

## Protected Attributes

- `GUI::CircularLinkedList visualizer`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `Animation::AnimationController< CLLAnimation >::Ptr animController`

## Private Member Functions

- `std::function< GUI::CircularLinkedList(GUI::CircularLinkedList, float, Vector2) > HighlightArrowFromCur (int index, bool drawVisualizer=true, bool reverse=false)`
- `std::function< GUI::CircularLinkedList(GUI::CircularLinkedList, float, Vector2) > HighlightCircularArrow (bool drawVisualizer=true, bool reverse=false)`
- `void ResetVisualizer ()`

## 6.9.1 Constructor & Destructor Documentation

### 6.9.1.1 CircularLinkedList() [1/2]

```
Algorithm::CircularLinkedList::CircularLinkedList ( )
```

### 6.9.1.2 CircularLinkedList() [2/2]

```
Algorithm::CircularLinkedList::CircularLinkedList (
    GUI::CodeHighlighter::Ptr codeHighlighter,
    CLLAnimationController::Ptr animController,
    FontHolder * fonts )
```

### 6.9.1.3 ~CircularLinkedList()

```
Algorithm::CircularLinkedList::~CircularLinkedList ( )
```

## 6.9.2 Member Function Documentation

### 6.9.2.1 ApplyInput()

```
void Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::ApplyInput (
    std::vector< int > input,
    std::size_t nMaxSize = 10 ) [virtual], [inherited]
```

### 6.9.2.2 DeleteHead()

```
void Algorithm::CircularLinkedList::DeleteHead ( )
```

### 6.9.2.3 DeleteMiddle()

```
void Algorithm::CircularLinkedList::DeleteMiddle (
    int index )
```

#### 6.9.2.4 DeleteTail()

```
void Algorithm::CircularLinkedList::DeleteTail ( )
```

#### 6.9.2.5 Empty()

```
void Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::Empty [virtual], [inherited]
```

#### 6.9.2.6 EmptyGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::Empty<->  
Generator [protected], [inherited]
```

#### 6.9.2.7 GenerateAnimation()

```
CLLAnimation Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::GenerateAnimation  
(  
    float duration,  
    int highlightLine,  
    std::string actionDescription ) [protected], [virtual], [inherited]
```

#### 6.9.2.8 GenerateRelayoutAnimation()

```
void Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::GenerateRelayoutAnimation  
(  
    Vector2 newPosition ) [inline], [protected], [virtual], [inherited]
```

#### 6.9.2.9 HighlightArrowFromCur()

```
std::function< GUI::CircularLinkedList(GUI::CircularLinkedList, float, Vector2) > Algorithm<->  
::CircularLinkedList::HighlightArrowFromCur (  
    int index,  
    bool drawVisualizer = true,  
    bool reverse = false ) [private]
```

### 6.9.2.10 **HighlightCircularArrow()**

```
std::function< GUI::CircularLinkedList(GUI::CircularLinkedList, float, Vector2) > Algorithm::CircularLinkedList::HighlightCircularArrow (
    bool drawVisualizer = true,
    bool reverse = false ) [private]
```

### 6.9.2.11 **InitAction()**

```
void Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::InitAction (
    std::vector< std::string > code ) [protected], [virtual], [inherited]
```

### 6.9.2.12 **InsertAfterTail()**

```
void Algorithm::CircularLinkedList::InsertAfterTail (
    int value )
```

### 6.9.2.13 **InsertHead()**

```
void Algorithm::CircularLinkedList::InsertHead (
    int value )
```

### 6.9.2.14 **InsertMiddle()**

```
void Algorithm::CircularLinkedList::InsertMiddle (
    int index,
    int value )
```

### 6.9.2.15 **Random()**

```
void Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::Random [virtual], [inherited]
```

### 6.9.2.16 **RandomFixedSize()**

```
void Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::RandomFixedSize (
    int N ) [virtual], [inherited]
```

**6.9.2.17 RandomFixedSizeGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::RandomFixedSizeModeGenerator ( int nSize ) [protected], [inherited]
```

**6.9.2.18 RandomGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::RandomGenerator [protected], [inherited]
```

**6.9.2.19 ReadFromExternalFile()**

```
void Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::ReadFromExternalFile ( std::string path ) [virtual], [inherited]
```

**6.9.2.20 ReadFromFileGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::ReadFromFileGenerator ( std::string inputFile ) [protected], [inherited]
```

**6.9.2.21 ResetVisualizer()**

```
void Algorithm::CircularLinkedList::ResetVisualizer ( ) [private]
```

**6.9.2.22 Search()**

```
void Algorithm::CircularLinkedList::Search ( int value )
```

**6.9.2.23 size()**

```
std::size_t Algorithm::CircularLinkedList::size ( ) const
```

### 6.9.2.24 Update()

```
void Algorithm::CircularLinkedList::Update (
    int index,
    int value )
```

### 6.9.2.25 UserDefined()

```
void Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::UserDefined (
    std::string input ) [virtual], [inherited]
```

### 6.9.2.26 UserDefinedGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::UserDefinedGenerator (
    std::string input ) [protected], [inherited]
```

## 6.9.3 Member Data Documentation

### 6.9.3.1 animController

```
Animation::AnimationController<CLLAnimation >::Ptr Algorithm::Algorithm< GUI::CircularLinkedList ,
, CLLAnimation >::animController [protected], [inherited]
```

### 6.9.3.2 codeHighlighter

```
GUI::CodeHighlighter::Ptr Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::codeHighlighter [protected], [inherited]
```

### 6.9.3.3 maxN

```
constexpr int Algorithm::CircularLinkedList::maxN = 10 [static], [constexpr]
```

### 6.9.3.4 visualizer

```
GUI::CircularLinkedList Algorithm::Algorithm< GUI::CircularLinkedList , CLLAnimation >::visualizer  
[protected], [inherited]
```

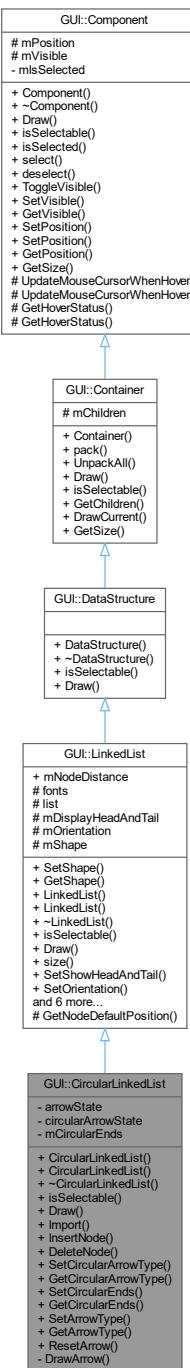
The documentation for this class was generated from the following files:

- src/Algorithms/LinkedList/CircularLinkedList.hpp
- src/Algorithms/LinkedList/CircularLinkedList.cpp

## 6.10 GUI::CircularLinkedList Class Reference

```
#include <CircularLinkedList.hpp>
```

Inheritance diagram for GUI::CircularLinkedList:



## Public Types

- enum **ArrowType** {
 Default , Hidden , Active , Skip ,
 ArrowTypeCount
 }
- enum **Orientation** { Horizontal , Vertical , OrientationCount }
- typedef std::shared\_ptr< **DataStructure** > **Ptr**

## Public Member Functions

- `CircularLinkedList ()`
- `CircularLinkedList (FontHolder *fonts)`
- `~CircularLinkedList ()`
- `bool IsSelectable () const`
- `void Draw (Vector2 base=(Vector2){0, 0}, float t=1.0f, bool init=false)`
- `void Import (std::vector< int > nodes)`
- `void InsertNode (std::size_t index, GUI::Node node, bool rePosition=true)`
- `void DeleteNode (std::size_t index, bool rePosition=true)`
- `void SetCircularArrowType (ArrowType type)`
- `ArrowType GetCircularArrowType (std::size_t index)`
- `void SetCircularEnds (std::size_t from, std::size_t to)`
- `std::pair< std::size_t, std::size_t > GetCircularEnds ()`
- `void SetArrowType (std::size_t index, ArrowType type)`
- `ArrowType GetArrowType (std::size_t index)`
- `void ResetArrow ()`
- `void SetShape (GUI::Node::Shape shape)`
- `GUI::Node::Shape GetShape () const`
- `virtual void Draw (Vector2 base)`
- `virtual std::size_t size () const`
- `virtual void SetShowHeadAndTail (bool show)`
- `virtual void SetOrientation (Orientation orientation)`
- `virtual std::vector< GUI::Node > & GetList ()`
- `virtual GUI::Node GenerateNode (int value)`
- `virtual void Relayout ()`
- `void pack (Component::Ptr component)`
- `void UnpackAll ()`
- `std::vector< Component::Ptr > GetChildren ()`
- `virtual void DrawCurrent (Vector2 base)`
- `virtual Vector2 GetSize ()`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Static Public Attributes

- `static constexpr float mNodeDistance = 20`

## Protected Member Functions

- `Vector2 GetNodeDefaultPosition (std::size_t index)`
- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `FontHolder * fonts`
- `std::vector< GUI::Node > list`
- `bool mDisplayHeadAndTail`
- `Orientation mOrientation = Orientation::Horizontal`
- `GUI::Node::Shape mShape = GUI::Node::Shape::Circle`
- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `void DrawArrow (Vector2 base, float t)`

## Private Attributes

- `std::vector< ArrowType > arrowState`
- `ArrowType circularArrowState`
- `std::pair< std::size_t, std::size_t > mCircularEnds`
- `bool mIsSelected`

### 6.10.1 Member Typedef Documentation

#### 6.10.1.1 Ptr

```
typedef std::shared_ptr< DataStructure > GUI::DataStructure::Ptr [inherited]
```

### 6.10.2 Member Enumeration Documentation

#### 6.10.2.1 ArrowType

```
enum GUI::LinkedList::ArrowType [inherited]
```

##### Enumerator

Default	
Hidden	
Active	
Skip	
ArrowTypeCount	

### 6.10.2.2 Orientation

```
enum GUI::LinkedList::Orientation [inherited]
```

Enumerator

Horizontal	
Vertical	
OrientationCount	

### 6.10.3 Constructor & Destructor Documentation

#### 6.10.3.1 CircularLinkedList() [1/2]

```
GUI::CircularLinkedList::CircularLinkedList ()
```

#### 6.10.3.2 CircularLinkedList() [2/2]

```
GUI::CircularLinkedList::CircularLinkedList (
    FontHolder * fonts )
```

#### 6.10.3.3 ~CircularLinkedList()

```
GUI::CircularLinkedList::~CircularLinkedList ()
```

### 6.10.4 Member Function Documentation

#### 6.10.4.1 DeleteNode()

```
void GUI::CircularLinkedList::DeleteNode (
    std::size_t index,
    bool rePosition = true ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

#### 6.10.4.2 **deselect()**

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

#### 6.10.4.3 **Draw() [1/2]**

```
void GUI::DataStructure::Draw (
    Vector2 base ) [virtual], [inherited]
```

Reimplemented from [GUI::Container](#).

#### 6.10.4.4 **Draw() [2/2]**

```
void GUI::CircularLinkedList::Draw (
    Vector2 base = (Vector2){0, 0},
    float t = 1.0f,
    bool init = false ) [virtual]
```

Implements [GUI::LinkedList](#).

#### 6.10.4.5 **DrawArrow()**

```
void GUI::CircularLinkedList::DrawArrow (
    Vector2 base,
    float t ) [private]
```

#### 6.10.4.6 **DrawCurrent()**

```
void GUI::Container::DrawCurrent (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.10.4.7 **GenerateNode()**

```
GUI::Node GUI::LinkedList::GenerateNode (
    int value ) [virtual], [inherited]
```

#### 6.10.4.8 GetArrowType()

```
GUI::LinkedList::ArrowType GUI::CircularLinkedList::GetArrowType (
    std::size_t index )
```

#### 6.10.4.9 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( ) [inherited]
```

#### 6.10.4.10 GetCircularArrowType()

```
GUI::LinkedList::ArrowType GUI::CircularLinkedList::GetCircularArrowType (
    std::size_t index )
```

#### 6.10.4.11 GetCircularEnds()

```
std::pair< std::size_t, std::size_t > GUI::CircularLinkedList::GetCircularEnds ( )
```

#### 6.10.4.12 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.10.4.13 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.10.4.14 GetList()

```
std::vector< GUI::Node > & GUI::LinkedList::GetList ( ) [virtual], [inherited]
```

#### 6.10.4.15 GetNodeDefaultPosition()

```
Vector2 GUI::LinkedList::GetNodeDefaultPosition (
    std::size_t index ) [protected], [inherited]
```

#### 6.10.4.16 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

#### 6.10.4.17 GetShape()

```
GUI::Node::Shape GUI::LinkedList::GetShape () const [inherited]
```

#### 6.10.4.18 GetSize()

```
Vector2 GUI::Container::GetSize () [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

#### 6.10.4.19 GetVisible()

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

#### 6.10.4.20 Import()

```
void GUI::CircularLinkedList::Import (
    std::vector< int > nodes ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

**6.10.4.21 InsertNode()**

```
void GUI::CircularLinkedList::InsertNode ( 
    std::size_t index,
    GUI::Node node,
    bool rePosition = true ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

**6.10.4.22 isSelectable()**

```
bool GUI::CircularLinkedList::isSelectable ( ) const [virtual]
```

Reimplemented from [GUI::DataStructure](#).

**6.10.4.23 isSelected()**

```
bool GUI::Component::isSelected ( ) const [inherited]
```

**6.10.4.24 pack()**

```
void GUI::Container::pack ( 
    Component::Ptr component ) [inherited]
```

**6.10.4.25 Relayout()**

```
void GUI::LinkedList::Relayout ( ) [virtual], [inherited]
```

**6.10.4.26 ResetArrow()**

```
void GUI::CircularLinkedList::ResetArrow ( )
```

**6.10.4.27 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.10.4.28 SetArrowType()**

```
void GUI::CircularLinkedList::SetArrowType (
    std::size_t index,
    ArrowType type )
```

**6.10.4.29 SetCircularArrowType()**

```
void GUI::CircularLinkedList::SetCircularArrowType (
    ArrowType type )
```

**6.10.4.30 SetCircularEnds()**

```
void GUI::CircularLinkedList::SetCircularEnds (
    std::size_t from,
    std::size_t to )
```

**6.10.4.31 SetOrientation()**

```
void GUI::LinkedList::SetOrientation (
    Orientation orientation ) [virtual], [inherited]
```

**6.10.4.32 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.10.4.33 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

**6.10.4.34 SetShape()**

```
void GUI::LinkedList::SetShape (
    GUI::Node::Shape shape ) [inherited]
```

**6.10.4.35 SetShowHeadAndTail()**

```
void GUI::LinkedList::SetShowHeadAndTail (
    bool show ) [virtual], [inherited]
```

**6.10.4.36 SetVisible()**

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

**6.10.4.37 size()**

```
std::size_t GUI::LinkedList::size () const [virtual], [inherited]
```

**6.10.4.38 ToggleVisible()**

```
void GUI::Component::ToggleVisible () [virtual], [inherited]
```

**6.10.4.39 UnpackAll()**

```
void GUI::Container::UnpackAll () [inherited]
```

**6.10.4.40 UpdateMouseCursorWhenHover() [1/2]**

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.10.4.41 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.10.5 Member Data Documentation

#### 6.10.5.1 arrowState

```
std::vector< ArrowType > GUI::CircularLinkedList::arrowState [private]
```

#### 6.10.5.2 circularArrowState

```
ArrowType GUI::CircularLinkedList::circularArrowState [private]
```

#### 6.10.5.3 fonts

```
FontHolder* GUI::LinkedList::fonts [protected], [inherited]
```

#### 6.10.5.4 list

```
std::vector< GUI::Node > GUI::LinkedList::list [protected], [inherited]
```

#### 6.10.5.5 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

#### 6.10.5.6 mCircularEnds

```
std::pair< std::size_t, std::size_t > GUI::CircularLinkedList::mCircularEnds [private]
```

### 6.10.5.7 mDisplayHeadAndTail

```
bool GUI::LinkedList::mDisplayHeadAndTail [protected], [inherited]
```

### 6.10.5.8 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

### 6.10.5.9 mNodeDistance

```
constexpr float GUI::LinkedList::mNodeDistance = 20 [static], [constexpr], [inherited]
```

### 6.10.5.10 mOrientation

```
Orientation GUI::LinkedList::mOrientation = Orientation::Horizontal [protected], [inherited]
```

### 6.10.5.11 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

### 6.10.5.12 mShape

```
GUI::Node::Shape GUI::LinkedList::mShape = GUI::Node::Shape::Circle [protected], [inherited]
```

### 6.10.5.13 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

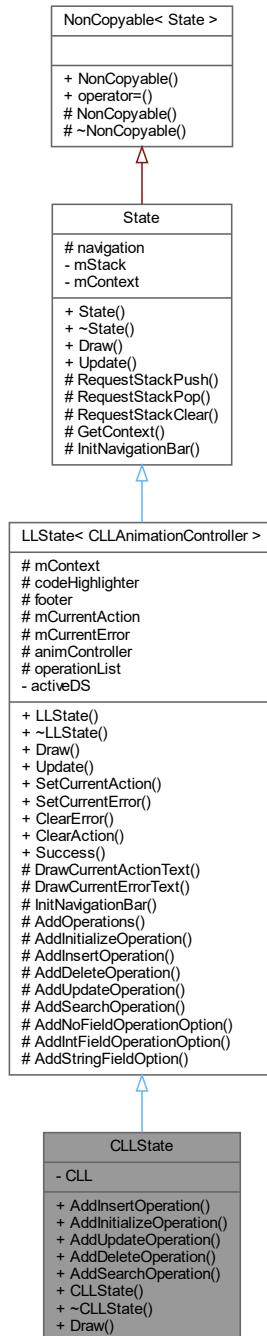
The documentation for this class was generated from the following files:

- src/Components/Visualization/CircularLinkedList.hpp
- src/Components/Visualization/CircularLinkedList.cpp

## 6.11 CLLState Class Reference

```
#include <CLLState.hpp>
```

Inheritance diagram for CLLState:



### Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- void `AddInsertOperation ()`
- void `AddInitializeOperation ()`
- void `AddUpdateOperation ()`
- void `AddDeleteOperation ()`
- void `AddSearchOperation ()`
- `CLLState (StateStack &stack, Context context)`
- `~CLLState ()`
- void `Draw ()`
- virtual bool `Update (float dt)`
- virtual void `SetCurrentAction (std::string action)`
- virtual void `SetCurrentError (std::string error)`
- virtual void `ClearError ()`
- virtual void `ClearAction ()`
- virtual void `Success ()`

## Protected Member Functions

- virtual void `DrawCurrentActionText ()`
- virtual void `DrawCurrentErrorText ()`
- void `InitNavigationBar ()`
- virtual void `AddOperations ()`
- virtual void `AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void() > action)`
- virtual void `AddIntFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >) > action)`
- virtual void `AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >) > action)`
- void `RequestStackPush (States::ID stateID)`
- void `RequestStackPop ()`
- void `RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< CLLAnimationController > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Attributes

- `Algorithm::CircularLinkedList CLL`
- `DataStructures::ID activeDS`
- `StateStack * mStack`

### 6.11.1 Member Typedef Documentation

#### 6.11.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

### 6.11.2 Constructor & Destructor Documentation

#### 6.11.2.1 CLLState()

```
CLLState::CLLState (
    StateStack & stack,
    Context context )
```

#### 6.11.2.2 ~CLLState()

```
CLLState::~CLLState ( )
```

### 6.11.3 Member Function Documentation

#### 6.11.3.1 AddDeleteOperation()

```
void CLLState::AddDeleteOperation ( ) [virtual]
```

Reimplemented from [LLState< CLLAnimationController >](#).

#### 6.11.3.2 AddInitializeOperation()

```
void CLLState::AddInitializeOperation ( ) [virtual]
```

Reimplemented from [LLState< CLLAnimationController >](#).

### 6.11.3.3 AddInsertOperation()

```
void CLLState::AddInsertOperation ( ) [virtual]
```

Reimplemented from [LLState< CLLAnimationController >](#).

### 6.11.3.4 AddIntFieldOperationOption()

```
void LLState< CLLAnimationController >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

### 6.11.3.5 AddNoFieldOperationOption()

```
void LLState< CLLAnimationController >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual], [inherited]
```

### 6.11.3.6 AddOperations()

```
void LLState< CLLAnimationController >::AddOperations [protected], [virtual], [inherited]
```

### 6.11.3.7 AddSearchOperation()

```
void CLLState::AddSearchOperation ( ) [virtual]
```

Reimplemented from [LLState< CLLAnimationController >](#).

### 6.11.3.8 AddStringFieldOption()

```
void LLState< CLLAnimationController >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

### 6.11.3.9 AddUpdateOperation()

```
void CLLState::AddUpdateOperation ( ) [virtual]
```

Reimplemented from [LLState< CLLAnimationController >](#).

### 6.11.3.10 ClearAction()

```
void LLState< CLLAnimationController >::ClearAction [inline], [virtual], [inherited]
```

### 6.11.3.11 ClearError()

```
void LLState< CLLAnimationController >::ClearError [inline], [virtual], [inherited]
```

### 6.11.3.12 Draw()

```
void CLLState::Draw ( ) [virtual]
```

Implements [LLState< CLLAnimationController >](#).

### 6.11.3.13 DrawCurrentActionText()

```
void LLState< CLLAnimationController >::DrawCurrentActionText [inline], [protected], [virtual], [inherited]
```

### 6.11.3.14 DrawCurrentErrorText()

```
void LLState< CLLAnimationController >::DrawCurrentErrorText [inline], [protected], [virtual], [inherited]
```

### 6.11.3.15 GetContext()

```
State::Context State::GetContext ( ) const [protected], [inherited]
```

**6.11.3.16 InitNavigationBar()**

```
void LLState< CLLAnimationController >::InitNavigationBar [protected], [inherited]
```

**6.11.3.17 RequestStackClear()**

```
void State::RequestStackClear () [protected], [inherited]
```

**6.11.3.18 RequestStackPop()**

```
void State::RequestStackPop () [protected], [inherited]
```

**6.11.3.19 RequestStackPush()**

```
void State::RequestStackPush (
    States::ID stateID ) [protected], [inherited]
```

**6.11.3.20 SetCurrentAction()**

```
void LLState< CLLAnimationController >::SetCurrentAction (
    std::string action ) [inline], [virtual], [inherited]
```

**6.11.3.21 SetCurrentError()**

```
void LLState< CLLAnimationController >::SetCurrentError (
    std::string error ) [inline], [virtual], [inherited]
```

**6.11.3.22 Success()**

```
void LLState< CLLAnimationController >::Success [inline], [virtual], [inherited]
```

### 6.11.3.23 Update()

```
bool LLState< CLLAnimationController >::Update (
    float dt ) [virtual], [inherited]
```

Implements [State](#).

## 6.11.4 Member Data Documentation

### 6.11.4.1 activeDS

```
DataStructures::ID LLState< CLLAnimationController >::activeDS [private], [inherited]
```

### 6.11.4.2 animController

```
T::Ptr LLState< CLLAnimationController >::animController [protected], [inherited]
```

### 6.11.4.3 CLL

```
Algorithm::CircularLinkedList CLLState::CLL [private]
```

### 6.11.4.4 codeHighlighter

```
GUI::CodeHighlighter::Ptr LLState< CLLAnimationController >::codeHighlighter [protected],
[inherited]
```

### 6.11.4.5 footer

```
GUI::Footer< CLLAnimationController > LLState< CLLAnimationController >::footer [protected],
[inherited]
```

#### 6.11.4.6 mContext

```
Context LLState< CLLAnimationController >::mContext [protected], [inherited]
```

#### 6.11.4.7 mCurrentAction

```
std::string LLState< CLLAnimationController >::mCurrentAction [protected], [inherited]
```

#### 6.11.4.8 mCurrentError

```
std::string LLState< CLLAnimationController >::mCurrentError [protected], [inherited]
```

#### 6.11.4.9 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.11.4.10 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

#### 6.11.4.11 operationList

```
GUI::OperationList LLState< CLLAnimationController >::operationList [protected], [inherited]
```

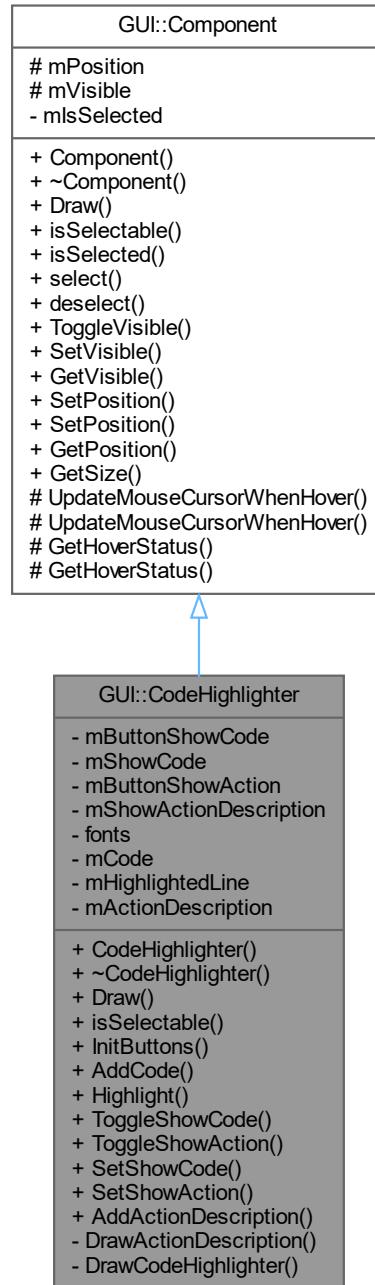
The documentation for this class was generated from the following files:

- src/States/LinkedList/[CLLState.hpp](#)
- src/States/LinkedList/[CLLState.cpp](#)

## 6.12 GUI::CodeHighlighter Class Reference

```
#include <CodeHighlighter.hpp>
```

Inheritance diagram for GUI::CodeHighlighter:



### Public Types

- `typedef std::shared_ptr<CodeHighlighter> Ptr`

## Public Member Functions

- `CodeHighlighter (FontHolder *fonts)`
- `~CodeHighlighter ()`
- `void Draw (Vector2 base=(Vector2){0, 0})`
- `bool IsSelectable () const`
- `void InitButtons ()`
- `void AddCode (std::vector< std::string > code)`
- `void Highlight (int line)`
- `void ToggleShowCode ()`
- `void ToggleShowAction ()`
- `void SetShowCode (bool show)`
- `void SetShowAction (bool show)`
- `void AddActionDescription (std::string description)`
- `bool isSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`
- `virtual Vector2 GetSize ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `void DrawActionDescription (Vector2 base=(Vector2){0, 0})`
- `void DrawCodeHighlighter (Vector2 base=(Vector2){0, 0})`

## Private Attributes

- `GUI::Button mButtonShowCode`
- `bool mShowCode`
- `GUI::Button mButtonShowAction`
- `bool mShowActionDescription`
- `FontHolder * fonts`
- `std::vector< std::string > mCode`
- `int mHighlightedLine`
- `std::string mActionDescription`
- `bool mIsSelected`

### 6.12.1 Member Typedef Documentation

#### 6.12.1.1 Ptr

```
typedef std::shared_ptr< CodeHighlighter > GUI::CodeHighlighter::Ptr
```

### 6.12.2 Constructor & Destructor Documentation

#### 6.12.2.1 CodeHighlighter()

```
GUI::CodeHighlighter::CodeHighlighter (
    FontHolder * fonts )
```

#### 6.12.2.2 ~CodeHighlighter()

```
GUI::CodeHighlighter::~CodeHighlighter ( )
```

### 6.12.3 Member Function Documentation

#### 6.12.3.1 AddActionDescription()

```
void GUI::CodeHighlighter::AddActionDescription (
    std::string description )
```

#### 6.12.3.2 AddCode()

```
void GUI::CodeHighlighter::AddCode (
    std::vector< std::string > code )
```

### 6.12.3.3 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.12.3.4 Draw()

```
void GUI::CodeHighlighter::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Component](#).

### 6.12.3.5 DrawActionDescription()

```
void GUI::CodeHighlighter::DrawActionDescription (
    Vector2 base = (Vector2){0, 0} ) [private]
```

### 6.12.3.6 DrawCodeHighlighter()

```
void GUI::CodeHighlighter::DrawCodeHighlighter (
    Vector2 base = (Vector2){0, 0} ) [private]
```

### 6.12.3.7 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.12.3.8 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.12.3.9 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

### 6.12.3.10 GetSize()

```
Vector2 GUI::Component::GetSize () [virtual], [inherited]
```

Reimplemented in [GUI::Button](#), [GUI::InputField](#), [GUI::OperationList](#), [GUI::OptionInputField](#), and [GUI::Container](#).

### 6.12.3.11 GetVisible()

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

### 6.12.3.12 Highlight()

```
void GUI::CodeHighlighter::Highlight (
    int line )
```

### 6.12.3.13 InitButtons()

```
void GUI::CodeHighlighter::InitButtons ()
```

### 6.12.3.14 IsSelectable()

```
bool GUI::CodeHighlighter::IsSelectable () const [virtual]
```

Implements [GUI::Component](#).

### 6.12.3.15 IsSelected()

```
bool GUI::Component::IsSelected () const [inherited]
```

**6.12.3.16 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.12.3.17 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.12.3.18 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

**6.12.3.19 SetShowAction()**

```
void GUI::CodeHighlighter::SetShowAction (
    bool show )
```

**6.12.3.20 SetShowCode()**

```
void GUI::CodeHighlighter::SetShowCode (
    bool show )
```

**6.12.3.21 SetVisible()**

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

**6.12.3.22 ToggleShowAction()**

```
void GUI::CodeHighlighter::ToggleShowAction ( )
```

### 6.12.3.23 ToggleShowCode()

```
void GUI::CodeHighlighter::ToggleShowCode ( )
```

### 6.12.3.24 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

### 6.12.3.25 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.12.3.26 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.12.4 Member Data Documentation

### 6.12.4.1 fonts

```
FontHolder* GUI::CodeHighlighter::fonts [private]
```

### 6.12.4.2 mActionDescription

```
std::string GUI::CodeHighlighter::mActionDescription [private]
```

#### 6.12.4.3 mButtonShowAction

```
GUI::Button GUI::CodeHighlighter::mButtonShowAction [private]
```

#### 6.12.4.4 mButtonShowCode

```
GUI::Button GUI::CodeHighlighter::mButtonShowCode [private]
```

#### 6.12.4.5 mCode

```
std::vector< std::string > GUI::CodeHighlighter::mCode [private]
```

#### 6.12.4.6 mHighlightedLine

```
int GUI::CodeHighlighter::mHighlightedLine [private]
```

#### 6.12.4.7 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.12.4.8 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.12.4.9 mShowActionDescription

```
bool GUI::CodeHighlighter::mShowActionDescription [private]
```

#### 6.12.4.10 mShowCode

```
bool GUI::CodeHighlighter::mShowCode [private]
```

#### 6.12.4.11 mVisible

bool GUI::Component::mVisible [protected], [inherited]

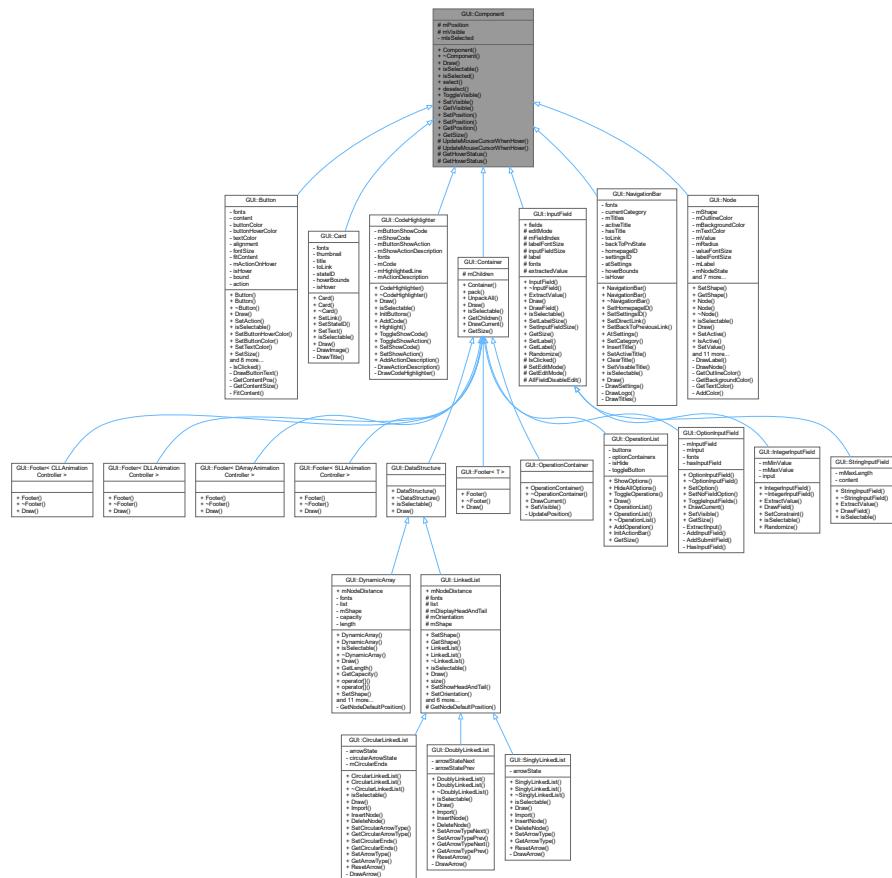
The documentation for this class was generated from the following files:

- src/Components/Common/CodeHighlighter.hpp
  - src/Components/Common/CodeHighlighter.cpp

## 6.13 GUI::Component Class Reference

```
#include <Component.hpp>
```

## Inheritance diagram for GUI::Component:



## Public Types

- `typedef std::shared_ptr< Component > Ptr`

## Public Member Functions

- `Component ()`
- `virtual ~Component ()`
- `virtual void Draw (Vector2 base=(Vector2){0, 0})`
- `virtual bool IsSelectable () const =0`
- `bool IsSelected () const`
- `virtual void Select ()`
- `virtual void Deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`
- `virtual Vector2 GetSize ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `Vector2 mPosition`
- `bool mVisible`

## Private Attributes

- `bool mIsSelected`

### 6.13.1 Member Typedef Documentation

#### 6.13.1.1 Ptr

```
typedef std::shared_ptr< Component > GUI::Component::Ptr
```

### 6.13.2 Constructor & Destructor Documentation

### 6.13.2.1 Component()

```
GUI::Component::Component ( )
```

### 6.13.2.2 ~Component()

```
GUI::Component::~Component ( ) [virtual]
```

## 6.13.3 Member Function Documentation

### 6.13.3.1 deselect()

```
void GUI::Component::deselect ( ) [virtual]
```

### 6.13.3.2 Draw()

```
void GUI::Component::Draw (  
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented in [GUI::DataStructure](#), [GUI::Button](#), [GUI::Card](#), [GUI::CodeHighlighter](#), [GUI::InputField](#), [GUI::NavigationBar](#), [GUI::OperationList](#), and [GUI::Container](#).

### 6.13.3.3 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (   
    Rectangle bound,  
    bool hover,  
    bool noHover ) [protected], [virtual]
```

### 6.13.3.4 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (   
    std::map< std::string, Rectangle > bounds,  
    bool hover,  
    bool noHover ) [protected], [virtual]
```

### 6.13.3.5 GetPosition()

```
Vector2 GUI::Component::GetPosition ( )
```

### 6.13.3.6 GetSize()

```
Vector2 GUI::Component::GetSize ( ) [virtual]
```

Reimplemented in [GUI::Button](#), [GUI::InputField](#), [GUI::OperationList](#), [GUI::OptionInputField](#), and [GUI::Container](#).

### 6.13.3.7 GetVisible()

```
bool GUI::Component::GetVisible ( ) [virtual]
```

### 6.13.3.8 isSelectable()

```
virtual bool GUI::Component::isSelectable ( ) const [pure virtual]
```

Implemented in [GUI::Button](#), [GUI::Card](#), [GUI::CodeHighlighter](#), [GUI::InputField](#), [GUI::IntegerInputField](#), [GUI::NavigationBar](#), [GUI::StringInputField](#), [GUI::CircularLinkedList](#), [GUI::DataStructure](#), [GUI::DoublyLinkedList](#), [GUI::DynamicArray](#), [GUI::LinkedList](#), [GUI::Node](#), [GUI::SinglyLinkedList](#), and [GUI::Container](#).

### 6.13.3.9 isSelected()

```
bool GUI::Component::isSelected ( ) const
```

### 6.13.3.10 select()

```
void GUI::Component::select ( ) [virtual]
```

### 6.13.3.11 SetPosition() [1/2]

```
void GUI::Component::SetPosition (
    float x,
    float y )
```

### 6.13.3.12 SetPosition() [2/2]

```
void GUI::Component::SetPosition (
    Vector2 position )
```

### 6.13.3.13 SetVisible()

```
void GUI::Component::SetVisible (
    bool visible ) [virtual]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.13.3.14 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual]
```

### 6.13.3.15 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual]
```

### 6.13.3.16 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual]
```

## 6.13.4 Member Data Documentation

### 6.13.4.1 mIsSelected

```
bool GUI::Component::mIsSelected [private]
```

### 6.13.4.2 mPosition

```
Vector2 GUI::Component::mPosition [protected]
```

### 6.13.4.3 mVisible

```
bool GUI::Component::mVisible [protected]
```

The documentation for this class was generated from the following files:

- [src/Component.hpp](#)
- [src/Component.cpp](#)

## 6.14 Core::List< T >::const\_iterator Class Reference

```
#include <List.hpp>
```

### Public Types

- using [iterator\\_category](#) = std::bidirectional\_iterator\_tag
- using [value\\_type](#) = T
- using [difference\\_type](#) = std::ptrdiff\_t
- using [pointer](#) = const T \*
- using [reference](#) = const T &

### Public Member Functions

- [const\\_iterator \(\)](#)
- [const\\_iterator \(Node< value\\_type > \\*const &p\)](#)
- [const\\_iterator \(const iterator &other\)](#)
- [const\\_iterator \(const const\\_iterator &other\)](#)
- [reference operator\\* \(\) const](#)
- [pointer operator-> \(\) const](#)
- [const\\_iterator & operator++ \(\)](#)
- [const\\_iterator operator++ \(int\)](#)
- [const\\_iterator & operator-- \(\)](#)
- [const\\_iterator operator-- \(int\)](#)
- [const\\_iterator operator+ \(const int &step\)](#)
- [const\\_iterator operator- \(const int &step\)](#)
- [const\\_iterator & operator= \(Node< value\\_type > \\*const &p\)](#)
- [const\\_iterator & operator= \(const iterator &other\)](#)
- [bool operator== \(const const\\_iterator &it\) const](#)
- [bool operator!= \(const const\\_iterator &it\) const](#)
- [void swap \(const\\_iterator &other\)](#)

## Private Attributes

- const `Node< T > * ptr`

## Friends

- class `List`

### 6.14.1 Member Typedef Documentation

#### 6.14.1.1 difference\_type

```
template<typename T >
using Core::List< T >::const_iterator::difference_type = std::ptrdiff_t
```

#### 6.14.1.2 iterator\_category

```
template<typename T >
using Core::List< T >::const_iterator::iterator_category = std::bidirectional_iterator_tag
```

#### 6.14.1.3 pointer

```
template<typename T >
using Core::List< T >::const_iterator::pointer = const T*
```

#### 6.14.1.4 reference

```
template<typename T >
using Core::List< T >::const_iterator::reference = const T&
```

#### 6.14.1.5 value\_type

```
template<typename T >
using Core::List< T >::const_iterator::value_type = T
```

## 6.14.2 Constructor & Destructor Documentation

### 6.14.2.1 const\_iterator() [1/4]

```
template<typename T >
Core::List< T >::const_iterator::const_iterator ( ) [inline]
```

### 6.14.2.2 const\_iterator() [2/4]

```
template<typename T >
Core::List< T >::const_iterator::const_iterator (
    Node< value_type > *const & p ) [inline]
```

### 6.14.2.3 const\_iterator() [3/4]

```
template<typename T >
Core::List< T >::const_iterator::const_iterator (
    const iterator & other ) [inline]
```

### 6.14.2.4 const\_iterator() [4/4]

```
template<typename T >
Core::List< T >::const_iterator::const_iterator (
    const const_iterator & other ) [inline]
```

## 6.14.3 Member Function Documentation

### 6.14.3.1 operator"!=()

```
template<typename T >
bool Core::List< T >::const_iterator::operator!= (
    const const_iterator & it ) const [inline]
```

### 6.14.3.2 operator\*()

```
template<typename T >
reference Core::List< T >::const_iterator::operator* ( ) const [inline]
```

### 6.14.3.3 operator+()

```
template<typename T >
const_iterator Core::List< T >::const_iterator::operator+ (
    const int & step ) [inline]
```

### 6.14.3.4 operator++() [1/2]

```
template<typename T >
const_iterator & Core::List< T >::const_iterator::operator++ ( ) [inline]
```

### 6.14.3.5 operator++() [2/2]

```
template<typename T >
const_iterator Core::List< T >::const_iterator::operator++ (
    int ) [inline]
```

### 6.14.3.6 operator-()

```
template<typename T >
const_iterator Core::List< T >::const_iterator::operator- (
    const int & step ) [inline]
```

### 6.14.3.7 operator--() [1/2]

```
template<typename T >
const_iterator & Core::List< T >::const_iterator::operator-- ( ) [inline]
```

**6.14.3.8 operator--() [2/2]**

```
template<typename T >
const_iterator Core::List< T >::const_iterator::operator-- (
    int   )  [inline]
```

**6.14.3.9 operator->()**

```
template<typename T >
pointer Core::List< T >::const_iterator::operator-> ( ) const  [inline]
```

**6.14.3.10 operator=() [1/2]**

```
template<typename T >
const_iterator & Core::List< T >::const_iterator::operator= (
    const iterator & other )  [inline]
```

**6.14.3.11 operator=() [2/2]**

```
template<typename T >
const_iterator & Core::List< T >::const_iterator::operator= (
    Node< value_type > *const & p )  [inline]
```

**6.14.3.12 operator==( )**

```
template<typename T >
bool Core::List< T >::const_iterator::operator== (
    const const_iterator & it ) const  [inline]
```

**6.14.3.13 swap()**

```
template<typename T >
void Core::List< T >::const_iterator::swap (
    const_iterator & other )  [inline]
```

**6.14.4 Friends And Related Function Documentation**

#### 6.14.4.1 List

```
template<typename T >
friend class List [friend]
```

### 6.14.5 Member Data Documentation

#### 6.14.5.1 ptr

```
template<typename T >
const Node< T >* Core::List< T >::const_iterator::ptr [private]
```

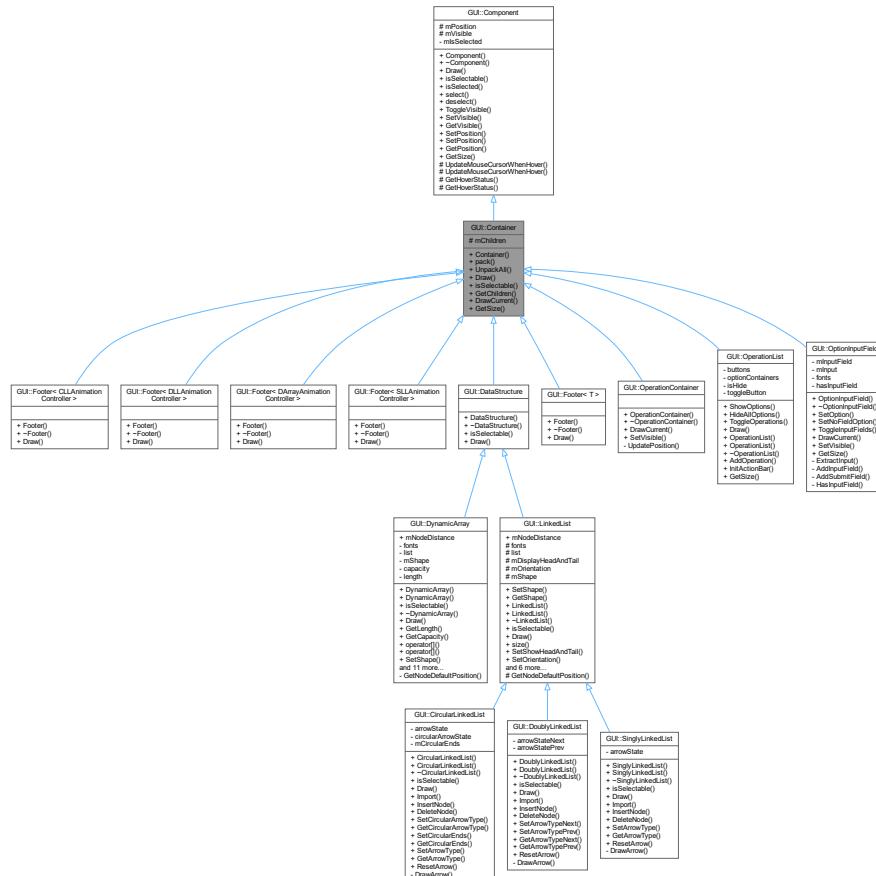
The documentation for this class was generated from the following file:

- src/Core/[List.hpp](#)

## 6.15 GUI::Container Class Reference

```
#include <Container.hpp>
```

Inheritance diagram for GUI::Container:



## Public Types

- `typedef std::shared_ptr< Container > Ptr`

## Public Member Functions

- `Container ()`
- `void pack (Component::Ptr component)`
- `void UnpackAll ()`
- `virtual void Draw (Vector2 base=(Vector2){0, 0})`
- `virtual bool isSelectable () const`
- `std::vector< Component::Ptr > GetChildren ()`
- `virtual void DrawCurrent (Vector2 base)`
- `virtual Vector2 GetSize ()`
- `bool isSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Attributes

- `bool mIsSelected`

### 6.15.1 Member Typedef Documentation

#### 6.15.1.1 Ptr

```
typedef std::shared_ptr< Container > GUI::Container::Ptr
```

## 6.15.2 Constructor & Destructor Documentation

### 6.15.2.1 Container()

```
GUI::Container::Container ( )
```

## 6.15.3 Member Function Documentation

### 6.15.3.1 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.15.3.2 Draw()

```
void GUI::Container::Draw ( 
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::DataStructure](#), and [GUI::OperationList](#).

### 6.15.3.3 DrawCurrent()

```
void GUI::Container::DrawCurrent ( 
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.15.3.4 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( )
```

### 6.15.3.5 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.15.3.6 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.15.3.7 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

### 6.15.3.8 GetSize()

```
Vector2 GUI::Container::GetSize () [virtual]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

### 6.15.3.9 GetVisible()

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

### 6.15.3.10 isSelectable()

```
bool GUI::Container::isSelectable () const [virtual]
```

Implements [GUI::Component](#).

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DataStructure](#), [GUI::DoublyLinkedList](#), [GUI::DynamicArray](#), [GUI::LinkedList](#), and [GUI::SinglyLinkedList](#).

### 6.15.3.11 isSelected()

```
bool GUI::Component::isSelected ( ) const [inherited]
```

### 6.15.3.12 pack()

```
void GUI::Container::pack ( Component::Ptr component )
```

### 6.15.3.13 select()

```
void GUI::Component::select ( ) [virtual], [inherited]
```

### 6.15.3.14 SetPosition() [1/2]

```
void GUI::Component::SetPosition ( float x, float y ) [inherited]
```

### 6.15.3.15 SetPosition() [2/2]

```
void GUI::Component::SetPosition ( Vector2 position ) [inherited]
```

### 6.15.3.16 SetVisible()

```
void GUI::Component::SetVisible ( bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.15.3.17 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

### 6.15.3.18 UnpackAll()

```
void GUI::Container::UnpackAll ( )
```

### 6.15.3.19 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.15.3.20 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.15.4 Member Data Documentation

### 6.15.4.1 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected]
```

### 6.15.4.2 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

### 6.15.4.3 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.15.4.4 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

The documentation for this class was generated from the following files:

- [src/Container.hpp](#)
- [src/Container.cpp](#)

## 6.16 State::Context Struct Reference

```
#include <State.hpp>
```

### Public Member Functions

- [Context \(\)](#)
- [Context \(FontHolder \\*fonts, TextureHolder \\*textures, CategoryInfo \\*categories, DSInfo \\*dsInfo\)](#)

### Public Attributes

- [FontHolder \\* fonts](#)
- [TextureHolder \\* textures](#)
- [CategoryInfo \\* categories](#)
- [DSInfo \\* dsInfo](#)

#### 6.16.1 Constructor & Destructor Documentation

##### 6.16.1.1 Context() [1/2]

```
State::Context::Context ( )
```

##### 6.16.1.2 Context() [2/2]

```
State::Context::Context (
    FontHolder * fonts,
    TextureHolder * textures,
    CategoryInfo * categories,
    DSInfo * dsInfo )
```

## 6.16.2 Member Data Documentation

### 6.16.2.1 categories

```
CategoryInfo* State::Context::categories
```

### 6.16.2.2 dsInfo

```
DSInfo* State::Context::dsInfo
```

### 6.16.2.3 fonts

```
FontHolder* State::Context::fonts
```

### 6.16.2.4 textures

```
TextureHolder* State::Context::textures
```

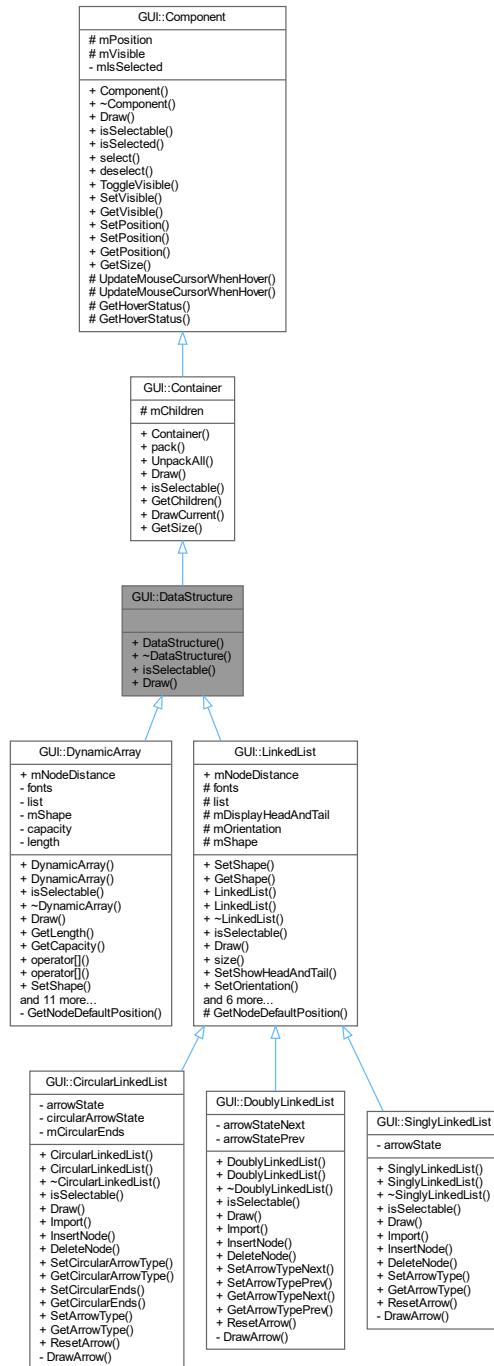
The documentation for this struct was generated from the following files:

- src/[State.hpp](#)
- src/[State.cpp](#)

## 6.17 GUI::DataStructure Class Reference

```
#include <DataStructure.hpp>
```

Inheritance diagram for GUI::DataStructure:



### Public Types

- `typedef std::shared_ptr< DataStructure > Ptr`

## Public Member Functions

- `DataStructure ()`
- `virtual ~DataStructure ()`
- `virtual bool IsSelectable () const`
- `virtual void Draw (Vector2 base)`
- `void pack (Component::Ptr component)`
- `void UnpackAll ()`
- `std::vector< Component::Ptr > GetChildren ()`
- `virtual void DrawCurrent (Vector2 base)`
- `virtual Vector2 GetSize ()`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Attributes

- `bool mIsSelected`

### 6.17.1 Member Typedef Documentation

#### 6.17.1.1 Ptr

```
typedef std::shared_ptr< DataStructure > GUI::DataStructure::Ptr
```

## 6.17.2 Constructor & Destructor Documentation

### 6.17.2.1 DataStructure()

```
GUI::DataStructure::DataStructure ( )
```

### 6.17.2.2 ~DataStructure()

```
GUI::DataStructure::~DataStructure ( ) [virtual]
```

## 6.17.3 Member Function Documentation

### 6.17.3.1 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.17.3.2 Draw()

```
void GUI::DataStructure::Draw ( 
    Vector2 base ) [virtual]
```

Reimplemented from [GUI::Container](#).

### 6.17.3.3 DrawCurrent()

```
void GUI::Container::DrawCurrent ( 
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.17.3.4 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( ) [inherited]
```

### 6.17.3.5 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.17.3.6 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.17.3.7 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

### 6.17.3.8 GetSize()

```
Vector2 GUI::Container::GetSize () [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

### 6.17.3.9 GetVisible()

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

### 6.17.3.10 isSelectable()

```
bool GUI::DataStructure::isSelectable () const [virtual]
```

Reimplemented from [GUI::Container](#).

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DoublyLinkedList](#), [GUI::DynamicArray](#), [GUI::LinkedList](#), and [GUI::SinglyLinkedList](#).

### 6.17.3.11 isSelected()

```
bool GUI::Component::isSelected ( ) const [inherited]
```

### 6.17.3.12 pack()

```
void GUI::Container::pack ( Component::Ptr component ) [inherited]
```

### 6.17.3.13 select()

```
void GUI::Component::select ( ) [virtual], [inherited]
```

### 6.17.3.14 SetPosition() [1/2]

```
void GUI::Component::SetPosition ( float x, float y ) [inherited]
```

### 6.17.3.15 SetPosition() [2/2]

```
void GUI::Component::SetPosition ( Vector2 position ) [inherited]
```

### 6.17.3.16 SetVisible()

```
void GUI::Component::SetVisible ( bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.17.3.17 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

### 6.17.3.18 UnpackAll()

```
void GUI::Container::UnpackAll ( ) [inherited]
```

### 6.17.3.19 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.17.3.20 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.17.4 Member Data Documentation

### 6.17.4.1 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

### 6.17.4.2 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

### 6.17.4.3 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.17.4.4 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

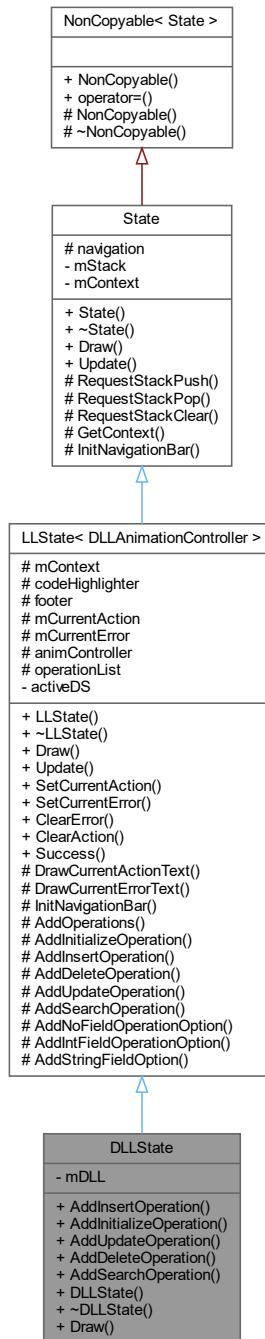
The documentation for this class was generated from the following files:

- src/Components/Visualization/[DataStructure.hpp](#)
- src/Components/Visualization/[DataStructure.cpp](#)

## 6.18 DLLState Class Reference

```
#include <DLLState.hpp>
```

Inheritance diagram for DLLState:



## Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `void AddInsertOperation ()`

- void `AddInitializeOperation ()`
- void `AddUpdateOperation ()`
- void `AddDeleteOperation ()`
- void `AddSearchOperation ()`
- `DLLState (StateStack &stack, Context context)`
- `~DLLState ()`
- void `Draw ()`
- virtual bool `Update (float dt)`
- virtual void `SetCurrentAction (std::string action)`
- virtual void `SetCurrentError (std::string error)`
- virtual void `ClearError ()`
- virtual void `ClearAction ()`
- virtual void `Success ()`

## Protected Member Functions

- virtual void `DrawCurrentActionText ()`
- virtual void `DrawCurrentErrorText ()`
- void `InitNavigationBar ()`
- virtual void `AddOperations ()`
- virtual void `AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void()> action)`
- virtual void `AddIntFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >)> action)`
- virtual void `AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >)> action)`
- void `RequestStackPush (States::ID stateID)`
- void `RequestStackPop ()`
- void `RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< DLLAnimationController > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Attributes

- `Algorithm::DoublyLinkedList mDLL`
- `DataStructures::ID activeDS`
- `StateStack * mStack`

### 6.18.1 Member Typedef Documentation

### 6.18.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

## 6.18.2 Constructor & Destructor Documentation

### 6.18.2.1 DLLState()

```
DLLState::DLLState (
    StateStack & stack,
    Context context )
```

### 6.18.2.2 ~DLLState()

```
DLLState::~DLLState ( )
```

## 6.18.3 Member Function Documentation

### 6.18.3.1 AddDeleteOperation()

```
void DLLState::AddDeleteOperation ( ) [virtual]
```

Reimplemented from [LLState< DLLAnimationController >](#).

### 6.18.3.2 AddInitializeOperation()

```
void DLLState::AddInitializeOperation ( ) [virtual]
```

Reimplemented from [LLState< DLLAnimationController >](#).

### 6.18.3.3 AddInsertOperation()

```
void DLLState::AddInsertOperation ( ) [virtual]
```

Reimplemented from [LLState< DLLAnimationController >](#).

#### 6.18.3.4 AddIntFieldOperationOption()

```
void LLState< DLLAnimationController >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

#### 6.18.3.5 AddNoFieldOperationOption()

```
void LLState< DLLAnimationController >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual], [inherited]
```

#### 6.18.3.6 AddOperations()

```
void LLState< DLLAnimationController >::AddOperations [protected], [virtual], [inherited]
```

#### 6.18.3.7 AddSearchOperation()

```
void DLLState::AddSearchOperation () [virtual]
```

Reimplemented from [LLState< DLLAnimationController >](#).

#### 6.18.3.8 AddStringFieldOption()

```
void LLState< DLLAnimationController >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

#### 6.18.3.9 AddUpdateOperation()

```
void DLLState::AddUpdateOperation () [virtual]
```

Reimplemented from [LLState< DLLAnimationController >](#).

**6.18.3.10 ClearAction()**

```
void LLState< DLLAnimationController >::ClearAction [inline], [virtual], [inherited]
```

**6.18.3.11 ClearError()**

```
void LLState< DLLAnimationController >::ClearError [inline], [virtual], [inherited]
```

**6.18.3.12 Draw()**

```
void DLLState::Draw () [virtual]
```

Implements [LLState< DLLAnimationController >](#).

**6.18.3.13 DrawCurrentActionText()**

```
void LLState< DLLAnimationController >::DrawCurrentActionText [inline], [protected], [virtual], [inherited]
```

**6.18.3.14 DrawCurrentErrorText()**

```
void LLState< DLLAnimationController >::DrawCurrentErrorText [inline], [protected], [virtual], [inherited]
```

**6.18.3.15 GetContext()**

```
State::Context State::GetContext () const [protected], [inherited]
```

**6.18.3.16 InitNavigationBar()**

```
void LLState< DLLAnimationController >::InitNavigationBar [protected], [inherited]
```

**6.18.3.17 RequestStackClear()**

```
void State::RequestStackClear ( ) [protected], [inherited]
```

**6.18.3.18 RequestStackPop()**

```
void State::RequestStackPop ( ) [protected], [inherited]
```

**6.18.3.19 RequestStackPush()**

```
void State::RequestStackPush (
    States::ID stateID ) [protected], [inherited]
```

**6.18.3.20 SetCurrentAction()**

```
void LLState< DLLAnimationController >::SetCurrentAction (
    std::string action ) [inline], [virtual], [inherited]
```

**6.18.3.21 SetCurrentError()**

```
void LLState< DLLAnimationController >::SetCurrentError (
    std::string error ) [inline], [virtual], [inherited]
```

**6.18.3.22 Success()**

```
void LLState< DLLAnimationController >::Success [inline], [virtual], [inherited]
```

**6.18.3.23 Update()**

```
bool LLState< DLLAnimationController >::Update (
    float dt ) [virtual], [inherited]
```

Implements [State](#).

## 6.18.4 Member Data Documentation

### 6.18.4.1 activeDS

```
DataStructures::ID LLState< DLLAnimationController >::activeDS [private], [inherited]
```

### 6.18.4.2 animController

```
T::Ptr LLState< DLLAnimationController >::animController [protected], [inherited]
```

### 6.18.4.3 codeHighlighter

```
GUI::CodeHighlighter::Ptr LLState< DLLAnimationController >::codeHighlighter [protected], [inherited]
```

### 6.18.4.4 footer

```
GUI::Footer< DLLAnimationController > LLState< DLLAnimationController >::footer [protected], [inherited]
```

### 6.18.4.5 mContext

```
Context LLState< DLLAnimationController >::mContext [protected], [inherited]
```

### 6.18.4.6 mCurrentAction

```
std::string LLState< DLLAnimationController >::mCurrentAction [protected], [inherited]
```

### 6.18.4.7 mCurrentError

```
std::string LLState< DLLAnimationController >::mCurrentError [protected], [inherited]
```

#### 6.18.4.8 mDLL

```
Algorithm::DoublyLinkedList DLLState::mDLL [private]
```

#### 6.18.4.9 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.18.4.10 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

#### 6.18.4.11 operationList

```
GUI::OperationList LLState< DLLAnimationController >::operationList [protected], [inherited]
```

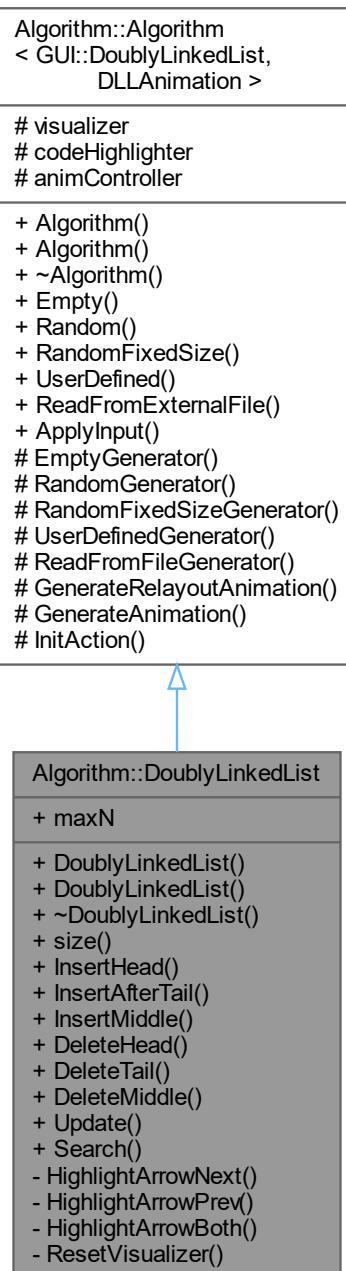
The documentation for this class was generated from the following files:

- src/States/LinkedList/[DLLState.hpp](#)
- src/States/LinkedList/[DLLState.cpp](#)

## 6.19 Algorithm::DoublyLinkedList Class Reference

```
#include <DoublyLinkedList.hpp>
```

Inheritance diagram for Algorithm::DoublyLinkedList:



## Public Member Functions

- [DoublyLinkedList \(\)](#)
- [DoublyLinkedList \(GUI::CodeHighlighter::Ptr codeHighlighter, DLLAnimationController::Ptr animController, FontHolder \\*fonts\)](#)
- [~DoublyLinkedList \(\)](#)
- [std::size\\_t size \(\) const](#)

- void `InsertHead` (int value)
- void `InsertAfterTail` (int value)
- void `InsertMiddle` (int index, int value)
- void `DeleteHead` ()
- void `DeleteTail` ()
- void `DeleteMiddle` (int index)
- void `Update` (int index, int value)
- void `Search` (int value)
- virtual void `Empty` ()
- virtual void `Random` ()
- virtual void `RandomFixedSize` (int N)
- virtual void `UserDefined` (std::string input)
- virtual void `ReadFromExternalFile` (std::string path)
- virtual void `ApplyInput` (std::vector< int > input, std::size\_t nMaxSize=10)

## Static Public Attributes

- static constexpr int `maxN` = 10

## Protected Member Functions

- std::vector< int > `EmptyGenerator` ()
- std::vector< int > `RandomGenerator` ()
- std::vector< int > `RandomFixedSizeGenerator` (int nSize)
- std::vector< int > `UserDefinedGenerator` (std::string input)
- std::vector< int > `ReadFromFileGenerator` (std::string inputFile)
- virtual void `GenerateRelayoutAnimation` (Vector2 newPosition)
- virtual `DLLAnimation` `GenerateAnimation` (float duration, int highlightLine, std::string actionDescription)
- virtual void `InitAction` (std::vector< std::string > code)

## Protected Attributes

- `GUI::DoublyLinkedList` visualizer
- `GUI::CodeHighlighter::Ptr` codeHighlighter
- `Animation::AnimationController< DLLAnimation >::Ptr` animController

## Private Member Functions

- std::function< `GUI::DoublyLinkedList`(`GUI::DoublyLinkedList`, float, Vector2) > `HighlightArrowNext` (int index, bool drawVisualizer=true, bool reverse=false)
- std::function< `GUI::DoublyLinkedList`(`GUI::DoublyLinkedList`, float, Vector2) > `HighlightArrowPrev` (int index, bool drawVisualizer=true, bool reverse=false)
- std::function< `GUI::DoublyLinkedList`(`GUI::DoublyLinkedList`, float, Vector2) > `HighlightArrowBoth` (int index, bool drawVisualizer=true, bool reverse=false)
- void `ResetVisualizer` ()

### 6.19.1 Constructor & Destructor Documentation

### 6.19.1.1 DoublyLinkedList() [1/2]

```
Algorithm::DoublyLinkedList::DoublyLinkedList ( )
```

### 6.19.1.2 DoublyLinkedList() [2/2]

```
Algorithm::DoublyLinkedList::DoublyLinkedList (
    GUI::CodeHighlighter::Ptr codeHighlighter,
    DLLAnimationController::Ptr animController,
    FontHolder * fonts )
```

### 6.19.1.3 ~DoublyLinkedList()

```
Algorithm::DoublyLinkedList::~DoublyLinkedList ( )
```

## 6.19.2 Member Function Documentation

### 6.19.2.1 ApplyInput()

```
void Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::ApplyInput (
    std::vector< int > input,
    std::size_t nMaxSize = 10 ) [virtual], [inherited]
```

### 6.19.2.2 DeleteHead()

```
void Algorithm::DoublyLinkedList::DeleteHead ( )
```

### 6.19.2.3 DeleteMiddle()

```
void Algorithm::DoublyLinkedList::DeleteMiddle (
    int index )
```

#### 6.19.2.4 DeleteTail()

```
void Algorithm::DoublyLinkedList::DeleteTail ( )
```

#### 6.19.2.5 Empty()

```
void Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::Empty [virtual], [inherited]
```

#### 6.19.2.6 EmptyGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::Empty<->  
Generator [protected], [inherited]
```

#### 6.19.2.7 GenerateAnimation()

```
DLLAnimation Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::GenerateAnimation (   
    float duration,  
    int highlightLine,  
    std::string actionDescription ) [protected], [virtual], [inherited]
```

#### 6.19.2.8 GenerateRelayoutAnimation()

```
void Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::GenerateRelayoutAnimation (   
    Vector2 newPosition ) [inline], [protected], [virtual], [inherited]
```

#### 6.19.2.9 HighlightArrowBoth()

```
std::function< GUI::DoublyLinkedList(GUI::DoublyLinkedList, float, Vector2) > Algorithm::->  
DoublyLinkedList::HighlightArrowBoth (   
    int index,  
    bool drawVisualizer = true,  
    bool reverse = false ) [private]
```

**6.19.2.10 HighlightArrowNext()**

```
std::function< GUI::DoublyLinkedList(GUI::DoublyLinkedList, float, Vector2) > Algorithm::->
DoublyLinkedList::HighlightArrowNext (
    int index,
    bool drawVisualizer = true,
    bool reverse = false ) [private]
```

**6.19.2.11 HighlightArrowPrev()**

```
std::function< GUI::DoublyLinkedList(GUI::DoublyLinkedList, float, Vector2) > Algorithm::->
DoublyLinkedList::HighlightArrowPrev (
    int index,
    bool drawVisualizer = true,
    bool reverse = false ) [private]
```

**6.19.2.12 InitAction()**

```
void Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::InitAction (
    std::vector< std::string > code ) [protected], [virtual], [inherited]
```

**6.19.2.13 InsertAfterTail()**

```
void Algorithm::DoublyLinkedList::InsertAfterTail (
    int value )
```

**6.19.2.14 InsertHead()**

```
void Algorithm::DoublyLinkedList::InsertHead (
    int value )
```

**6.19.2.15 InsertMiddle()**

```
void Algorithm::DoublyLinkedList::InsertMiddle (
    int index,
    int value )
```

### 6.19.2.16 Random()

```
void Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::Random [virtual], [inherited]
```

### 6.19.2.17 RandomFixedSize()

```
void Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::RandomFixedSize (
    int N ) [virtual], [inherited]
```

### 6.19.2.18 RandomFixedSizeGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::Random←
FixedSizeGenerator (
    int nSize ) [protected], [inherited]
```

### 6.19.2.19 RandomGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::Random←
Generator [protected], [inherited]
```

### 6.19.2.20 ReadFromExternalFile()

```
void Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::ReadFromExternalFile (
    std::string path ) [virtual], [inherited]
```

### 6.19.2.21 ReadFromFileGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::ReadFrom←
FileGenerator (
    std::string inputFile ) [protected], [inherited]
```

### 6.19.2.22 ResetVisualizer()

```
void Algorithm::DoublyLinkedList::ResetVisualizer ( ) [private]
```

**6.19.2.23 Search()**

```
void Algorithm::DoublyLinkedList::Search (
    int value )
```

**6.19.2.24 size()**

```
std::size_t Algorithm::DoublyLinkedList::size ( ) const
```

**6.19.2.25 Update()**

```
void Algorithm::DoublyLinkedList::Update (
    int index,
    int value )
```

**6.19.2.26 UserDefined()**

```
void Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::UserDefined (
    std::string input ) [virtual], [inherited]
```

**6.19.2.27 UserDefinedGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::UserDefined<->
Generator (
    std::string input ) [protected], [inherited]
```

**6.19.3 Member Data Documentation****6.19.3.1 animController**

```
Animation::AnimationController< DLLAnimation >::Ptr Algorithm::Algorithm< GUI::DoublyLinkedList ,
, DLLAnimation >::animController [protected], [inherited]
```

### 6.19.3.2 codeHighlighter

```
GUI::CodeHighlighter::Ptr Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::code←  
Highlighter [protected], [inherited]
```

### 6.19.3.3 maxN

```
constexpr int Algorithm::DoublyLinkedList::maxN = 10 [static], [constexpr]
```

### 6.19.3.4 visualizer

```
GUI::DoublyLinkedList Algorithm::Algorithm< GUI::DoublyLinkedList , DLLAnimation >::visualizer  
[protected], [inherited]
```

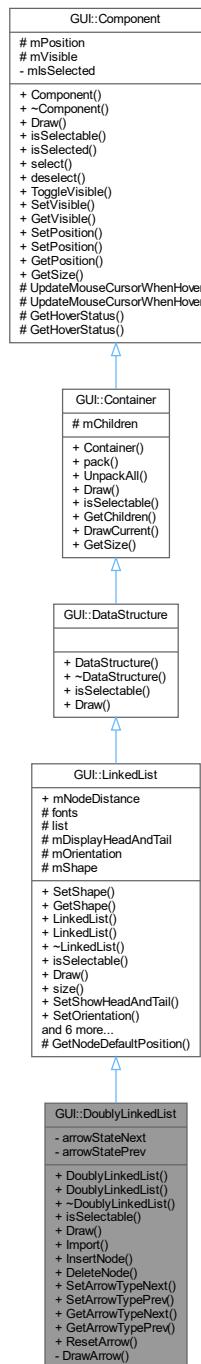
The documentation for this class was generated from the following files:

- src/Algorithms/LinkedList/DoublyLinkedList.hpp
- src/Algorithms/LinkedList/DoublyLinkedList.cpp

## 6.20 GUI::DoublyLinkedList Class Reference

```
#include <DoublyLinkedList.hpp>
```

Inheritance diagram for GUI::DoublyLinkedList:



## Public Types

- enum `ArrowType` {
 `Default` , `Hidden` , `Active` , `Skip` ,
 `ArrowTypeCount` }
- enum `Orientation` { `Horizontal` , `Vertical` , `OrientationCount` }
- typedef `std::shared_ptr< DataStructure >` `Ptr`

## Public Member Functions

- `DoublyLinkedList ()`
- `DoublyLinkedList (FontHolder *fonts)`
- `~DoublyLinkedList ()`
- `bool isSelectable () const`
- `void Draw (Vector2 base=(Vector2){0, 0}, float t=1.0f, bool init=false)`
- `void Import (std::vector< int > nodes)`
- `void InsertNode (std::size_t index, GUI::Node node, bool rePosition=true)`
- `void DeleteNode (std::size_t index, bool rePosition=true)`
- `void SetArrowTypeNext (std::size_t index, ArrowType type)`
- `void SetArrowTypePrev (std::size_t index, ArrowType type)`
- `ArrowType GetArrowTypeNext (std::size_t index)`
- `ArrowType GetArrowTypePrev (std::size_t index)`
- `void ResetArrow ()`
- `void SetShape (GUI::Node::Shape shape)`
- `GUI::Node::Shape GetShape () const`
- `virtual void Draw (Vector2 base)`
- `virtual std::size_t size () const`
- `virtual void SetShowHeadAndTail (bool show)`
- `virtual void SetOrientation (Orientation orientation)`
- `virtual std::vector< GUI::Node > & GetList ()`
- `virtual GUI::Node GenerateNode (int value)`
- `virtual void Relayout ()`
- `void pack (Component::Ptr component)`
- `void UnpackAll ()`
- `std::vector< Component::Ptr > GetChildren ()`
- `virtual void DrawCurrent (Vector2 base)`
- `virtual Vector2 GetSize ()`
- `bool isSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Static Public Attributes

- `static constexpr float mNodeDistance = 20`

## Protected Member Functions

- `Vector2 GetNodeDefaultPosition (std::size_t index)`
- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `FontHolder * fonts`
- `std::vector< GUI::Node > list`
- `bool mDisplayHeadAndTail`
- `Orientation mOrientation = Orientation::Horizontal`
- `GUI::Node::Shape mShape = GUI::Node::Shape::Circle`
- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `void DrawArrow (Vector2 base, float t)`

## Private Attributes

- `std::vector< ArrowType > arrowStateNext`
- `std::vector< ArrowType > arrowStatePrev`
- `bool mIsSelected`

### 6.20.1 Member Typedef Documentation

#### 6.20.1.1 Ptr

```
typedef std::shared_ptr< DataStructure > GUI::DataStructure::Ptr [inherited]
```

### 6.20.2 Member Enumeration Documentation

#### 6.20.2.1 ArrowType

```
enum GUI::LinkedList::ArrowType [inherited]
```

##### Enumerator

Default	
Hidden	
Active	
Skip	
ArrowTypeCount	

### 6.20.2.2 Orientation

```
enum GUI::LinkedList::Orientation [inherited]
```

Enumerator

Horizontal	
Vertical	
OrientationCount	

### 6.20.3 Constructor & Destructor Documentation

#### 6.20.3.1 DoublyLinkedList() [1/2]

```
GUI::DoublyLinkedList::DoublyLinkedList ()
```

#### 6.20.3.2 DoublyLinkedList() [2/2]

```
GUI::DoublyLinkedList::DoublyLinkedList (
    FontHolder * fonts )
```

#### 6.20.3.3 ~DoublyLinkedList()

```
GUI::DoublyLinkedList::~DoublyLinkedList ()
```

### 6.20.4 Member Function Documentation

#### 6.20.4.1 DeleteNode()

```
void GUI::DoublyLinkedList::DeleteNode (
    std::size_t index,
    bool rePosition = true ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

#### 6.20.4.2 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

#### 6.20.4.3 Draw() [1/2]

```
void GUI::DataStructure::Draw (
    Vector2 base ) [virtual], [inherited]
```

Reimplemented from [GUI::Container](#).

#### 6.20.4.4 Draw() [2/2]

```
void GUI::DoublyLinkedList::Draw (
    Vector2 base = (Vector2){0, 0},
    float t = 1.0f,
    bool init = false ) [virtual]
```

Implements [GUI::LinkedList](#).

#### 6.20.4.5 DrawArrow()

```
void GUI::DoublyLinkedList::DrawArrow (
    Vector2 base,
    float t ) [private]
```

#### 6.20.4.6 DrawCurrent()

```
void GUI::Container::DrawCurrent (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.20.4.7 GenerateNode()

```
GUI::Node GUI::LinkedList::GenerateNode (
    int value ) [virtual], [inherited]
```

#### 6.20.4.8 GetArrowTypeNext()

```
GUI::DoublyLinkedList::ArrowType GUI::DoublyLinkedList::GetArrowTypeNext (
    std::size_t index )
```

#### 6.20.4.9 GetArrowTypePrev()

```
GUI::DoublyLinkedList::ArrowType GUI::DoublyLinkedList::GetArrowTypePrev (
    std::size_t index )
```

#### 6.20.4.10 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren () [inherited]
```

#### 6.20.4.11 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.20.4.12 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.20.4.13 GetList()

```
std::vector< GUI::Node > & GUI::LinkedList::GetList () [virtual], [inherited]
```

#### 6.20.4.14 GetNodeDefaultPosition()

```
Vector2 GUI::LinkedList::GetNodeDefaultPosition (
    std::size_t index ) [protected], [inherited]
```

#### 6.20.4.15 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

#### 6.20.4.16 GetShape()

```
GUI::Node::Shape GUI::LinkedList::GetShape () const [inherited]
```

#### 6.20.4.17 GetSize()

```
Vector2 GUI::Container::GetSize () [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

#### 6.20.4.18 GetVisible()

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

#### 6.20.4.19 Import()

```
void GUI::DoublyLinkedList::Import (
    std::vector< int > nodes ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

#### 6.20.4.20 InsertNode()

```
void GUI::DoublyLinkedList::InsertNode (
    std::size_t index,
    GUI::Node node,
    bool rePosition = true ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

#### 6.20.4.21 **isSelectable()**

```
bool GUI::DoublyLinkedList::isSelectable ( ) const [virtual]
```

Reimplemented from [GUI::DataStructure](#).

#### 6.20.4.22 **isSelected()**

```
bool GUI::Component::isSelected ( ) const [inherited]
```

#### 6.20.4.23 **pack()**

```
void GUI::Container::pack (
    Component::Ptr component ) [inherited]
```

#### 6.20.4.24 **Relayout()**

```
void GUI::LinkedList::Relayout ( ) [virtual], [inherited]
```

#### 6.20.4.25 **ResetArrow()**

```
void GUI::DoublyLinkedList::ResetArrow ( )
```

#### 6.20.4.26 **select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

#### 6.20.4.27 **SetArrowTypeNext()**

```
void GUI::DoublyLinkedList::SetArrowTypeNext (
    std::size_t index,
    ArrowType type )
```

**6.20.4.28 SetArrowTypePrev()**

```
void GUI::DoublyLinkedList::SetArrowTypePrev (
    std::size_t index,
    ArrowType type )
```

**6.20.4.29 SetOrientation()**

```
void GUI::LinkedList::SetOrientation (
    Orientation orientation ) [virtual], [inherited]
```

**6.20.4.30 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.20.4.31 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

**6.20.4.32 SetShape()**

```
void GUI::LinkedList::SetShape (
    GUI::Node::Shape shape ) [inherited]
```

**6.20.4.33 SetShowHeadAndTail()**

```
void GUI::LinkedList::SetShowHeadAndTail (
    bool show ) [virtual], [inherited]
```

#### 6.20.4.34 SetVisible()

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.20.4.35 size()

```
std::size_t GUI::LinkedList::size () const [virtual], [inherited]
```

#### 6.20.4.36 ToggleVisible()

```
void GUI::Component::ToggleVisible () [virtual], [inherited]
```

#### 6.20.4.37 UnpackAll()

```
void GUI::Container::UnpackAll () [inherited]
```

#### 6.20.4.38 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.20.4.39 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.20.5 Member Data Documentation

### 6.20.5.1 arrowStateNext

```
std::vector< ArrowType > GUI::DoublyLinkedList::arrowStateNext [private]
```

### 6.20.5.2 arrowStatePrev

```
std::vector< ArrowType > GUI::DoublyLinkedList::arrowStatePrev [private]
```

### 6.20.5.3 fonts

```
FontHolder* GUI::LinkedList::fonts [protected], [inherited]
```

### 6.20.5.4 list

```
std::vector< GUI::Node > GUI::LinkedList::list [protected], [inherited]
```

### 6.20.5.5 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

### 6.20.5.6 mDisplayHeadAndTail

```
bool GUI::LinkedList::mDisplayHeadAndTail [protected], [inherited]
```

### 6.20.5.7 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

### 6.20.5.8 mNodeDistance

```
constexpr float GUI::LinkedList::mNodeDistance = 20 [static], [constexpr], [inherited]
```

### 6.20.5.9 mOrientation

```
Orientation GUI::LinkedList::mOrientation = Orientation::Horizontal [protected], [inherited]
```

### 6.20.5.10 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

### 6.20.5.11 mShape

```
GUI::Node::Shape GUI::LinkedList::mShape = GUI::Node::Shape::Circle [protected], [inherited]
```

### 6.20.5.12 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

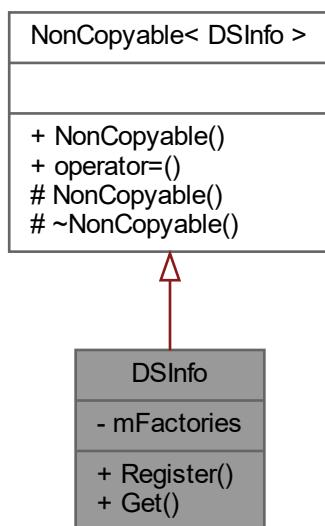
The documentation for this class was generated from the following files:

- src/Components/Visualization/[DoublyLinkedList.hpp](#)
- src/Components/Visualization/[DoublyLinkedList.cpp](#)

## 6.21 DSInfo Class Reference

```
#include <DSInfo.hpp>
```

Inheritance diagram for DSInfo:



## Classes

- struct [Info](#)

## Public Member Functions

- void [Register \(DataStructures::ID, Info info\)](#)
- [Info Get \(DataStructures::ID id\) const](#)

## Private Attributes

- std::map< [DataStructures::ID](#), [Info](#) > [mFactories](#)

### 6.21.1 Member Function Documentation

#### 6.21.1.1 [Get\(\)](#)

```
DSInfo::Info DSInfo::Get (
    DataStructures::ID id ) const
```

#### 6.21.1.2 [Register\(\)](#)

```
void DSInfo::Register (
    DataStructures::ID id,
    Info info )
```

### 6.21.2 Member Data Documentation

#### 6.21.2.1 [mFactories](#)

```
std::map< DataStructures::ID, Info > DSInfo::mFactories [private]
```

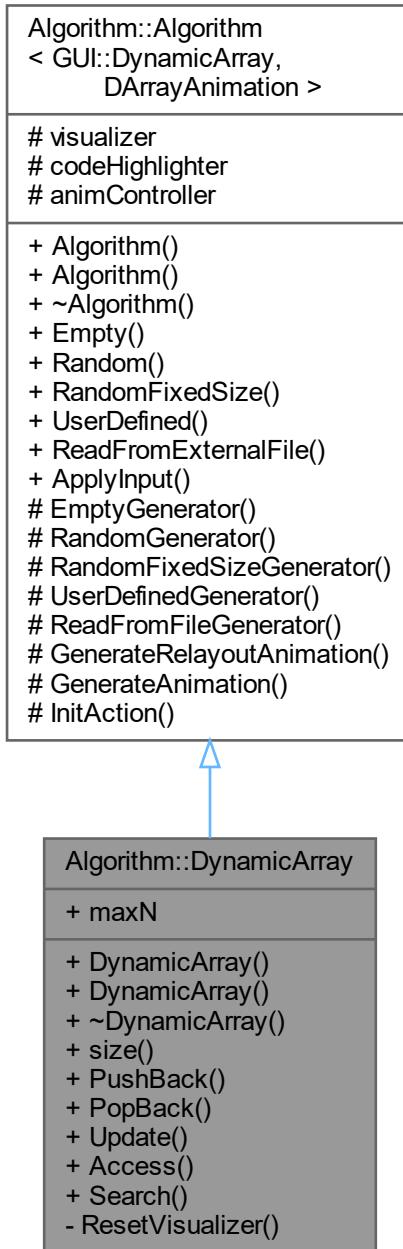
The documentation for this class was generated from the following files:

- src/Identifiers/[DSInfo.hpp](#)
- src/Identifiers/[DSInfo.cpp](#)

## 6.22 Algorithm::DynamicArray Class Reference

```
#include <DynamicArray.hpp>
```

Inheritance diagram for Algorithm::DynamicArray:



### Public Member Functions

- [DynamicArray \(\)](#)

- `DynamicArray (GUI::CodeHighlighter::Ptr codeHighlighter, DArrayAnimationController::Ptr animController, FontHolder *fonts)`
- `~DynamicArray ()`
- `std::size_t size () const`
- `void PushBack (int value)`
- `void PopBack ()`
- `void Update (int index, int value)`
- `void Access (int index)`
- `void Search (int value)`
- `virtual void Empty ()`
- `virtual void Random ()`
- `virtual void RandomFixedSize (int N)`
- `virtual void UserDefined (std::string input)`
- `virtual void ReadFromExternalFile (std::string path)`
- `virtual void ApplyInput (std::vector< int > input, std::size_t nMaxSize=10)`

## Static Public Attributes

- `static constexpr int maxN = 16`

## Protected Member Functions

- `std::vector< int > EmptyGenerator ()`
- `std::vector< int > RandomGenerator ()`
- `std::vector< int > RandomFixedSizeGenerator (int nSize)`
- `std::vector< int > UserDefinedGenerator (std::string input)`
- `std::vector< int > ReadFromFileGenerator (std::string inputFile)`
- `virtual void GenerateRelayoutAnimation (Vector2 newPosition)`
- `virtual DArrayAnimation GenerateAnimation (float duration, int highlightLine, std::string actionDescription)`
- `virtual void InitAction (std::vector< std::string > code)`

## Protected Attributes

- `GUI::DynamicArray visualizer`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `Animation::AnimationController< DArrayAnimation >::Ptr animController`

## Private Member Functions

- `void ResetVisualizer ()`

### 6.22.1 Constructor & Destructor Documentation

#### 6.22.1.1 DynamicArray() [1/2]

```
Algorithm::DynamicArray::DynamicArray ( )
```

### 6.22.1.2 DynamicArray() [2/2]

```
Algorithm::DynamicArray::DynamicArray (
    GUI::CodeHighlighter::Ptr codeHighlighter,
    DArrayAnimationController::Ptr animController,
    FontHolder * fonts )
```

### 6.22.1.3 ~DynamicArray()

```
Algorithm::DynamicArray::~DynamicArray ()
```

## 6.22.2 Member Function Documentation

### 6.22.2.1 Access()

```
void Algorithm::DynamicArray::Access (
    int index )
```

### 6.22.2.2 ApplyInput()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::ApplyInput (
    std::vector< int > input,
    std::size_t nMaxSize = 10 ) [virtual], [inherited]
```

Reimplemented in [Algorithm::StaticArray](#).

### 6.22.2.3 Empty()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::Empty [virtual], [inherited]
```

### 6.22.2.4 EmptyGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::EmptyGenerator
[protected], [inherited]
```

**6.22.2.5 GenerateAnimation()**

```
DArrayAnimation Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::GenerateAnimation  
(  
    float duration,  
    int highlightLine,  
    std::string actionDescription ) [protected], [virtual], [inherited]
```

**6.22.2.6 GenerateRelayoutAnimation()**

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::GenerateRelayoutAnimation (   
    Vector2 newPosition ) [inline], [protected], [virtual], [inherited]
```

**6.22.2.7 InitAction()**

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::InitAction (   
    std::vector< std::string > code ) [protected], [virtual], [inherited]
```

**6.22.2.8 PopBack()**

```
void Algorithm::DynamicArray::PopBack ( )
```

**6.22.2.9 PushBack()**

```
void Algorithm::DynamicArray::PushBack (   
    int value )
```

**6.22.2.10 Random()**

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::Random [virtual], [inherited]
```

**6.22.2.11 RandomFixedSize()**

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::RandomFixedSize (   
    int N ) [virtual], [inherited]
```

**6.22.2.12 RandomFixedSizeGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::RandomFixed←
SizeGenerator (
    int nSize ) [protected], [inherited]
```

**6.22.2.13 RandomGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::Random←
Generator [protected], [inherited]
```

**6.22.2.14 ReadFromExternalFile()**

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::ReadFromExternalFile (
    std::string path ) [virtual], [inherited]
```

**6.22.2.15 ReadFromFileGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::ReadFrom←
FileGenerator (
    std::string inputFile ) [protected], [inherited]
```

**6.22.2.16 ResetVisualizer()**

```
void Algorithm::DynamicArray::ResetVisualizer ( ) [private]
```

**6.22.2.17 Search()**

```
void Algorithm::DynamicArray::Search (
    int value )
```

**6.22.2.18 size()**

```
std::size_t Algorithm::DynamicArray::size ( ) const
```

### 6.22.2.19 Update()

```
void Algorithm::DynamicArray::Update (
    int index,
    int value )
```

### 6.22.2.20 UserDefined()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::UserDefined (
    std::string input ) [virtual], [inherited]
```

### 6.22.2.21 UserDefinedGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::UserDefined<-
Generator (
    std::string input ) [protected], [inherited]
```

## 6.22.3 Member Data Documentation

### 6.22.3.1 animController

```
Animation::AnimationController<DArrayAnimation >::Ptr Algorithm::Algorithm< GUI::DynamicArray ,
, DArrayAnimation >::animController [protected], [inherited]
```

### 6.22.3.2 codeHighlighter

```
GUI::CodeHighlighter::Ptr Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::code<-
Highlighter [protected], [inherited]
```

### 6.22.3.3 maxN

```
constexpr int Algorithm::DynamicArray::maxN = 16 [static], [constexpr]
```

#### 6.22.3.4 visualizer

```
GUI::DynamicArray Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::visualizer  
[protected], [inherited]
```

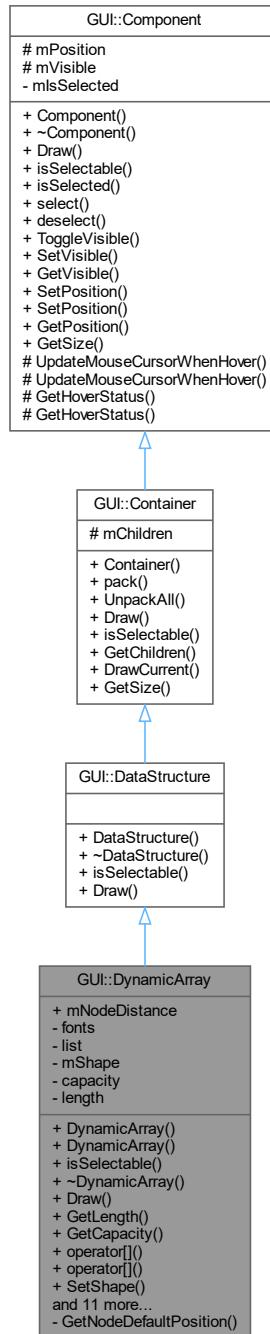
The documentation for this class was generated from the following files:

- src/Algorithms/Array/[DynamicArray.hpp](#)
- src/Algorithms/Array/[DynamicArray.cpp](#)

## 6.23 GUI::DynamicArray Class Reference

```
#include <DynamicArray.hpp>
```

Inheritance diagram for GUI::DynamicArray:



## Public Types

- `typedef std::shared_ptr< DataStructure > Ptr`

## Public Member Functions

- `DynamicArray ()`

- `DynamicArray (FontHolder *fonts)`
- `bool IsSelectable () const`
- `~DynamicArray ()`
- `void Draw (Vector2 base=(Vector2){0, 0}, float t=1.0f, bool init=false)`
- `std::size_t GetLength () const`
- `std::size_t GetCapacity () const`
- `GUI::Node & operator[] (std::size_t index)`
- `const GUI::Node & operator[] (std::size_t index) const`
- `void SetShape (GUI::Node::Shape shape)`
- `GUI::Node::Shape GetShape () const`
- `void Reserve (std::size_t size)`
- `void Resize (std::size_t size)`
- `void Clear ()`
- `std::vector< GUI::Node > & GetList ()`
- `GUI::Node GenerateNode (int value)`
- `void Import (std::vector< int > nodes)`
- `void InsertNode (std::size_t index, GUI::Node node, bool rePosition=true)`
- `void Relayout ()`
- `void DeleteNode (std::size_t index, bool rePosition=true)`
- `std::size_t GetCapacityFromLength (std::size_t length)`
- `virtual void Draw (Vector2 base)`
- `void pack (Component::Ptr component)`
- `void UnpackAll ()`
- `std::vector< Component::Ptr > GetChildren ()`
- `virtual void DrawCurrent (Vector2 base)`
- `virtual Vector2 GetSize ()`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2GetPosition ()`

## Static Public Attributes

- `static constexpr float mNodeDistance = 20`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- Vector2 [GetNodeDefaultPosition](#) (std::size\_t index)

## Private Attributes

- FontHolder \* [fonts](#)
- std::vector< [GUI::Node](#) > [list](#)
- [GUI::Node::Shape](#) [mShape](#)
- std::size\_t [capacity](#)
- std::size\_t [length](#)
- bool [mIsSelected](#)

### 6.23.1 Member Typedef Documentation

#### 6.23.1.1 Ptr

```
typedef std::shared_ptr< DataStructure > GUI::DataStructure::Ptr [inherited]
```

### 6.23.2 Constructor & Destructor Documentation

#### 6.23.2.1 DynamicArray() [1/2]

```
GUI::DynamicArray::DynamicArray ( )
```

#### 6.23.2.2 DynamicArray() [2/2]

```
GUI::DynamicArray::DynamicArray ( FontHolder * fonts )
```

#### 6.23.2.3 ~DynamicArray()

```
GUI::DynamicArray::~DynamicArray ( )
```

### 6.23.3 Member Function Documentation

### 6.23.3.1 Clear()

```
void GUI::DynamicArray::Clear ( )
```

### 6.23.3.2 DeleteNode()

```
void GUI::DynamicArray::DeleteNode (
    std::size_t index,
    bool rePosition = true )
```

### 6.23.3.3 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.23.3.4 Draw() [1/2]

```
void GUI::DataStructure::Draw (
    Vector2 base ) [virtual], [inherited]
```

Reimplemented from [GUI::Container](#).

### 6.23.3.5 Draw() [2/2]

```
void GUI::DynamicArray::Draw (
    Vector2 base = (Vector2){0, 0},
    float t = 1.0f,
    bool init = false )
```

### 6.23.3.6 DrawCurrent()

```
void GUI::Container::DrawCurrent (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.23.3.7 GenerateNode()

```
GUI::Node GUI::DynamicArray::GenerateNode (
    int value )
```

### 6.23.3.8 GetCapacity()

```
std::size_t GUI::DynamicArray::GetCapacity ( ) const
```

### 6.23.3.9 GetCapacityFromLength()

```
std::size_t GUI::DynamicArray::GetCapacityFromLength (
    std::size_t length )
```

### 6.23.3.10 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( ) [inherited]
```

### 6.23.3.11 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.23.3.12 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.23.3.13 GetLength()

```
std::size_t GUI::DynamicArray::GetLength ( ) const
```

### 6.23.3.14 `GetList()`

```
std::vector< GUI::Node > & GUI::DynamicArray::GetList ( )
```

### 6.23.3.15 `GetNodeDefaultPosition()`

```
Vector2 GUI::DynamicArray::GetNodeDefaultPosition (   
    std::size_t index ) [private]
```

### 6.23.3.16 `GetPosition()`

```
Vector2 GUI::Component::GetPosition ( ) [inherited]
```

### 6.23.3.17 `GetShape()`

```
GUI::Node::Shape GUI::DynamicArray::GetShape ( ) const
```

### 6.23.3.18 `GetSize()`

```
Vector2 GUI::Container::GetSize ( ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

### 6.23.3.19 `GetVisible()`

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

### 6.23.3.20 `Import()`

```
void GUI::DynamicArray::Import (   
    std::vector< int > nodes )
```

### 6.23.3.21 InsertNode()

```
void GUI::DynamicArray::InsertNode (
    std::size_t index,
    GUI::Node node,
    bool rePosition = true )
```

### 6.23.3.22 isSelectable()

```
bool GUI::DynamicArray::isSelectable () const [virtual]
```

Reimplemented from [GUI::DataStructure](#).

### 6.23.3.23 isSelected()

```
bool GUI::Component::isSelected () const [inherited]
```

### 6.23.3.24 operator[](1/2)

```
GUI::Node & GUI::DynamicArray::operator[] (
    std::size_t index )
```

### 6.23.3.25 operator[](2/2)

```
const GUI::Node & GUI::DynamicArray::operator[] (
    std::size_t index ) const
```

### 6.23.3.26 pack()

```
void GUI::Container::pack (
    Component::Ptr component ) [inherited]
```

### 6.23.3.27 Relayout()

```
void GUI::DynamicArray::Relayout ()
```

**6.23.3.28 Reserve()**

```
void GUI::DynamicArray::Reserve (
    std::size_t size )
```

**6.23.3.29 Resize()**

```
void GUI::DynamicArray::Resize (
    std::size_t size )
```

**6.23.3.30 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.23.3.31 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.23.3.32 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

**6.23.3.33 SetShape()**

```
void GUI::DynamicArray::SetShape (
    GUI::Node::Shape shape )
```

**6.23.3.34 SetVisible()**

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.23.3.35 ToggleVisible()

```
void GUI::Component::ToggleVisible () [virtual], [inherited]
```

### 6.23.3.36 UnpackAll()

```
void GUI::Container::UnpackAll () [inherited]
```

### 6.23.3.37 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.23.3.38 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.23.4 Member Data Documentation

### 6.23.4.1 capacity

```
std::size_t GUI::DynamicArray::capacity [private]
```

### 6.23.4.2 fonts

```
FontHolder* GUI::DynamicArray::fonts [private]
```

#### 6.23.4.3 length

```
std::size_t GUI::DynamicArray::length [private]
```

#### 6.23.4.4 list

```
std::vector< GUI::Node > GUI::DynamicArray::list [private]
```

#### 6.23.4.5 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

#### 6.23.4.6 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.23.4.7 mNodeDistance

```
constexpr float GUI::DynamicArray::mNodeDistance = 20 [static], [constexpr]
```

#### 6.23.4.8 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.23.4.9 mShape

```
GUI::Node::Shape GUI::DynamicArray::mShape [private]
```

### 6.23.4.10 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

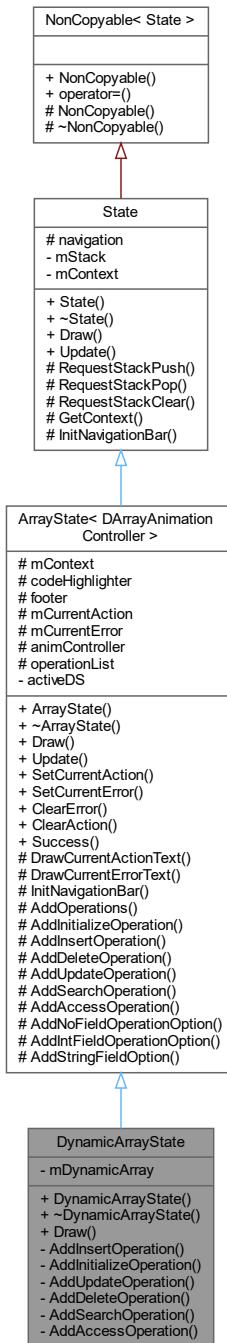
The documentation for this class was generated from the following files:

- src/Components/Visualization/[DynamicArray.hpp](#)
- src/Components/Visualization/[DynamicArray.cpp](#)

## 6.24 DynamicArrayState Class Reference

```
#include <DynamicArrayState.hpp>
```

Inheritance diagram for DynamicArrayState:



## Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `DynamicArrayState (StateStack &stack, Context context)`

- `~DynamicArrayState ()`
- `void Draw ()`
- `virtual bool Update (float dt)`
- `virtual void SetCurrentAction (std::string action)`
- `virtual void SetCurrentError (std::string error)`
- `virtual void ClearError ()`
- `virtual void ClearAction ()`
- `virtual void Success ()`

## Protected Member Functions

- `virtual void DrawCurrentActionText ()`
- `virtual void DrawCurrentErrorText ()`
- `void InitNavigationBar ()`
- `virtual void AddOperations ()`
- `virtual void AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void() > action)`
- `virtual void AddIntFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >) > action)`
- `virtual void AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >) > action)`
- `void RequestStackPush (States::ID stateID)`
- `void RequestStackPop ()`
- `void RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< DArrayAnimationController > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Member Functions

- `void AddInsertOperation ()`
- `void AddInitializeOperation ()`
- `void AddUpdateOperation ()`
- `void AddDeleteOperation ()`
- `void AddSearchOperation ()`
- `void AddAccessOperation ()`

## Private Attributes

- `Algorithm::DynamicArray mDynamicArray`
- `DataStructures::ID activeDS`
- `StateStack * mStack`

## 6.24.1 Member Typedef Documentation

### 6.24.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

## 6.24.2 Constructor & Destructor Documentation

### 6.24.2.1 DynamicArrayState()

```
DynamicArrayState::DynamicArrayState (
    StateStack & stack,
    Context context )
```

### 6.24.2.2 ~DynamicArrayState()

```
DynamicArrayState::~DynamicArrayState ( )
```

## 6.24.3 Member Function Documentation

### 6.24.3.1 AddAccessOperation()

```
void DynamicArrayState::AddAccessOperation ( ) [private], [virtual]
```

Reimplemented from [ArrayState< DArrayAnimationController >](#).

### 6.24.3.2 AddDeleteOperation()

```
void DynamicArrayState::AddDeleteOperation ( ) [private], [virtual]
```

Reimplemented from [ArrayState< DArrayAnimationController >](#).

#### 6.24.3.3 AddInitializeOperation()

```
void DynamicArrayState::AddInitializeOperation ( ) [private], [virtual]
```

Reimplemented from [AppState< DArrayAnimationController >](#).

#### 6.24.3.4 AddInsertOperation()

```
void DynamicArrayState::AddInsertOperation ( ) [private], [virtual]
```

Reimplemented from [AppState< DArrayAnimationController >](#).

#### 6.24.3.5 AddIntFieldOperationOption()

```
void AppState< DArrayAnimationController >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

#### 6.24.3.6 AddNoFieldOperationOption()

```
void AppState< DArrayAnimationController >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual], [inherited]
```

#### 6.24.3.7 AddOperations()

```
void AppState< DArrayAnimationController >::AddOperations [protected], [virtual], [inherited]
```

#### 6.24.3.8 AddSearchOperation()

```
void DynamicArrayState::AddSearchOperation ( ) [private], [virtual]
```

Reimplemented from [AppState< DArrayAnimationController >](#).

### 6.24.3.9 AddStringFieldOption()

```
void ArrayState< DArrayAnimationController >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

### 6.24.3.10 AddUpdateOperation()

```
void DynamicArrayState::AddUpdateOperation () [private], [virtual]
```

Reimplemented from [ArrayState< DArrayAnimationController >](#).

### 6.24.3.11 ClearAction()

```
void ArrayState< DArrayAnimationController >::ClearAction [inline], [virtual], [inherited]
```

### 6.24.3.12 ClearError()

```
void ArrayState< DArrayAnimationController >::ClearError [inline], [virtual], [inherited]
```

### 6.24.3.13 Draw()

```
void DynamicArrayState::Draw () [virtual]
```

Implements [ArrayState< DArrayAnimationController >](#).

### 6.24.3.14 DrawCurrentActionText()

```
void ArrayState< DArrayAnimationController >::DrawCurrentActionText [inline], [protected],
[virtual], [inherited]
```

**6.24.3.15 DrawCurrentErrorText()**

```
void ArrayType< DArrayAnimationController >::DrawCurrentErrorText [inline], [protected],  
[virtual], [inherited]
```

**6.24.3.16 GetContext()**

```
State::Context State::GetContext () const [protected], [inherited]
```

**6.24.3.17 InitNavigationBar()**

```
void ArrayType< DArrayAnimationController >::InitNavigationBar [protected], [inherited]
```

**6.24.3.18 RequestStackClear()**

```
void State::RequestStackClear () [protected], [inherited]
```

**6.24.3.19 RequestStackPop()**

```
void State::RequestStackPop () [protected], [inherited]
```

**6.24.3.20 RequestStackPush()**

```
void State::RequestStackPush (  
    States::ID stateID ) [protected], [inherited]
```

**6.24.3.21 SetCurrentAction()**

```
void ArrayType< DArrayAnimationController >::SetCurrentAction (  
    std::string action ) [inline], [virtual], [inherited]
```

### 6.24.3.22 SetCurrentError()

```
void ArrayType< DArrayAnimationController >::SetCurrentError (
    std::string error) [inline], [virtual], [inherited]
```

### 6.24.3.23 Success()

```
void ArrayType< DArrayAnimationController >::Success [inline], [virtual], [inherited]
```

### 6.24.3.24 Update()

```
bool ArrayType< DArrayAnimationController >::Update (
    float dt) [virtual], [inherited]
```

Implements [State](#).

## 6.24.4 Member Data Documentation

### 6.24.4.1 activeDS

```
DataStructures::ID ArrayType< DArrayAnimationController >::activeDS [private], [inherited]
```

### 6.24.4.2 animController

```
T::Ptr ArrayType< DArrayAnimationController >::animController [protected], [inherited]
```

### 6.24.4.3 codeHighlighter

```
GUI::CodeHighlighter::Ptr ArrayType< DArrayAnimationController >::codeHighlighter [protected], [inherited]
```

#### 6.24.4.4 footer

```
GUI::Footer< DArrayAnimationController > AppState< DArrayAnimationController >::footer  
[protected], [inherited]
```

#### 6.24.4.5 mContext

```
Context AppState< DArrayAnimationController >::mContext [protected], [inherited]
```

#### 6.24.4.6 mCurrentAction

```
std::string AppState< DArrayAnimationController >::mCurrentAction [protected], [inherited]
```

#### 6.24.4.7 mCurrentError

```
std::string AppState< DArrayAnimationController >::mCurrentError [protected], [inherited]
```

#### 6.24.4.8 mDynamicArray

```
Algorithm::DynamicArray DynamicArrayState::mDynamicArray [private]
```

#### 6.24.4.9 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.24.4.10 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

#### 6.24.4.11 operationList

```
GUI::OperationList ArrayState< DArrayAnimationController >::operationList [protected], [inherited]
```

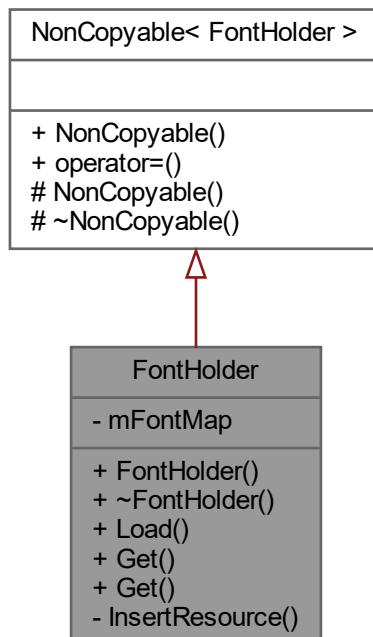
The documentation for this class was generated from the following files:

- src/States/Array/[DynamicArrayList.hpp](#)
- src/States/Array/[DynamicArrayList.cpp](#)

## 6.25 FontHolder Class Reference

```
#include <FontHolder.hpp>
```

Inheritance diagram for FontHolder:



### Public Member Functions

- [FontHolder \(\)](#)
- [~FontHolder \(\)](#)
- void [Load \(Fonts::ID id, const std::string &filename\)](#)
- [Font & Get \(Fonts::ID id\)](#)
- [const Font & Get \(Fonts::ID id\) const](#)

## Private Member Functions

- void `InsertResource (Fonts::ID id, std::unique_ptr<Font> font)`

## Private Attributes

- `std::map<Fonts::ID, std::unique_ptr<Font>> mFontMap`

### 6.25.1 Constructor & Destructor Documentation

#### 6.25.1.1 `FontHolder()`

```
FontHolder::FontHolder ( )
```

#### 6.25.1.2 `~FontHolder()`

```
FontHolder::~FontHolder ( )
```

### 6.25.2 Member Function Documentation

#### 6.25.2.1 `Get() [1/2]`

```
Font & FontHolder::Get (
    Fonts::ID id )
```

#### 6.25.2.2 `Get() [2/2]`

```
const Font & FontHolder::Get (
    Fonts::ID id ) const
```

#### 6.25.2.3 `InsertResource()`

```
void FontHolder::InsertResource (
    Fonts::ID id,
    std::unique_ptr<Font> font ) [private]
```

#### 6.25.2.4 Load()

```
void FontHolder::Load (
    Fonts::ID id,
    const std::string & filename )
```

### 6.25.3 Member Data Documentation

#### 6.25.3.1 mFontMap

```
std::map< Fonts::ID, std::unique_ptr< Font > > FontHolder::mFontMap [private]
```

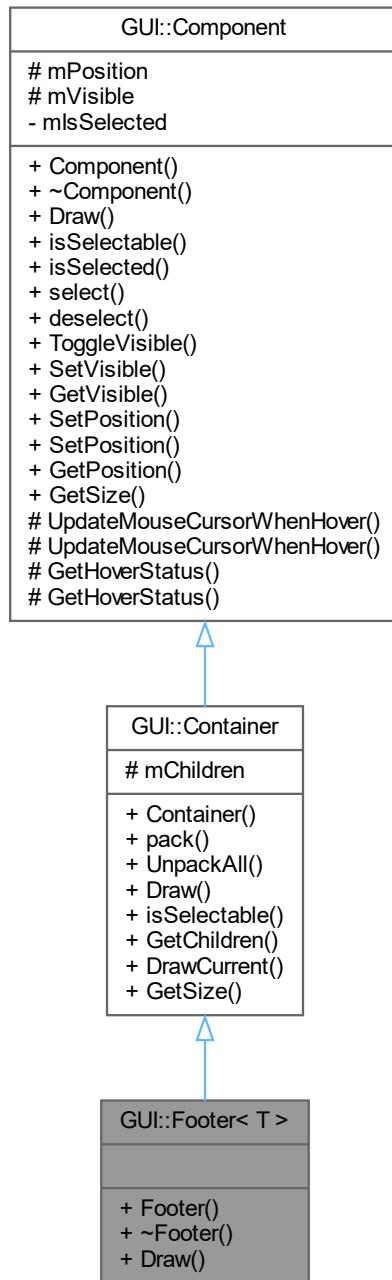
The documentation for this class was generated from the following files:

- src/[FontHolder.hpp](#)
- src/[FontHolder.cpp](#)

## 6.26 GUI::Footer< T > Class Template Reference

```
#include <Footer.hpp>
```

Inheritance diagram for GUI::Footer< T >:



## Public Types

- `typedef std::shared_ptr< Container > Ptr`

## Public Member Functions

- `Footer ()`

- `~Footer ()`
- void `Draw (T *animController, Vector2 base=(Vector2){0, 0})`
- void `pack (Component::Ptr component)`
- void `UnpackAll ()`
- virtual void `Draw (Vector2 base=(Vector2){0, 0})`
- virtual bool `isSelectable () const`
- std::vector< Component::Ptr > `GetChildren ()`
- virtual void `DrawCurrent (Vector2 base)`
- virtual Vector2 `GetSize ()`
- bool `isSelected () const`
- virtual void `select ()`
- virtual void `deselect ()`
- virtual void `ToggleVisible ()`
- virtual void `SetVisible (bool visible)`
- virtual bool `GetVisible ()`
- void `SetPosition (float x, float y)`
- void `SetPosition (Vector2 position)`
- Vector2 `GetPosition ()`

## Protected Member Functions

- virtual void `UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- virtual void `UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- virtual bool `GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- virtual bool `GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- std::vector< Component::Ptr > `mChildren`
- Vector2 `mPosition`
- bool `mVisible`

## Private Attributes

- bool `mIsSelected`

### 6.26.1 Member Typedef Documentation

#### 6.26.1.1 Ptr

```
typedef std::shared_ptr< Container > GUI::Container::Ptr [inherited]
```

### 6.26.2 Constructor & Destructor Documentation

### 6.26.2.1 Footer()

```
template<typename T >
GUI::Footer< T >::Footer
```

### 6.26.2.2 ~Footer()

```
template<typename T >
GUI::Footer< T >::~Footer
```

## 6.26.3 Member Function Documentation

### 6.26.3.1 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.26.3.2 Draw() [1/2]

```
template<typename T >
void GUI::Footer< T >::Draw (
    T * animController,
    Vector2 base = (Vector2){0, 0} )
```

### 6.26.3.3 Draw() [2/2]

```
void GUI::Container::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::DataStructure](#), and [GUI::OperationList](#).

### 6.26.3.4 DrawCurrent()

```
void GUI::Container::DrawCurrent (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.26.3.5 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( ) [inherited]
```

### 6.26.3.6 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.26.3.7 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.26.3.8 GetPosition()

```
Vector2 GUI::Component::GetPosition ( ) [inherited]
```

### 6.26.3.9 GetSize()

```
Vector2 GUI::Container::GetSize ( ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

### 6.26.3.10 GetVisible()

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

**6.26.3.11 isSelectable()**

```
bool GUI::Container::isSelectable () const [virtual], [inherited]
```

Implements [GUI::Component](#).

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DataStructure](#), [GUI::DoublyLinkedList](#), [GUI::DynamicArray](#), [GUI::LinkedList](#), and [GUI::SinglyLinkedList](#).

**6.26.3.12 isSelected()**

```
bool GUI::Component::isSelected () const [inherited]
```

**6.26.3.13 pack()**

```
void GUI::Container::pack (
    Component::Ptr component ) [inherited]
```

**6.26.3.14 select()**

```
void GUI::Component::select () [virtual], [inherited]
```

**6.26.3.15 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.26.3.16 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

### 6.26.3.17 SetVisible()

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.26.3.18 ToggleVisible()

```
void GUI::Component::ToggleVisible () [virtual], [inherited]
```

### 6.26.3.19 UnpackAll()

```
void GUI::Container::UnpackAll () [inherited]
```

### 6.26.3.20 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.26.3.21 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.26.4 Member Data Documentation

### 6.26.4.1 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

#### 6.26.4.2 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.26.4.3 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.26.4.4 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

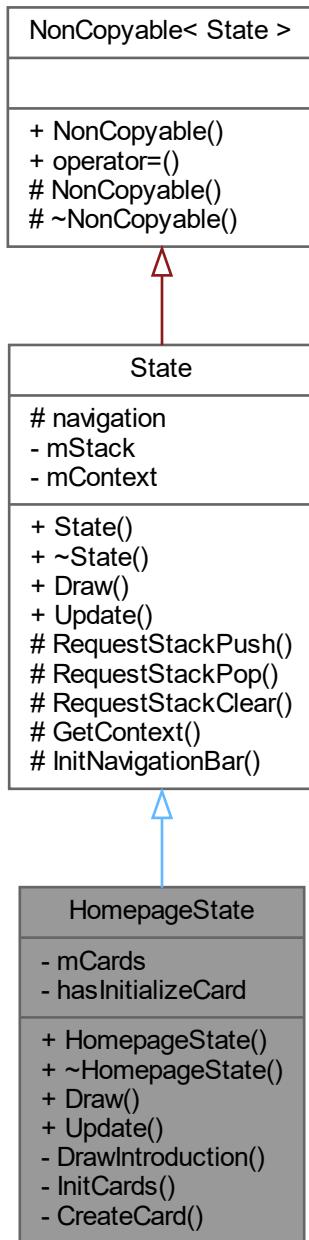
The documentation for this class was generated from the following file:

- src/Components/Common/[Footer.hpp](#)

## 6.27 HomepageState Class Reference

```
#include <HomepageState.hpp>
```

Inheritance diagram for HomepageState:



## Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `HomepageState (StateStack &stack, Context context)`

- `~HomepageState ()`
- `void Draw ()`
- `bool Update (float dt)`

## Protected Member Functions

- `void RequestStackPush (States::ID stateID)`
- `void RequestStackPop ()`
- `void RequestStackClear ()`
- `Context GetContext () const`
- `void InitNavigationBar ()`

## Protected Attributes

- `GUI::NavigationBar navigation`

## Private Member Functions

- `void DrawIntroduction ()`
- `void InitCards ()`
- `void CreateCard (States::ID stateID, std::string title, Textures::ID textureID, int x, int y)`

## Private Attributes

- `GUI::Container mCards`
- `bool hasInitializeCard`
- `StateStack * mStack`
- `Context mContext`

### 6.27.1 Member Typedef Documentation

#### 6.27.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

### 6.27.2 Constructor & Destructor Documentation

### 6.27.2.1 HomepageState()

```
HomepageState::HomepageState (
    StateStack & stack,
    Context context )
```

### 6.27.2.2 ~HomepageState()

```
HomepageState::~HomepageState ( )
```

## 6.27.3 Member Function Documentation

### 6.27.3.1 CreateCard()

```
void HomepageState::CreateCard (
    States::ID stateID,
    std::string title,
    Textures::ID textureID,
    int x,
    int y ) [private]
```

### 6.27.3.2 Draw()

```
void HomepageState::Draw ( ) [virtual]
```

Implements [State](#).

### 6.27.3.3 DrawIntroduction()

```
void HomepageState::DrawIntroduction ( ) [private]
```

### 6.27.3.4 GetContext()

```
State::Context State::GetContext ( ) const [protected], [inherited]
```

### 6.27.3.5 InitCards()

```
void HomepageState::InitCards ( ) [private]
```

### 6.27.3.6 InitNavigationBar()

```
void State::InitNavigationBar ( ) [protected], [inherited]
```

### 6.27.3.7 RequestStackClear()

```
void State::RequestStackClear ( ) [protected], [inherited]
```

### 6.27.3.8 RequestStackPop()

```
void State::RequestStackPop ( ) [protected], [inherited]
```

### 6.27.3.9 RequestStackPush()

```
void State::RequestStackPush (
    States::ID stateID ) [protected], [inherited]
```

### 6.27.3.10 Update()

```
bool HomepageState::Update (
    float dt ) [virtual]
```

Implements [State](#).

## 6.27.4 Member Data Documentation

### 6.27.4.1 hasInitializeCard

```
bool HomepageState::hasInitializeCard [private]
```

#### 6.27.4.2 mCards

```
GUI::Container HomepageState::mCards [private]
```

#### 6.27.4.3 mContext

```
Context State::mContext [private], [inherited]
```

#### 6.27.4.4 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.27.4.5 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

The documentation for this class was generated from the following files:

- src/States/[HomepageState.hpp](#)
- src/States/[HomepageState.cpp](#)

## 6.28 CategoryInfo::Info Struct Reference

### Public Member Functions

- [Info \(Category::ID categoryID, std::vector< DataStructures::ID > mDS, std::string name\)](#)

### Public Attributes

- [Category::ID categoryID](#)
- [std::string categoryName](#)
- [std::vector< DataStructures::ID > mDS](#)

#### 6.28.1 Constructor & Destructor Documentation

### 6.28.1.1 Info()

```
CategoryInfo::Info::Info (
    Category::ID categoryID,
    std::vector< DataStructures::ID > mDS,
    std::string name )
```

## 6.28.2 Member Data Documentation

### 6.28.2.1 categoryID

```
Category::ID CategoryInfo::Info::categoryID
```

### 6.28.2.2 categoryName

```
std::string CategoryInfo::Info::categoryName
```

### 6.28.2.3 mDS

```
std::vector< DataStructures::ID > CategoryInfo::Info::mDS
```

The documentation for this struct was generated from the following files:

- src/Identifiers/CategoryInfo.hpp
- src/Identifiers/CategoryInfo.cpp

## 6.29 DSInfo::Info Struct Reference

### Public Member Functions

- [Info \(DataStructures::ID ID, States::ID stateID, Category::ID categoryID, Textures::ID thumbnail, std::string name, std::string abbr\)](#)

### Public Attributes

- [DataStructures::ID ID](#)
- [States::ID stateID](#)
- [Category::ID categoryID](#)
- [Textures::ID thumbnail](#)
- [std::string name](#)
- [std::string abbr](#)

## 6.29.1 Constructor & Destructor Documentation

### 6.29.1.1 Info()

```
DSInfo::Info::Info (
    DataStructures::ID ID,
    States::ID stateID,
    Category::ID categoryID,
    Textures::ID thumbnail,
    std::string name,
    std::string abbr )
```

## 6.29.2 Member Data Documentation

### 6.29.2.1 abbr

```
std::string DSInfo::Info::abbr
```

### 6.29.2.2 categoryID

```
Category::ID DSInfo::Info::categoryID
```

### 6.29.2.3 ID

```
DataStructures::ID DSInfo::Info::ID
```

### 6.29.2.4 name

```
std::string DSInfo::Info::name
```

### 6.29.2.5 stateID

```
States::ID DSInfo::Info::stateID
```

### 6.29.2.6 thumbnail

`Textures::ID DSInfo::Info::thumbnail`

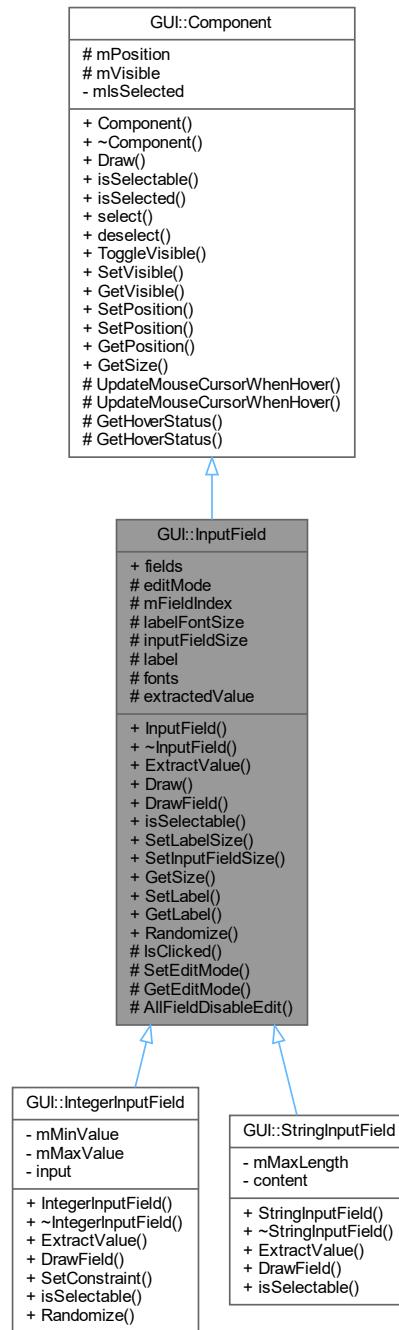
The documentation for this struct was generated from the following files:

- src/Identifiers/[DSInfo.hpp](#)
- src/Identifiers/[DSInfo.cpp](#)

## 6.30 GUI::InputField Class Reference

`#include <InputField.hpp>`

Inheritance diagram for GUI::InputField:



## Public Types

- `typedef std::shared_ptr< InputField > Ptr`

## Public Member Functions

- `InputField (FontHolder *fonts)`

- `~InputField ()`
- `virtual std::string ExtractValue ()=0`
- `virtual void Draw (Vector2 base=(Vector2){0, 0})`
- `virtual void DrawField (Vector2 base=(Vector2){0, 0})=0`
- `virtual bool IsSelectable () const`
- `virtual void SetLabelSize (float fontSize)`
- `virtual void SetInputFieldSize (Vector2 size)`
- `virtual Vector2 GetSize ()`
- `virtual void SetLabel (std::string labelContent)`
- `virtual std::string GetLabel () const`
- `virtual void Randomize ()`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2GetPosition ()`

## Static Public Attributes

- `static std::vector< bool > fields`

## Protected Member Functions

- `virtual bool IsClicked (Vector2 base=(Vector2){0, 0}) const`
- `virtual void SetEditMode (bool canEdit)`
- `virtual bool GetEditMode () const`
- `virtual void AllFieldDisableEdit ()`
- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `bool editMode`
- `std::size_t mFieldIndex`
- `float labelFontSize`
- `Vector2 inputFieldSize`
- `std::string label`
- `FontHolder * fonts`
- `std::string extractedValue`
- `Vector2 mPosition`
- `bool mVisible`

## Private Attributes

- `bool mIsSelected`

### 6.30.1 Member Typedef Documentation

#### 6.30.1.1 Ptr

```
typedef std::shared_ptr< InputField > GUI::InputField::Ptr
```

### 6.30.2 Constructor & Destructor Documentation

#### 6.30.2.1 InputField()

```
GUI::InputField::InputField (
    FontHolder * fonts )
```

#### 6.30.2.2 ~InputField()

```
GUI::InputField::~InputField ( )
```

### 6.30.3 Member Function Documentation

#### 6.30.3.1 AllFieldDisableEdit()

```
void GUI::InputField::AllFieldDisableEdit ( ) [protected], [virtual]
```

#### 6.30.3.2 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

#### 6.30.3.3 Draw()

```
void GUI::InputField::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Component](#).

### 6.30.3.4 DrawField()

```
virtual void GUI::InputField::DrawField (
    Vector2 base = (Vector2){0, 0} ) [pure virtual]
```

Implemented in [GUI::IntegerField](#), and [GUI::StringInputField](#).

### 6.30.3.5 ExtractValue()

```
virtual std::string GUI::InputField::ExtractValue () [pure virtual]
```

Implemented in [GUI::IntegerField](#), and [GUI::StringInputField](#).

### 6.30.3.6 GetEditMode()

```
bool GUI::InputField::GetEditMode () const [protected], [virtual]
```

### 6.30.3.7 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.30.3.8 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.30.3.9 GetLabel()

```
std::string GUI::InputField::GetLabel () const [virtual]
```

### 6.30.3.10 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

### 6.30.3.11 GetSize()

```
Vector2 GUI::InputField::GetSize () [virtual]
```

Reimplemented from [GUI::Component](#).

### 6.30.3.12 GetVisible()

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

### 6.30.3.13 IsClicked()

```
bool GUI::InputField::IsClicked (
    Vector2 base = (Vector2){0, 0} ) const [protected], [virtual]
```

### 6.30.3.14 isSelectable()

```
bool GUI::InputField::isSelectable () const [virtual]
```

Implements [GUI::Component](#).

Reimplemented in [GUI::IntegerInputField](#), and [GUI::StringInputField](#).

### 6.30.3.15 isSelected()

```
bool GUI::Component::isSelected () const [inherited]
```

### 6.30.3.16 Randomize()

```
void GUI::InputField::Randomize () [virtual]
```

Reimplemented in [GUI::IntegerInputField](#).

**6.30.3.17 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.30.3.18 SetEditMode()**

```
void GUI::InputField::SetEditMode (
    bool canEdit ) [protected], [virtual]
```

**6.30.3.19 SetInputFieldSize()**

```
void GUI::InputField::SetInputFieldSize (
    Vector2 size ) [virtual]
```

**6.30.3.20 SetLabel()**

```
void GUI::InputField::SetLabel (
    std::string labelContent ) [virtual]
```

**6.30.3.21 SetLabelSize()**

```
void GUI::InputField::SetLabelSize (
    float fontSize ) [virtual]
```

**6.30.3.22 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.30.3.23 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

### 6.30.3.24 SetVisible()

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.30.3.25 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

### 6.30.3.26 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.30.3.27 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.30.4 Member Data Documentation

### 6.30.4.1 editMode

```
bool GUI::InputField::editMode [protected]
```

### 6.30.4.2 extractedValue

```
std::string GUI::InputField::extractedValue [protected]
```

#### 6.30.4.3 fields

```
std::vector< bool > GUI::InputField::fields [static]
```

#### 6.30.4.4 fonts

```
FontHolder* GUI::InputField::fonts [protected]
```

#### 6.30.4.5 inputFieldSize

```
Vector2 GUI::InputField::inputFieldSize [protected]
```

#### 6.30.4.6 label

```
std::string GUI::InputField::label [protected]
```

#### 6.30.4.7 labelFontSize

```
float GUI::InputField::labelFontSize [protected]
```

#### 6.30.4.8 mFieldIndex

```
std::size_t GUI::InputField::mFieldIndex [protected]
```

#### 6.30.4.9 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.30.4.10 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

### 6.30.4.11 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

The documentation for this class was generated from the following files:

- src/Components/Common/InputField.hpp
- src/Application.cpp
- src/Components/Common/InputField.cpp

## 6.31 **ArrayType< T >::IntegerInput Struct Reference**

```
#include <ArrayType.hpp>
```

### Public Attributes

- std::string [label](#)
- int [width](#)
- int [minValue](#)
- int [maxValue](#)

### 6.31.1 Member Data Documentation

#### 6.31.1.1 [label](#)

```
template<typename T >
std::string ArrayType< T >::IntegerInput::label
```

#### 6.31.1.2 [maxValue](#)

```
template<typename T >
int ArrayType< T >::IntegerInput::maxValue
```

#### 6.31.1.3 [minValue](#)

```
template<typename T >
int ArrayType< T >::IntegerInput::minValue
```

### 6.31.1.4 width

```
template<typename T >
int ArrayState< T >::IntegerInput::width
```

The documentation for this struct was generated from the following file:

- [src/States/Array/ArrayState.hpp](#)

## 6.32 LLState< T >::IntegerInput Struct Reference

```
#include <LLState.hpp>
```

### Public Attributes

- std::string [label](#)
- int [width](#)
- int [minValue](#)
- int [maxValue](#)

#### 6.32.1 Member Data Documentation

##### 6.32.1.1 [label](#)

```
template<typename T >
std::string LLState< T >::IntegerInput::label
```

##### 6.32.1.2 [maxValue](#)

```
template<typename T >
int LLState< T >::IntegerInput::maxValue
```

##### 6.32.1.3 [minValue](#)

```
template<typename T >
int LLState< T >::IntegerInput::minValue
```

### 6.32.1.4 width

```
template<typename T >
int LLState< T >::IntegerInput::width
```

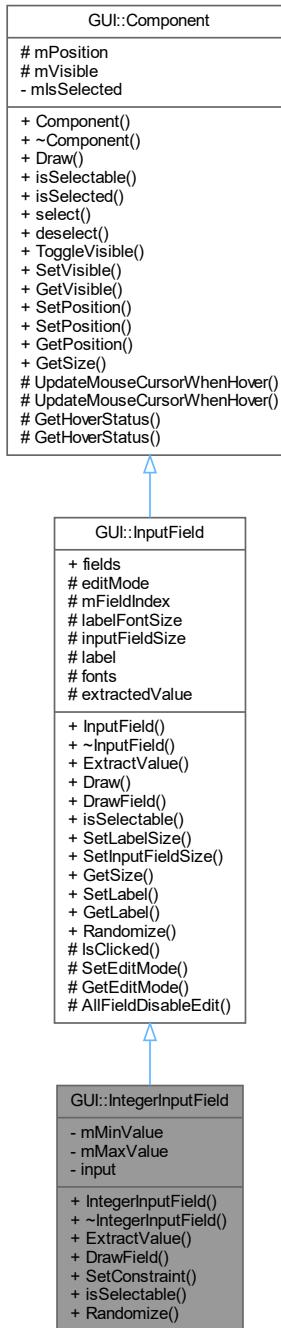
The documentation for this struct was generated from the following file:

- src/States/LinkedList/LLState.hpp

## 6.33 GUI::IntegerField Class Reference

```
#include <IntegerField.hpp>
```

Inheritance diagram for GUI::IntegerField:



## Public Types

- `typedef std::shared_ptr< IntegerIntegerField > Ptr`

## Public Member Functions

- `IntegerField (FontHolder *fonts)`

- `~IntegerInputField ()`
- `std::string ExtractValue ()`
- `void DrawField (Vector2 base=(Vector2){0, 0})`
- `void SetConstraint (int minValue, int maxValue)`
- `bool IsSelectable () const`
- `void Randomize ()`
- `virtual void Draw (Vector2 base=(Vector2){0, 0})`
- `virtual void SetLabelSize (float fontSize)`
- `virtual void SetInputFieldSize (Vector2 size)`
- `virtual Vector2 GetSize ()`
- `virtual void SetLabel (std::string labelContent)`
- `virtual std::string GetLabel () const`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Static Public Attributes

- `static std::vector< bool > fields`

## Protected Member Functions

- `virtual bool IsClicked (Vector2 base=(Vector2){0, 0}) const`
- `virtual void SetEditMode (bool canEdit)`
- `virtual bool GetEditMode () const`
- `virtual void AllFieldDisableEdit ()`
- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `bool editMode`
- `std::size_t mFieldIndex`
- `float labelFontSize`
- `Vector2 inputFieldSize`
- `std::string label`
- `FontHolder * fonts`
- `std::string extractedValue`
- `Vector2 mPosition`
- `bool mVisible`

## Private Attributes

- int `mMinValue`
- int `mMaxValue`
- int `input`
- bool `mlsSelected`

### 6.33.1 Member Typedef Documentation

#### 6.33.1.1 Ptr

```
typedef std::shared_ptr< IntegerInputField > GUI::IntegerField::Ptr
```

### 6.33.2 Constructor & Destructor Documentation

#### 6.33.2.1 IntegerInputField()

```
GUI::IntegerField::IntegerField ( FontHolder * fonts )
```

#### 6.33.2.2 ~IntegerInputField()

```
GUI::IntegerField::~IntegerField ( )
```

### 6.33.3 Member Function Documentation

#### 6.33.3.1 AllFieldDisableEdit()

```
void GUI::InputField::AllFieldDisableEdit ( ) [protected], [virtual], [inherited]
```

#### 6.33.3.2 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.33.3.3 Draw()

```
void GUI::InputField::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

### 6.33.3.4 DrawField()

```
void GUI::IntegerInputField::DrawField (
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Implements [GUI::InputField](#).

### 6.33.3.5 ExtractValue()

```
std::string GUI::IntegerInputField::ExtractValue ( ) [virtual]
```

Implements [GUI::InputField](#).

### 6.33.3.6 GetEditMode()

```
bool GUI::InputField::GetEditMode ( ) const [protected], [virtual], [inherited]
```

### 6.33.3.7 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.33.3.8 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

**6.33.3.9 GetLabel()**

```
std::string GUI::InputField::GetLabel ( ) const [virtual], [inherited]
```

**6.33.3.10GetPosition()**

```
Vector2 GUI::Component::GetPosition ( ) [inherited]
```

**6.33.3.11GetSize()**

```
Vector2 GUI::InputField::GetSize ( ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

**6.33.3.12GetVisible()**

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

**6.33.3.13IsClicked()**

```
bool GUI::InputField::IsClicked ( 
    Vector2 base = (Vector2){0, 0} ) const [protected], [virtual], [inherited]
```

**6.33.3.14isSelectable()**

```
bool GUI::IntegerField::isSelectable ( ) const [virtual]
```

Reimplemented from [GUI::InputField](#).

**6.33.3.15isSelected()**

```
bool GUI::Component::isSelected ( ) const [inherited]
```

### 6.33.3.16 Randomize()

```
void GUI::IntegerField::Randomize ( ) [virtual]
```

Reimplemented from [GUI::InputField](#).

### 6.33.3.17 select()

```
void GUI::Component::select ( ) [virtual], [inherited]
```

### 6.33.3.18 SetConstraint()

```
void GUI::IntegerField::SetConstraint (
    int minValue,
    int maxValue )
```

### 6.33.3.19 SetEditMode()

```
void GUI::InputField::SetEditMode (
    bool canEdit ) [protected], [virtual], [inherited]
```

### 6.33.3.20 SetInputFieldSize()

```
void GUI::InputField::SetInputFieldSize (
    Vector2 size ) [virtual], [inherited]
```

### 6.33.3.21 SetLabel()

```
void GUI::InputField::SetLabel (
    std::string labelText ) [virtual], [inherited]
```

### 6.33.3.22 SetLabelSize()

```
void GUI::InputField::SetLabelSize (
    float fontSize ) [virtual], [inherited]
```

**6.33.3.23 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y )  [inherited]
```

**6.33.3.24 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position )  [inherited]
```

**6.33.3.25 SetVisible()**

```
void GUI::Component::SetVisible (
    bool visible )  [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

**6.33.3.26 ToggleVisible()**

```
void GUI::Component::ToggleVisible ( )  [virtual], [inherited]
```

**6.33.3.27 UpdateMouseCursorWhenHover() [1/2]**

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover )  [protected], [virtual], [inherited]
```

**6.33.3.28 UpdateMouseCursorWhenHover() [2/2]**

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover )  [protected], [virtual], [inherited]
```

## 6.33.4 Member Data Documentation

### 6.33.4.1 editMode

```
bool GUI::InputField::editMode [protected], [inherited]
```

### 6.33.4.2 extractedValue

```
std::string GUI::InputField::extractedValue [protected], [inherited]
```

### 6.33.4.3 fields

```
std::vector< bool > GUI::InputField::fields [static], [inherited]
```

### 6.33.4.4 fonts

```
FontHolder* GUI::InputField::fonts [protected], [inherited]
```

### 6.33.4.5 input

```
int GUI::IntegerInputField::input [private]
```

### 6.33.4.6 inputFieldSize

```
Vector2 GUI::InputField::inputFieldSize [protected], [inherited]
```

### 6.33.4.7 label

```
std::string GUI::InputField::label [protected], [inherited]
```

#### 6.33.4.8 **labelFontSize**

```
float GUI::InputField::labelFontSize [protected], [inherited]
```

#### 6.33.4.9 **mFieldIndex**

```
std::size_t GUI::InputField::mFieldIndex [protected], [inherited]
```

#### 6.33.4.10 **mIsSelected**

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.33.4.11 **m.MaxValue**

```
int GUI::IntegerInputField::m.MaxValue [private]
```

#### 6.33.4.12 **m.MinValue**

```
int GUI::IntegerInputField::m.MinValue [private]
```

#### 6.33.4.13 **mPosition**

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.33.4.14 **mVisible**

```
bool GUI::Component::mVisible [protected], [inherited]
```

The documentation for this class was generated from the following files:

- src/Components/Common/[IntegerField.hpp](#)
- src/Components/Common/[IntegerField.cpp](#)

## 6.34 Core::List< T >::iterator Class Reference

```
#include <List.hpp>
```

### Public Types

- using `iterator_category` = std::bidirectional\_iterator\_tag
- using `value_type` = T
- using `difference_type` = std::ptrdiff\_t
- using `pointer` = T \*
- using `reference` = T &

### Public Member Functions

- `iterator ()`
- `iterator (Node< value_type > *const &p)`
- `reference operator* () const`
- `pointer operator-> () const`
- `iterator & operator++ ()`
- `iterator operator++ (int)`
- `iterator & operator-- ()`
- `iterator operator-- (int)`
- `iterator & operator= (Node< value_type > *const &p)`
- `iterator operator+ (const int &step)`
- `iterator operator- (const int &step)`
- `bool operator== (const iterator &it) const`
- `bool operator!= (const iterator &it) const`
- `void swap (iterator &other)`

### Private Attributes

- `Node< T > * ptr`

### Friends

- class `List`

### 6.34.1 Member Typedef Documentation

#### 6.34.1.1 difference\_type

```
template<typename T >
using Core::List< T >::iterator::difference_type = std::ptrdiff_t
```

### 6.34.1.2 iterator\_category

```
template<typename T >
using Core::List< T >::iterator::iterator_category = std::bidirectional_iterator_tag
```

### 6.34.1.3 pointer

```
template<typename T >
using Core::List< T >::iterator::pointer = T*
```

### 6.34.1.4 reference

```
template<typename T >
using Core::List< T >::iterator::reference = T&
```

### 6.34.1.5 value\_type

```
template<typename T >
using Core::List< T >::iterator::value_type = T
```

## 6.34.2 Constructor & Destructor Documentation

### 6.34.2.1 iterator() [1/2]

```
template<typename T >
Core::List< T >::iterator::iterator ( ) [inline]
```

### 6.34.2.2 iterator() [2/2]

```
template<typename T >
Core::List< T >::iterator::iterator (
    Node< value_type > *const & p ) [inline]
```

## 6.34.3 Member Function Documentation

### 6.34.3.1 operator"!=()

```
template<typename T >
bool Core::List< T >::iterator::operator!= (
    const iterator & it ) const [inline]
```

### 6.34.3.2 operator\*()

```
template<typename T >
reference Core::List< T >::iterator::operator* ( ) const [inline]
```

### 6.34.3.3 operator+()

```
template<typename T >
iterator Core::List< T >::iterator::operator+ (
    const int & step ) [inline]
```

### 6.34.3.4 operator++() [1/2]

```
template<typename T >
iterator & Core::List< T >::iterator::operator++ ( ) [inline]
```

### 6.34.3.5 operator++() [2/2]

```
template<typename T >
iterator Core::List< T >::iterator::operator++ (
    int ) [inline]
```

### 6.34.3.6 operator-()

```
template<typename T >
iterator Core::List< T >::iterator::operator- (
    const int & step ) [inline]
```

### 6.34.3.7 operator--() [1/2]

```
template<typename T >
iterator & Core::List< T >::iterator::operator-- ( ) [inline]
```

### 6.34.3.8 operator--() [2/2]

```
template<typename T >
iterator Core::List< T >::iterator::operator-- (
    int ) [inline]
```

### 6.34.3.9 operator->()

```
template<typename T >
pointer Core::List< T >::iterator::operator-> ( ) const [inline]
```

### 6.34.3.10 operator=()

```
template<typename T >
iterator & Core::List< T >::iterator::operator= (
    Node< value_type > *const & p ) [inline]
```

### 6.34.3.11 operator==( )

```
template<typename T >
bool Core::List< T >::iterator::operator== (
    const iterator & it ) const [inline]
```

### 6.34.3.12 swap()

```
template<typename T >
void Core::List< T >::iterator::swap (
    iterator & other ) [inline]
```

## 6.34.4 Friends And Related Function Documentation

#### 6.34.4.1 List

```
template<typename T >
friend class List [friend]
```

### 6.34.5 Member Data Documentation

#### 6.34.5.1 ptr

```
template<typename T >
Node< T >* Core::List< T >::iterator::ptr [private]
```

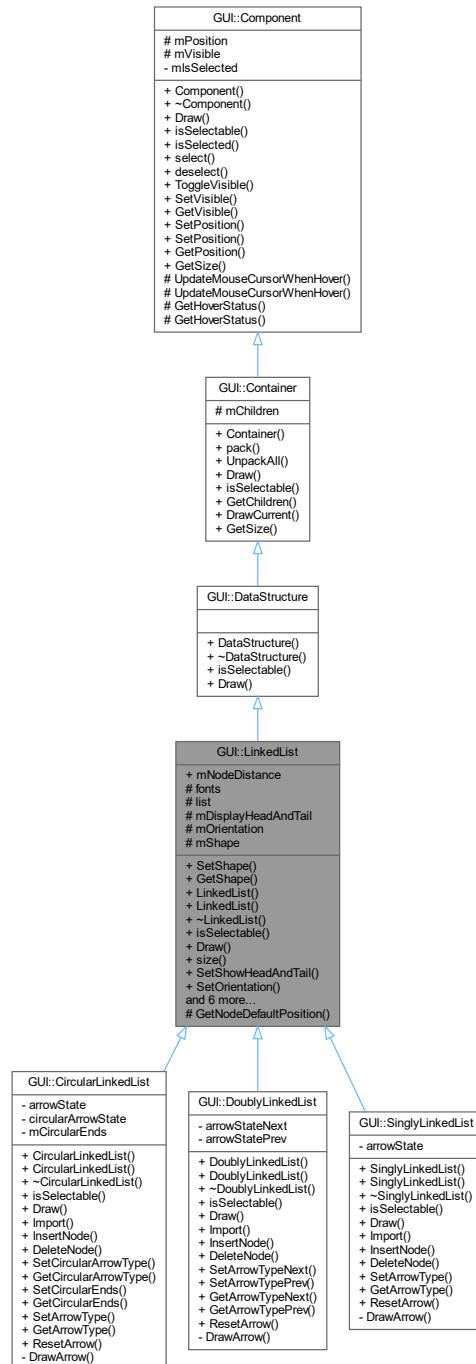
The documentation for this class was generated from the following file:

- src/Core/[List.hpp](#)

## 6.35 GUI::LinkedList Class Reference

```
#include <LinkedList.hpp>
```

Inheritance diagram for GUI::LinkedList:



## Public Types

- enum `ArrowType` {
 `Default` , `Hidden` , `Active` , `Skip` ,
 `ArrowTypeCount` }
- enum `Orientation` { `Horizontal` , `Vertical` , `OrientationCount` }
- typedef std::shared\_ptr<`DataStructure`> `Ptr`

## Public Member Functions

- void `SetShape` (`GUI::Node::Shape` shape)
- `GUI::Node::Shape GetShape () const`
- `LinkedList ()`
- `LinkedList (FontHolder *fonts)`
- `~LinkedList ()`
- bool `isSelectable () const`
- virtual void `Draw` (`Vector2 base=(Vector2){0, 0}`, float `t=1.0f`, bool `init=false`)=0
- virtual std::size\_t `size () const`
- virtual void `SetShowHeadAndTail` (bool show)
- virtual void `SetOrientation` (`Orientation orientation`)
- virtual std::vector<`GUI::Node`> & `GetList ()`
- virtual `GUI::Node GenerateNode` (int value)
- virtual void `Import` (std::vector<int> nodes)
- virtual void `InsertNode` (std::size\_t index, `GUI::Node` node, bool rePosition=true)
- virtual void `DeleteNode` (std::size\_t index, bool rePosition=true)
- virtual void `Relayout ()`
- virtual void `Draw` (`Vector2 base`)
- void `pack` (`Component::Ptr component`)
- void `UnpackAll ()`
- std::vector<`Component::Ptr`> `GetChildren ()`
- virtual void `DrawCurrent` (`Vector2 base`)
- virtual `Vector2 GetSize ()`
- bool `isSelected () const`
- virtual void `select ()`
- virtual void `deselect ()`
- virtual void `ToggleVisible ()`
- virtual void `SetVisible` (bool visible)
- virtual bool `GetVisible ()`
- void `SetPosition` (float x, float y)
- void `SetPosition` (`Vector2 position`)
- `Vector2 GetPosition ()`

## Static Public Attributes

- static constexpr float `mNodeDistance` = 20

## Protected Member Functions

- `Vector2 GetNodeDefaultPosition (std::size_t index)`
- virtual void `UpdateMouseCursorWhenHover` (std::map<std::string, Rectangle> bounds, bool hover, bool noHover)
- virtual void `UpdateMouseCursorWhenHover` (Rectangle bound, bool hover, bool noHover)
- virtual bool `GetHoverStatus` (std::map<std::string, Rectangle> bounds, bool hover, bool noHover)
- virtual bool `GetHoverStatus` (Rectangle bound, bool hover, bool noHover)

## Protected Attributes

- `FontHolder * fonts`
- `std::vector< GUI::Node > list`
- `bool mDisplayHeadAndTail`
- `Orientation mOrientation = Orientation::Horizontal`
- `GUI::Node::Shape mShape = GUI::Node::Shape::Circle`
- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Attributes

- `bool mIsSelected`

### 6.35.1 Member Typedef Documentation

#### 6.35.1.1 Ptr

```
typedef std::shared_ptr< DataStructure > GUI::DataStructure::Ptr [inherited]
```

### 6.35.2 Member Enumeration Documentation

#### 6.35.2.1 ArrowType

```
enum GUI::LinkedList::ArrowType
```

##### Enumerator

Default	
Hidden	
Active	
Skip	
ArrowTypeCount	

#### 6.35.2.2 Orientation

```
enum GUI::LinkedList::Orientation
```

**Enumerator**

Horizontal	
Vertical	
OrientationCount	

### 6.35.3 Constructor & Destructor Documentation

#### 6.35.3.1 `LinkedList()` [1/2]

```
GUI::LinkedList::LinkedList ( )
```

#### 6.35.3.2 `LinkedList()` [2/2]

```
GUI::LinkedList::LinkedList (
    FontHolder * fonts )
```

#### 6.35.3.3 `~LinkedList()`

```
GUI::LinkedList::~LinkedList ( )
```

### 6.35.4 Member Function Documentation

#### 6.35.4.1 `DeleteNode()`

```
void GUI::LinkedList::DeleteNode (
    std::size_t index,
    bool rePosition = true ) [virtual]
```

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DoublyLinkedList](#), and [GUI::SinglyLinkedList](#).

#### 6.35.4.2 `deselect()`

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

#### 6.35.4.3 Draw() [1/2]

```
void GUI::DataStructure::Draw (
    Vector2 base )  [virtual], [inherited]
```

Reimplemented from [GUI::Container](#).

#### 6.35.4.4 Draw() [2/2]

```
virtual void GUI::LinkedList::Draw (
    Vector2 base = (Vector2){0, 0},
    float t = 1.0f,
    bool init = false )  [pure virtual]
```

Implemented in [GUI::CircularLinkedList](#), [GUI::DoublyLinkedList](#), and [GUI::SinglyLinkedList](#).

#### 6.35.4.5 DrawCurrent()

```
void GUI::Container::DrawCurrent (
    Vector2 base = (Vector2){0, 0} )  [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.35.4.6 GenerateNode()

```
GUI::Node GUI::LinkedList::GenerateNode (
    int value )  [virtual]
```

#### 6.35.4.7 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( )  [inherited]
```

#### 6.35.4.8 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover )  [protected], [virtual], [inherited]
```

#### 6.35.4.9 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.35.4.10 GetList()

```
std::vector< GUI::Node > & GUI::LinkedList::GetList () [virtual]
```

#### 6.35.4.11 GetNodeDefaultPosition()

```
Vector2 GUI::LinkedList::GetNodeDefaultPosition (
    std::size_t index ) [protected]
```

#### 6.35.4.12 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

#### 6.35.4.13 GetShape()

```
GUI::Node::Shape GUI::LinkedList::GetShape () const
```

#### 6.35.4.14 GetSize()

```
Vector2 GUI::Container::GetSize () [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

#### 6.35.4.15 GetVisible()

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

#### 6.35.4.16 Import()

```
void GUI::LinkedList::Import (
    std::vector< int > nodes ) [virtual]
```

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DoublyLinkedList](#), and [GUI::SinglyLinkedList](#).

#### 6.35.4.17 InsertNode()

```
void GUI::LinkedList::InsertNode (
    std::size_t index,
    GUI::Node node,
    bool rePosition = true ) [virtual]
```

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DoublyLinkedList](#), and [GUI::SinglyLinkedList](#).

#### 6.35.4.18 isSelectable()

```
bool GUI::LinkedList::isSelectable ( ) const [virtual]
```

Reimplemented from [GUI::DataStructure](#).

Reimplemented in [GUI::SinglyLinkedList](#).

#### 6.35.4.19 isSelected()

```
bool GUI::Component::isSelected ( ) const [inherited]
```

#### 6.35.4.20 pack()

```
void GUI::Container::pack (
    Component::Ptr component ) [inherited]
```

#### 6.35.4.21 Relayout()

```
void GUI::LinkedList::Relayout ( ) [virtual]
```

**6.35.4.22 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.35.4.23 SetOrientation()**

```
void GUI::LinkedList::SetOrientation (
    Orientation orientation ) [virtual]
```

**6.35.4.24 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.35.4.25 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

**6.35.4.26 SetShape()**

```
void GUI::LinkedList::SetShape (
    GUI::Node::Shape shape )
```

**6.35.4.27 SetShowHeadAndTail()**

```
void GUI::LinkedList::SetShowHeadAndTail (
    bool show ) [virtual]
```

**6.35.4.28 SetVisible()**

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.35.4.29 size()

```
std::size_t GUI::LinkedList::size () const [virtual]
```

#### 6.35.4.30 ToggleVisible()

```
void GUI::Component::ToggleVisible () [virtual], [inherited]
```

#### 6.35.4.31 UnpackAll()

```
void GUI::Container::UnpackAll () [inherited]
```

#### 6.35.4.32 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.35.4.33 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.35.5 Member Data Documentation

#### 6.35.5.1 fonts

```
FontHolder* GUI::LinkedList::fonts [protected]
```

### 6.35.5.2 list

```
std::vector< GUI::Node > GUI::LinkedList::list [protected]
```

### 6.35.5.3 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

### 6.35.5.4 mDisplayHeadAndTail

```
bool GUI::LinkedList::mDisplayHeadAndTail [protected]
```

### 6.35.5.5 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

### 6.35.5.6 mNodeDistance

```
constexpr float GUI::LinkedList::mNodeDistance = 20 [static], [constexpr]
```

### 6.35.5.7 mOrientation

```
Orientation GUI::LinkedList::mOrientation = Orientation::Horizontal [protected]
```

### 6.35.5.8 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

### 6.35.5.9 mShape

```
GUI::Node::Shape GUI::LinkedList::mShape = GUI::Node::Shape::Circle [protected]
```

### 6.35.5.10 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

The documentation for this class was generated from the following files:

- src/Components/Visualization/[LinkedList.hpp](#)
- src/Components/Visualization/[LinkedList.cpp](#)

## 6.36 Core::List< T > Class Template Reference

```
#include <List.hpp>
```

### Classes

- class [const\\_iterator](#)
- class [iterator](#)

### Public Member Functions

- [List \(\)](#)
- [List \(std::initializer\\_list< T > list\)](#)
- [List \(const List< T > &list\)](#)
- [List \(List< T > &&list\)](#)
- [List< T > & operator= \(const List< T > &&list\)](#)
- [iterator begin \(\)](#)
- [const\\_iterator begin \(\) const](#)
- [iterator end \(\)](#)
- [const\\_iterator end \(\) const](#)
- [bool empty \(\) const](#)
- [std::size\\_t size \(\) const](#)
- [T & front \(\)](#)
- [const T & front \(\) const](#)
- [T & back \(\)](#)
- [const T & back \(\) const](#)
- [T & operator\[\] \(std::size\\_t index\)](#)
- [const T & operator\[\] \(std::size\\_t index\) const](#)
- [T & at \(std::size\\_t index\)](#)
- [const T & at \(std::size\\_t index\) const](#)
- [void push\\_front \(const T &value\)](#)
- [void push\\_back \(const T &value\)](#)
- [void pop\\_front \(\)](#)
- [void pop\\_back \(\)](#)
- [iterator remove \(const iterator &it\)](#)
- [int remove \(const T &value, const iterator &begin, const iterator &end\)](#)
- [int remove \(const T &value\)](#)
- [int remove\\_if \(std::function< bool\(const T &\) > predicate, const iterator &begin, const iterator &end\)](#)
- [int remove\\_if \(std::function< bool\(const T &\) > predicate\)](#)
- [void clear \(\)](#)

## Private Member Functions

- void `reset ()`
- void `insert_previous (const iterator &it, const iterator &it_prev)`
- iterator `move_previous (const iterator &it, const iterator &first, const iterator &last)`
- iterator `get_iterator (std::size_t index)`

## Private Attributes

- iterator `mBegin`
- iterator `mEnd`
- std::size\_t `mSize`

### 6.36.1 Constructor & Destructor Documentation

#### 6.36.1.1 List() [1/4]

```
template<typename T >
Core::List< T >::List ( ) [inline]
```

#### 6.36.1.2 List() [2/4]

```
template<typename T >
Core::List< T >::List (
    std::initializer_list< T > list ) [inline]
```

#### 6.36.1.3 List() [3/4]

```
template<typename T >
Core::List< T >::List (
    const List< T > & list ) [inline]
```

#### 6.36.1.4 List() [4/4]

```
template<typename T >
Core::List< T >::List (
    List< T > && list ) [inline]
```

## 6.36.2 Member Function Documentation

### 6.36.2.1 at() [1/2]

```
template<typename T >
T & Core::List< T >::at (
    std::size_t index ) [inline]
```

### 6.36.2.2 at() [2/2]

```
template<typename T >
const T & Core::List< T >::at (
    std::size_t index ) const [inline]
```

### 6.36.2.3 back() [1/2]

```
template<typename T >
T & Core::List< T >::back ( ) [inline]
```

### 6.36.2.4 back() [2/2]

```
template<typename T >
const T & Core::List< T >::back ( ) const [inline]
```

### 6.36.2.5 begin() [1/2]

```
template<typename T >
iterator Core::List< T >::begin ( ) [inline]
```

### 6.36.2.6 begin() [2/2]

```
template<typename T >
const_iterator Core::List< T >::begin ( ) const [inline]
```

### 6.36.2.7 `clear()`

```
template<typename T >
void Core::List< T >::clear ( ) [inline]
```

### 6.36.2.8 `empty()`

```
template<typename T >
bool Core::List< T >::empty ( ) const [inline]
```

### 6.36.2.9 `end() [1/2]`

```
template<typename T >
iterator Core::List< T >::end ( ) [inline]
```

### 6.36.2.10 `end() [2/2]`

```
template<typename T >
const_iterator Core::List< T >::end ( ) const [inline]
```

### 6.36.2.11 `front() [1/2]`

```
template<typename T >
T & Core::List< T >::front ( ) [inline]
```

### 6.36.2.12 `front() [2/2]`

```
template<typename T >
const T & Core::List< T >::front ( ) const [inline]
```

### 6.36.2.13 `get_iterator()`

```
template<typename T >
iterator Core::List< T >::get_iterator (
    std::size_t index ) [inline], [private]
```

### 6.36.2.14 insert\_previous()

```
template<typename T >
void Core::List< T >::insert_previous (
    const iterator & it,
    const iterator & it_prev ) [inline], [private]
```

### 6.36.2.15 move\_previous()

```
template<typename T >
iterator Core::List< T >::move_previous (
    const iterator & it,
    const iterator & first,
    const iterator & last ) [inline], [private]
```

### 6.36.2.16 operator=()

```
template<typename T >
List< T > & Core::List< T >::operator= (
    const List< T > && list ) [inline]
```

### 6.36.2.17 operator[](1/2)

```
template<typename T >
T & Core::List< T >::operator[] (
    std::size_t index ) [inline]
```

### 6.36.2.18 operator[](2/2)

```
template<typename T >
const T & Core::List< T >::operator[] (
    std::size_t index ) const [inline]
```

### 6.36.2.19 pop\_back()

```
template<typename T >
void Core::List< T >::pop_back ( ) [inline]
```

### 6.36.2.20 `pop_front()`

```
template<typename T >
void Core::List< T >::pop_front ( ) [inline]
```

### 6.36.2.21 `push_back()`

```
template<typename T >
void Core::List< T >::push_back (
    const T & value ) [inline]
```

### 6.36.2.22 `push_front()`

```
template<typename T >
void Core::List< T >::push_front (
    const T & value ) [inline]
```

### 6.36.2.23 `remove()` [1/3]

```
template<typename T >
iterator Core::List< T >::remove (
    const iterator & it ) [inline]
```

### 6.36.2.24 `remove()` [2/3]

```
template<typename T >
int Core::List< T >::remove (
    const T & value ) [inline]
```

### 6.36.2.25 `remove()` [3/3]

```
template<typename T >
int Core::List< T >::remove (
    const T & value,
    const iterator & begin,
    const iterator & end ) [inline]
```

**6.36.2.26 remove\_if() [1/2]**

```
template<typename T >
int Core::List< T >::remove_if (
    std::function< bool(const T &) > predicate ) [inline]
```

**6.36.2.27 remove\_if() [2/2]**

```
template<typename T >
int Core::List< T >::remove_if (
    std::function< bool(const T &) > predicate,
    const iterator & begin,
    const iterator & end ) [inline]
```

**6.36.2.28 reset()**

```
template<typename T >
void Core::List< T >::reset () [inline], [private]
```

**6.36.2.29 size()**

```
template<typename T >
std::size_t Core::List< T >::size () const [inline]
```

**6.36.3 Member Data Documentation****6.36.3.1 mBegin**

```
template<typename T >
iterator Core::List< T >::mBegin [private]
```

**6.36.3.2 mEnd**

```
template<typename T >
iterator Core::List< T >::mEnd [private]
```

### 6.36.3.3 mSize

```
template<typename T >
std::size_t Core::List< T >::mSize [private]
```

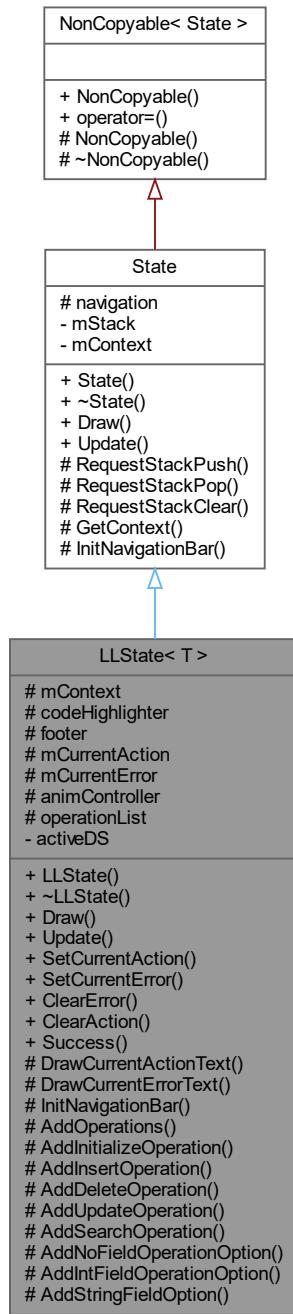
The documentation for this class was generated from the following file:

- src/Core/[List.hpp](#)

## 6.37 LLState< T > Class Template Reference

```
#include <LLState.hpp>
```

Inheritance diagram for LLState< T >:



## Classes

- struct [IntegerInput](#)

## Public Types

- typedef std::unique\_ptr< State > [Ptr](#)

## Public Member Functions

- `LLState (StateStack &stack, Context context, DataStructures::ID activeDS)`
- `~LLState ()`
- `virtual void Draw ()=0`
- `virtual bool Update (float dt)`
- `virtual void SetCurrentAction (std::string action)`
- `virtual void SetCurrentError (std::string error)`
- `virtual void ClearError ()`
- `virtual void ClearAction ()`
- `virtual void Success ()`

## Protected Member Functions

- `virtual void DrawCurrentActionText ()`
- `virtual void DrawCurrentErrorText ()`
- `void InitNavigationBar ()`
- `virtual void AddOperations ()`
- `virtual void AddInitializeOperation ()`
- `virtual void AddInsertOperation ()`
- `virtual void AddDeleteOperation ()`
- `virtual void AddUpdateOperation ()`
- `virtual void AddSearchOperation ()`
- `virtual void AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void() > action)`
- `virtual void AddIntFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >) > action)`
- `virtual void AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >) > action)`
- `void RequestStackPush (States::ID stateID)`
- `void RequestStackPop ()`
- `void RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< T > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Attributes

- `DataStructures::ID activeDS`
- `StateStack * mStack`

### 6.37.1 Member Typedef Documentation

#### 6.37.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

### 6.37.2 Constructor & Destructor Documentation

#### 6.37.2.1 LLState()

```
template<typename T >
LLState< T >::LLState (
    StateStack & stack,
    Context context,
    DataStructures::ID activeDS )
```

#### 6.37.2.2 ~LLState()

```
template<typename T >
LLState< T >::~LLState
```

### 6.37.3 Member Function Documentation

#### 6.37.3.1 AddDeleteOperation()

```
template<typename T >
void LLState< T >::AddDeleteOperation [protected], [virtual]
```

Reimplemented in [CLLState](#), [DLLState](#), [QueueState](#), [SLLState](#), and [StackState](#).

#### 6.37.3.2 AddInitializeOperation()

```
template<typename T >
void LLState< T >::AddInitializeOperation [protected], [virtual]
```

Reimplemented in [CLLState](#), [DLLState](#), [QueueState](#), [SLLState](#), and [StackState](#).

### 6.37.3.3 AddInsertOperation()

```
template<typename T >
void LLState< T >::AddInsertOperation [protected], [virtual]
```

Reimplemented in [CLLState](#), [DLLState](#), [QueueState](#), [SLLState](#), and [StackState](#).

### 6.37.3.4 AddIntFieldOperationOption()

```
template<typename T >
void LLState< T >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual]
```

### 6.37.3.5 AddNoFieldOperationOption()

```
template<typename T >
void LLState< T >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual]
```

### 6.37.3.6 AddOperations()

```
template<typename T >
void LLState< T >::AddOperations [protected], [virtual]
```

### 6.37.3.7 AddSearchOperation()

```
template<typename T >
void LLState< T >::AddSearchOperation [protected], [virtual]
```

Reimplemented in [CLLState](#), [DLLState](#), [QueueState](#), [SLLState](#), and [StackState](#).

### 6.37.3.8 AddStringFieldOption()

```
template<typename T >
void LLState< T >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual]
```

### 6.37.3.9 AddUpdateOperation()

```
template<typename T >
void LLState< T >::AddUpdateOperation [protected], [virtual]
```

Reimplemented in [CLLState](#), [DLLState](#), and [SLLState](#).

### 6.37.3.10 ClearAction()

```
template<typename T >
void LLState< T >::ClearAction [inline], [virtual]
```

### 6.37.3.11 ClearError()

```
template<typename T >
void LLState< T >::ClearError [inline], [virtual]
```

### 6.37.3.12 Draw()

```
template<typename T >
virtual void LLState< T >::Draw ( ) [pure virtual]
```

Implements [State](#).

Implemented in [CLLState](#), [DLLState](#), [QueueState](#), [SLLState](#), and [StackState](#).

### 6.37.3.13 DrawCurrentActionText()

```
template<typename T >
void LLState< T >::DrawCurrentActionText [inline], [protected], [virtual]
```

### 6.37.3.14 DrawCurrentErrorText()

```
template<typename T >
void LLState< T >::DrawCurrentErrorText [inline], [protected], [virtual]
```

### 6.37.3.15 GetContext()

```
State::Context State::GetContext () const [protected], [inherited]
```

### 6.37.3.16 InitNavigationBar()

```
template<typename T >
void LLState< T >::InitNavigationBar [protected]
```

### 6.37.3.17 RequestStackClear()

```
void State::RequestStackClear () [protected], [inherited]
```

### 6.37.3.18 RequestStackPop()

```
void State::RequestStackPop () [protected], [inherited]
```

### 6.37.3.19 RequestStackPush()

```
void State::RequestStackPush (
    States::ID stateID ) [protected], [inherited]
```

### 6.37.3.20 SetCurrentAction()

```
template<typename T >
void LLState< T >::SetCurrentAction (
    std::string action ) [inline], [virtual]
```

### 6.37.3.21 SetCurrentError()

```
template<typename T >
void LLState< T >::SetCurrentError (
    std::string error ) [inline], [virtual]
```

### 6.37.3.22 Success()

```
template<typename T >
void LLState< T >::Success [inline], [virtual]
```

### 6.37.3.23 Update()

```
template<typename T >
bool LLState< T >::Update (
    float dt ) [virtual]
```

Implements [State](#).

## 6.37.4 Member Data Documentation

### 6.37.4.1 activeDS

```
template<typename T >
DataStructures::ID LLState< T >::activeDS [private]
```

### 6.37.4.2 animController

```
template<typename T >
T::Ptr LLState< T >::animController [protected]
```

### 6.37.4.3 codeHighlighter

```
template<typename T >
GUI::CodeHighlighter::Ptr LLState< T >::codeHighlighter [protected]
```

#### 6.37.4.4 footer

```
template<typename T >
GUI::Footer< T >::footer [protected]
```

#### 6.37.4.5 mContext

```
template<typename T >
Context LLState< T >::mContext [protected]
```

#### 6.37.4.6 mCurrentAction

```
template<typename T >
std::string LLState< T >::mCurrentAction [protected]
```

#### 6.37.4.7 mCurrentError

```
template<typename T >
std::string LLState< T >::mCurrentError [protected]
```

#### 6.37.4.8 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.37.4.9 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

#### 6.37.4.10 operationList

```
template<typename T >
GUI::OperationList LLState< T >::operationList [protected]
```

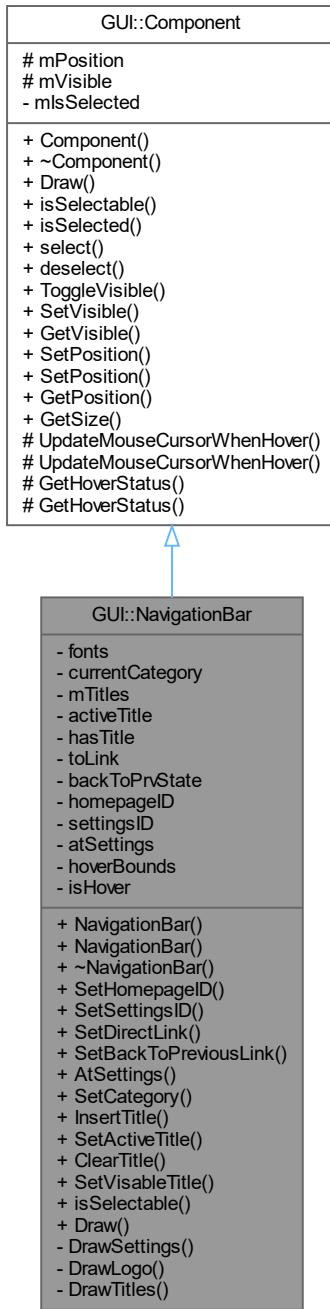
The documentation for this class was generated from the following file:

- src/States/LinkedList/LLState.hpp

## 6.38 GUI::NavigationBar Class Reference

```
#include <NavigationBar.hpp>
```

Inheritance diagram for GUI::NavigationBar:



## Classes

- struct [TitleInfo](#)

## Public Types

- `typedef std::shared_ptr< Component > Ptr`

## Public Member Functions

- `NavigationBar (FontHolder *fonts)`
- `NavigationBar ()`
- `~NavigationBar ()`
- `void SetHomepagelD (States::ID id)`
- `void SetSettingsID (States::ID id)`
- `void SetDirectLink (std::function< void(States::ID) > link)`
- `void SetBackToPreviousLink (std::function< void() > link)`
- `void AtSettings (bool settings)`
- `void SetCategory (std::string category)`
- `void InsertTitle (DataStructures::ID titleID, States::ID stateID, std::string abbrTitle, std::string titleName)`
- `void SetActiveTitle (DataStructures::ID title)`
- `void ClearTitle ()`
- `void SetVisibleTitle (bool visible)`
- `bool IsSelectable () const`
- `void Draw (Vector2 base=(Vector2){0, 0})`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`
- `virtual Vector2 GetSize ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `bool DrawSettings ()`
- `bool DrawLogo ()`
- `States::ID DrawTitles ()`

## Private Attributes

- `FontHolder * fonts`
- `std::string currentCategory`
- `std::map< DataStructures::ID, TitleInfo > mTitles`
- `DataStructures::ID activeTitle`
- `bool hasTitle`
- `std::function< void(States::ID) > toLink`
- `std::function< void() > backToPrvState`
- `States::ID homepageID`
- `States::ID settingsID`
- `bool atSettings`
- `std::map< std::string, Rectangle > hoverBounds`
- `bool isHover`
- `bool mIsSelected`

### 6.38.1 Member Typedef Documentation

#### 6.38.1.1 Ptr

```
typedef std::shared_ptr< Component > GUI::Component::Ptr [inherited]
```

### 6.38.2 Constructor & Destructor Documentation

#### 6.38.2.1 NavigationBar() [1/2]

```
GUI::NavigationBar::NavigationBar ( FontHolder * fonts )
```

#### 6.38.2.2 NavigationBar() [2/2]

```
GUI::NavigationBar::NavigationBar ( )
```

#### 6.38.2.3 ~NavigationBar()

```
GUI::NavigationBar::~NavigationBar ( )
```

### 6.38.3 Member Function Documentation

#### 6.38.3.1 AtSettings()

```
void GUI::NavigationBar::AtSettings (
    bool settings )
```

#### 6.38.3.2 ClearTitle()

```
void GUI::NavigationBar::ClearTitle ( )
```

#### 6.38.3.3 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

#### 6.38.3.4 Draw()

```
void GUI::NavigationBar::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Component](#).

#### 6.38.3.5 DrawLogo()

```
bool GUI::NavigationBar::DrawLogo ( ) [private]
```

#### 6.38.3.6 DrawSettings()

```
bool GUI::NavigationBar::DrawSettings ( ) [private]
```

### 6.38.3.7 DrawTitles()

```
States::ID GUI::NavigationBar::DrawTitles ( ) [private]
```

### 6.38.3.8 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.38.3.9 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.38.3.10 GetPosition()

```
Vector2 GUI::Component::GetPosition ( ) [inherited]
```

### 6.38.3.11 GetSize()

```
Vector2 GUI::Component::GetSize ( ) [virtual], [inherited]
```

Reimplemented in [GUI::Button](#), [GUI::InputField](#), [GUI::OperationList](#), [GUI::OptionInputField](#), and [GUI::Container](#).

### 6.38.3.12 GetVisible()

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

**6.38.3.13 InsertTitle()**

```
void GUI::NavigationBar::InsertTitle (
    DataStructures::ID titleID,
    States::ID stateID,
    std::string abbrTitle,
    std::string titleName )
```

**6.38.3.14 isSelectable()**

```
bool GUI::NavigationBar::isSelectable () const [virtual]
```

Implements [GUI::Component](#).

**6.38.3.15 isSelected()**

```
bool GUI::Component::isSelected () const [inherited]
```

**6.38.3.16 select()**

```
void GUI::Component::select () [virtual], [inherited]
```

**6.38.3.17 SetActiveTitle()**

```
void GUI::NavigationBar::SetActiveTitle (
    DataStructures::ID title )
```

**6.38.3.18 SetBackToPreviousLink()**

```
void GUI::NavigationBar::SetBackToPreviousLink (
    std::function< void() > link )
```

**6.38.3.19 SetCategory()**

```
void GUI::NavigationBar::SetCategory (
    std::string category )
```

**6.38.3.20 SetDirectLink()**

```
void GUI::NavigationBar::SetDirectLink (
    std::function< void(States::ID) > link )
```

**6.38.3.21 SetHomepageID()**

```
void GUI::NavigationBar::SetHomepageID (
    States::ID id )
```

**6.38.3.22 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.38.3.23 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

**6.38.3.24 SetSettingsID()**

```
void GUI::NavigationBar::SetSettingsID (
    States::ID id )
```

**6.38.3.25 SetVisableTitle()**

```
void GUI::NavigationBar::SetVisableTitle (
    bool visible )
```

### 6.38.3.26 SetVisible()

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.38.3.27 ToggleVisible()

```
void GUI::Component::ToggleVisible () [virtual], [inherited]
```

### 6.38.3.28 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.38.3.29 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.38.4 Member Data Documentation

### 6.38.4.1 activeTitle

```
DataStructures::ID GUI::NavigationBar::activeTitle [private]
```

### 6.38.4.2 atSettings

```
bool GUI::NavigationBar::atSettings [private]
```

#### 6.38.4.3 backToPrvState

```
std::function< void() > GUI::NavigationBar::backToPrvState [private]
```

#### 6.38.4.4 currentCategory

```
std::string GUI::NavigationBar::currentCategory [private]
```

#### 6.38.4.5 fonts

```
FontHolder* GUI::NavigationBar::fonts [private]
```

#### 6.38.4.6 hasTitle

```
bool GUI::NavigationBar::hasTitle [private]
```

#### 6.38.4.7 homepageID

```
States::ID GUI::NavigationBar::homepageID [private]
```

#### 6.38.4.8 hoverBounds

```
std::map< std::string, Rectangle > GUI::NavigationBar::hoverBounds [private]
```

#### 6.38.4.9 isHover

```
bool GUI::NavigationBar::isHover [private]
```

#### 6.38.4.10 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.38.4.11 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.38.4.12 mTitles

```
std::map< DataStructures::ID, TitleInfo > GUI::NavigationBar::mTitles [private]
```

#### 6.38.4.13 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

#### 6.38.4.14 settingsID

```
States::ID GUI::NavigationBar::settingsID [private]
```

#### 6.38.4.15 toLink

```
std::function< void(States::ID) > GUI::NavigationBar::toLink [private]
```

The documentation for this class was generated from the following files:

- src/Components/Common/[NavigationBar.hpp](#)
- src/Components/Common/[NavigationBar.cpp](#)

## 6.39 Core::Node< T > Class Template Reference

```
#include <Node.hpp>
```

### Public Member Functions

- [Node \(\)](#)
- [Node \(const T &value\)](#)
- [Node \(const Node< T > &node\)](#)
- [Node \(const T &value, Node< T > \\*const &prev, Node< T > \\*const &next\)](#)

## Public Attributes

- T `mValue`
- `Node< T > * mPrev`
- `Node< T > * mNext`

### 6.39.1 Constructor & Destructor Documentation

#### 6.39.1.1 `Node()` [1/4]

```
template<typename T >
Core::Node< T >::Node ( ) [inline]
```

#### 6.39.1.2 `Node()` [2/4]

```
template<typename T >
Core::Node< T >::Node (
    const T & value ) [inline]
```

#### 6.39.1.3 `Node()` [3/4]

```
template<typename T >
Core::Node< T >::Node (
    const Node< T > & node ) [inline]
```

#### 6.39.1.4 `Node()` [4/4]

```
template<typename T >
Core::Node< T >::Node (
    const T & value,
    Node< T > *const & prev,
    Node< T > *const & next ) [inline]
```

### 6.39.2 Member Data Documentation

### 6.39.2.1 mNext

```
template<typename T >
Node< T > * Core::Node< T >::mNext
```

### 6.39.2.2 mPrev

```
template<typename T >
Node< T >* Core::Node< T >::mPrev
```

### 6.39.2.3 mValue

```
template<typename T >
T Core::Node< T >::mValue
```

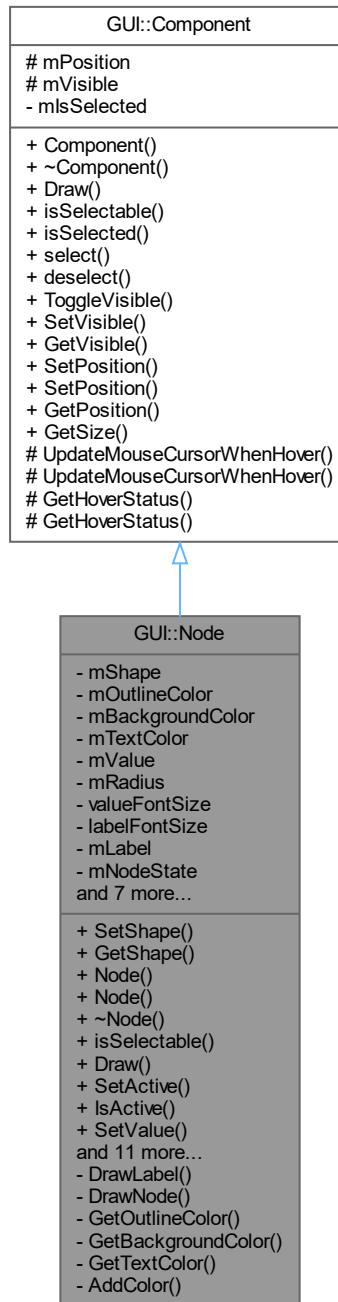
The documentation for this class was generated from the following file:

- src/Core/[Node.hpp](#)

## 6.40 GUI::Node Class Reference

```
#include <Node.hpp>
```

Inheritance diagram for GUI::Node:



## Public Types

- enum `State` {
 `Default` , `Active` , `ActiveBlue` , `ActiveGreen` ,
 `ActiveRed` , `Iterated` , `Hide` , `StateCount` }
- enum `Shape` { `Circle` , `Square` , `ShapeCount` }
- typedef `std::shared_ptr<Component>` `Ptr`

## Public Member Functions

- void `SetShape (Shape shape)`
- `Shape GetShape () const`
- `Node (int value, FontHolder *fonts)`
- `Node ()`
- `~Node ()`
- bool `isSelectable () const`
- void `Draw (Vector2 base=(Vector2){0, 0}, float t=1.0f)`
- void `SetActive (bool active)`
- bool `IsActive ()`
- void `SetValue (int value)`
- int `GetValue () const`
- void `SetLabel (std::string label)`
- void `ClearLabel ()`
- void `AnimationOnNode (bool animate)`
- void `SetRadius (float radius)`
- void `SetValueFontSize (int fontSize)`
- void `SetLabelFontSize (int fontSize)`
- void `SetNodeState (State state)`
- `State GetNodeState () const`
- void `SetReachable (bool reachable)`
- bool `GetReachable () const`
- virtual void `Draw (Vector2 base=(Vector2){0, 0})`
- bool `isSelected () const`
- virtual void `select ()`
- virtual void `deselect ()`
- virtual void `ToggleVisible ()`
- virtual void `SetVisible (bool visible)`
- virtual bool `GetVisible ()`
- void `SetPosition (float x, float y)`
- void `SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`
- virtual `Vector2 GetSize ()`

## Protected Member Functions

- virtual void `UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- virtual void `UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- virtual bool `GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- virtual bool `GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- void `DrawLabel` (Vector2 base=(Vector2){0, 0})
- void `DrawNode` (Vector2 base=(Vector2){0, 0}, float t=1.0f)
- Color `GetOutlineColor` (float t=1.0f)
- Color `GetBackgroundColor` (float t=1.0f)
- Color `GetTextColor` (float t=1.0f)
- void `AddColor` ()

## Private Attributes

- Shape `mShape` = Shape::Circle
- std::map< State, std::pair< Color, Color > > `mOutlineColor`
- std::map< State, std::pair< Color, Color > > `mBackgroundColor`
- std::map< State, std::pair< Color, Color > > `mTextColor`
- int `mValue`
- float `mRadius`
- float `valueFontSize`
- float `labelFontSize`
- std::string `mLabel`
- State `mNodeState`
- bool `mReachable`
- bool `animateNode`
- bool `mActive`
- FontHolder \* `fonts`
- Color `mDefaultColor`
- Color `mActiveColor`
- Color `mBorderColor`
- bool `mIsSelected`

### 6.40.1 Member Typedef Documentation

#### 6.40.1.1 Ptr

```
typedef std::shared_ptr< Component > GUI::Component::Ptr [inherited]
```

### 6.40.2 Member Enumeration Documentation

#### 6.40.2.1 Shape

```
enum GUI::Node::Shape
```

**Enumerator**

Circle	
Square	
ShapeCount	

**6.40.2.2 State**

```
enum GUI::Node::State
```

**Enumerator**

Default	
Active	
ActiveBlue	
ActiveGreen	
ActiveRed	
Iterated	
Hide	
StateCount	

**6.40.3 Constructor & Destructor Documentation****6.40.3.1 Node() [1/2]**

```
GUI::Node::Node (
    int value,
    FontHolder * fonts )
```

**6.40.3.2 Node() [2/2]**

```
GUI::Node::Node ( )
```

**6.40.3.3 ~Node()**

```
GUI::Node::~Node ( )
```

## 6.40.4 Member Function Documentation

### 6.40.4.1 AddColor()

```
void GUI::Node::AddColor ( ) [private]
```

### 6.40.4.2 AnimationOnNode()

```
void GUI::Node::AnimationOnNode (
    bool animate )
```

### 6.40.4.3 ClearLabel()

```
void GUI::Node::ClearLabel ( )
```

### 6.40.4.4 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.40.4.5 Draw() [1/2]

```
void GUI::Component::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented in [GUI::DataStructure](#), [GUI::Button](#), [GUI::Card](#), [GUI::CodeHighlighter](#), [GUI::InputField](#), [GUI::NavigationBar](#), [GUI::OperationList](#), and [GUI::Container](#).

### 6.40.4.6 Draw() [2/2]

```
void GUI::Node::Draw (
    Vector2 base = (Vector2){0, 0},
    float t = 1.0f )
```

#### 6.40.4.7 DrawLabel()

```
void GUI::Node::DrawLabel (
    Vector2 base = (Vector2){0, 0} ) [private]
```

#### 6.40.4.8 DrawNode()

```
void GUI::Node::DrawNode (
    Vector2 base = (Vector2){0, 0},
    float t = 1.0f ) [private]
```

#### 6.40.4.9 GetBackgroundColor()

```
Color GUI::Node::GetBackgroundColor (
    float t = 1.0f ) [private]
```

#### 6.40.4.10 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.40.4.11 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.40.4.12 GetNodeState()

```
GUI::Node::State GUI::Node::GetNodeState ( ) const
```

#### 6.40.4.13 GetOutlineColor()

```
Color GUI::Node::GetOutlineColor (
    float t = 1.0f ) [private]
```

#### 6.40.4.14 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

#### 6.40.4.15 GetReachable()

```
bool GUI::Node::GetReachable () const
```

#### 6.40.4.16 GetShape()

```
GUI::Node::Shape GUI::Node::GetShape () const
```

#### 6.40.4.17 GetSize()

```
Vector2 GUI::Component::GetSize () [virtual], [inherited]
```

Reimplemented in [GUI::Button](#), [GUI::InputField](#), [GUI::OperationList](#), [GUI::OptionInputField](#), and [GUI::Container](#).

#### 6.40.4.18 GetTextColor()

```
Color GUI::Node::GetTextColor (
    float t = 1.0f ) [private]
```

#### 6.40.4.19 GetValue()

```
int GUI::Node::GetValue () const
```

#### 6.40.4.20 GetVisible()

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

#### 6.40.4.21 IsActive()

```
bool GUI::Node::IsActive ( )
```

#### 6.40.4.22 isSelectable()

```
bool GUI::Node::isSelectable ( ) const [virtual]
```

Implements [GUI::Component](#).

#### 6.40.4.23 isSelected()

```
bool GUI::Component::isSelected ( ) const [inherited]
```

#### 6.40.4.24 select()

```
void GUI::Component::select ( ) [virtual], [inherited]
```

#### 6.40.4.25 SetActive()

```
void GUI::Node::SetActive ( bool active )
```

#### 6.40.4.26 SetLabel()

```
void GUI::Node::SetLabel ( std::string label )
```

**6.40.4.27 SetLabelFontSize()**

```
void GUI::Node::SetLabelFontSize (
    int fontSize )
```

**6.40.4.28 SetNodeState()**

```
void GUI::Node::SetNodeState (
    State state )
```

**6.40.4.29 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y )  [inherited]
```

**6.40.4.30 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position )  [inherited]
```

**6.40.4.31 SetRadius()**

```
void GUI::Node::SetRadius (
    float radius )
```

**6.40.4.32 SetReachable()**

```
void GUI::Node::SetReachable (
    bool reachable )
```

**6.40.4.33 SetShape()**

```
void GUI::Node::SetShape (
    Shape shape )
```

#### 6.40.4.34 SetValue()

```
void GUI::Node::SetValue (
    int value )
```

#### 6.40.4.35 SetValueFontSize()

```
void GUI::Node::SetValueFontSize (
    int fontSize )
```

#### 6.40.4.36 SetVisible()

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.40.4.37 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

#### 6.40.4.38 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.40.4.39 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.40.5 Member Data Documentation

#### 6.40.5.1 animateNode

```
bool GUI::Node::animateNode [private]
```

#### 6.40.5.2 fonts

```
FontHolder* GUI::Node::fonts [private]
```

#### 6.40.5.3 labelFontSize

```
float GUI::Node::labelFontSize [private]
```

#### 6.40.5.4 mActive

```
bool GUI::Node::mActive [private]
```

#### 6.40.5.5 mActiveColor

```
Color GUI::Node::mActiveColor [private]
```

#### 6.40.5.6 mBackgroundColor

```
std::map< State, std::pair< Color, Color > > GUI::Node::mBackgroundColor [private]
```

#### 6.40.5.7 mBorderColor

```
Color GUI::Node::mBorderColor [private]
```

#### 6.40.5.8 mDefaultColor

```
Color GUI::Node::mDefaultColor [private]
```

#### 6.40.5.9 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.40.5.10 mLabel

```
std::string GUI::Node::mLabel [private]
```

#### 6.40.5.11 mNodeState

```
State GUI::Node::mNodeState [private]
```

#### 6.40.5.12 mOutlineColor

```
std::map< State, std::pair< Color, Color > > GUI::Node::mOutlineColor [private]
```

#### 6.40.5.13 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.40.5.14 mRadius

```
float GUI::Node::mRadius [private]
```

#### 6.40.5.15 mReachable

```
bool GUI::Node::mReachable [private]
```

#### 6.40.5.16 mShape

```
Shape GUI::Node::mShape = Shape::Circle [private]
```

#### 6.40.5.17 mTextColor

```
std::map< State, std::pair< Color, Color > > GUI::Node::mTextColor [private]
```

#### 6.40.5.18 mValue

```
int GUI::Node::mValue [private]
```

#### 6.40.5.19 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

#### 6.40.5.20 valueFontSize

```
float GUI::Node::valueFontSize [private]
```

The documentation for this class was generated from the following files:

- src/Components/Visualization/[Node.hpp](#)
- src/Components/Visualization/[Node.cpp](#)

## 6.41 NonCopyable< T > Class Template Reference

```
#include <NonCopyable.hpp>
```

### Public Member Functions

- [NonCopyable](#) (const [NonCopyable](#) &)=delete
- T & [operator=](#) (const T &)=delete

### Protected Member Functions

- [NonCopyable](#) ()=default
- [~NonCopyable](#) ()=default

#### 6.41.1 Constructor & Destructor Documentation

#### 6.41.1.1 NonCopyable() [1/2]

```
template<class T >
NonCopyable< T >::NonCopyable (
    const NonCopyable< T > & )  [delete]
```

#### 6.41.1.2 NonCopyable() [2/2]

```
template<class T >
NonCopyable< T >::NonCopyable ()  [protected], [default]
```

#### 6.41.1.3 ~NonCopyable()

```
template<class T >
NonCopyable< T >::~NonCopyable ()  [protected], [default]
```

### 6.41.2 Member Function Documentation

#### 6.41.2.1 operator=()

```
template<class T >
T & NonCopyable< T >::operator= (
    const T & )  [delete]
```

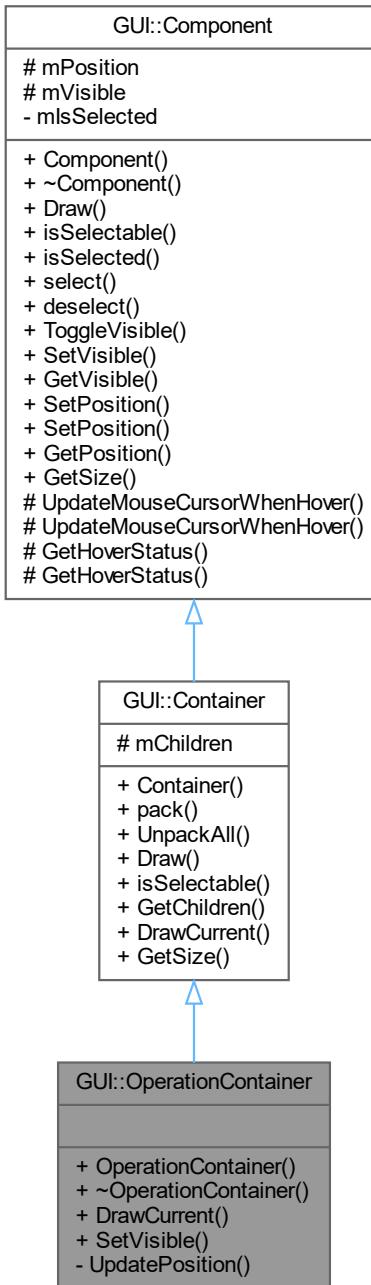
The documentation for this class was generated from the following file:

- src/[NonCopyable.hpp](#)

## 6.42 GUI::OperationContainer Class Reference

```
#include <OperationContainer.hpp>
```

Inheritance diagram for GUI::OperationContainer:



### Public Types

- `typedef std::shared_ptr< OperationContainer > Ptr`

## Public Member Functions

- `OperationContainer ()`
- `~OperationContainer ()`
- `void DrawCurrent (Vector2 base=(Vector2){0, 0})`
- `void SetVisible (bool visible)`
- `void pack (Component::Ptr component)`
- `void UnpackAll ()`
- `virtual void Draw (Vector2 base=(Vector2){0, 0})`
- `virtual bool IsSelectable () const`
- `std::vector< Component::Ptr > GetChildren ()`
- `virtual Vector2 GetSize ()`
- `bool IsSelected () const`
- `virtual void Select ()`
- `virtual void Deselect ()`
- `virtual void ToggleVisible ()`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `void UpdatePosition ()`

## Private Attributes

- `bool mIsSelected`

### 6.42.1 Member Typedef Documentation

### 6.42.1.1 Ptr

```
typedef std::shared_ptr< OperationContainer > GUI::OperationContainer::Ptr
```

## 6.42.2 Constructor & Destructor Documentation

### 6.42.2.1 OperationContainer()

```
GUI::OperationContainer::OperationContainer ( )
```

### 6.42.2.2 ~OperationContainer()

```
GUI::OperationContainer::~OperationContainer ( )
```

## 6.42.3 Member Function Documentation

### 6.42.3.1 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.42.3.2 Draw()

```
void GUI::Container::Draw ( 
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::DataStructure](#), and [GUI::OperationList](#).

### 6.42.3.3 DrawCurrent()

```
void GUI::OperationContainer::DrawCurrent ( 
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Container](#).

#### 6.42.3.4 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( ) [inherited]
```

#### 6.42.3.5 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.42.3.6 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.42.3.7 GetPosition()

```
Vector2 GUI::Component::GetPosition ( ) [inherited]
```

#### 6.42.3.8 GetSize()

```
Vector2 GUI::Container::GetSize ( ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

#### 6.42.3.9 GetVisible()

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

**6.42.3.10 isSelectable()**

```
bool GUI::Container::isSelectable () const [virtual], [inherited]
```

Implements [GUI::Component](#).

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DataStructure](#), [GUI::DoublyLinkedList](#), [GUI::DynamicArray](#), [GUI::LinkedList](#), and [GUI::SinglyLinkedList](#).

**6.42.3.11 isSelected()**

```
bool GUI::Component::isSelected () const [inherited]
```

**6.42.3.12 pack()**

```
void GUI::Container::pack (
    Component::Ptr component ) [inherited]
```

**6.42.3.13 select()**

```
void GUI::Component::select () [virtual], [inherited]
```

**6.42.3.14 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.42.3.15 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

#### 6.42.3.16 SetVisible()

```
void GUI::OperationContainer::SetVisible (
    bool visible ) [virtual]
```

Reimplemented from [GUI::Component](#).

#### 6.42.3.17 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

#### 6.42.3.18 UnpackAll()

```
void GUI::Container::UnpackAll ( ) [inherited]
```

#### 6.42.3.19 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.42.3.20 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.42.3.21 UpdatePosition()

```
void GUI::OperationContainer::UpdatePosition ( ) [private]
```

### 6.42.4 Member Data Documentation

#### 6.42.4.1 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

#### 6.42.4.2 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.42.4.3 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.42.4.4 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

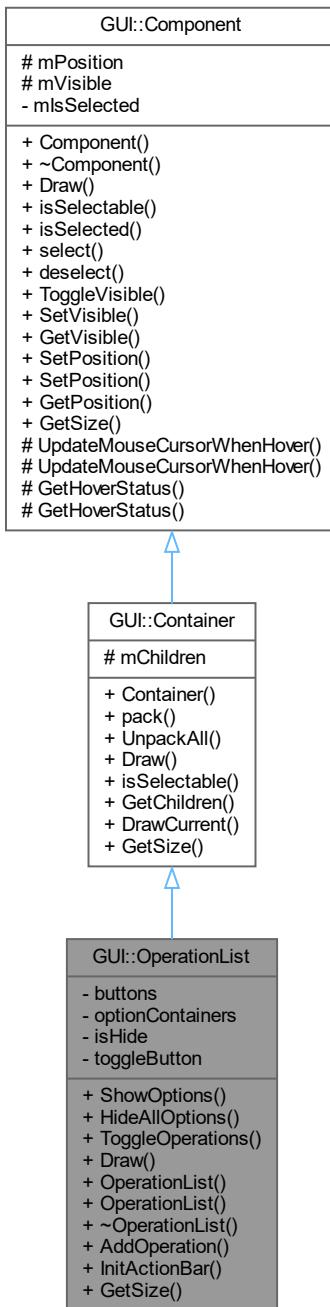
The documentation for this class was generated from the following files:

- src/Components/Common/[OperationContainer.hpp](#)
- src/Components/Common/[OperationContainer.cpp](#)

## 6.43 GUI::OperationList Class Reference

```
#include <OperationList.hpp>
```

Inheritance diagram for GUI::OperationList:



## Public Types

- `typedef std::shared_ptr< Container > Ptr`

## Public Member Functions

- `void ShowOptions (std::size_t index)`

- void `HideAllOptions ()`
- void `ToggleOperations ()`
- void `Draw (Vector2 base=(Vector2){0, 0})`
- `OperationList ()`
- `OperationList (FontHolder *fonts)`
- `~OperationList ()`
- void `AddOperation (GUI::Button::Ptr action, GUI::Container::Ptr optionContainer)`
- void `InitActionBar ()`
- `Vector2 GetSize ()`
- void `pack (Component::Ptr component)`
- void `UnpackAll ()`
- virtual bool `isSelectable () const`
- std::vector< Component::Ptr > `GetChildren ()`
- virtual void `DrawCurrent (Vector2 base)`
- bool `isSelected () const`
- virtual void `select ()`
- virtual void `deselect ()`
- virtual void `ToggleVisible ()`
- virtual void `SetVisible (bool visible)`
- virtual bool `GetVisible ()`
- void `SetPosition (float x, float y)`
- void `SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Protected Member Functions

- virtual void `UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- virtual void `UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- virtual bool `GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- virtual bool `GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- std::vector< Component::Ptr > `mChildren`
- `Vector2 mPosition`
- bool `mVisible`

## Private Attributes

- `GUI::Container buttons`
- `GUI::Container optionContainers`
- bool `isHide`
- `GUI::Button toggleButton`
- bool `mlsSelected`

### 6.43.1 Member Typedef Documentation

### 6.43.1.1 Ptr

```
typedef std::shared_ptr< Container > GUI::Container::Ptr [inherited]
```

## 6.43.2 Constructor & Destructor Documentation

### 6.43.2.1 OperationList() [1/2]

```
GUI::OperationList::OperationList ( )
```

### 6.43.2.2 OperationList() [2/2]

```
GUI::OperationList::OperationList (
    FontHolder * fonts )
```

### 6.43.2.3 ~OperationList()

```
GUI::OperationList::~OperationList ( )
```

## 6.43.3 Member Function Documentation

### 6.43.3.1 AddOperation()

```
void GUI::OperationList::AddOperation (
    GUI::Button::Ptr action,
    GUI::Container::Ptr optionContainer )
```

### 6.43.3.2 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

#### 6.43.3.3 Draw()

```
void GUI::OperationList::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Container](#).

#### 6.43.3.4 DrawCurrent()

```
void GUI::Container::DrawCurrent (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.43.3.5 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren () [inherited]
```

#### 6.43.3.6 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.43.3.7 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.43.3.8 GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

### 6.43.3.9 **GetSize()**

```
Vector2 GUI::OperationList::GetSize ( ) [virtual]
```

Reimplemented from [GUI::Container](#).

### 6.43.3.10 **GetVisible()**

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

### 6.43.3.11 **HideAllOptions()**

```
void GUI::OperationList::HideAllOptions ( )
```

### 6.43.3.12 **InitActionBar()**

```
void GUI::OperationList::InitActionBar ( )
```

### 6.43.3.13 **isSelectable()**

```
bool GUI::Container::isSelectable ( ) const [virtual], [inherited]
```

Implements [GUI::Component](#).

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DataStructure](#), [GUI::DoublyLinkedList](#), [GUI::DynamicArray](#), [GUI::LinkedList](#), and [GUI::SinglyLinkedList](#).

### 6.43.3.14 **isSelected()**

```
bool GUI::Component::isSelected ( ) const [inherited]
```

### 6.43.3.15 **pack()**

```
void GUI::Container::pack (
    Component::Ptr component ) [inherited]
```

**6.43.3.16 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.43.3.17 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.43.3.18 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

**6.43.3.19 SetVisible()**

```
void GUI::Component::SetVisible (
    bool visible ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

**6.43.3.20 ShowOptions()**

```
void GUI::OperationList::ShowOptions (
    std::size_t index )
```

**6.43.3.21 ToggleOperations()**

```
void GUI::OperationList::ToggleOperations ( )
```

**6.43.3.22 ToggleVisible()**

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

### 6.43.3.23 UnpackAll()

```
void GUI::Container::UnpackAll ( ) [inherited]
```

### 6.43.3.24 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.43.3.25 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

## 6.43.4 Member Data Documentation

### 6.43.4.1 buttons

```
GUI::Container GUI::OperationList::buttons [private]
```

### 6.43.4.2 isHide

```
bool GUI::OperationList::isHide [private]
```

### 6.43.4.3 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

#### 6.43.4.4 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.43.4.5 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.43.4.6 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

#### 6.43.4.7 optionContainers

```
GUI::Container GUI::OperationList::optionContainers [private]
```

#### 6.43.4.8 toggleButton

```
GUI::Button GUI::OperationList::toggleButton [private]
```

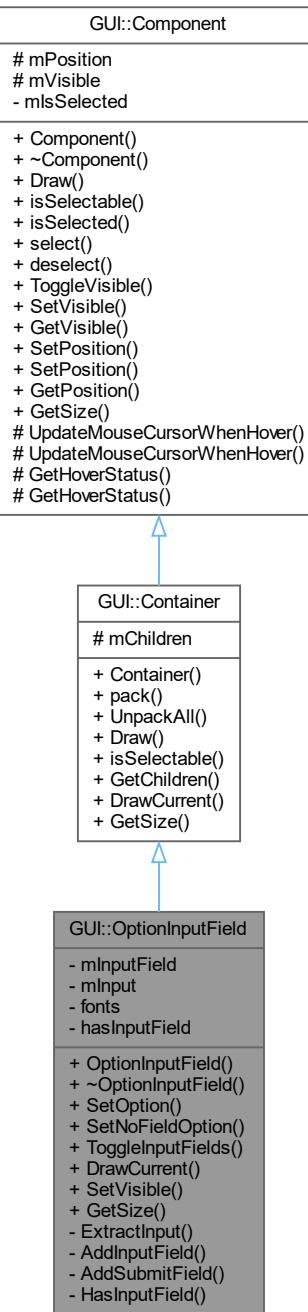
The documentation for this class was generated from the following files:

- src/Components/Common/[OperationList.hpp](#)
- src/Components/Common/[OperationList.cpp](#)

## 6.44 GUI::OptionInputField Class Reference

```
#include <OptionInputField.hpp>
```

Inheritance diagram for GUI::OptionInputField:



### Public Types

- `typedef std::shared_ptr< OptionInputField > Ptr`

## Public Member Functions

- `OptionInputField (FontHolder *fonts)`
- `~OptionInputField ()`
- `void SetOption (std::string content, std::vector< InputField::Ptr > fields, std::function< void(std::map< std::string, std::string >) > action)`
- `void SetNoFieldOption (std::string content, std::function< void() > action)`
- `void ToggleInputFields ()`
- `void DrawCurrent (Vector2 base=(Vector2){0, 0})`
- `void SetVisible (bool visible)`
- `virtual Vector2 GetSize ()`
- `void pack (Component::Ptr component)`
- `void UnpackAll ()`
- `virtual void Draw (Vector2 base=(Vector2){0, 0})`
- `virtual bool IsSelectable () const`
- `std::vector< Component::Ptr > GetChildren ()`
- `bool IsSelected () const`
- `virtual void Select ()`
- `virtual void Deselect ()`
- `virtual void ToggleVisible ()`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2GetPosition ()`

## Protected Member Functions

- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `std::map< std::string, std::string > ExtractInput ()`
- `void AddInputField (InputField::Ptr inputField)`
- `void AddSubmitField (std::function< void(std::map< std::string, std::string >) > action)`
- `bool HasInputField ()`

## Private Attributes

- `GUI::Container::Ptr mInputField`
- `std::map< std::string, std::string > mInput`
- `FontHolder * fonts`
- `bool hasInputField`
- `bool mIsSelected`

## 6.44.1 Member Typedef Documentation

### 6.44.1.1 Ptr

```
typedef std::shared_ptr< OptionInputField > GUI::OptionInputField::Ptr
```

## 6.44.2 Constructor & Destructor Documentation

### 6.44.2.1 OptionInputField()

```
GUI::OptionInputField::OptionInputField (
    FontHolder * fonts )
```

### 6.44.2.2 ~OptionInputField()

```
GUI::OptionInputField::~OptionInputField ( )
```

## 6.44.3 Member Function Documentation

### 6.44.3.1 AddInputField()

```
void GUI::OptionInputField::AddInputField (
    InputField::Ptr inputField ) [private]
```

### 6.44.3.2 AddSubmitField()

```
void GUI::OptionInputField::AddSubmitField (
    std::function< void(std::map< std::string, std::string >) > action ) [private]
```

#### 6.44.3.3 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

#### 6.44.3.4 Draw()

```
void GUI::Container::Draw ( 
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::DataStructure](#), and [GUI::OperationList](#).

#### 6.44.3.5 DrawCurrent()

```
void GUI::OptionInputField::DrawCurrent ( 
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Reimplemented from [GUI::Container](#).

#### 6.44.3.6 ExtractInput()

```
std::map< std::string, std::string > GUI::OptionInputField::ExtractInput ( ) [private]
```

#### 6.44.3.7 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( ) [inherited]
```

#### 6.44.3.8 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus ( 
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.44.3.9 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.44.3.10GetPosition()

```
Vector2 GUI::Component::GetPosition () [inherited]
```

#### 6.44.3.11GetSize()

```
Vector2 GUI::OptionInputField::GetSize () [virtual]
```

Reimplemented from [GUI::Container](#).

#### 6.44.3.12GetVisible()

```
bool GUI::Component::GetVisible () [virtual], [inherited]
```

#### 6.44.3.13HasInputField()

```
bool GUI::OptionInputField::HasInputField () [private]
```

#### 6.44.3.14isSelectable()

```
bool GUI::Container::isSelectable () const [virtual], [inherited]
```

Implements [GUI::Component](#).

Reimplemented in [GUI::CircularLinkedList](#), [GUI::DataStructure](#), [GUI::DoublyLinkedList](#), [GUI::DynamicArray](#), [GUI::LinkedList](#), and [GUI::SinglyLinkedList](#).

**6.44.3.15 isSelected()**

```
bool GUI::Component::isSelected ( ) const [inherited]
```

**6.44.3.16 pack()**

```
void GUI::Container::pack (
    Component::Ptr component ) [inherited]
```

**6.44.3.17 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.44.3.18 SetNoFieldOption()**

```
void GUI::OptionInputField::SetNoFieldOption (
    std::string content,
    std::function< void() > action )
```

**6.44.3.19 SetOption()**

```
void GUI::OptionInputField::SetOption (
    std::string content,
    std::vector< InputField::Ptr > fields,
    std::function< void(std::map< std::string, std::string >) > action )
```

**6.44.3.20 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

**6.44.3.21 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position ) [inherited]
```

#### 6.44.3.22 SetVisible()

```
void GUI::OptionInputField::SetVisible (
    bool visible ) [virtual]
```

Reimplemented from [GUI::Component](#).

#### 6.44.3.23 ToggleInputFields()

```
void GUI::OptionInputField::ToggleInputFields ( )
```

#### 6.44.3.24 ToggleVisible()

```
void GUI::Component::ToggleVisible ( ) [virtual], [inherited]
```

#### 6.44.3.25 UnpackAll()

```
void GUI::Container::UnpackAll ( ) [inherited]
```

#### 6.44.3.26 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.44.3.27 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.44.4 Member Data Documentation

#### 6.44.4.1 fonts

```
FontHolder* GUI::OptionInputField::fonts [private]
```

#### 6.44.4.2 hasInputField

```
bool GUI::OptionInputField::hasInputField [private]
```

#### 6.44.4.3 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

#### 6.44.4.4 mInput

```
std::map< std::string, std::string > GUI::OptionInputField::mInput [private]
```

#### 6.44.4.5 mInputField

```
GUI::Container::Ptr GUI::OptionInputField::mInputField [private]
```

#### 6.44.4.6 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.44.4.7 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.44.4.8 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

The documentation for this class was generated from the following files:

- src/Components/Common/[OptionInputField.hpp](#)
- src/Components/Common/[OptionInputField.cpp](#)

## 6.45 StateStack::PendingChange Struct Reference

### Public Member Functions

- [PendingChange \(Action action, States::ID stateID=States::None\)](#)

### Public Attributes

- [Action action](#)
- [States::ID stateID](#)

### 6.45.1 Constructor & Destructor Documentation

#### 6.45.1.1 PendingChange()

```
StateStack::PendingChange::PendingChange (
    Action action,
    States::ID stateID = States::None ) [explicit]
```

### 6.45.2 Member Data Documentation

#### 6.45.2.1 action

```
Action StateStack::PendingChange::action
```

### 6.45.2.2 stateID

```
States::ID StateStack::PendingChange::stateID
```

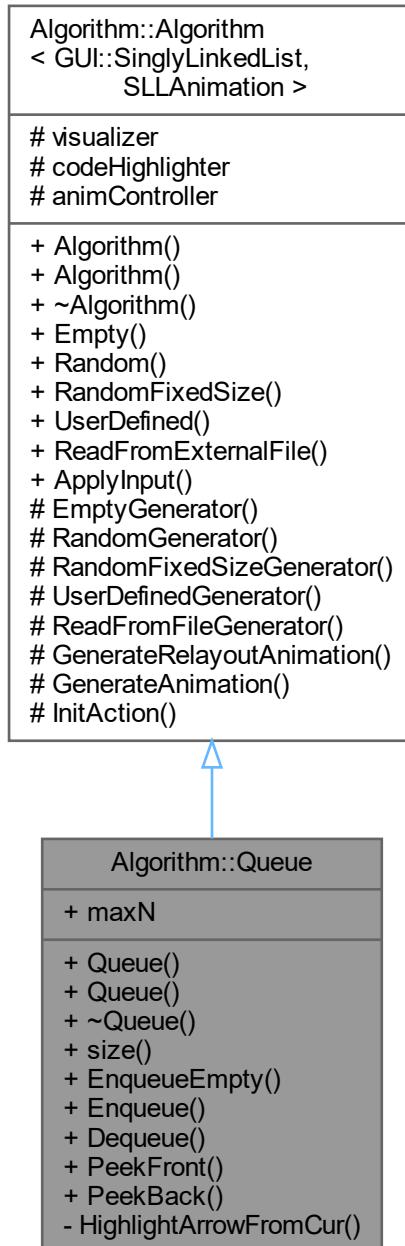
The documentation for this struct was generated from the following files:

- src/[StateStack.hpp](#)
- src/[StateStack.cpp](#)

## 6.46 Algorithm::Queue Class Reference

```
#include <Queue.hpp>
```

Inheritance diagram for Algorithm::Queue:



## Public Member Functions

- `Queue (GUI::CodeHighlighter::Ptr _codeHighlighter, SLLAnimationController::Ptr animController, FontHolder *fonts)`
- `Queue ()`
- `~Queue ()`
- `std::size_t size () const`

- void [EnqueueEmpty](#) (int value)
- void [Enqueue](#) (int value)
- void [Dequeue](#) ()
- void [PeekFront](#) ()
- void [PeekBack](#) ()
- virtual void [Empty](#) ()
- virtual void [Random](#) ()
- virtual void [RandomFixedSize](#) (int N)
- virtual void [UserDefined](#) (std::string input)
- virtual void [ReadFromExternalFile](#) (std::string path)
- virtual void [ApplyInput](#) (std::vector< int > input, std::size\_t nMaxSize=10)

## Static Public Attributes

- static constexpr int [maxN](#) = 10

## Protected Member Functions

- std::vector< int > [EmptyGenerator](#) ()
- std::vector< int > [RandomGenerator](#) ()
- std::vector< int > [RandomFixedSizeGenerator](#) (int nSize)
- std::vector< int > [UserDefinedGenerator](#) (std::string input)
- std::vector< int > [ReadFromFileGenerator](#) (std::string inputFile)
- virtual void [GenerateRelayoutAnimation](#) (Vector2 newPosition)
- virtual [SLLAnimation](#) [GenerateAnimation](#) (float duration, int highlightLine, std::string actionDescription)
- virtual void [InitAction](#) (std::vector< std::string > code)

## Protected Attributes

- [GUI::SinglyLinkedList](#) visualizer
- [GUI::CodeHighlighter::Ptr](#) codeHighlighter
- [Animation::AnimationController< SLLAnimation >::Ptr](#) animController

## Private Member Functions

- std::function< [GUI::SinglyLinkedList](#)([GUI::SinglyLinkedList](#), float, Vector2) > [HighlightArrowFromCur](#) (int index, bool drawVisualizer=true, bool reverse=false)

### 6.46.1 Constructor & Destructor Documentation

#### 6.46.1.1 Queue() [1/2]

```
Algorithm::Queue::Queue (
    GUI::CodeHighlighter::Ptr _codeHighlighter,
    SLLAnimationController::Ptr animController,
    FontHolder * fonts )
```

#### 6.46.1.2 Queue() [2/2]

```
Algorithm::Queue::Queue ( )
```

#### 6.46.1.3 ~Queue()

```
Algorithm::Queue::~Queue ( )
```

### 6.46.2 Member Function Documentation

#### 6.46.2.1 ApplyInput()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ApplyInput (
    std::vector< int > input,
    std::size_t nMaxSize = 10 ) [virtual], [inherited]
```

#### 6.46.2.2 Dequeue()

```
void Algorithm::Queue::Dequeue ( )
```

#### 6.46.2.3 Empty()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Empty [virtual], [inherited]
```

#### 6.46.2.4 EmptyGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Empty ←
Generator [protected], [inherited]
```

#### 6.46.2.5 Enqueue()

```
void Algorithm::Queue::Enqueue (
    int value )
```

#### 6.46.2.6 EnqueueEmpty()

```
void Algorithm::Queue::EnqueueEmpty (
    int value )
```

#### 6.46.2.7 GenerateAnimation()

```
SLLAnimation Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::GenerateAnimation (
    float duration,
    int highlightLine,
    std::string actionDescription ) [protected], [virtual], [inherited]
```

#### 6.46.2.8 GenerateRelayoutAnimation()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::GenerateRelayoutAnimation (
    Vector2 newPosition ) [inline], [protected], [virtual], [inherited]
```

#### 6.46.2.9 HighlightArrowFromCur()

```
std::function< GUI::SinglyLinkedList(GUI::SinglyLinkedList, float, Vector2) > Algorithm::Queue::HighlightArrowFromCur (
    int index,
    bool drawVisualizer = true,
    bool reverse = false ) [private]
```

#### 6.46.2.10 InitAction()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::InitAction (
    std::vector< std::string > code ) [protected], [virtual], [inherited]
```

#### 6.46.2.11 PeekBack()

```
void Algorithm::Queue::PeekBack ( )
```

### 6.46.2.12 PeekFront()

```
void Algorithm::Queue::PeekFront ( )
```

### 6.46.2.13 Random()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random [virtual], [inherited]
```

### 6.46.2.14 RandomFixedSize()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::RandomFixedSize (
    int N ) [virtual], [inherited]
```

### 6.46.2.15 RandomFixedSizeGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random←
FixedSizeGenerator (
    int nSize ) [protected], [inherited]
```

### 6.46.2.16 RandomGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random←
Generator [protected], [inherited]
```

### 6.46.2.17 ReadFromExternalFile()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ReadFromExternalFile (
    std::string path ) [virtual], [inherited]
```

### 6.46.2.18 ReadFromFileGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ReadFrom←
FileGenerator (
    std::string inputFile ) [protected], [inherited]
```

### 6.46.2.19 size()

```
std::size_t Algorithm::Queue::size () const
```

### 6.46.2.20 UserDefined()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::UserDefined ( std::string input ) [virtual], [inherited]
```

### 6.46.2.21 UserDefinedGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::UserDefined↔ Generator ( std::string input ) [protected], [inherited]
```

## 6.46.3 Member Data Documentation

### 6.46.3.1 animController

```
Animation::AnimationController< SLLAnimation >::Ptr Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::animController [protected], [inherited]
```

### 6.46.3.2 codeHighlighter

```
GUI::CodeHighlighter::Ptr Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::code↔ Highlighter [protected], [inherited]
```

### 6.46.3.3 maxN

```
constexpr int Algorithm::Queue::maxN = 10 [static], [constexpr]
```

#### 6.46.3.4 visualizer

```
GUI::SinglyLinkedList Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::visualizer  
[protected], [inherited]
```

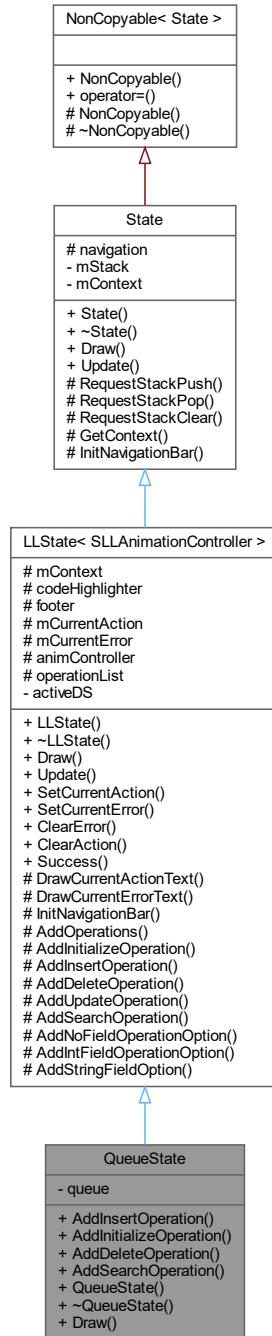
The documentation for this class was generated from the following files:

- src/Algorithms/LinkedList/Queue.hpp
- src/Algorithms/LinkedList/Queue.cpp

## 6.47 QueueState Class Reference

```
#include <QueueState.hpp>
```

Inheritance diagram for QueueState:



## Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `void AddInsertOperation ()`

- void `AddInitializeOperation ()`
- void `AddDeleteOperation ()`
- void `AddSearchOperation ()`
- `QueueState (StateStack &stack, Context context)`
- `~QueueState ()`
- void `Draw ()`
- virtual bool `Update (float dt)`
- virtual void `SetCurrentAction (std::string action)`
- virtual void `SetCurrentError (std::string error)`
- virtual void `ClearError ()`
- virtual void `ClearAction ()`
- virtual void `Success ()`

## Protected Member Functions

- virtual void `DrawCurrentActionText ()`
- virtual void `DrawCurrentErrorText ()`
- void `InitNavigationBar ()`
- virtual void `AddOperations ()`
- virtual void `AddUpdateOperation ()`
- virtual void `AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void()> action)`
- virtual void `AddIntFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >)> action)`
- virtual void `AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >)> action)`
- void `RequestStackPush (States::ID stateID)`
- void `RequestStackPop ()`
- void `RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< SLLAnimationController > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Attributes

- `Algorithm::Queue queue`
- `DataStructures::ID activeDS`
- `StateStack * mStack`

### 6.47.1 Member Typedef Documentation

### 6.47.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

## 6.47.2 Constructor & Destructor Documentation

### 6.47.2.1 QueueState()

```
QueueState::QueueState (
    StateStack & stack,
    Context context )
```

### 6.47.2.2 ~QueueState()

```
QueueState::~QueueState ( )
```

## 6.47.3 Member Function Documentation

### 6.47.3.1 AddDeleteOperation()

```
void QueueState::AddDeleteOperation ( ) [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

### 6.47.3.2 AddInitializeOperation()

```
void QueueState::AddInitializeOperation ( ) [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

### 6.47.3.3 AddInsertOperation()

```
void QueueState::AddInsertOperation ( ) [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

#### 6.47.3.4 AddIntFieldOperationOption()

```
void LLState< SLLAnimationController >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

#### 6.47.3.5 AddNoFieldOperationOption()

```
void LLState< SLLAnimationController >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual], [inherited]
```

#### 6.47.3.6 AddOperations()

```
void LLState< SLLAnimationController >::AddOperations [protected], [virtual], [inherited]
```

#### 6.47.3.7 AddSearchOperation()

```
void QueueState::AddSearchOperation () [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

#### 6.47.3.8 AddStringFieldOption()

```
void LLState< SLLAnimationController >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

#### 6.47.3.9 AddUpdateOperation()

```
void LLState< SLLAnimationController >::AddUpdateOperation [protected], [virtual], [inherited]
```

Reimplemented in [SLLState](#).

#### 6.47.3.10 ClearAction()

```
void LLState< SLLAnimationController >::ClearAction [inline], [virtual], [inherited]
```

#### 6.47.3.11 ClearError()

```
void LLState< SLLAnimationController >::ClearError [inline], [virtual], [inherited]
```

#### 6.47.3.12 Draw()

```
void QueueState::Draw () [virtual]
```

Implements [LLState< SLLAnimationController >](#).

#### 6.47.3.13 DrawCurrentActionText()

```
void LLState< SLLAnimationController >::DrawCurrentActionText [inline], [protected], [virtual], [inherited]
```

#### 6.47.3.14 DrawCurrentErrorText()

```
void LLState< SLLAnimationController >::DrawCurrentErrorText [inline], [protected], [virtual], [inherited]
```

#### 6.47.3.15 GetContext()

```
State::Context State::GetContext () const [protected], [inherited]
```

#### 6.47.3.16 InitNavigationBar()

```
void LLState< SLLAnimationController >::InitNavigationBar [protected], [inherited]
```

**6.47.3.17 RequestStackClear()**

```
void State::RequestStackClear ( ) [protected], [inherited]
```

**6.47.3.18 RequestStackPop()**

```
void State::RequestStackPop ( ) [protected], [inherited]
```

**6.47.3.19 RequestStackPush()**

```
void State::RequestStackPush (
    States::ID stateID ) [protected], [inherited]
```

**6.47.3.20 SetCurrentAction()**

```
void LLState< SLLAnimationController >::SetCurrentAction (
    std::string action ) [inline], [virtual], [inherited]
```

**6.47.3.21 SetCurrentError()**

```
void LLState< SLLAnimationController >::SetCurrentError (
    std::string error ) [inline], [virtual], [inherited]
```

**6.47.3.22 Success()**

```
void LLState< SLLAnimationController >::Success [inline], [virtual], [inherited]
```

**6.47.3.23 Update()**

```
bool LLState< SLLAnimationController >::Update (
    float dt ) [virtual], [inherited]
```

Implements [State](#).

## 6.47.4 Member Data Documentation

### 6.47.4.1 activeDS

```
DataStructures::ID LLState< SLLAnimationController >::activeDS [private], [inherited]
```

### 6.47.4.2 animController

```
T::Ptr LLState< SLLAnimationController >::animController [protected], [inherited]
```

### 6.47.4.3 codeHighlighter

```
GUI::CodeHighlighter::Ptr LLState< SLLAnimationController >::codeHighlighter [protected], [inherited]
```

### 6.47.4.4 footer

```
GUI::Footer< SLLAnimationController > LLState< SLLAnimationController >::footer [protected], [inherited]
```

### 6.47.4.5 mContext

```
Context LLState< SLLAnimationController >::mContext [protected], [inherited]
```

### 6.47.4.6 mCurrentAction

```
std::string LLState< SLLAnimationController >::mCurrentAction [protected], [inherited]
```

### 6.47.4.7 mCurrentError

```
std::string LLState< SLLAnimationController >::mCurrentError [protected], [inherited]
```

#### 6.47.4.8 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.47.4.9 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

#### 6.47.4.10 operationList

```
GUI::OperationList LLState< SLLAnimationController >::operationList [protected], [inherited]
```

#### 6.47.4.11 queue

```
Algorithm::Queue QueueState::queue [private]
```

The documentation for this class was generated from the following files:

- src/States/LinkedList/QueueState.hpp
- src/States/LinkedList/QueueState.cpp

## 6.48 SceneNode Class Reference

```
#include <SceneNode.hpp>
```

### Public Types

- `typedef std::shared_ptr< SceneNode > Ptr`

### Public Member Functions

- `SceneNode ()`
- `virtual ~SceneNode ()`
- `void AttachChild (Ptr child)`
- `Ptr DetachChild (const SceneNode &node)`
- `virtual void Draw ()`
- `virtual void DrawCurrent ()=0`

## Private Attributes

- std::vector< [Ptr](#) > mChildren
- [SceneNode](#) \* mParent

### 6.48.1 Member Typedef Documentation

#### 6.48.1.1 [Ptr](#)

```
typedef std::shared_ptr< SceneNode > SceneNode::Ptr
```

### 6.48.2 Constructor & Destructor Documentation

#### 6.48.2.1 [SceneNode\(\)](#)

```
SceneNode::SceneNode ( )
```

#### 6.48.2.2 [~SceneNode\(\)](#)

```
virtual SceneNode::~SceneNode ( ) [inline], [virtual]
```

### 6.48.3 Member Function Documentation

#### 6.48.3.1 [AttachChild\(\)](#)

```
void SceneNode::AttachChild (  
    Ptr child )
```

#### 6.48.3.2 [DetachChild\(\)](#)

```
SceneNode::Ptr SceneNode::DetachChild (   
    const SceneNode & node )
```

#### 6.48.3.3 Draw()

```
void SceneNode::Draw ( ) [virtual]
```

#### 6.48.3.4 DrawCurrent()

```
virtual void SceneNode::DrawCurrent ( ) [pure virtual]
```

### 6.48.4 Member Data Documentation

#### 6.48.4.1 mChildren

```
std::vector< Ptr > SceneNode::mChildren [private]
```

#### 6.48.4.2 mParent

```
SceneNode* SceneNode::mParent [private]
```

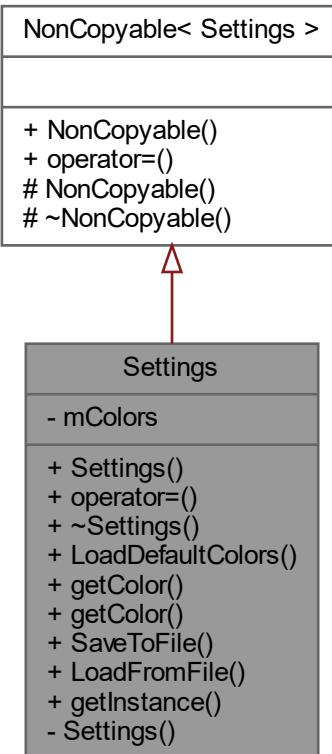
The documentation for this class was generated from the following files:

- src/[SceneNode.hpp](#)
- src/[SceneNode.cpp](#)

## 6.49 Settings Class Reference

```
#include <Settings.hpp>
```

Inheritance diagram for Settings:



## Public Member Functions

- `Settings (Settings &&)=delete`
- `Settings & operator= (Settings &&)=delete`
- `~Settings ()`
- `void LoadDefaultColors ()`
- `Color & getColor (ColorTheme::ID id)`
- `Color getColor (ColorTheme::ID id) const`
- `void SaveToFile (const std::string &path)`
- `void LoadFromFile (const std::string &path)`

## Static Public Member Functions

- `static Settings & getInstance ()`

## Private Member Functions

- `Settings ()=default`

## Private Attributes

- std::map< ColorTheme::ID, Color > mColors

### 6.49.1 Constructor & Destructor Documentation

#### 6.49.1.1 Settings() [1/2]

```
Settings::Settings ( )  [private], [default]
```

#### 6.49.1.2 Settings() [2/2]

```
Settings::Settings (
    Settings && )  [delete]
```

#### 6.49.1.3 ~Settings()

```
Settings::~Settings ( )
```

### 6.49.2 Member Function Documentation

#### 6.49.2.1 getColor() [1/2]

```
Color & Settings::getColor (
    ColorTheme::ID id )
```

#### 6.49.2.2 getColor() [2/2]

```
Color Settings::getColor (
    ColorTheme::ID id ) const
```

#### 6.49.2.3 getInstance()

```
Settings & Settings::getInstance ( ) [static]
```

#### 6.49.2.4 LoadDefaultColors()

```
void Settings::LoadDefaultColors ( )
```

#### 6.49.2.5 LoadFromFile()

```
void Settings::LoadFromFile (
    const std::string & path )
```

#### 6.49.2.6 operator=( )

```
Settings & Settings::operator= (
    Settings && ) [delete]
```

#### 6.49.2.7 SaveToFile()

```
void Settings::SaveToFile (
    const std::string & path )
```

### 6.49.3 Member Data Documentation

#### 6.49.3.1 mColors

```
std::map< ColorTheme::ID, Color > Settings::mColors [private]
```

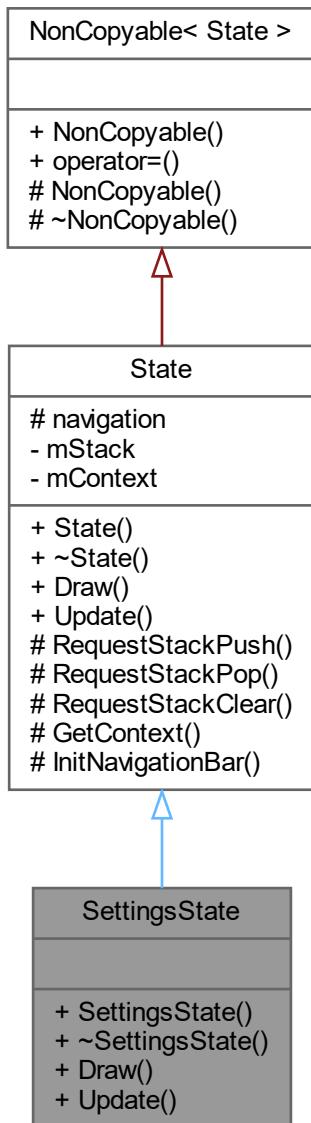
The documentation for this class was generated from the following files:

- src/[Settings.hpp](#)
- src/[Settings.cpp](#)

## 6.50 SettingsState Class Reference

```
#include <SettingsState.hpp>
```

Inheritance diagram for SettingsState:



### Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `SettingsState (StateStack &stack, Context context)`
- `~SettingsState ()`
- `void Draw ()`
- `bool Update (float dt)`

## Protected Member Functions

- `void RequestStackPush (States::ID stateID)`
- `void RequestStackPop ()`
- `void RequestStackClear ()`
- `Context GetContext () const`
- `void InitNavigationBar ()`

## Protected Attributes

- `GUI::NavigationBar navigation`

## Private Attributes

- `StateStack * mStack`
- `Context mContext`

### 6.50.1 Member Typedef Documentation

#### 6.50.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

### 6.50.2 Constructor & Destructor Documentation

#### 6.50.2.1 SettingsState()

```
SettingsState::SettingsState (
    StateStack & stack,
    Context context )
```

### 6.50.2.2 ~SettingsState()

```
SettingsState::~SettingsState ( )
```

## 6.50.3 Member Function Documentation

### 6.50.3.1 Draw()

```
void SettingsState::Draw ( ) [virtual]
```

Implements [State](#).

### 6.50.3.2 GetContext()

```
State::Context State::GetContext ( ) const [protected], [inherited]
```

### 6.50.3.3 InitNavigationBar()

```
void State::InitNavigationBar ( ) [protected], [inherited]
```

### 6.50.3.4 RequestStackClear()

```
void State::RequestStackClear ( ) [protected], [inherited]
```

### 6.50.3.5 RequestStackPop()

```
void State::RequestStackPop ( ) [protected], [inherited]
```

### 6.50.3.6 RequestStackPush()

```
void State::RequestStackPush ( States::ID stateID ) [protected], [inherited]
```

### 6.50.3.7 Update()

```
bool SettingsState::Update (
    float dt ) [virtual]
```

Implements [State](#).

## 6.50.4 Member Data Documentation

### 6.50.4.1 mContext

```
Context State::mContext [private], [inherited]
```

### 6.50.4.2 mStack

```
StateStack* State::mStack [private], [inherited]
```

### 6.50.4.3 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

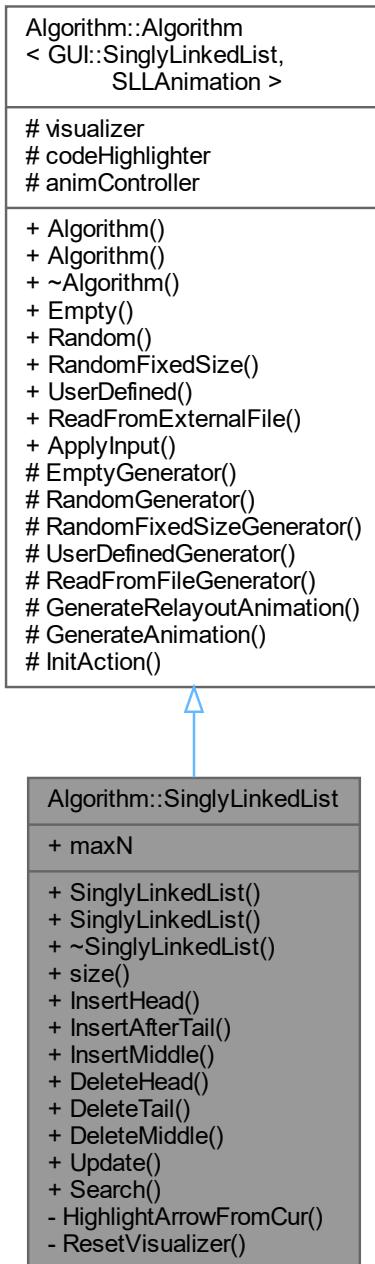
The documentation for this class was generated from the following files:

- src/States/[SettingsState.hpp](#)
- src/States/[SettingsState.cpp](#)

## 6.51 Algorithm::SinglyLinkedList Class Reference

```
#include <SinglyLinkedList.hpp>
```

Inheritance diagram for Algorithm::SinglyLinkedList:



### Public Member Functions

- [SinglyLinkedList \(\)](#)

- `SinglyLinkedList (GUI::CodeHighlighter::Ptr codeHighlighter, SLLAnimationController::Ptr animController, FontHolder *fonts)`
- `~SinglyLinkedList ()`
- `std::size_t size () const`
- `void InsertHead (int value)`
- `void InsertAfterTail (int value)`
- `void InsertMiddle (int index, int value)`
- `void DeleteHead ()`
- `void DeleteTail ()`
- `void DeleteMiddle (int index)`
- `void Update (int index, int value)`
- `void Search (int value)`
- `virtual void Empty ()`
- `virtual void Random ()`
- `virtual void RandomFixedSize (int N)`
- `virtual void UserDefined (std::string input)`
- `virtual void ReadFromExternalFile (std::string path)`
- `virtual void ApplyInput (std::vector< int > input, std::size_t nMaxSize=10)`

## Static Public Attributes

- `static constexpr int maxN = 10`

## Protected Member Functions

- `std::vector< int > EmptyGenerator ()`
- `std::vector< int > RandomGenerator ()`
- `std::vector< int > RandomFixedSizeGenerator (int nSize)`
- `std::vector< int > UserDefinedGenerator (std::string input)`
- `std::vector< int > ReadFromFileGenerator (std::string inputFile)`
- `virtual void GenerateRelayoutAnimation (Vector2 newPosition)`
- `virtual SLLAnimation GenerateAnimation (float duration, int highlightLine, std::string actionDescription)`
- `virtual void InitAction (std::vector< std::string > code)`

## Protected Attributes

- `GUI::SinglyLinkedList visualizer`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `Animation::AnimationController< SLLAnimation >::Ptr animController`

## Private Member Functions

- `std::function< GUI::SinglyLinkedList(GUI::SinglyLinkedList, float, Vector2) > HighlightArrowFromCur (int index, bool drawVisualizer=true, bool reverse=false)`
- `void ResetVisualizer ()`

### 6.51.1 Constructor & Destructor Documentation

### 6.51.1.1 **SinglyLinkedList()** [1/2]

```
Algorithm::SinglyLinkedList::SinglyLinkedList ( )
```

### 6.51.1.2 **SinglyLinkedList()** [2/2]

```
Algorithm::SinglyLinkedList::SinglyLinkedList (
    GUI::CodeHighlighter::Ptr codeHighlighter,
    SLLAnimationController::Ptr animController,
    FontHolder * fonts )
```

### 6.51.1.3 **~SinglyLinkedList()**

```
Algorithm::SinglyLinkedList::~SinglyLinkedList ( )
```

## 6.51.2 Member Function Documentation

### 6.51.2.1 **ApplyInput()**

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ApplyInput (
    std::vector< int > input,
    std::size_t nMaxSize = 10 ) [virtual], [inherited]
```

### 6.51.2.2 **DeleteHead()**

```
void Algorithm::SinglyLinkedList::DeleteHead ( )
```

### 6.51.2.3 **DeleteMiddle()**

```
void Algorithm::SinglyLinkedList::DeleteMiddle (
    int index )
```

#### 6.51.2.4 DeleteTail()

```
void Algorithm::SinglyLinkedList::DeleteTail ( )
```

#### 6.51.2.5 Empty()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Empty [virtual], [inherited]
```

#### 6.51.2.6 EmptyGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Empty<->  
Generator [protected], [inherited]
```

#### 6.51.2.7 GenerateAnimation()

```
SLLAnimation Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::GenerateAnimation (  
    float duration,  
    int highlightLine,  
    std::string actionDescription ) [protected], [virtual], [inherited]
```

#### 6.51.2.8 GenerateRelayoutAnimation()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::GenerateRelayoutAnimation (   
    Vector2 newPosition ) [inline], [protected], [virtual], [inherited]
```

#### 6.51.2.9 HighlightArrowFromCur()

```
std::function< GUI::SinglyLinkedList(GUI::SinglyLinkedList, float, Vector2) > Algorithm::->  
SinglyLinkedList::HighlightArrowFromCur (   
    int index,  
    bool drawVisualizer = true,  
    bool reverse = false ) [private]
```

### 6.51.2.10 **InitAction()**

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::InitAction (
    std::vector< std::string > code ) [protected], [virtual], [inherited]
```

### 6.51.2.11 **InsertAfterTail()**

```
void Algorithm::SinglyLinkedList::InsertAfterTail (
    int value )
```

### 6.51.2.12 **InsertHead()**

```
void Algorithm::SinglyLinkedList::InsertHead (
    int value )
```

### 6.51.2.13 **InsertMiddle()**

```
void Algorithm::SinglyLinkedList::InsertMiddle (
    int index,
    int value )
```

### 6.51.2.14 **Random()**

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random [virtual], [inherited]
```

### 6.51.2.15 **RandomFixedSize()**

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::RandomFixedSize (
    int N ) [virtual], [inherited]
```

### 6.51.2.16 **RandomFixedSizeGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random↔
FixedSizeGenerator (
    int nSize ) [protected], [inherited]
```

**6.51.2.17 RandomGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random←
Generator [protected], [inherited]
```

**6.51.2.18 ReadFromExternalFile()**

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ReadFromExternalFile (
    std::string path ) [virtual], [inherited]
```

**6.51.2.19 ReadFromFileGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ReadFrom←
FileGenerator (
    std::string inputFile ) [protected], [inherited]
```

**6.51.2.20 ResetVisualizer()**

```
void Algorithm::SinglyLinkedList::ResetVisualizer ( ) [private]
```

**6.51.2.21 Search()**

```
void Algorithm::SinglyLinkedList::Search (
    int value )
```

**6.51.2.22 size()**

```
std::size_t Algorithm::SinglyLinkedList::size ( ) const
```

**6.51.2.23 Update()**

```
void Algorithm::SinglyLinkedList::Update (
    int index,
    int value )
```

### 6.51.2.24 UserDefined()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::UserDefined (
    std::string input ) [virtual], [inherited]
```

### 6.51.2.25 UserDefinedGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::UserDefined↔
Generator (
    std::string input ) [protected], [inherited]
```

## 6.51.3 Member Data Documentation

### 6.51.3.1 animController

```
Animation::AnimationController< SLLAnimation >::Ptr Algorithm::Algorithm< GUI::SinglyLinkedList ,
, SLLAnimation >::animController [protected], [inherited]
```

### 6.51.3.2 codeHighlighter

```
GUI::CodeHighlighter::Ptr Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::code↔
Highlighter [protected], [inherited]
```

### 6.51.3.3 maxN

```
constexpr int Algorithm::SinglyLinkedList::maxN = 10 [static], [constexpr]
```

### 6.51.3.4 visualizer

```
GUI::SinglyLinkedList Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::visualizer
[protected], [inherited]
```

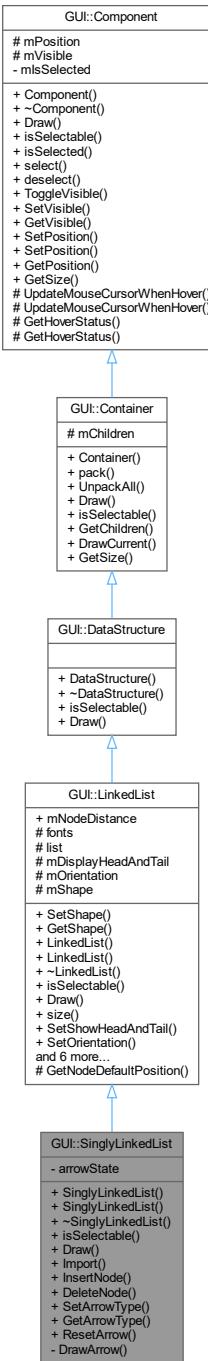
The documentation for this class was generated from the following files:

- src/Algorithms/LinkedList/[SinglyLinkedList.hpp](#)
- src/Algorithms/LinkedList/[SinglyLinkedList.cpp](#)

## 6.52 GUI::SinglyLinkedList Class Reference

```
#include <SinglyLinkedList.hpp>
```

Inheritance diagram for GUI::SinglyLinkedList:



### Public Types

- enum **ArrowType** {
 Default , Hidden , Active , Skip ,
 ArrowTypeCount
 }

- enum `Orientation` { `Horizontal` , `Vertical` , `OrientationCount` }
- typedef `std::shared_ptr< DataStructure > Ptr`

## Public Member Functions

- `SinglyLinkedList ()`
- `SinglyLinkedList (FontHolder *fonts)`
- `~SinglyLinkedList ()`
- bool `isSelectable () const`
- void `Draw (Vector2 base=(Vector2){0, 0}, float t=1.0f, bool init=false)`
- void `Import (std::vector< int > nodes)`
- void `InsertNode (std::size_t index, GUI::Node node, bool rePosition=true)`
- void `DeleteNode (std::size_t index, bool rePosition=true)`
- void `SetArrowType (std::size_t index, ArrowType type)`
- `ArrowType GetArrowType (std::size_t index)`
- void `ResetArrow ()`
- void `SetShape (GUI::Node::Shape shape)`
- `GUI::Node::Shape GetShape () const`
- virtual void `Draw (Vector2 base)`
- virtual std::size\_t `size () const`
- virtual void `SetShowHeadAndTail (bool show)`
- virtual void `SetOrientation (Orientation orientation)`
- virtual std::vector< `GUI::Node` > `& GetList ()`
- virtual `GUI::Node GenerateNode (int value)`
- virtual void `Relayout ()`
- void `pack (Component::Ptr component)`
- void `UnpackAll ()`
- `std::vector< Component::Ptr > GetChildren ()`
- virtual void `DrawCurrent (Vector2 base)`
- virtual Vector2 `GetSize ()`
- bool `isSelected () const`
- virtual void `select ()`
- virtual void `deselect ()`
- virtual void `ToggleVisible ()`
- virtual void `SetVisible (bool visible)`
- virtual bool `GetVisible ()`
- void `SetPosition (float x, float y)`
- void `SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Static Public Attributes

- static constexpr float `mNodeDistance` = 20

## Protected Member Functions

- `Vector2 GetNodeDefaultPosition (std::size_t index)`
- virtual void `UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- virtual void `UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- virtual bool `GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- virtual bool `GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `FontHolder * fonts`
- `std::vector< GUI::Node > list`
- `bool mDisplayHeadAndTail`
- `Orientation mOrientation = Orientation::Horizontal`
- `GUI::Node::Shape mShape = GUI::Node::Shape::Circle`
- `std::vector< Component::Ptr > mChildren`
- `Vector2 mPosition`
- `bool mVisible`

## Private Member Functions

- `void DrawArrow (Vector2 base, float t)`

## Private Attributes

- `std::vector< ArrowType > arrowState`
- `bool mIsSelected`

### 6.52.1 Member Typedef Documentation

#### 6.52.1.1 Ptr

```
typedef std::shared_ptr< DataStructure > GUI::DataStructure::Ptr [inherited]
```

### 6.52.2 Member Enumeration Documentation

#### 6.52.2.1 ArrowType

```
enum GUI::LinkedList::ArrowType [inherited]
```

##### Enumerator

Default	
Hidden	
Active	
Skip	
ArrowTypeCount	

### 6.52.2.2 Orientation

```
enum GUI::LinkedList::Orientation [inherited]
```

Enumerator

Horizontal	
Vertical	
OrientationCount	

### 6.52.3 Constructor & Destructor Documentation

#### 6.52.3.1 SinglyLinkedList() [1/2]

```
GUI::SinglyLinkedList::SinglyLinkedList ()
```

#### 6.52.3.2 SinglyLinkedList() [2/2]

```
GUI::SinglyLinkedList::SinglyLinkedList (
    FontHolder * fonts )
```

#### 6.52.3.3 ~SinglyLinkedList()

```
GUI::SinglyLinkedList::~SinglyLinkedList ()
```

### 6.52.4 Member Function Documentation

#### 6.52.4.1 DeleteNode()

```
void GUI::SinglyLinkedList::DeleteNode (
    std::size_t index,
    bool rePosition = true ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

#### 6.52.4.2 deselect()

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

#### 6.52.4.3 Draw() [1/2]

```
void GUI::DataStructure::Draw (
    Vector2 base ) [virtual], [inherited]
```

Reimplemented from [GUI::Container](#).

#### 6.52.4.4 Draw() [2/2]

```
void GUI::SinglyLinkedList::Draw (
    Vector2 base = (Vector2){0, 0},
    float t = 1.0f,
    bool init = false ) [virtual]
```

Implements [GUI::LinkedList](#).

#### 6.52.4.5 DrawArrow()

```
void GUI::SinglyLinkedList::DrawArrow (
    Vector2 base,
    float t ) [private]
```

#### 6.52.4.6 DrawCurrent()

```
void GUI::Container::DrawCurrent (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

#### 6.52.4.7 GenerateNode()

```
GUI::Node GUI::LinkedList::GenerateNode (
    int value ) [virtual], [inherited]
```

#### 6.52.4.8 GetArrowType()

```
GUI::SinglyLinkedList::ArrowType GUI::SinglyLinkedList::GetArrowType (
    std::size_t index )
```

#### 6.52.4.9 GetChildren()

```
std::vector< GUI::Component::Ptr > GUI::Container::GetChildren ( ) [inherited]
```

#### 6.52.4.10 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.52.4.11 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.52.4.12 GetList()

```
std::vector< GUI::Node > & GUI::LinkedList::GetList ( ) [virtual], [inherited]
```

#### 6.52.4.13 GetNodeDefaultPosition()

```
Vector2 GUI::LinkedList::GetNodeDefaultPosition (
    std::size_t index ) [protected], [inherited]
```

#### 6.52.4.14 GetPosition()

```
Vector2 GUI::Component::GetPosition ( ) [inherited]
```

#### 6.52.4.15 GetShape()

```
GUI::Node::Shape GUI::LinkedList::GetShape ( ) const [inherited]
```

#### 6.52.4.16 GetSize()

```
Vector2 GUI::Container::GetSize ( ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

Reimplemented in [GUI::OperationList](#), and [GUI::OptionInputField](#).

#### 6.52.4.17 GetVisible()

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

#### 6.52.4.18 Import()

```
void GUI::SinglyLinkedList::Import (
    std::vector< int > nodes ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

#### 6.52.4.19 InsertNode()

```
void GUI::SinglyLinkedList::InsertNode (
    std::size_t index,
    GUI::Node node,
    bool rePosition = true ) [virtual]
```

Reimplemented from [GUI::LinkedList](#).

#### 6.52.4.20 isSelectable()

```
bool GUI::SinglyLinkedList::isSelectable ( ) const [virtual]
```

Reimplemented from [GUI::LinkedList](#).

**6.52.4.21 isSelected()**

```
bool GUI::Component::isSelected ( ) const [inherited]
```

**6.52.4.22 pack()**

```
void GUI::Container::pack ( Component::Ptr component ) [inherited]
```

**6.52.4.23 Relayout()**

```
void GUI::LinkedList::Relayout ( ) [virtual], [inherited]
```

**6.52.4.24 ResetArrow()**

```
void GUI::SinglyLinkedList::ResetArrow ( )
```

**6.52.4.25 select()**

```
void GUI::Component::select ( ) [virtual], [inherited]
```

**6.52.4.26 SetArrowType()**

```
void GUI::SinglyLinkedList::SetArrowType ( std::size_t index, ArrowType type )
```

**6.52.4.27 SetOrientation()**

```
void GUI::LinkedList::SetOrientation ( Orientation orientation ) [virtual], [inherited]
```

**6.52.4.28 SetPosition() [1/2]**

```
void GUI::Component::SetPosition (
    float x,
    float y )  [inherited]
```

**6.52.4.29 SetPosition() [2/2]**

```
void GUI::Component::SetPosition (
    Vector2 position )  [inherited]
```

**6.52.4.30 SetShape()**

```
void GUI::LinkedList::SetShape (
    GUI::Node::Shape shape )  [inherited]
```

**6.52.4.31 SetShowHeadAndTail()**

```
void GUI::LinkedList::SetShowHeadAndTail (
    bool show )  [virtual], [inherited]
```

**6.52.4.32 SetVisible()**

```
void GUI::Component::SetVisible (
    bool visible )  [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

**6.52.4.33 size()**

```
std::size_t GUI::LinkedList::size ( ) const  [virtual], [inherited]
```

**6.52.4.34 ToggleVisible()**

```
void GUI::Component::ToggleVisible ( )  [virtual], [inherited]
```

#### 6.52.4.35 UnpackAll()

```
void GUI::Container::UnpackAll ( ) [inherited]
```

#### 6.52.4.36 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

#### 6.52.4.37 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.52.5 Member Data Documentation

#### 6.52.5.1 arrowState

```
std::vector< ArrowType > GUI::SinglyLinkedList::arrowState [private]
```

#### 6.52.5.2 fonts

```
FontHolder* GUI::LinkedList::fonts [protected], [inherited]
```

#### 6.52.5.3 list

```
std::vector< GUI::Node > GUI::LinkedList::list [protected], [inherited]
```

#### 6.52.5.4 mChildren

```
std::vector< Component::Ptr > GUI::Container::mChildren [protected], [inherited]
```

#### 6.52.5.5 mDisplayHeadAndTail

```
bool GUI::LinkedList::mDisplayHeadAndTail [protected], [inherited]
```

#### 6.52.5.6 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.52.5.7 mNodeDistance

```
constexpr float GUI::LinkedList::mNodeDistance = 20 [static], [constexpr], [inherited]
```

#### 6.52.5.8 mOrientation

```
Orientation GUI::LinkedList::mOrientation = Orientation::Horizontal [protected], [inherited]
```

#### 6.52.5.9 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.52.5.10 mShape

```
GUI::Node::Shape GUI::LinkedList::mShape = GUI::Node::Shape::Circle [protected], [inherited]
```

### 6.52.5.11 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

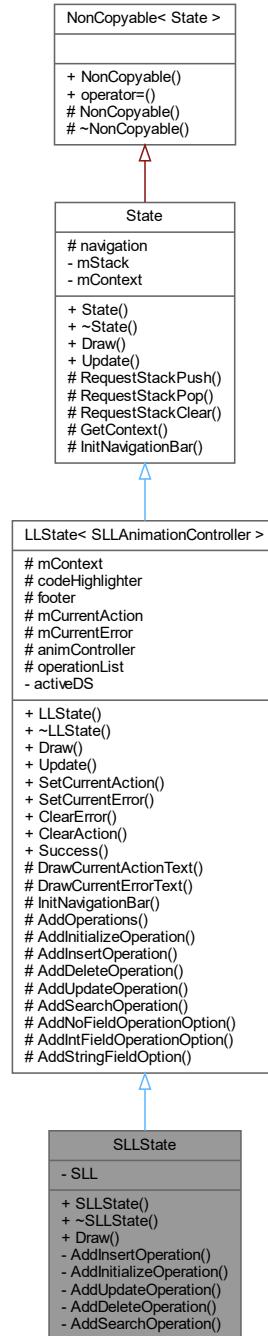
The documentation for this class was generated from the following files:

- src/Components/Visualization/[SinglyLinkedList.hpp](#)
- src/Components/Visualization/[SinglyLinkedList.cpp](#)

## 6.53 SLLState Class Reference

```
#include <SLLState.hpp>
```

Inheritance diagram for SLLState:



## Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `SLLState (StateStack &stack, Context context)`

- `~SLLState ()`
- `void Draw ()`
- `virtual bool Update (float dt)`
- `virtual void SetCurrentAction (std::string action)`
- `virtual void SetCurrentError (std::string error)`
- `virtual void ClearError ()`
- `virtual void ClearAction ()`
- `virtual void Success ()`

## Protected Member Functions

- `virtual void DrawCurrentActionText ()`
- `virtual void DrawCurrentErrorText ()`
- `void InitNavigationBar ()`
- `virtual void AddOperations ()`
- `virtual void AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void() > action)`
- `virtual void AddIntFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >) > action)`
- `virtual void AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >) > action)`
- `void RequestStackPush (States::ID stateID)`
- `void RequestStackPop ()`
- `void RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< SLLAnimationController > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Member Functions

- `void AddInsertOperation ()`
- `void AddInitializeOperation ()`
- `void AddUpdateOperation ()`
- `void AddDeleteOperation ()`
- `void AddSearchOperation ()`

## Private Attributes

- `Algorithm::SinglyLinkedList SLL`
- `DataStructures::ID activeDS`
- `StateStack * mStack`

### 6.53.1 Member Typedef Documentation

#### 6.53.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

### 6.53.2 Constructor & Destructor Documentation

#### 6.53.2.1 SLLState()

```
SLLState::SLLState (
    StateStack & stack,
    Context context )
```

#### 6.53.2.2 ~SLLState()

```
SLLState::~SLLState ( )
```

### 6.53.3 Member Function Documentation

#### 6.53.3.1 AddDeleteOperation()

```
void SLLState::AddDeleteOperation ( ) [private], [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

#### 6.53.3.2 AddInitializeOperation()

```
void SLLState::AddInitializeOperation ( ) [private], [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

### 6.53.3.3 AddInsertOperation()

```
void SLLState::AddInsertOperation ( ) [private], [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

### 6.53.3.4 AddIntFieldOperationOption()

```
void LLState< SLLAnimationController >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

### 6.53.3.5 AddNoFieldOperationOption()

```
void LLState< SLLAnimationController >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual], [inherited]
```

### 6.53.3.6 AddOperations()

```
void LLState< SLLAnimationController >::AddOperations [protected], [virtual], [inherited]
```

### 6.53.3.7 AddSearchOperation()

```
void SLLState::AddSearchOperation ( ) [private], [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

### 6.53.3.8 AddStringFieldOption()

```
void LLState< SLLAnimationController >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

### 6.53.3.9 AddUpdateOperation()

```
void SLLState::AddUpdateOperation ( ) [private], [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

### 6.53.3.10 ClearAction()

```
void LLState< SLLAnimationController >::ClearAction [inline], [virtual], [inherited]
```

### 6.53.3.11 ClearError()

```
void LLState< SLLAnimationController >::ClearError [inline], [virtual], [inherited]
```

### 6.53.3.12 Draw()

```
void SLLState::Draw ( ) [virtual]
```

Implements [LLState< SLLAnimationController >](#).

### 6.53.3.13 DrawCurrentActionText()

```
void LLState< SLLAnimationController >::DrawCurrentActionText [inline], [protected], [virtual], [inherited]
```

### 6.53.3.14 DrawCurrentErrorText()

```
void LLState< SLLAnimationController >::DrawCurrentErrorText [inline], [protected], [virtual], [inherited]
```

### 6.53.3.15 GetContext()

```
State::Context State::GetContext ( ) const [protected], [inherited]
```

**6.53.3.16 InitNavigationBar()**

```
void LLState< SLLAnimationController >::InitNavigationBar [protected], [inherited]
```

**6.53.3.17 RequestStackClear()**

```
void State::RequestStackClear () [protected], [inherited]
```

**6.53.3.18 RequestStackPop()**

```
void State::RequestStackPop () [protected], [inherited]
```

**6.53.3.19 RequestStackPush()**

```
void State::RequestStackPush (
    States::ID stateID ) [protected], [inherited]
```

**6.53.3.20 SetCurrentAction()**

```
void LLState< SLLAnimationController >::SetCurrentAction (
    std::string action ) [inline], [virtual], [inherited]
```

**6.53.3.21 SetCurrentError()**

```
void LLState< SLLAnimationController >::SetCurrentError (
    std::string error ) [inline], [virtual], [inherited]
```

**6.53.3.22 Success()**

```
void LLState< SLLAnimationController >::Success [inline], [virtual], [inherited]
```

### 6.53.3.23 Update()

```
bool LLState< SLLAnimationController >::Update (
    float dt ) [virtual], [inherited]
```

Implements [State](#).

## 6.53.4 Member Data Documentation

### 6.53.4.1 activeDS

```
DataStructures::ID LLState< SLLAnimationController >::activeDS [private], [inherited]
```

### 6.53.4.2 animController

```
T::Ptr LLState< SLLAnimationController >::animController [protected], [inherited]
```

### 6.53.4.3 codeHighlighter

```
GUI::CodeHighlighter::Ptr LLState< SLLAnimationController >::codeHighlighter [protected],
[inherited]
```

### 6.53.4.4 footer

```
GUI::Footer< SLLAnimationController > LLState< SLLAnimationController >::footer [protected],
[inherited]
```

### 6.53.4.5 mContext

```
Context LLState< SLLAnimationController >::mContext [protected], [inherited]
```

#### 6.53.4.6 mCurrentAction

```
std::string LLState< SLLAnimationController >::mCurrentAction [protected], [inherited]
```

#### 6.53.4.7 mCurrentError

```
std::string LLState< SLLAnimationController >::mCurrentError [protected], [inherited]
```

#### 6.53.4.8 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.53.4.9 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

#### 6.53.4.10 operationList

```
GUI::OperationList LLState< SLLAnimationController >::operationList [protected], [inherited]
```

#### 6.53.4.11 SLL

```
Algorithm::SinglyLinkedList SLLState::SLL [private]
```

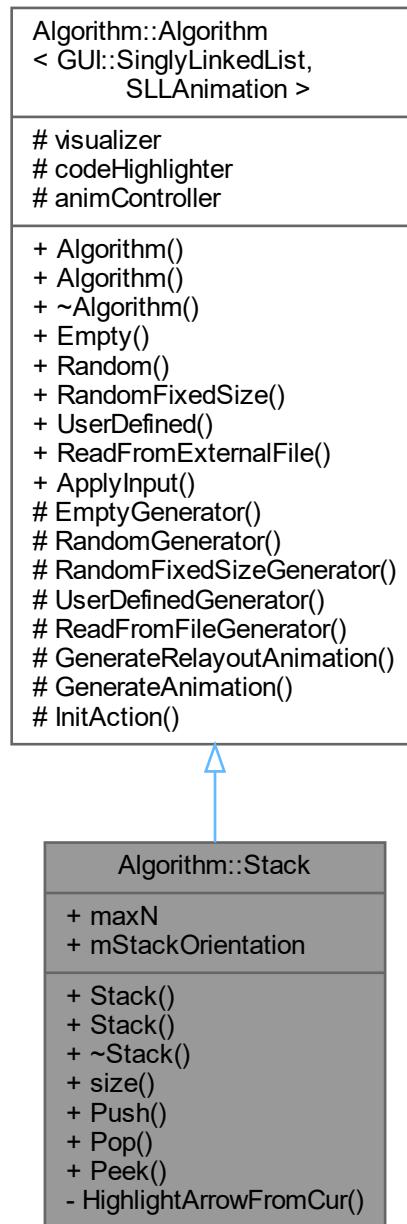
The documentation for this class was generated from the following files:

- src/States/LinkedList/[SLLState.hpp](#)
- src/States/LinkedList/[SLLState.cpp](#)

## 6.54 Algorithm::Stack Class Reference

```
#include <Stack.hpp>
```

Inheritance diagram for Algorithm::Stack:



### Public Member Functions

- `Stack (GUI::CodeHighlighter::Ptr codeHighlighter, SLLAnimationController::Ptr animController, FontHolder *fonts)`

- `Stack ()`
- `~Stack ()`
- `std::size_t size () const`
- `void Push (int value)`
- `void Pop ()`
- `void Peek ()`
- `virtual void Empty ()`
- `virtual void Random ()`
- `virtual void RandomFixedSize (int N)`
- `virtual void UserDefined (std::string input)`
- `virtual void ReadFromExternalFile (std::string path)`
- `virtual void ApplyInput (std::vector< int > input, std::size_t nMaxSize=10)`

## Static Public Attributes

- `static constexpr int maxN = 10`
- `static constexpr Orientation mStackOrientation = Orientation::Vertical`

## Protected Member Functions

- `std::vector< int > EmptyGenerator ()`
- `std::vector< int > RandomGenerator ()`
- `std::vector< int > RandomFixedSizeGenerator (int nSize)`
- `std::vector< int > UserDefinedGenerator (std::string input)`
- `std::vector< int > ReadFromFileGenerator (std::string inputFile)`
- `virtual void GenerateRelayoutAnimation (Vector2 newPosition)`
- `virtual SLLAnimation GenerateAnimation (float duration, int highlightLine, std::string actionDescription)`
- `virtual void InitAction (std::vector< std::string > code)`

## Protected Attributes

- `GUI::SinglyLinkedList visualizer`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `Animation::AnimationController< SLLAnimation >::Ptr animController`

## Private Member Functions

- `std::function< GUI::SinglyLinkedList(GUI::SinglyLinkedList, float, Vector2) > HighlightArrowFromCur (int index, bool drawVisualizer=true, bool reverse=false)`

### 6.54.1 Constructor & Destructor Documentation

### 6.54.1.1 Stack() [1/2]

```
Algorithm::Stack::Stack (
    GUI::CodeHighlighter::Ptr codeHighlighter,
    SLLAnimationController::Ptr animController,
    FontHolder * fonts )
```

### 6.54.1.2 Stack() [2/2]

```
Algorithm::Stack::Stack ( )
```

### 6.54.1.3 ~Stack()

```
Algorithm::Stack::~Stack ( )
```

## 6.54.2 Member Function Documentation

### 6.54.2.1 ApplyInput()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ApplyInput (
    std::vector< int > input,
    std::size_t nMaxSize = 10 ) [virtual], [inherited]
```

### 6.54.2.2 Empty()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Empty [virtual], [inherited]
```

### 6.54.2.3 EmptyGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Empty<->
Generator [protected], [inherited]
```

#### 6.54.2.4 GenerateAnimation()

```
SLLAnimation Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::GenerateAnimation (
    float duration,
    int highlightLine,
    std::string actionDescription ) [protected], [virtual], [inherited]
```

#### 6.54.2.5 GenerateRelayoutAnimation()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::GenerateRelayoutAnimation (
    Vector2 newPosition ) [inline], [protected], [virtual], [inherited]
```

#### 6.54.2.6 HighlightArrowFromCur()

```
std::function< GUI::SinglyLinkedList(GUI::SinglyLinkedList, float, Vector2) > Algorithm::->
Stack::HighlightArrowFromCur (
    int index,
    bool drawVisualizer = true,
    bool reverse = false ) [private]
```

#### 6.54.2.7 InitAction()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::InitAction (
    std::vector< std::string > code ) [protected], [virtual], [inherited]
```

#### 6.54.2.8 Peek()

```
void Algorithm::Stack::Peek ( )
```

#### 6.54.2.9 Pop()

```
void Algorithm::Stack::Pop ( )
```

**6.54.2.10 Push()**

```
void Algorithm::Stack::Push (
    int value )
```

**6.54.2.11 Random()**

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random [virtual], [inherited]
```

**6.54.2.12 RandomFixedSize()**

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::RandomFixedSize (
    int N ) [virtual], [inherited]
```

**6.54.2.13 RandomFixedSizeGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random←
FixedSizeGenerator (
    int nSize ) [protected], [inherited]
```

**6.54.2.14 RandomGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::Random←
Generator [protected], [inherited]
```

**6.54.2.15 ReadFromExternalFile()**

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ReadFromExternalFile (
    std::string path ) [virtual], [inherited]
```

**6.54.2.16 ReadFromFileGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::ReadFrom←
FileGenerator (
    std::string inputFile ) [protected], [inherited]
```

### 6.54.2.17 size()

```
std::size_t Algorithm::Stack::size () const
```

### 6.54.2.18 UserDefined()

```
void Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::UserDefined (
    std::string input ) [virtual], [inherited]
```

### 6.54.2.19 UserDefinedGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::UserDefined↔
Generator (
    std::string input ) [protected], [inherited]
```

## 6.54.3 Member Data Documentation

### 6.54.3.1 animController

```
Animation::AnimationController< SLLAnimation >::Ptr Algorithm::Algorithm< GUI::SinglyLinkedList ,
, SLLAnimation >::animController [protected], [inherited]
```

### 6.54.3.2 codeHighlighter

```
GUI::CodeHighlighter::Ptr Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::code↔
Highlighter [protected], [inherited]
```

### 6.54.3.3 maxN

```
constexpr int Algorithm::Stack::maxN = 10 [static], [constexpr]
```

#### 6.54.3.4 mStackOrientation

```
constexpr Orientation Algorithm::Stack::mStackOrientation = Orientation::Vertical [static],  
[constexpr]
```

#### 6.54.3.5 visualizer

```
GUI::SinglyLinkedList Algorithm::Algorithm< GUI::SinglyLinkedList , SLLAnimation >::visualizer  
[protected], [inherited]
```

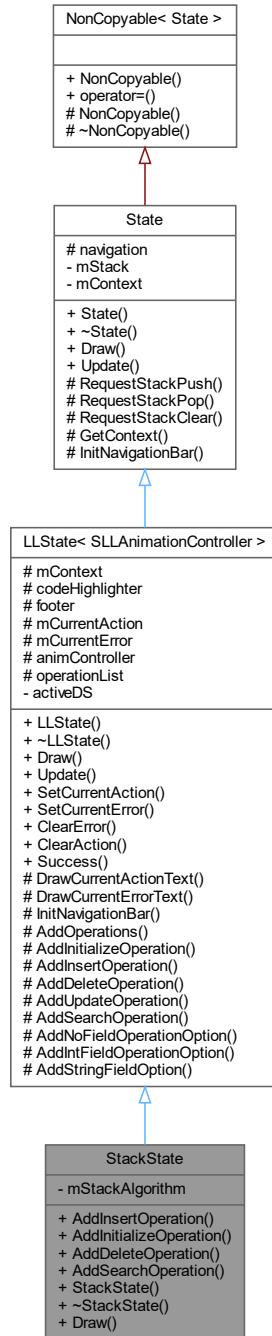
The documentation for this class was generated from the following files:

- src/Algorithms/LinkedList/[Stack.hpp](#)
- src/Algorithms/LinkedList/[Stack.cpp](#)

## 6.55 StackState Class Reference

```
#include <StackState.hpp>
```

Inheritance diagram for StackState:



## Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `void AddInsertOperation ()`

- void `AddInitializeOperation ()`
- void `AddDeleteOperation ()`
- void `AddSearchOperation ()`
- `StackState (StateStack &stack, Context context)`
- `~StackState ()`
- void `Draw ()`
- virtual bool `Update (float dt)`
- virtual void `SetCurrentAction (std::string action)`
- virtual void `SetCurrentError (std::string error)`
- virtual void `ClearError ()`
- virtual void `ClearAction ()`
- virtual void `Success ()`

## Protected Member Functions

- virtual void `DrawCurrentActionText ()`
- virtual void `DrawCurrentErrorText ()`
- void `InitNavigationBar ()`
- virtual void `AddOperations ()`
- virtual void `AddUpdateOperation ()`
- virtual void `AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void()> action)`
- virtual void `AddIntFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >)> action)`
- virtual void `AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >)> action)`
- void `RequestStackPush (States::ID stateID)`
- void `RequestStackPop ()`
- void `RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< SLLAnimationController > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Attributes

- `Algorithm::Stack mStackAlgorithm`
- `DataStructures::ID activeDS`
- `StateStack * mStack`

### 6.55.1 Member Typedef Documentation

### 6.55.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

## 6.55.2 Constructor & Destructor Documentation

### 6.55.2.1 StackState()

```
StackState::StackState (
    StateStack & stack,
    Context context )
```

### 6.55.2.2 ~StackState()

```
StackState::~StackState ( )
```

## 6.55.3 Member Function Documentation

### 6.55.3.1 AddDeleteOperation()

```
void StackState::AddDeleteOperation ( ) [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

### 6.55.3.2 AddInitializeOperation()

```
void StackState::AddInitializeOperation ( ) [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

### 6.55.3.3 AddInsertOperation()

```
void StackState::AddInsertOperation ( ) [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

#### 6.55.3.4 AddIntFieldOperationOption()

```
void LLState< SLLAnimationController >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

#### 6.55.3.5 AddNoFieldOperationOption()

```
void LLState< SLLAnimationController >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual], [inherited]
```

#### 6.55.3.6 AddOperations()

```
void LLState< SLLAnimationController >::AddOperations [protected], [virtual], [inherited]
```

#### 6.55.3.7 AddSearchOperation()

```
void StackState::AddSearchOperation ( ) [virtual]
```

Reimplemented from [LLState< SLLAnimationController >](#).

#### 6.55.3.8 AddStringFieldOption()

```
void LLState< SLLAnimationController >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

#### 6.55.3.9 AddUpdateOperation()

```
void LLState< SLLAnimationController >::AddUpdateOperation [protected], [virtual], [inherited]
```

Reimplemented in [SLLState](#).

### 6.55.3.10 ClearAction()

```
void LLState< SLLAnimationController >::ClearAction [inline], [virtual], [inherited]
```

### 6.55.3.11 ClearError()

```
void LLState< SLLAnimationController >::ClearError [inline], [virtual], [inherited]
```

### 6.55.3.12 Draw()

```
void StackState::Draw () [virtual]
```

Implements [LLState< SLLAnimationController >](#).

### 6.55.3.13 DrawCurrentActionText()

```
void LLState< SLLAnimationController >::DrawCurrentActionText [inline], [protected], [virtual], [inherited]
```

### 6.55.3.14 DrawCurrentErrorText()

```
void LLState< SLLAnimationController >::DrawCurrentErrorText [inline], [protected], [virtual], [inherited]
```

### 6.55.3.15 GetContext()

```
State::Context State::GetContext () const [protected], [inherited]
```

### 6.55.3.16 InitNavigationBar()

```
void LLState< SLLAnimationController >::InitNavigationBar [protected], [inherited]
```

**6.55.3.17 RequestStackClear()**

```
void State::RequestStackClear ( ) [protected], [inherited]
```

**6.55.3.18 RequestStackPop()**

```
void State::RequestStackPop ( ) [protected], [inherited]
```

**6.55.3.19 RequestStackPush()**

```
void State::RequestStackPush (
    States::ID stateID ) [protected], [inherited]
```

**6.55.3.20 SetCurrentAction()**

```
void LLState< SLLAnimationController >::SetCurrentAction (
    std::string action ) [inline], [virtual], [inherited]
```

**6.55.3.21 SetCurrentError()**

```
void LLState< SLLAnimationController >::SetCurrentError (
    std::string error ) [inline], [virtual], [inherited]
```

**6.55.3.22 Success()**

```
void LLState< SLLAnimationController >::Success [inline], [virtual], [inherited]
```

**6.55.3.23 Update()**

```
bool LLState< SLLAnimationController >::Update (
    float dt ) [virtual], [inherited]
```

Implements [State](#).

## 6.55.4 Member Data Documentation

### 6.55.4.1 activeDS

```
DataStructures::ID LLState< SLLAnimationController >::activeDS [private], [inherited]
```

### 6.55.4.2 animController

```
T::Ptr LLState< SLLAnimationController >::animController [protected], [inherited]
```

### 6.55.4.3 codeHighlighter

```
GUI::CodeHighlighter::Ptr LLState< SLLAnimationController >::codeHighlighter [protected], [inherited]
```

### 6.55.4.4 footer

```
GUI::Footer< SLLAnimationController > LLState< SLLAnimationController >::footer [protected], [inherited]
```

### 6.55.4.5 mContext

```
Context LLState< SLLAnimationController >::mContext [protected], [inherited]
```

### 6.55.4.6 mCurrentAction

```
std::string LLState< SLLAnimationController >::mCurrentAction [protected], [inherited]
```

### 6.55.4.7 mCurrentError

```
std::string LLState< SLLAnimationController >::mCurrentError [protected], [inherited]
```

### 6.55.4.8 mStack

```
StateStack* State::mStack [private], [inherited]
```

### 6.55.4.9 mStackAlgorithm

```
Algorithm::Stack StackState::mStackAlgorithm [private]
```

### 6.55.4.10 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

### 6.55.4.11 operationList

```
GUI::OperationList LLState< SLLAnimationController >::operationList [protected], [inherited]
```

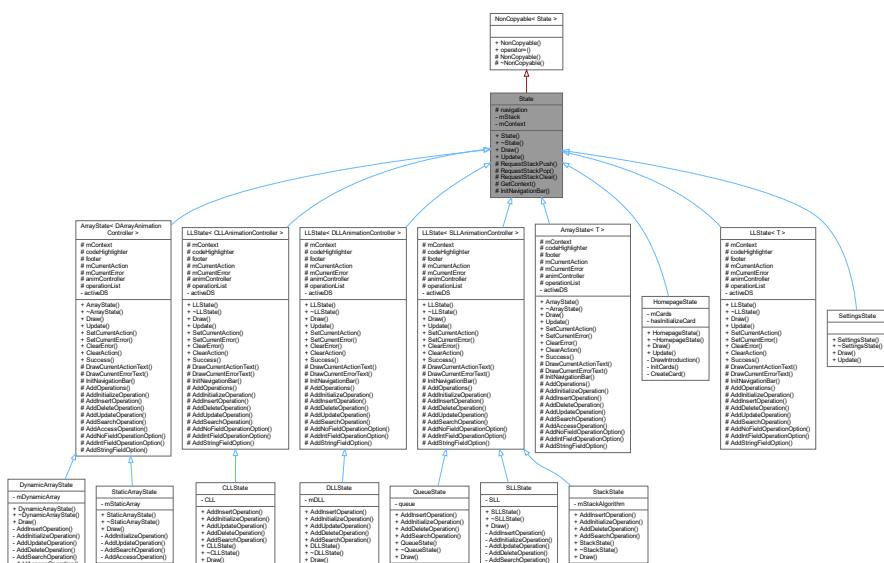
The documentation for this class was generated from the following files:

- src/States/LinkedList/StackState.hpp
- src/States/LinkedList/StackState.cpp

## 6.56 State Class Reference

```
#include <State.hpp>
```

Inheritance diagram for State:



## Classes

- struct [Context](#)

## Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `State (StateStack &stack, Context context)`
- `virtual ~State ()`
- `virtual void Draw ()=0`
- `virtual bool Update (float dt)=0`

## Protected Member Functions

- `void RequestStackPush (States::ID stateID)`
- `void RequestStackPop ()`
- `void RequestStackClear ()`
- `Context GetContext () const`
- `void InitNavigationBar ()`

## Protected Attributes

- `GUI::NavigationBar navigation`

## Private Attributes

- `StateStack * mStack`
- `Context mContext`

### 6.56.1 Member Typedef Documentation

#### 6.56.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr
```

### 6.56.2 Constructor & Destructor Documentation

### 6.56.2.1 State()

```
State::State (
    StateStack & stack,
    Context context )
```

### 6.56.2.2 ~State()

```
State::~State ( ) [virtual]
```

## 6.56.3 Member Function Documentation

### 6.56.3.1 Draw()

```
virtual void State::Draw ( ) [pure virtual]
```

Implemented in [DynamicArrayList](#), [StaticArrayList](#), [HomepageList](#), [CLLList](#), [DLLList](#), [QueueList](#), [SLLList](#), [StackList](#), [SettingsList](#), [ArrayList< T >](#), [ArrayList< DArrayAnimationController >](#), [LLList< T >](#), [LLList< CLLAnimationController >](#), [LLList< DLLAnimationController >](#), and [LLList< SLLAnimationController >](#).

### 6.56.3.2 GetContext()

```
State::Context State::GetContext ( ) const [protected]
```

### 6.56.3.3 InitNavigationBar()

```
void State::InitNavigationBar ( ) [protected]
```

### 6.56.3.4 RequestStackClear()

```
void State::RequestStackClear ( ) [protected]
```

### 6.56.3.5 RequestStackPop()

```
void State::RequestStackPop ( ) [protected]
```

### 6.56.3.6 RequestStackPush()

```
void State::RequestStackPush (
    States::ID stateID ) [protected]
```

### 6.56.3.7 Update()

```
virtual bool State::Update (
    float dt ) [pure virtual]
```

Implemented in [ArrayState< T >](#), [ArrayState< DArrayAnimationController >](#), [HomepageState](#), [LLState< T >](#), [LLState< CLAnimationController >](#), [LLState< DLLAnimationController >](#), [LLState< SLLAnimationController >](#), and [SettingsState](#).

## 6.56.4 Member Data Documentation

### 6.56.4.1 mContext

```
Context State::mContext [private]
```

### 6.56.4.2 mStack

```
StateStack* State::mStack [private]
```

### 6.56.4.3 navigation

```
GUI::NavigationBar State::navigation [protected]
```

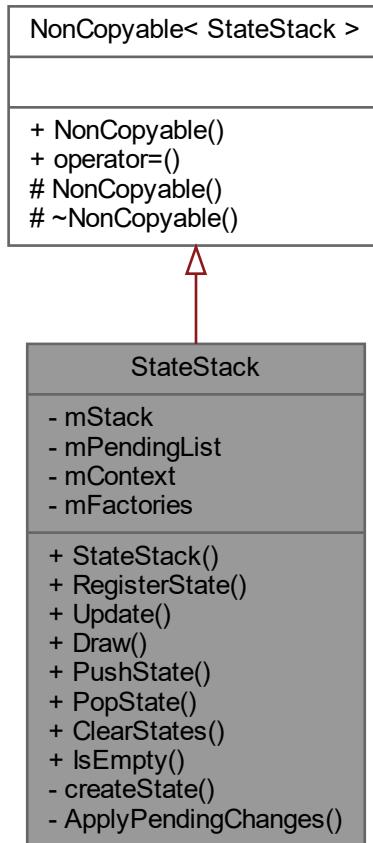
The documentation for this class was generated from the following files:

- src/[State.hpp](#)
- src/[State.cpp](#)

## 6.57 StateStack Class Reference

```
#include <StateStack.hpp>
```

Inheritance diagram for StateStack:



### Classes

- struct [PendingChange](#)

### Public Types

- enum class [Action](#) { `Push` , `Pop` , `Clear` }

## Public Member Functions

- `StateStack (State::Context context)`
- template<class T >  
  `void RegisterState (States::ID stateID)`
- `void Update (float dt)`
- `void Draw ()`
- `void PushState (States::ID stateID)`
- `void PopState ()`
- `void ClearStates ()`
- `bool IsEmpty () const`

## Private Member Functions

- `State::Ptr createState (States::ID stateID)`
- `void ApplyPendingChanges ()`

## Private Attributes

- `std::vector< State::Ptr > mStack`
- `std::vector< PendingChange > mPendingList`
- `State::Context mContext`
- `std::map< States::ID, std::function< State::Ptr() > > mFactories`

### 6.57.1 Member Enumeration Documentation

#### 6.57.1.1 Action

```
enum class StateStack::Action [strong]
```

Enumerator

Push	
Pop	
Clear	

### 6.57.2 Constructor & Destructor Documentation

#### 6.57.2.1 StateStack()

```
StateStack::StateStack (
    State::Context context ) [explicit]
```

### 6.57.3 Member Function Documentation

#### 6.57.3.1 ApplyPendingChanges()

```
void StateStack::ApplyPendingChanges ( ) [private]
```

#### 6.57.3.2 ClearStates()

```
void StateStack::ClearStates ( )
```

#### 6.57.3.3 createState()

```
State::Ptr StateStack::createState (
    States::ID stateID ) [private]
```

#### 6.57.3.4 Draw()

```
void StateStack::Draw ( )
```

#### 6.57.3.5 IsEmpty()

```
bool StateStack::IsEmpty ( ) const
```

#### 6.57.3.6 PopState()

```
void StateStack::PopState ( )
```

#### 6.57.3.7 PushState()

```
void StateStack::PushState (
    States::ID stateID )
```

### 6.57.3.8 RegisterState()

```
template<class T >
void StateStack::RegisterState (
    States::ID stateID )
```

### 6.57.3.9 Update()

```
void StateStack::Update (
    float dt )
```

## 6.57.4 Member Data Documentation

### 6.57.4.1 mContext

```
State::Context StateStack::mContext [private]
```

### 6.57.4.2 mFactories

```
std::map< States::ID, std::function< State::Ptr() > > StateStack::mFactories [private]
```

### 6.57.4.3 mPendingList

```
std::vector< PendingChange > StateStack::mPendingList [private]
```

### 6.57.4.4 mStack

```
std::vector< State::Ptr > StateStack::mStack [private]
```

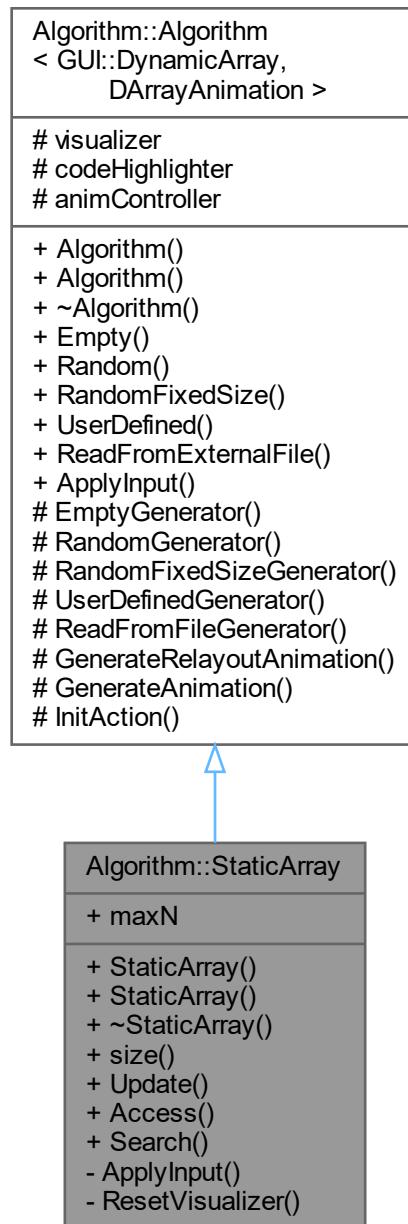
The documentation for this class was generated from the following files:

- src/[StateStack.hpp](#)
- src/[StateStack.cpp](#)

## 6.58 Algorithm::StaticArray Class Reference

```
#include <StaticArray.hpp>
```

Inheritance diagram for Algorithm::StaticArray:



### Public Member Functions

- [StaticArray \(\)](#)

- `StaticArray (GUI::CodeHighlighter::Ptr codeHighlighter, DArrayAnimationController::Ptr animController, FontHolder *fonts)`
- `~StaticArray ()`
- `std::size_t size () const`
- `void Update (int index, int value)`
- `void Access (int index)`
- `void Search (int value)`
- `virtual void Empty ()`
- `virtual void Random ()`
- `virtual void RandomFixedSize (int N)`
- `virtual void UserDefined (std::string input)`
- `virtual void ReadFromExternalFile (std::string path)`

## Static Public Attributes

- `static constexpr int maxN = 16`

## Protected Member Functions

- `std::vector< int > EmptyGenerator ()`
- `std::vector< int > RandomGenerator ()`
- `std::vector< int > RandomFixedSizeGenerator (int nSize)`
- `std::vector< int > UserDefinedGenerator (std::string input)`
- `std::vector< int > ReadFromFileGenerator (std::string inputFile)`
- `virtual void GenerateRelayoutAnimation (Vector2 newPosition)`
- `virtual DArrayAnimation GenerateAnimation (float duration, int highlightLine, std::string actionDescription)`
- `virtual void InitAction (std::vector< std::string > code)`

## Protected Attributes

- `GUI::DynamicArray visualizer`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `Animation::AnimationController< DArrayAnimation >::Ptr animController`

## Private Member Functions

- `void ApplyInput (std::vector< int > input, std::size_t nMaxSize)`
- `void ResetVisualizer ()`

### 6.58.1 Constructor & Destructor Documentation

#### 6.58.1.1 StaticArray() [1/2]

`Algorithm::StaticArray::StaticArray ( )`

### 6.58.1.2 StaticArray() [2/2]

```
Algorithm::StaticArray::StaticArray (
    GUI::CodeHighlighter::Ptr codeHighlighter,
    DArrayAnimationController::Ptr animController,
    FontHolder * fonts )
```

### 6.58.1.3 ~StaticArray()

```
Algorithm::StaticArray::~StaticArray ( )
```

## 6.58.2 Member Function Documentation

### 6.58.2.1 Access()

```
void Algorithm::StaticArray::Access (
    int index )
```

### 6.58.2.2 ApplyInput()

```
void Algorithm::StaticArray::ApplyInput (
    std::vector< int > input,
    std::size_t nMaxSize ) [private], [virtual]
```

Reimplemented from [Algorithm::Algorithm< GUI::DynamicArray, DArrayAnimation >](#).

### 6.58.2.3 Empty()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::Empty [virtual], [inherited]
```

### 6.58.2.4 EmptyGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::EmptyGenerator
[protected], [inherited]
```

### 6.58.2.5 GenerateAnimation()

```
DArrayAnimation Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::GenerateAnimation
(
    float duration,
    int highlightLine,
    std::string actionDescription ) [protected], [virtual], [inherited]
```

### 6.58.2.6 GenerateRelayoutAnimation()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::GenerateRelayoutAnimation (
    Vector2 newPosition ) [inline], [protected], [virtual], [inherited]
```

### 6.58.2.7 InitAction()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::InitAction (
    std::vector< std::string > code ) [protected], [virtual], [inherited]
```

### 6.58.2.8 Random()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::Random [virtual], [inherited]
```

### 6.58.2.9 RandomFixedSize()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::RandomFixedSize (
    int N ) [virtual], [inherited]
```

### 6.58.2.10 RandomFixedSizeGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::RandomFixedSizeGenerator (
    int nSize ) [protected], [inherited]
```

**6.58.2.11 RandomGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::Random←
Generator [protected], [inherited]
```

**6.58.2.12 ReadFromExternalFile()**

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::ReadFromExternalFile (
    std::string path ) [virtual], [inherited]
```

**6.58.2.13 ReadFromFileGenerator()**

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::ReadFrom←
FileGenerator (
    std::string inputFile ) [protected], [inherited]
```

**6.58.2.14 ResetVisualizer()**

```
void Algorithm::StaticArray::ResetVisualizer ( ) [private]
```

**6.58.2.15 Search()**

```
void Algorithm::StaticArray::Search (
    int value )
```

**6.58.2.16 size()**

```
std::size_t Algorithm::StaticArray::size ( ) const
```

**6.58.2.17 Update()**

```
void Algorithm::StaticArray::Update (
    int index,
    int value )
```

### 6.58.2.18 UserDefined()

```
void Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::UserDefined (
    std::string input ) [virtual], [inherited]
```

### 6.58.2.19 UserDefinedGenerator()

```
std::vector< int > Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::UserDefined←
Generator (
    std::string input ) [protected], [inherited]
```

## 6.58.3 Member Data Documentation

### 6.58.3.1 animController

```
Animation::AnimationController<DArrayAnimation >::Ptr Algorithm::Algorithm< GUI::DynamicArray ,
, DArrayAnimation >::animController [protected], [inherited]
```

### 6.58.3.2 codeHighlighter

```
GUI::CodeHighlighter::Ptr Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::code←
Highlighter [protected], [inherited]
```

### 6.58.3.3 maxN

```
constexpr int Algorithm::StaticArray::maxN = 16 [static], [constexpr]
```

### 6.58.3.4 visualizer

```
GUI::DynamicArray Algorithm::Algorithm< GUI::DynamicArray , DArrayAnimation >::visualizer
[protected], [inherited]
```

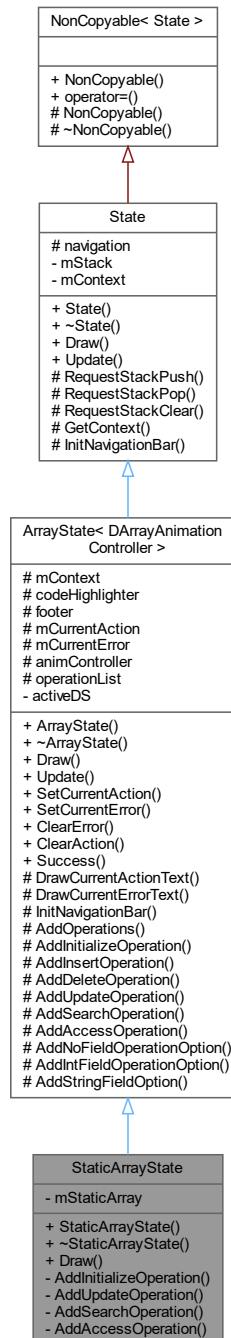
The documentation for this class was generated from the following files:

- src/Algorithms/Array/[StaticArray.hpp](#)
- src/Algorithms/Array/[StaticArray.cpp](#)

## 6.59 StaticArrayState Class Reference

```
#include <StaticArrayState.hpp>
```

Inheritance diagram for StaticArrayState:



### Public Types

- `typedef std::unique_ptr< State > Ptr`

## Public Member Functions

- `StaticArrayState (StateStack &stack, Context context)`
- `~StaticArrayState ()`
- `void Draw ()`
- `virtual bool Update (float dt)`
- `virtual void SetCurrentAction (std::string action)`
- `virtual void SetCurrentError (std::string error)`
- `virtual void ClearError ()`
- `virtual void ClearAction ()`
- `virtual void Success ()`

## Protected Member Functions

- `virtual void DrawCurrentActionText ()`
- `virtual void DrawCurrentErrorText ()`
- `void InitNavigationBar ()`
- `virtual void AddOperations ()`
- `virtual void AddInsertOperation ()`
- `virtual void AddDeleteOperation ()`
- `virtual void AddNoFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::function< void() > action)`
- `virtual void AddIntFieldOperationOption (GUI::OperationContainer::Ptr container, std::string title, std::vector< IntegerInput > fields, std::function< void(std::map< std::string, std::string >) > action)`
- `virtual void AddStringFieldOption (GUI::OperationContainer::Ptr container, std::string title, std::string label, std::function< void(std::map< std::string, std::string >) > action)`
- `void RequestStackPush (States::ID stateID)`
- `void RequestStackPop ()`
- `void RequestStackClear ()`
- `Context GetContext () const`

## Protected Attributes

- `Context mContext`
- `GUI::CodeHighlighter::Ptr codeHighlighter`
- `GUI::Footer< DArrayAnimationController > footer`
- `std::string mCurrentAction`
- `std::string mCurrentError`
- `T::Ptr animController`
- `GUI::OperationList operationList`
- `GUI::NavigationBar navigation`

## Private Member Functions

- `void AddInitializeOperation ()`
- `void AddUpdateOperation ()`
- `void AddSearchOperation ()`
- `void AddAccessOperation ()`

## Private Attributes

- Algorithm::StaticArray mStaticArray
- DataStructures::ID activeDS
- StateStack \* mStack

### 6.59.1 Member Typedef Documentation

#### 6.59.1.1 Ptr

```
typedef std::unique_ptr< State > State::Ptr [inherited]
```

### 6.59.2 Constructor & Destructor Documentation

#### 6.59.2.1 StaticArrayState()

```
StaticArrayState::StaticArrayState (
    StateStack & stack,
    Context context )
```

#### 6.59.2.2 ~StaticArrayState()

```
StaticArrayState::~StaticArrayState ()
```

### 6.59.3 Member Function Documentation

#### 6.59.3.1 AddAccessOperation()

```
void StaticArrayState::AddAccessOperation () [private], [virtual]
```

Reimplemented from [ArrayState< DArrayAnimationController >](#).

### 6.59.3.2 AddDeleteOperation()

```
void ArrayState< DArrayAnimationController >::AddDeleteOperation [protected], [virtual], [inherited]
```

Reimplemented in [DynamicArrayState](#).

### 6.59.3.3 AddInitializeOperation()

```
void StaticArrayState::AddInitializeOperation () [private], [virtual]
```

Reimplemented from [ArrayState< DArrayAnimationController >](#).

### 6.59.3.4 AddInsertOperation()

```
void ArrayState< DArrayAnimationController >::AddInsertOperation [protected], [virtual], [inherited]
```

Reimplemented in [DynamicArrayState](#).

### 6.59.3.5 AddIntFieldOperationOption()

```
void ArrayState< DArrayAnimationController >::AddIntFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::vector< IntegerInput > fields,
    std::function< void(std::map< std::string, std::string >) > action ) [protected], [virtual], [inherited]
```

### 6.59.3.6 AddNoFieldOperationOption()

```
void ArrayState< DArrayAnimationController >::AddNoFieldOperationOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::function< void() > action ) [protected], [virtual], [inherited]
```

### 6.59.3.7 AddOperations()

```
void ArrayState< DArrayAnimationController >::AddOperations [protected], [virtual], [inherited]
```

### 6.59.3.8 AddSearchOperation()

```
void StaticArrayState::AddSearchOperation ( ) [private], [virtual]
```

Reimplemented from [AppState< DArrayAnimationController >](#).

### 6.59.3.9 AddStringFieldOption()

```
void AppState< DArrayAnimationController >::AddStringFieldOption (
    GUI::OperationContainer::Ptr container,
    std::string title,
    std::string label,
    std::function< void(std::map< std::string, std::string >) > action ) [protected],
[virtual], [inherited]
```

### 6.59.3.10 AddUpdateOperation()

```
void StaticArrayState::AddUpdateOperation ( ) [private], [virtual]
```

Reimplemented from [AppState< DArrayAnimationController >](#).

### 6.59.3.11 ClearAction()

```
void AppState< DArrayAnimationController >::ClearAction [inline], [virtual], [inherited]
```

### 6.59.3.12 ClearError()

```
void AppState< DArrayAnimationController >::ClearError [inline], [virtual], [inherited]
```

### 6.59.3.13 Draw()

```
void StaticArrayState::Draw ( ) [virtual]
```

Implements [AppState< DArrayAnimationController >](#).

**6.59.3.14 DrawCurrentActionText()**

```
void ArrayState< DArrayAnimationController >::DrawCurrentActionText [inline], [protected],  
[virtual], [inherited]
```

**6.59.3.15 DrawCurrentErrorText()**

```
void ArrayState< DArrayAnimationController >::DrawCurrentErrorText [inline], [protected],  
[virtual], [inherited]
```

**6.59.3.16 GetContext()**

```
State::Context State::GetContext () const [protected], [inherited]
```

**6.59.3.17 InitNavigationBar()**

```
void ArrayState< DArrayAnimationController >::InitNavigationBar [protected], [inherited]
```

**6.59.3.18 RequestStackClear()**

```
void State::RequestStackClear () [protected], [inherited]
```

**6.59.3.19 RequestStackPop()**

```
void State::RequestStackPop () [protected], [inherited]
```

**6.59.3.20 RequestStackPush()**

```
void State::RequestStackPush (  
    States::ID stateID ) [protected], [inherited]
```

### 6.59.3.21 SetCurrentAction()

```
void ArrayType< DArrayAnimationController >::SetCurrentAction (
    std::string action) [inline], [virtual], [inherited]
```

### 6.59.3.22 SetCurrentError()

```
void ArrayType< DArrayAnimationController >::SetCurrentError (
    std::string error) [inline], [virtual], [inherited]
```

### 6.59.3.23 Success()

```
void ArrayType< DArrayAnimationController >::Success [inline], [virtual], [inherited]
```

### 6.59.3.24 Update()

```
bool ArrayType< DArrayAnimationController >::Update (
    float dt) [virtual], [inherited]
```

Implements [State](#).

## 6.59.4 Member Data Documentation

### 6.59.4.1 activeDS

```
DataStructures::ID ArrayType< DArrayAnimationController >::activeDS [private], [inherited]
```

### 6.59.4.2 animController

```
T::Ptr ArrayType< DArrayAnimationController >::animController [protected], [inherited]
```

#### 6.59.4.3 codeHighlighter

```
GUI::CodeHighlighter::Ptr ArrayState< DArrayAnimationController >::codeHighlighter [protected],  
[inherited]
```

#### 6.59.4.4 footer

```
GUI::Footer< DArrayAnimationController > ArrayState< DArrayAnimationController >::footer  
[protected], [inherited]
```

#### 6.59.4.5 mContext

```
Context ArrayState< DArrayAnimationController >::mContext [protected], [inherited]
```

#### 6.59.4.6 mCurrentAction

```
std::string ArrayState< DArrayAnimationController >::mCurrentAction [protected], [inherited]
```

#### 6.59.4.7 mCurrentError

```
std::string ArrayState< DArrayAnimationController >::mCurrentError [protected], [inherited]
```

#### 6.59.4.8 mStack

```
StateStack* State::mStack [private], [inherited]
```

#### 6.59.4.9 mStaticArray

```
Algorithm::StaticArray StaticArrayState::mStaticArray [private]
```

#### 6.59.4.10 navigation

```
GUI::NavigationBar State::navigation [protected], [inherited]
```

#### 6.59.4.11 operationList

```
GUI::OperationList ArrayState< DArrayAnimationController >::operationList [protected], [inherited]
```

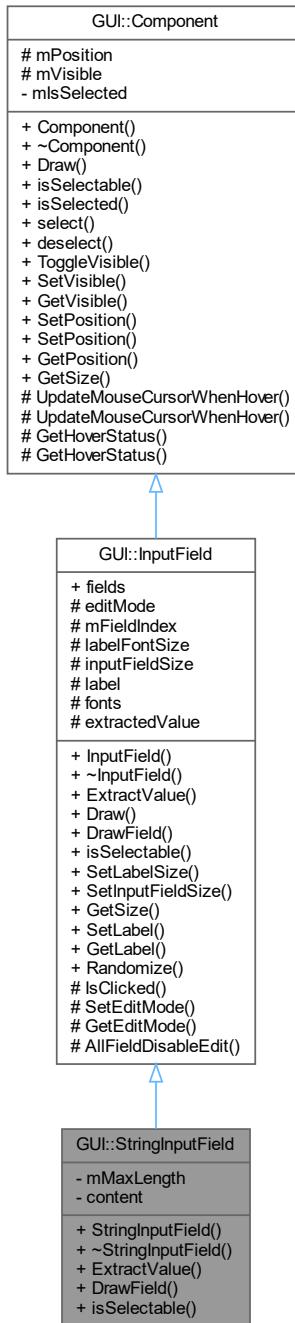
The documentation for this class was generated from the following files:

- src/States/Array/StaticArrayState.hpp
- src/States/Array/StaticArrayState.cpp

## 6.60 GUI::StringInputField Class Reference

```
#include <StringInputField.hpp>
```

Inheritance diagram for GUI::StringInputField:



## Public Types

- `typedef std::shared_ptr< StringInputField > Ptr`

## Public Member Functions

- `StringInputField (FontHolder *fonts)`

- `~StringInputField ()`
- `std::string ExtractValue ()`
- `void DrawField (Vector2 base=(Vector2){0, 0})`
- `bool IsSelectable () const`
- `virtual void Draw (Vector2 base=(Vector2){0, 0})`
- `virtual void SetLabelSize (float fontSize)`
- `virtual void SetInputFieldSize (Vector2 size)`
- `virtual Vector2 GetSize ()`
- `virtual void SetLabel (std::string labelContent)`
- `virtual std::string GetLabel () const`
- `virtual void Randomize ()`
- `bool IsSelected () const`
- `virtual void select ()`
- `virtual void deselect ()`
- `virtual void ToggleVisible ()`
- `virtual void SetVisible (bool visible)`
- `virtual bool GetVisible ()`
- `void SetPosition (float x, float y)`
- `void SetPosition (Vector2 position)`
- `Vector2 GetPosition ()`

## Static Public Attributes

- `static std::vector< bool > fields`

## Protected Member Functions

- `virtual bool IsClicked (Vector2 base=(Vector2){0, 0}) const`
- `virtual void SetEditMode (bool canEdit)`
- `virtual bool GetEditMode () const`
- `virtual void AllFieldDisableEdit ()`
- `virtual void UpdateMouseCursorWhenHover (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual void UpdateMouseCursorWhenHover (Rectangle bound, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (std::map< std::string, Rectangle > bounds, bool hover, bool noHover)`
- `virtual bool GetHoverStatus (Rectangle bound, bool hover, bool noHover)`

## Protected Attributes

- `bool editMode`
- `std::size_t mFieldIndex`
- `float labelFontSize`
- `Vector2 inputFieldSize`
- `std::string label`
- `FontHolder * fonts`
- `std::string extractedValue`
- `Vector2 mPosition`
- `bool mVisible`

## Private Attributes

- std::size\_t `mMaxLength`
- std::string `content`
- bool `isSelected`

### 6.60.1 Member Typedef Documentation

#### 6.60.1.1 Ptr

```
typedef std::shared_ptr< StringInputField > GUI::StringInputField::Ptr
```

### 6.60.2 Constructor & Destructor Documentation

#### 6.60.2.1 `StringInputField()`

```
GUI::StringInputField::StringInputField (  
    FontHolder * fonts )
```

#### 6.60.2.2 `~StringInputField()`

```
GUI::StringInputField::~StringInputField ( )
```

### 6.60.3 Member Function Documentation

#### 6.60.3.1 `AllFieldDisableEdit()`

```
void GUI::InputField::AllFieldDisableEdit ( ) [protected], [virtual], [inherited]
```

#### 6.60.3.2 `deselect()`

```
void GUI::Component::deselect ( ) [virtual], [inherited]
```

### 6.60.3.3 Draw()

```
void GUI::InputField::Draw (
    Vector2 base = (Vector2){0, 0} ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

### 6.60.3.4 DrawField()

```
void GUI::StringInputField::DrawField (
    Vector2 base = (Vector2){0, 0} ) [virtual]
```

Implements [GUI::InputField](#).

### 6.60.3.5 ExtractValue()

```
std::string GUI::StringInputField::ExtractValue () [virtual]
```

Implements [GUI::InputField](#).

### 6.60.3.6 GetEditMode()

```
bool GUI::InputField::GetEditMode () const [protected], [virtual], [inherited]
```

### 6.60.3.7 GetHoverStatus() [1/2]

```
bool GUI::Component::GetHoverStatus (
    Rectangle bound,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.60.3.8 GetHoverStatus() [2/2]

```
bool GUI::Component::GetHoverStatus (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover ) [protected], [virtual], [inherited]
```

### 6.60.3.9 GetLabel()

```
std::string GUI::InputField::GetLabel ( ) const [virtual], [inherited]
```

### 6.60.3.10GetPosition()

```
Vector2 GUI::Component::GetPosition ( ) [inherited]
```

### 6.60.3.11GetSize()

```
Vector2 GUI::InputField::GetSize ( ) [virtual], [inherited]
```

Reimplemented from [GUI::Component](#).

### 6.60.3.12GetVisible()

```
bool GUI::Component::GetVisible ( ) [virtual], [inherited]
```

### 6.60.3.13IsClicked()

```
bool GUI::InputField::IsClicked ( 
    Vector2 base = (Vector2){0, 0} ) const [protected], [virtual], [inherited]
```

### 6.60.3.14isSelectable()

```
bool GUI::StringInputField::isSelectable ( ) const [virtual]
```

Reimplemented from [GUI::InputField](#).

### 6.60.3.15isSelected()

```
bool GUI::Component::isSelected ( ) const [inherited]
```

### 6.60.3.16 Randomize()

```
void GUI::InputField::Randomize ( ) [virtual], [inherited]
```

Reimplemented in [GUI::IntegerInputField](#).

### 6.60.3.17 select()

```
void GUI::Component::select ( ) [virtual], [inherited]
```

### 6.60.3.18 SetEditMode()

```
void GUI::InputField::SetEditMode (
    bool canEdit ) [protected], [virtual], [inherited]
```

### 6.60.3.19 SetInputFieldSize()

```
void GUI::InputField::SetInputFieldSize (
    Vector2 size ) [virtual], [inherited]
```

### 6.60.3.20 SetLabel()

```
void GUI::InputField::SetLabel (
    std::string labelContent ) [virtual], [inherited]
```

### 6.60.3.21 SetLabelSize()

```
void GUI::InputField::SetLabelSize (
    float fontSize ) [virtual], [inherited]
```

### 6.60.3.22 SetPosition() [1/2]

```
void GUI::Component::SetPosition (
    float x,
    float y ) [inherited]
```

### 6.60.3.23 SetPosition() [2/2]

```
void GUI::Component::SetPosition (
    Vector2 position )  [inherited]
```

### 6.60.3.24 SetVisible()

```
void GUI::Component::SetVisible (
    bool visible )  [virtual], [inherited]
```

Reimplemented in [GUI::OperationContainer](#), and [GUI::OptionInputField](#).

### 6.60.3.25 ToggleVisible()

```
void GUI::Component::ToggleVisible ( )  [virtual], [inherited]
```

### 6.60.3.26 UpdateMouseCursorWhenHover() [1/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    Rectangle bound,
    bool hover,
    bool noHover )  [protected], [virtual], [inherited]
```

### 6.60.3.27 UpdateMouseCursorWhenHover() [2/2]

```
void GUI::Component::UpdateMouseCursorWhenHover (
    std::map< std::string, Rectangle > bounds,
    bool hover,
    bool noHover )  [protected], [virtual], [inherited]
```

## 6.60.4 Member Data Documentation

### 6.60.4.1 content

```
std::string GUI::StringInputField::content  [private]
```

#### 6.60.4.2 editMode

```
bool GUI::InputField::editMode [protected], [inherited]
```

#### 6.60.4.3 extractedValue

```
std::string GUI::InputField::extractedValue [protected], [inherited]
```

#### 6.60.4.4 fields

```
std::vector< bool > GUI::InputField::fields [static], [inherited]
```

#### 6.60.4.5 fonts

```
FontHolder* GUI::InputField::fonts [protected], [inherited]
```

#### 6.60.4.6 inputFontSize

```
Vector2 GUI::InputField::inputFontSize [protected], [inherited]
```

#### 6.60.4.7 label

```
std::string GUI::InputField::label [protected], [inherited]
```

#### 6.60.4.8 labelFontSize

```
float GUI::InputField::labelFontSize [protected], [inherited]
```

#### 6.60.4.9 mFieldIndex

```
std::size_t GUI::InputField::mFieldIndex [protected], [inherited]
```

#### 6.60.4.10 mIsSelected

```
bool GUI::Component::mIsSelected [private], [inherited]
```

#### 6.60.4.11 mMaxLength

```
std::size_t GUI::StringInputField::mMaxLength [private]
```

#### 6.60.4.12 mPosition

```
Vector2 GUI::Component::mPosition [protected], [inherited]
```

#### 6.60.4.13 mVisible

```
bool GUI::Component::mVisible [protected], [inherited]
```

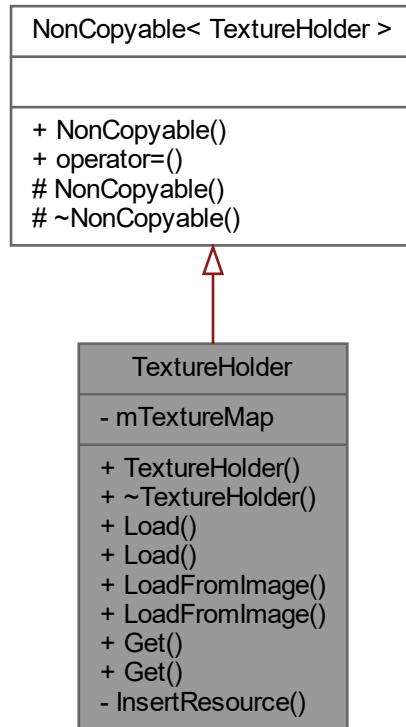
The documentation for this class was generated from the following files:

- src/Components/Common/[StringInputField.hpp](#)
- src/Components/Common/[StringInputField.cpp](#)

## 6.61 TextureHolder Class Reference

```
#include <TextureHolder.hpp>
```

Inheritance diagram for TextureHolder:



## Public Member Functions

- `TextureHolder ()`
- `~TextureHolder ()`
- `void Load (Textures::ID id, const std::string &filename)`
- `void Load (Textures::ID id, const std::string &filename, int width, int height, bool cropCenter=false)`
- `void LoadFromImage (Textures::ID id, Image &image)`
- `void LoadFromImage (Textures::ID id, Image &image, int width, int height)`
- `Texture & Get (Textures::ID id)`
- `const Texture & Get (Textures::ID id) const`

## Private Member Functions

- `void InsertResource (Textures::ID id, std::unique_ptr< Texture > texture)`

## Private Attributes

- `std::map< Textures::ID, std::unique_ptr< Texture > > mTextureMap`

## 6.61.1 Constructor & Destructor Documentation

### 6.61.1.1 TextureHolder()

```
TextureHolder::TextureHolder ( )
```

### 6.61.1.2 ~TextureHolder()

```
TextureHolder::~TextureHolder ( )
```

## 6.61.2 Member Function Documentation

### 6.61.2.1 Get() [1/2]

```
Texture & TextureHolder::Get (
    Textures::ID id )
```

### 6.61.2.2 Get() [2/2]

```
const Texture & TextureHolder::Get (
    Textures::ID id ) const
```

### 6.61.2.3 InsertResource()

```
void TextureHolder::InsertResource (
    Textures::ID id,
    std::unique_ptr< Texture > texture ) [private]
```

### 6.61.2.4 Load() [1/2]

```
void TextureHolder::Load (
    Textures::ID id,
    const std::string & filename )
```

### 6.61.2.5 Load() [2/2]

```
void TextureHolder::Load (
    Textures::ID id,
    const std::string & filename,
    int width,
    int height,
    bool cropCenter = false )
```

### 6.61.2.6 LoadFromImage() [1/2]

```
void TextureHolder::LoadFromImage (
    Textures::ID id,
    Image & image )
```

### 6.61.2.7 LoadFromImage() [2/2]

```
void TextureHolder::LoadFromImage (
    Textures::ID id,
    Image & image,
    int width,
    int height )
```

## 6.61.3 Member Data Documentation

### 6.61.3.1 mTextureMap

```
std::map< Textures::ID, std::unique_ptr< Texture > > TextureHolder::mTextureMap [private]
```

The documentation for this class was generated from the following files:

- src/[TextureHolder.hpp](#)
- src/[TextureHolder.cpp](#)

## 6.62 GUI::NavigationBar::TitleInfo Struct Reference

### Public Attributes

- States::ID stateID
- std::string abbrTitle
- std::string titleName

## 6.62.1 Member Data Documentation

### 6.62.1.1 abbrTitle

```
std::string GUI::NavigationBar::TitleInfo::abbrTitle
```

### 6.62.1.2 stateID

```
States::ID GUI::NavigationBar::TitleInfo::stateID
```

### 6.62.1.3 titleName

```
std::string GUI::NavigationBar::TitleInfo::titleName
```

The documentation for this struct was generated from the following file:

- src/Components/Common/[NavigationBar.hpp](#)

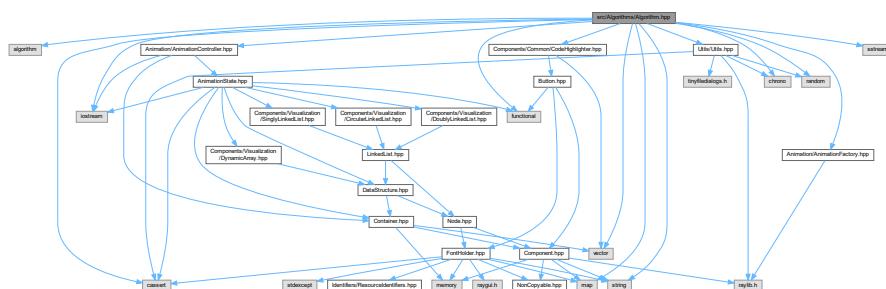
# Chapter 7

# File Documentation

## 7.1 src/Algorithms/Algorithm.hpp File Reference

```
#include <algorithm>
#include <cassert>
#include <chrono>
#include <functional>
#include <iostream>
#include <map>
#include <random>
#include <string>
#include <vector>
#include "Animation/AnimationController.hpp"
#include "Animation/AnimationFactory.hpp"
#include "Components/Common/CodeHighlighter.hpp"
#include "Utils/Utils.hpp"
#include <sstream>
```

Include dependency graph for Algorithm.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class Algorithm::Algorithm< GUIAlgorithm, AnimationState >

## Namespaces

- namespace Algorithm

## 7.2 Algorithm.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef CORE_ALGORITHMS_LINKEDLIST_ALGORITHM_HPP
00002 #define CORE_ALGORITHMS_LINKEDLIST_ALGORITHM_HPP
00003
00004 #include <algorithm>
00005 #include <cassert>
00006 #include <chrono>
00007 #include <functional>
00008 #include <iostream>
00009 #include <map>
00010 #include <random>
00011 #include <string>
00012 #include <vector>
00013
00014 #include "Animation/AnimationController.hpp"
00015 #include "Animation/AnimationFactory.hpp"
00016 #include "Components/Common/CodeHighlighter.hpp"
00017 #include "Utils/Utils.hpp"
00018
00019 namespace Algorithm {
00020     template< typename GUIAlgorithm, typename AnimationState >
00021     class Algorithm {
00022         public:
00023             Algorithm(GUI::CodeHighlighter::Ptr codeHighlighter,
00024                     typename Animation::AnimationController< AnimationState >::Ptr
00025                         animController,
00026                         FontHolder* fonts);
00027             Algorithm();
00028             ~Algorithm();
00029
00030         public:
00031         protected:
00032             GUIAlgorithm visualizer;
00033             GUI::CodeHighlighter::Ptr codeHighlighter;
00034             typename Animation::AnimationController< AnimationState >::Ptr
00035                 animController;
00036
00037         public:
00038             virtual void Empty();
00039             virtual void Random();
00040             virtual void RandomFixedSize(int N);
00041             virtual void UserDefined(std::string input);
00042             virtual void ReadFromExternalFile(std::string path);
00043             virtual void ApplyInput(std::vector< int > input,
00044                                     std::size_t nMaxSize = 10);
00045
00046         protected:
00047             std::vector< int > EmptyGenerator();
00048             std::vector< int > RandomGenerator();
00049
00050             std::vector< int > RandomFixedSizeGenerator(int nSize);
00051
00052             std::vector< int > UserDefinedGenerator(std::string input);
00053
00054             std::vector< int > ReadFromFileGenerator(std::string inputFile);
00055
00056         protected:
00057             virtual void GenerateRelayoutAnimation(Vector2 newPosition);
00058             virtual AnimationState GenerateAnimation(float duration,
00059                                             int highlightLine,
00060                                             std::string actionDescription);
00061             virtual void InitAction(std::vector< std::string > code);
00062     };
00063 }; // namespace Algorithm
00064
00065 #include <sstream>

```

```
00066 template< typename GUIAlgorithm, typename AnimationState >
00067 inline Algorithm::Algorithm< GUIAlgorithm, AnimationState >::Algorithm(
00068     GUI::CodeHighlighter::Ptr codeHighlighter,
00069     typename Animation::AnimationController< AnimationState >::Ptr
00070         animController,
00071     FontHolder* fonts)
00072 : codeHighlighter(codeHighlighter), animController(animController),
00073     visualizer(GUIAlgorithm(fonts)) {
00074     // visualizer.SetPosition(0, 150);
00075 }
00076
00077
00078 template< typename GUIAlgorithm, typename AnimationState >
00079 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::Algorithm() {}
00080
00081 template< typename GUIAlgorithm, typename AnimationState >
00082 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::~Algorithm() {}
00083
00084 template< typename GUIAlgorithm, typename AnimationState >
00085 std::vector< int >
00086 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::EmptyGenerator() {
00087     return std::vector< int >();
00088 }
00089
00090 template< typename GUIAlgorithm, typename AnimationState >
00091 std::vector< int >
00092 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::RandomGenerator() {
00093     int nSize = Utils::Rand(1, 10);
00094     // params["nSize"] = (char)(nSize + '0');
00095     return RandomFixedSizeGenerator(nSize);
00096 }
00097
00098 template< typename GUIAlgorithm, typename AnimationState >
00099 std::vector< int >
00100 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::RandomFixedSizeGenerator(
00101     int nSize) {
00102     std::vector< int > answer(nSize);
00103     for (int& v : answer) v = Utils::Rand(1, 99);
00104     return answer;
00105 }
00106
00107 template< typename GUIAlgorithm, typename AnimationState >
00108 std::vector< int >
00109 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::UserDefinedGenerator(
00110     std::string input) {
00111     bool canParse =
00112         (!input.empty() &&
00113             std::all_of(
00114                 input.begin(), input.end(),
00115                 [] (char c) { return ('0' <= c && c <= '9') || (c == ','); }) &&
00116             input.back() != ',');
00117     if (!canParse) return EmptyGenerator();
00118
00119     std::vector< int > answer;
00120
00121     std::istringstream f(input);
00122     std::string s;
00123     while (std::getline(f, s, ',')) {
00124         try {
00125             answer.push_back(atoi(s.c_str()));
00126         } catch (char* e) {
00127         }
00128     }
00129     return answer;
00130 }
00131
00132 template< typename GUIAlgorithm, typename AnimationState >
00133 std::vector< int >
00134 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::ReadFromFileGenerator(
00135     std::string inputFile) {
00136     // will be implemented later
00137     return std::vector< int >();
00138 }
00139
00140 template< typename GUIAlgorithm, typename AnimationState >
00141 inline void
00142 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::GenerateRelayoutAnimation(
00143     Vector2 newPosition) {
00144     AnimationState animRelayout =
00145         GenerateAnimation(0.5, -1,
00146                             "Re-layout the Linked List for visualization "
00147                             "(not in the actual Linked "
00148                             "List).\nThe whole process is still O(1).");
00149     animRelayout.SetAnimation(
00150         [this, newPosition](GUIAlgorithm srcDS, float playingAt, Vector2 base) {
00151             Vector2 newPos = srcDS.GetPosition();
00152             newPos.x += (newPosition.x - newPos.x) * playingAt;
```

```

00153         newPos.y += (newPosition.y - newPos.y) * playingAt;
00154         srcDS.SetPosition(newPos);
00155         srcDS.Draw(base, playingAt);
00156     }
00157     return srcDS;
00158 }
00159 });
00160 animController->AddAnimation(animRelayout);
00161 }
00162
00163 template< typename GUIAlgorithm, typename AnimationState >
00164 AnimationState
00165 Algorithm::Algorithm< GUIAlgorithm, AnimationState >::GenerateAnimation(
00166     float duration, int highlightLine, std::string actionDescription) {
00167     AnimationState animation;
00168     animation.SetDuration(duration);
00169     animation.SetHighlightLine(highlightLine);
00170     animation.SetSourceDataStructure(visualizer);
00171     animation.setActionDescription(actionDescription);
00172     return animation;
00173 }
00174
00175 template< typename GUIAlgorithm, typename AnimationState >
00176 void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::InitAction(
00177     std::vector< std::string > code) {
00178     animController->Reset();
00179     // animController->Pause();
00180     animController->InteractionAllow();
00181     animController->Clear();
00182     codeHighlighter->AddCode(code);
00183     codeHighlighter->SetShowCode(true);
00184     codeHighlighter->SetShowAction(true);
00185 }
00186
00187 template< typename GUIAlgorithm, typename AnimationState >
00188 void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::Empty() {
00189     ApplyInput(EmptyGenerator());
00190 }
00191
00192 template< typename GUIAlgorithm, typename AnimationState >
00193 void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::Random() {
00194     ApplyInput(RandomGenerator());
00195 }
00196
00197 template< typename GUIAlgorithm, typename AnimationState >
00198 void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::RandomFixedSize(
00199     int N) {
00200     ApplyInput(RandomFixedSizeGenerator(N));
00201 }
00202
00203 template< typename GUIAlgorithm, typename AnimationState >
00204 void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::UserDefined(
00205     std::string input) {
00206     ApplyInput(UserDefinedGenerator(input));
00207 }
00208
00209 template< typename GUIAlgorithm, typename AnimationState >
00210 void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::ReadFromExternalFile(
00211     std::string path) {
00212     ApplyInput(ReadFromFileGenerator(path));
00213 }
00214
00215 template< typename GUIAlgorithm, typename AnimationState >
00216 void Algorithm::Algorithm< GUIAlgorithm, AnimationState >::ApplyInput(
00217     std::vector< int > input, std::size_t nMaxSize) {
00218     if (input.size() > nMaxSize) input.resize(nMaxSize);
00219     InitAction({});
00220
00221     codeHighlighter->SetShowCode(false);
00222     codeHighlighter->SetShowAction(false);
00223
00224     visualizer.Import(input);
00225
00226     AnimationState state;
00227     state.SetDuration(0.5);
00228     state.SetHighlightLine(-1);
00229     state.SetSourceDataStructure(visualizer);
00230     state.SetAnimation([this] (GUIAlgorithm srcDS, float playingAt,
00231                             Vector2 base) {
00232         auto& nodes = srcDS.GetList();
00233         for (GUI::Node& node : nodes) {
00234             node.SetRadius(AnimationFactory::ElasticOut(playingAt) * 20);
00235             node.SetValueFontSize(AnimationFactory::ElasticOut(playingAt) * 24);
00236             node.SetLabelFontSize(AnimationFactory::ElasticOut(playingAt) * 20);
00237         }
00238         srcDS.Draw(base, playingAt, true);
00239     }
00240 }
```

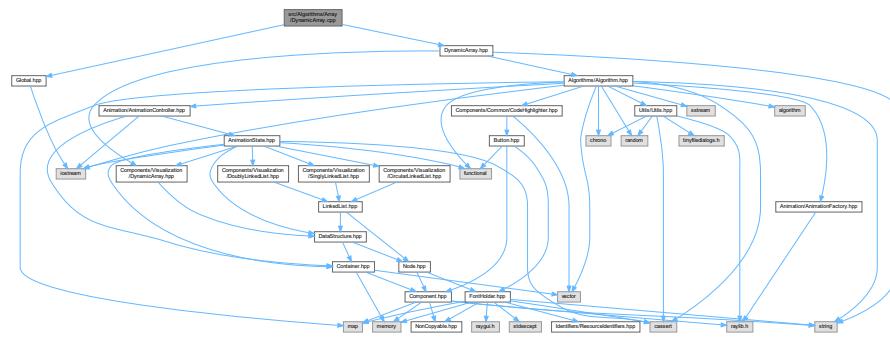
```

00240     });
00241     animController->AddAnimation(state);
00242     animController->Reset();
00243     animController->InteractionLock();
00244 }
00245
00246 #endif // CORE_ALGORITHMS_LINKEDLIST_ALGORITHM_HPP

```

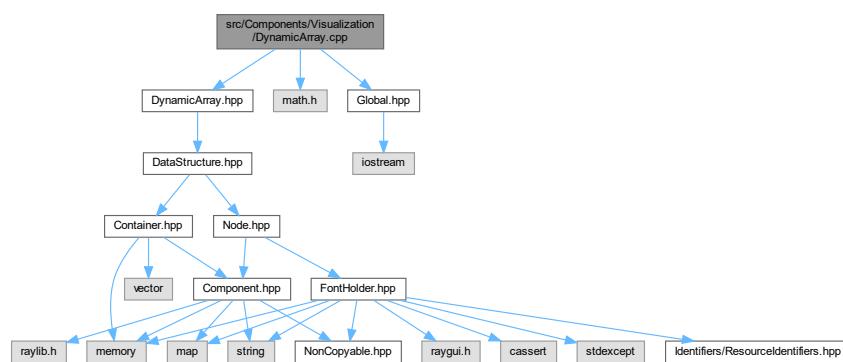
## 7.3 src/Algorithms/Array/DynamicArray.cpp File Reference

```
#include "DynamicArray.hpp"
#include "Global.hpp"
Include dependency graph for DynamicArray.cpp:
```



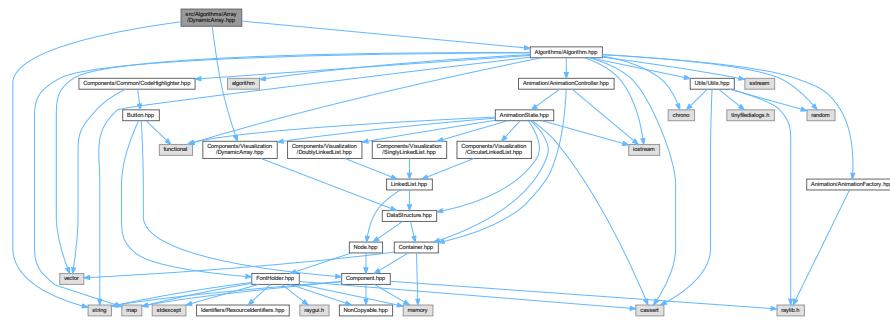
## 7.4 src/Components/Visualization/DynamicArray.cpp File Reference

```
#include "DynamicArray.hpp"
#include <math.h>
#include "Global.hpp"
Include dependency graph for DynamicArray.cpp:
```

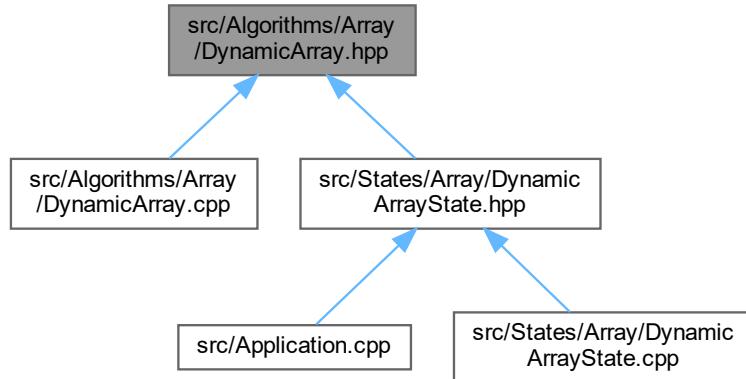


## 7.5 src/Algorithms/Array/DynamicArray.hpp File Reference

```
#include <string>
#include "Algorithms/Algorithm.hpp"
#include "Components/Visualization/DynamicArray.hpp"
Include dependency graph for DynamicArray.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class `Algorithm::DynamicArray`

## Namespaces

- namespace `Algorithm`

## 7.6 DynamicArray.hpp

[Go to the documentation of this file.](#)

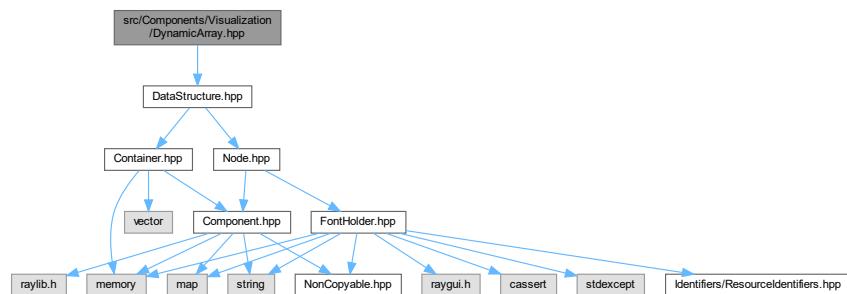
```

00001 #ifndef CORE_ALGORITHMS_ARRAY_DYNAMICARRAY_HPP
00002 #define CORE_ALGORITHMS_ARRAY_DYNAMICARRAY_HPP
00003
00004 #include <string>
00005
00006 // #include "BaseNode.hpp"
00007 #include "Algorithms/Algorithm.hpp"
00008 #include "Components/Visualization/DynamicArray.hpp"
00009
00010 namespace Algorithm {
00011     class DynamicArray
00012         : public Algorithm< GUI::DynamicArray, DArrayAnimation > {
00013     public:
00014         static constexpr int maxN = 16;
00015
00016     public:
00017         DynamicArray();
00018         DynamicArray(GUI::CodeHighlighter::Ptr codeHighlighter,
00019                         DArrayAnimationController::Ptr animController,
00020                         FontHolder* fonts);
00021         ~DynamicArray();
00022         std::size_t size() const;
00023
00024     public:
00025         void PushBack(int value);
00026
00027     public:
00028         void PopBack();
00029
00030     public:
00031         void Update(int index, int value);
00032
00033     public:
00034         void Access(int index);
00035
00036     public:
00037         void Search(int value);
00038
00039     private:
00040         void ResetVisualizer();
00041     };
00042 }; // namespace Algorithm
00043
00044 #endif // CORE_ALGORITHMS_ARRAY_DYNAMICARRAY_HPP

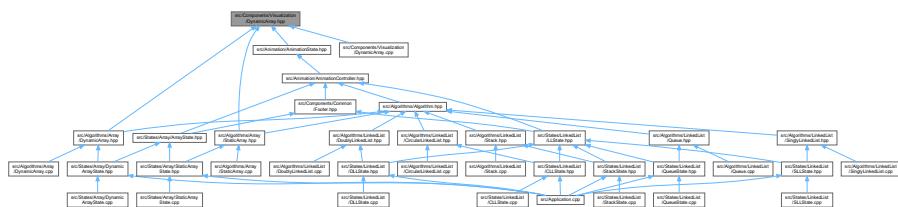
```

## 7.7 src/Components/Visualization/DynamicArray.hpp File Reference

#include "DataStructure.hpp"  
 Include dependency graph for DynamicArray.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::DynamicArray](#)

## Namespaces

- namespace [GUI](#)

## 7.8 DynamicArray.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef COMPONENTS_VISUALIZATION_DYNAMICARRAY_HPP
00002 #define COMPONENTS_VISUALIZATION_DYNAMICARRAY_HPP
00003
00004 #include "DataStructure.hpp"
00005
00006 namespace GUI {
00007     class DynamicArray : public GUI::DataStructure {
00008     public:
00009         DynamicArray();
00010         DynamicArray(FontHolder* fonts);
00011         bool isSelectable() const;
00012         ~DynamicArray();
00013
00014         void Draw(Vector2 base = (Vector2){0, 0}, float t = 1.0f,
00015                   bool init = false);
00016
00017     public:
00018         std::size_t GetLength() const;
00019         std::size_t GetCapacity() const;
00020         GUI::Node& operator[](std::size_t index);
00021         const GUI::Node& operator[](std::size_t index) const;
00022
00023     public:
00024         void SetShape(GUI::Node::Shape shape);
00025         GUI::Node::Shape GetShape() const;
00026
00027     public:
00028         void Reserve(std::size_t size);
00029         void Resize(std::size_t size);
00030         void Clear();
00031
00032     public:
00033         std::vector< GUI::Node >& GetList();
00034         GUI::Node GenerateNode(int value);
00035         void Import(std::vector< int > nodes);
00036         void InsertNode(std::size_t index, GUI::Node node,
00037                         bool rePosition = true);
00038         void Relayout();
00039         void DeleteNode(std::size_t index, bool rePosition = true);
00040
00041     private:
00042         Vector2 GetNodeDefaultPosition(std::size_t index);
00043
00044     public:
00045         std::size_t GetCapacityFromLength(std::size_t length);
00046

```

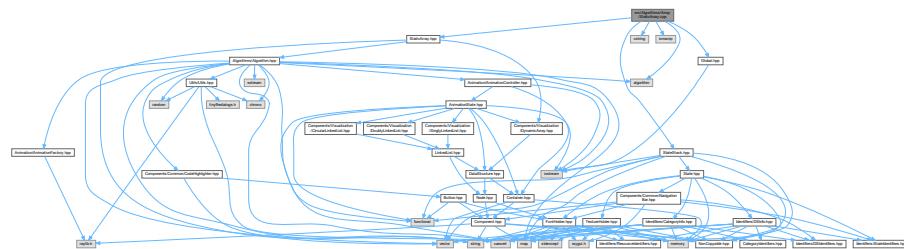
```

00047     public:
00048         static constexpr float mNodeDistance = 20;
00049
00050     private:
00051         FontHolder* fonts;
00052         std::vector< GUI::Node > list;
00053         GUI::Node::Shape mShape;
00054
00055     private:
00056         std::size_t capacity;
00057         std::size_t length;
00058     };
00059 }; // namespace GUI
00060
00061 #endif // COMPONENTS_VISUALIZATION_DYNAMICARRAY_HPP

```

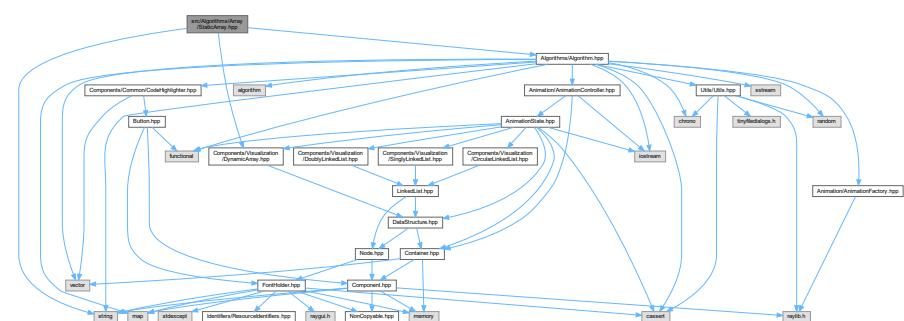
## 7.9 src/Algorithms/Array/StaticArray.cpp File Reference

```
#include "StaticArray.hpp"
#include <algorithm>
#include <cstring>
#include <iomanip>
#include "Global.hpp"
#include "StateStack.hpp"
Include dependency graph for StaticArray.cpp:
```

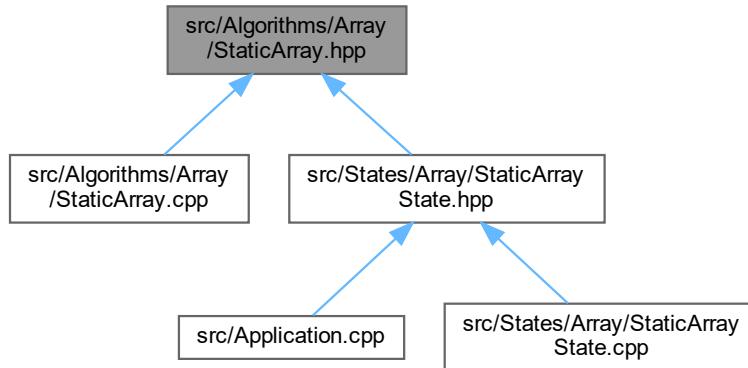


## 7.10 src/Algorithms/Array/StaticArray.hpp File Reference

```
#include <string>
#include "Algorithms/Algorithm.hpp"
#include "Components/Visualization/DynamicArray.hpp"
Include dependency graph for StaticArray.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Algorithm::StaticArray](#)

## Namespaces

- namespace [Algorithm](#)

## 7.11 StaticArray.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef CORE_ALGORITHMS_ARRAY_STATICARRAY_HPP
00002 #define CORE_ALGORITHMS_ARRAY_STATICARRAY_HPP
00003
00004 #include <string>
00005
00006 // #include "BaseNode.hpp"
00007 #include "Algorithms/Algorithm.hpp"
00008 #include "Components/Visualization/DynamicArray.hpp"
00009
00010 namespace Algorithm {
00011     class StaticArray : public Algorithm< GUI::DynamicArray, DArrayAnimation > {
00012     public:
00013         static constexpr int maxN = 16;
00014
00015     public:
00016         StaticArray();
00017         StaticArray(GUI::CodeHighlighter::Ptr codeHighlighter,
00018                     DArrayAnimationController::Ptr animController,
00019                     FontHolder* fonts);
00020         ~StaticArray();
00021         std::size_t size() const;
00022
00023     public:
00024         void Update(int index, int value);
00025
00026     public:
00027         void Access(int index);
00028
00029     public:
00030         void Search(int value);
00031
  
```

```

00032     private:
00033         void ApplyInput(std::vector< int > input, std::size_t nMaxSize);
00034
00035     private:
00036         void ResetVisualizer();
00037     };
00038 }; // namespace Algorithm
00039
00040 #endif // CORE_ALGORITHMS_ARRAY_STATICARRAY_HPP

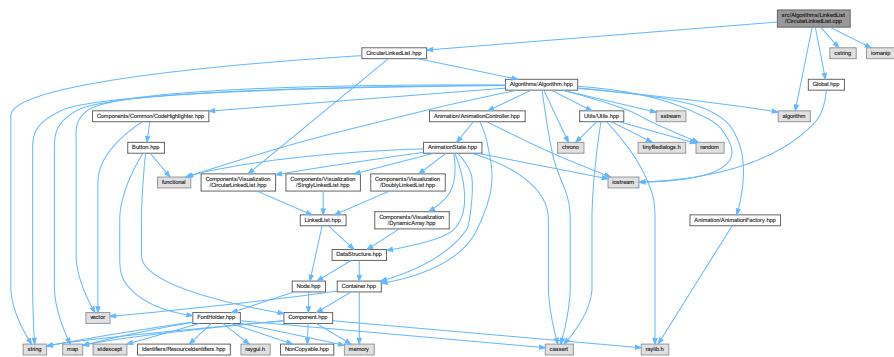
```

## 7.12 src/Algorithms/LinkedList/CircularLinkedList.cpp File Reference

```

#include "CircularLinkedList.hpp"
#include <algorithm>
#include <cstring>
#include <iomanip>
#include "Global.hpp"
Include dependency graph for CircularLinkedList.cpp:

```



## Typedefs

- using `ArrowType = GUI::LinkedList::ArrowType`

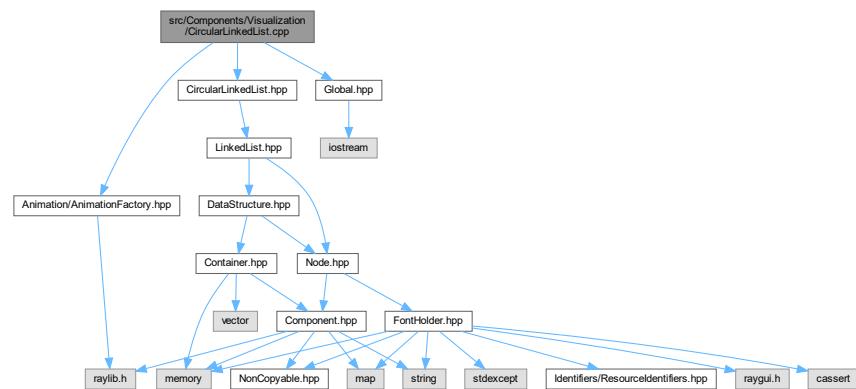
### 7.12.1 Typedef Documentation

#### 7.12.1.1 ArrowType

```
using ArrowType = GUI::LinkedList::ArrowType
```

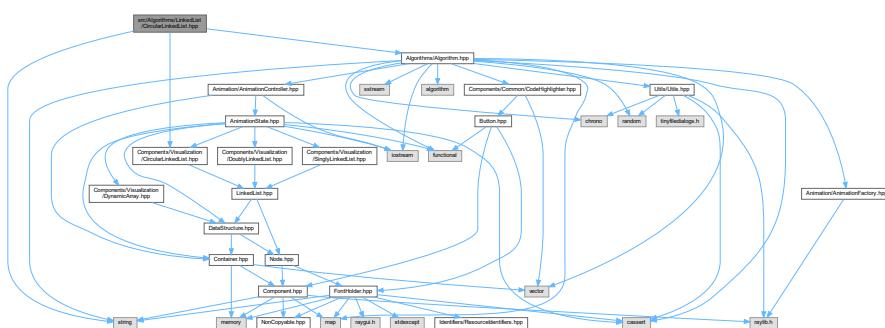
## 7.13 src/Components/Visualization/CircularLinkedList.cpp File Reference

```
#include "CircularLinkedList.hpp"
#include "Animation/AnimationFactory.hpp"
#include "Global.hpp"
Include dependency graph for CircularLinkedList.cpp:
```

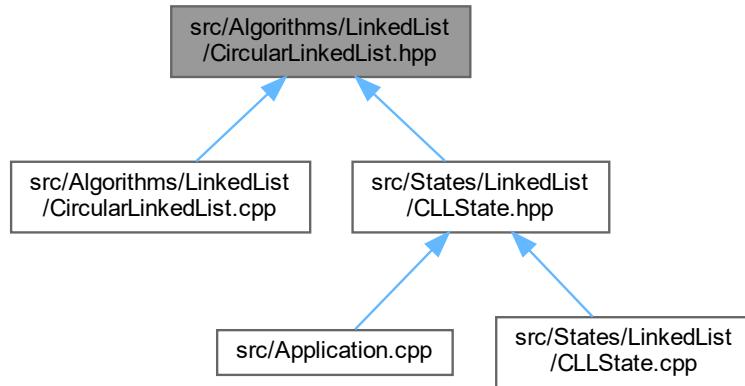


## 7.14 src/Algorithms/LinkedList/CircularLinkedList.hpp File Reference

```
#include <string>
#include "Algorithms/Algorithm.hpp"
#include "Components/Visualization/CircularLinkedList.hpp"
Include dependency graph for CircularLinkedList.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Algorithm::CircularLinkedList](#)

## Namespaces

- namespace [Algorithm](#)

## 7.15 CircularLinkedList.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef CORE_DATASTRUCTURES_LINKEDLIST_CIRCULARLINKEDLIST_HPP
00002 #define CORE_DATASTRUCTURES_LINKEDLIST_CIRCULARLINKEDLIST_HPP
00003
00004 #include <string>
00005
00006 // #include "BaseNode.hpp"
00007 #include "Algorithms/Algorithm.hpp"
00008 #include "Components/Visualization/CircularLinkedList.hpp"
00009
00010 namespace Algorithm {
00011     class CircularLinkedList
00012         : public Algorithm< GUI::CircularLinkedList, CLLAnimation > {
00013     public:
00014         static constexpr int maxN = 10;
00015
00016     public:
00017         CircularLinkedList();
00018         CircularLinkedList(GUI::CodeHighlighter::Ptr codeHighlighter,
00019                             CLLAnimationController::Ptr animController,
00020                             FontHolder* fonts);
00021         ~CircularLinkedList();
00022         std::size_t size() const;
00023
00024     public:
00025         void InsertHead(int value);
00026         void InsertAfterTail(int value);
00027         void InsertMiddle(int index, int value);
00028
00029     public:
00030         void DeleteHead();
  
```

```

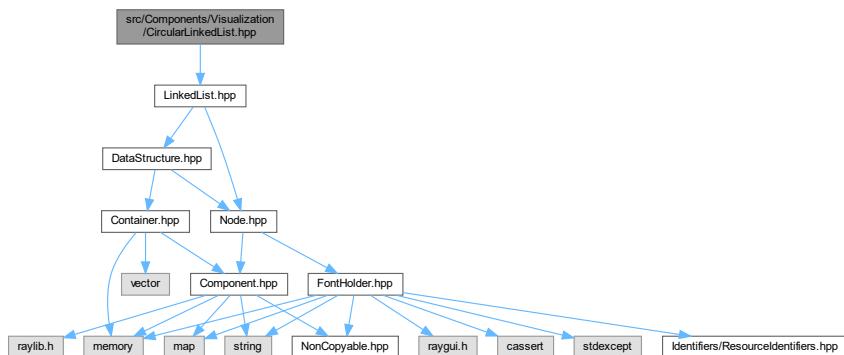
00031     void DeleteTail();
00032     void DeleteMiddle(int index);
00033
00034 public:
00035     void Update(int index, int value);
00036
00037 public:
00038     void Search(int value);
00039
00040 private:
00041     std::function< GUI::CircularLinkedList(GUI::CircularLinkedList, float,
00042                                         Vector2) >
00043         HighlightArrowFromCur(int index, bool drawVisualizer = true,
00044                                bool reverse = false);
00045     std::function< GUI::CircularLinkedList(GUI::CircularLinkedList, float,
00046                                         Vector2) >
00047         HighlightCircularArrow(bool drawVisualizer = true,
00048                                bool reverse = false);
00049
00050 private:
00051     void ResetVisualizer();
00052 };
00053 } // namespace Algorithm
00054
00055 #endif // CORE_DATASTRUCTURES_LINKEDLIST_CIRCULARLINKEDLIST_HPP

```

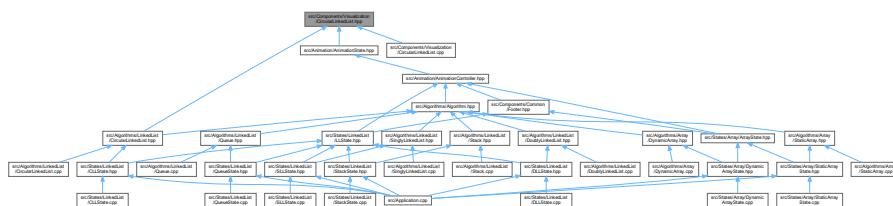
## 7.16 src/Components/Visualization/CircularLinkedList.hpp File Reference

#include "LinkedList.hpp"

Include dependency graph for CircularLinkedList.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::CircularLinkedList](#)

## Namespaces

- namespace GUI

## 7.17 CircularLinkedList.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef COMPONENTS_VISUALIZATION_CIRCULARLINKEDLIST_HPP
00002 #define COMPONENTS_VISUALIZATION_CIRCULARLINKEDLIST_HPP
00003
00004 #include "LinkedList.hpp"
00005
00006 namespace GUI {
00007     class CircularLinkedList : public GUI::LinkedList {
00008     private:
00009         std::vector< ArrowType > arrowState;
00010
00011     private:
00012         ArrowType circularArrowState;
00013         std::pair< std::size_t, std::size_t > mCircularEnds;
00014
00015     public:
00016         CircularLinkedList();
00017         CircularLinkedList(FontHolder* fonts);
00018         ~CircularLinkedList();
00019         bool isSelectable() const;
00020         void Draw(Vector2 base = (Vector2){0, 0}, float t = 1.0f,
00021                   bool init = false);
00022
00023     public:
00024         void Import(std::vector< int > nodes);
00025         void InsertNode(std::size_t index, GUI::Node node,
00026                          bool rePosition = true);
00027         void DeleteNode(std::size_t index, bool rePosition = true);
00028
00029     public:
00030         void SetCircularArrowType(ArrowType type);
00031         ArrowType GetCircularArrowType(std::size_t index);
00032         void SetCircularEnds(std::size_t from, std::size_t to);
00033         std::pair< std::size_t, std::size_t > GetCircularEnds();
00034
00035         void SetArrowType(std::size_t index, ArrowType type);
00036         ArrowType GetArrowType(std::size_t index);
00037         void ResetArrow();
00038
00039     private:
00040         void DrawArrow(Vector2 base, float t);
00041     };
00042 }; // namespace GUI
00043
00044 #endif // COMPONENTS_VISUALIZATION_CIRCULARLINKEDLIST_HPP

```

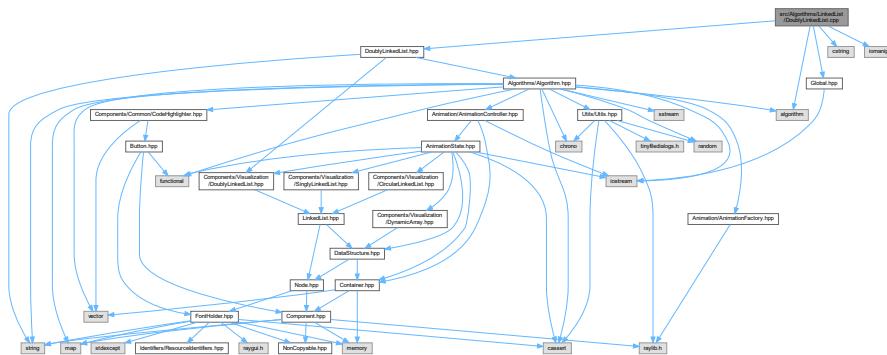
## 7.18 src/Algorithms/LinkedList/DoublyLinkedList.cpp File Reference

```

#include "DoublyLinkedList.hpp"
#include <algorithm>
#include <cstring>
#include <iomanip>
#include "Global.hpp"

```

Include dependency graph for DoublyLinkedList.cpp:



## TypeDefs

- using `ArrowType = GUI::LinkedList::ArrowType`

### 7.18.1 Typedef Documentation

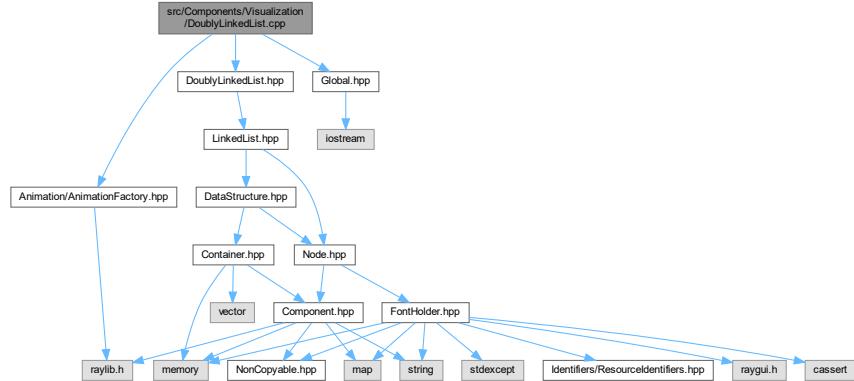
### 7.18.1.1 ArrowType

```
using ArrowType = GUI::LinkedList::ArrowType
```

## 7.19 src/Components/Visualization/DoublyLinkedList.cpp File Reference

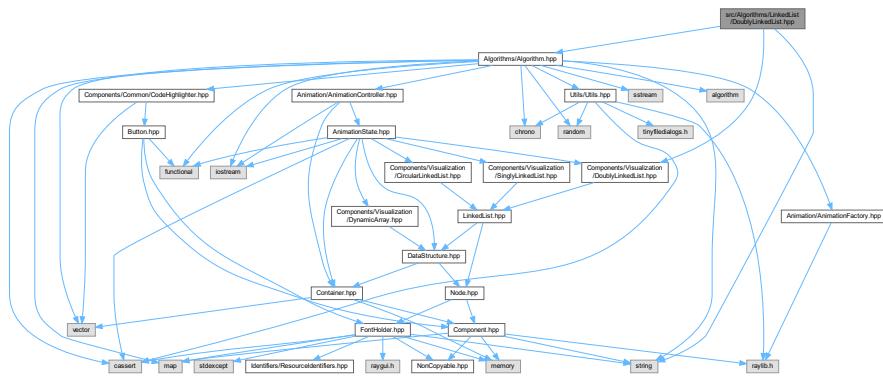
```
#include "DoublyLinkedList.hpp"
#include "Animation/AnimationFactory.hpp"
#include "Global.hpp"
Include dependency graph for DoublyLinkedList.cpp:
```

Include dependency graph for DoublyLinkedList.cpp:

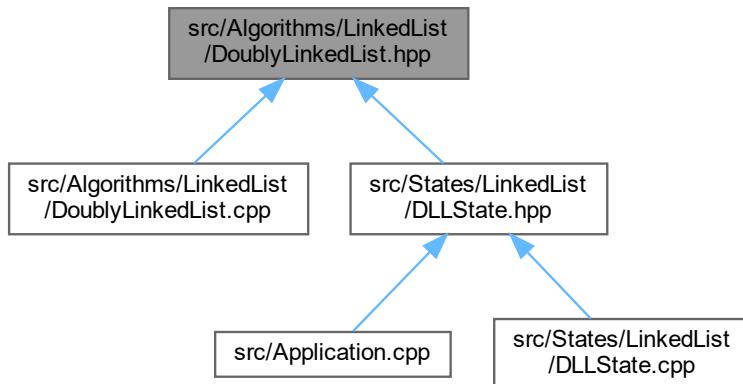


## 7.20 src/Algorithms/LinkedList/DoublyLinkedList.hpp File Reference

```
#include <string>
#include "Algorithms/Algorithm.hpp"
#include "Components/Visualization/DoublyLinkedList.hpp"
Include dependency graph for DoublyLinkedList.hpp:
```



This graph shows which files directly or indirectly include this file:



### Classes

- class [Algorithm::DoublyLinkedList](#)

### Namespaces

- namespace [Algorithm](#)

## 7.21 DoublyLinkedList.hpp

[Go to the documentation of this file.](#)

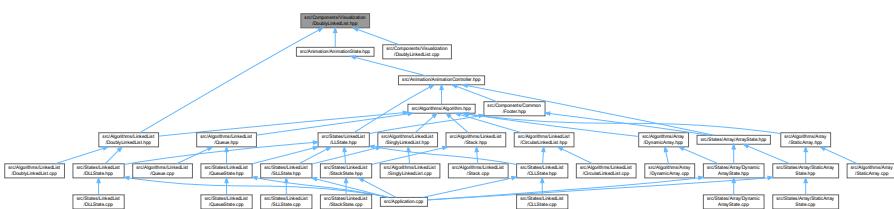
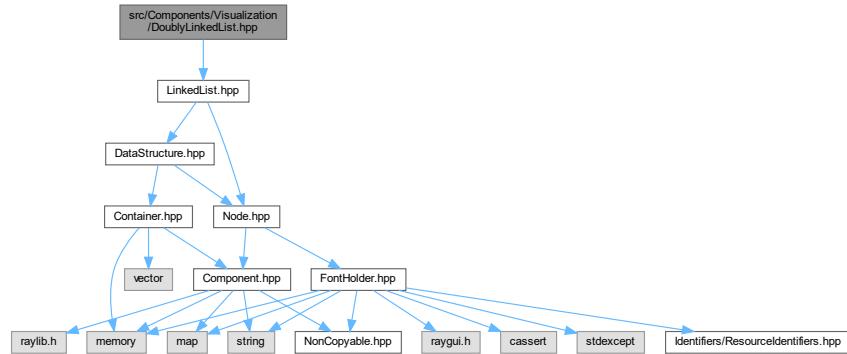
```

00001 #ifndef CORE_DATASTRUCTURES_LINKEDLIST_DOUBLYLINKEDLIST_HPP
00002 #define CORE_DATASTRUCTURES_LINKEDLIST_DOUBLYLINKEDLIST_HPP
00003
00004 #include <string>
00005
00006 // #include "BaseNode.hpp"
00007 #include "Algorithms/Algorithm.hpp"
00008 #include "Components/Visualization/DoublyLinkedList.hpp"
00009
00010 namespace Algorithm {
00011     class DoublyLinkedList
00012         : public Algorithm< GUI::DoublyLinkedList, DLLAnimation > {
00013     public:
00014         static constexpr int maxN = 10;
00015
00016     public:
00017         DoublyLinkedList();
00018         DoublyLinkedList(GUI::CodeHighlighter::Ptr codeHighlighter,
00019                         DLLAnimationController::Ptr animController,
00020                         FontHolder* fonts);
00021         ~DoublyLinkedList();
00022         std::size_t size() const;
00023
00024     public:
00025         void InsertHead(int value);
00026         void InsertAfterTail(int value);
00027         void InsertMiddle(int index, int value);
00028
00029     public:
00030         void DeleteHead();
00031         void DeleteTail();
00032         void DeleteMiddle(int index);
00033
00034     public:
00035         void Update(int index, int value);
00036
00037     public:
00038         void Search(int value);
00039
00040     private:
00041         std::function< GUI::DoublyLinkedList(GUI::DoublyLinkedList, float,
00042                                              Vector2) >
00043         HighlightArrowNext(int index, bool drawVisualizer = true,
00044                             bool reverse = false);
00045
00046         std::function< GUI::DoublyLinkedList(GUI::DoublyLinkedList, float,
00047                                              Vector2) >
00048         HighlightArrowPrev(int index, bool drawVisualizer = true,
00049                             bool reverse = false);
00050
00051         std::function< GUI::DoublyLinkedList(GUI::DoublyLinkedList, float,
00052                                              Vector2) >
00053         HighlightArrowBoth(int index, bool drawVisualizer = true,
00054                             bool reverse = false);
00055
00056     private:
00057         void ResetVisualizer();
00058     };
00059 }; // namespace Algorithm
00060
00061 #endif // CORE_DATASTRUCTURES_LINKEDLIST_DOUBLYLINKEDLIST_HPP

```

## 7.22 src/Components/Visualization/DoublyLinkedList.hpp File Reference

```
#include "LinkedList.hpp"
Include dependency graph for DoublyLinkedList.hpp:
```



## Classes

- class [GUI::DoublyLinkedList](#)

## Namespaces

- namespace [GUI](#)

## 7.23 DoublyLinkedList.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef COMPONENTS_VISUALIZATION_DOUBLYLINKEDLIST_HPP
00002 #define COMPONENTS_VISUALIZATION_DOUBLYLINKEDLIST_HPP
00003
00004 #include "LinkedList.hpp"
00005
00006 namespace GUI {
00007     class DoublyLinkedList : public GUI::LinkedList {
00008     public:
00009     private:
00010         std::vector< ArrowType > arrowStateNext;
00011         std::vector< ArrowType > arrowStatePrev;
  
```

```

00013     public:
00014     public:
00015         DoublyLinkedList();
00016         DoublyLinkedList(FontHolder* fonts);
00017         ~DoublyLinkedList();
00018         bool isSelectable() const;
00019         void Draw(Vector2 base = {Vector2}{0, 0}, float t = 1.0f,
00020                 bool init = false);
00021
00022     public:
00023         void Import(std::vector< int > nodes);
00024         void InsertNode(std::size_t index, GUI::Node node,
00025                         bool rePosition = true);
00026         void DeleteNode(std::size_t index, bool rePosition = true);
00027
00028     public:
00029         void SetArrowTypeNext(std::size_t index, ArrowType type);
00030         void SetArrowTypePrev(std::size_t index, ArrowType type);
00031         ArrowType GetArrowTypeNext(std::size_t index);
00032         ArrowType GetArrowTypePrev(std::size_t index);
00033         void ResetArrow();
00034
00035     private:
00036         void DrawArrow(Vector2 base, float t);
00037     };
00038 }; // namespace GUI
00039
00040 #endif // COMPONENTS_VISUALIZATION_DOUBLYLINKEDLIST_HPP

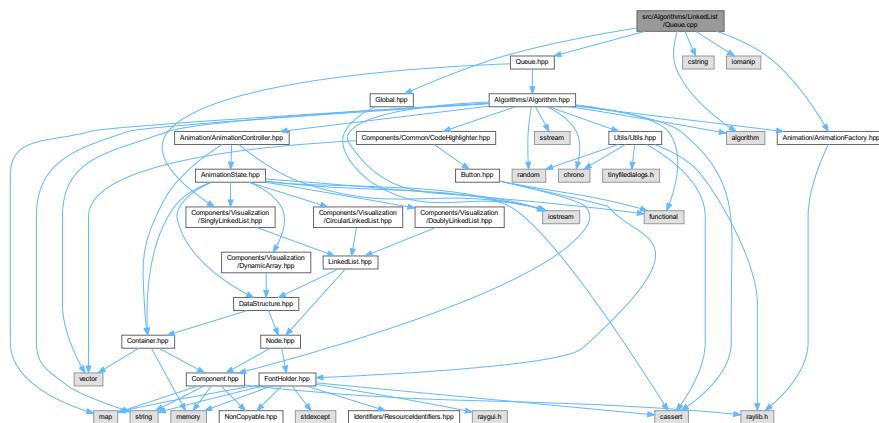
```

## 7.24 src/Algorithms/LinkedList/Queue.cpp File Reference

```

#include "Queue.hpp"
#include <algorithm>
#include <cstring>
#include <iomanip>
#include "Animation/AnimationFactory.hpp"
#include "Global.hpp"
Include dependency graph for Queue.cpp:

```



### Typedefs

- using `ArrowType` = `GUI::SinglyLinkedList::ArrowType`

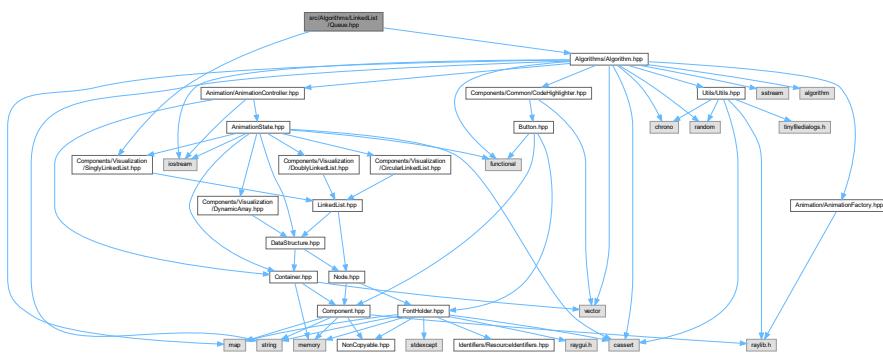
#### 7.24.1 Typedef Documentation

### 7.24.1.1 ArrowType

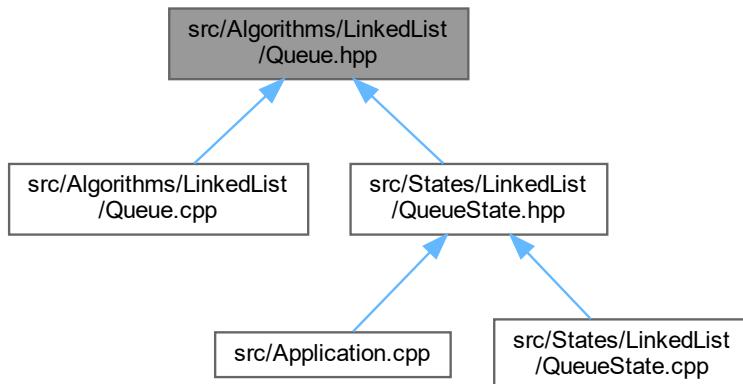
```
using ArrowType = GUI::SinglyLinkedList::ArrowType
```

## 7.25 src/Algorithms/LinkedList/Queue.hpp File Reference

```
#include "Algorithms/Algorithm.hpp"
#include "Components/Visualization/SinglyLinkedList.hpp"
Include dependency graph for Queue.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Algorithm::Queue](#)

## Namespaces

- namespace [Algorithm](#)

## 7.26 Queue.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef CORE_ALGORITHMS_LINKEDLIST_QUEUE_HPP
00002 #define CORE_ALGORITHMS_LINKEDLIST_QUEUE_HPP
00003
00004 #include "Algorithms/Algorithm.hpp"
00005 #include "Components/Visualization/SinglyLinkedList.hpp"
00006
00007 namespace Algorithm {
00008     class Queue : public Algorithm< GUI::SinglyLinkedList, SLLAnimation > {
00009     public:
00010         static constexpr int maxN = 10;
00011
00012     public:
00013         Queue(GUI::CodeHighlighter::Ptr _codeHighlighter,
00014               SLLAnimationController::Ptr animController, FontHolder* fonts);
00015         Queue();
00016         ~Queue();
00017         std::size_t size() const;
00018
00019     public:
00020         void EnqueueEmpty(int value);
00021         void Enqueue(int value);
00022
00023     public:
00024         void Dequeue();
00025
00026     public:
00027         void PeekFront();
00028         void PeekBack();
00029
00030     private:
00031         std::function< GUI::SinglyLinkedList(GUI::SinglyLinkedList, float,
00032                                              Vector2) >
00033             HighlightArrowFromCur(int index, bool drawVisualizer = true,
00034                                   bool reverse = false);
00035     };
00036 }; // namespace Algorithm
00037
00038 #endif // CORE_ALGORITHMS_LINKEDLIST_QUEUE_HPP

```

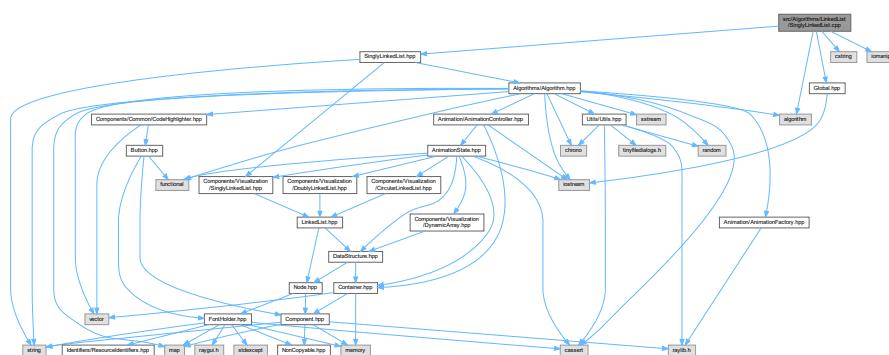
## 7.27 src/Algorithms/LinkedList/SinglyLinkedList.cpp File Reference

```

#include "SinglyLinkedList.hpp"
#include <algorithm>
#include <cstring>
#include <iomanip>
#include "Global.hpp"

```

Include dependency graph for SinglyLinkedList.cpp:



### Typedefs

- using ArrowType = GUI::LinkedList::ArrowType

## 7.27.1 Typedef Documentation

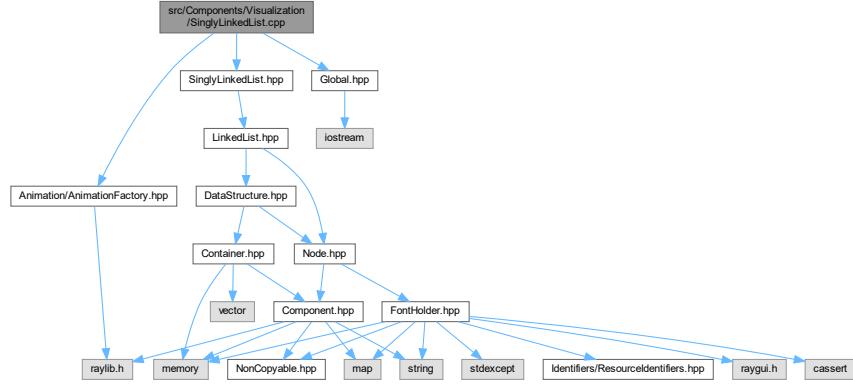
### 7.27.1.1 ArrowType

```
using ArrowType = GUI::LinkedList::ArrowType;
```

## 7.28 src/Components/Visualization/SinglyLinkedList.cpp File Reference

```
#include "SinglyLinkedList.hpp"
#include "Animation/AnimationFactory.hpp"
#include "Global.hpp"
Include dependency graph for SinglyLinkedList.cpp:
```

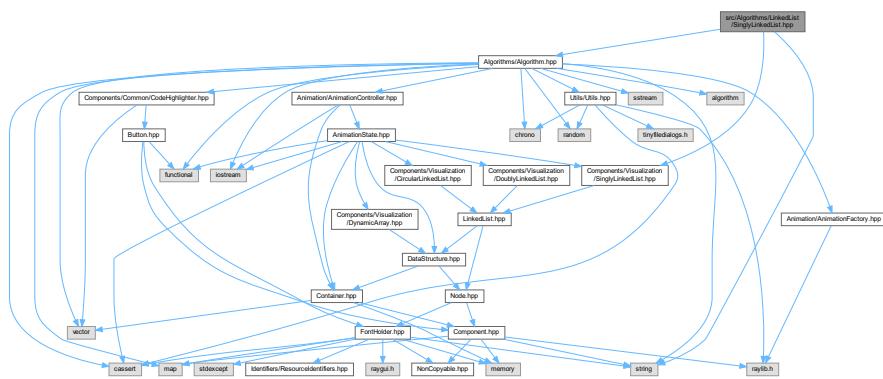
Include dependency graph for SinglyLinkedList.cpp:



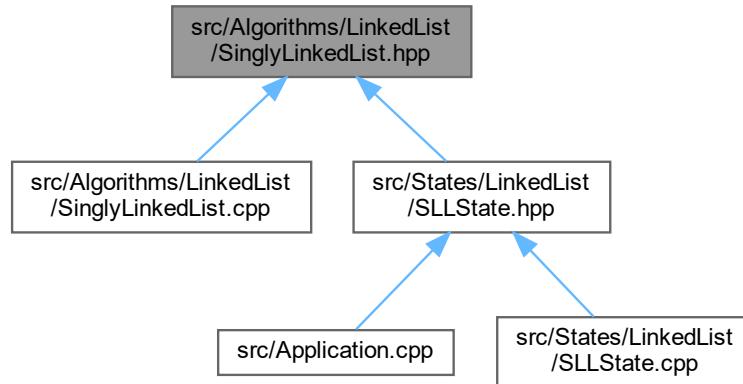
## 7.29 src/Algorithms/LinkedList/SinglyLinkedList.hpp File Reference

```
#include <string>
#include "Algorithms/Algorithm.hpp"
#include "Components/Visualization/SinglyLinkedList.hpp"
Include dependency graph for SinglyLinkedList.hpp:
```

Include dependency graph for SinglyLinkedList.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Algorithm::SinglyLinkedList](#)

## Namespaces

- namespace [Algorithm](#)

## 7.30 SinglyLinkedList.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef CORE_DATASTRUCTURES_LINKEDLIST_SINGLYLINKEDLIST_HPP
00002 #define CORE_DATASTRUCTURES_LINKEDLIST_SINGLYLINKEDLIST_HPP
00003
00004 #include <string>
00005
00006 // #include "BaseNode.hpp"
00007 #include "Algorithms/Algorithm.hpp"
00008 #include "Components/Visualization/SinglyLinkedList.hpp"
00009
00010 namespace Algorithm {
00011     class SinglyLinkedList
00012         : public Algorithm< GUI::SinglyLinkedList, SLLAnimation > {
00013     public:
00014         static constexpr int maxN = 10;
00015
00016     public:
00017         SinglyLinkedList();
00018         SinglyLinkedList(GUI::CodeHighlighter::Ptr codeHighlighter,
00019                         SLLAnimationController::Ptr animController,
00020                         FontHolder* fonts);
00021         ~SinglyLinkedList();
00022         std::size_t size() const;
00023
00024     public:
00025         void InsertHead(int value);
00026         void InsertAfterTail(int value);
00027         void InsertMiddle(int index, int value);
00028
00029     public:
00030         void DeleteHead();
  
```

```

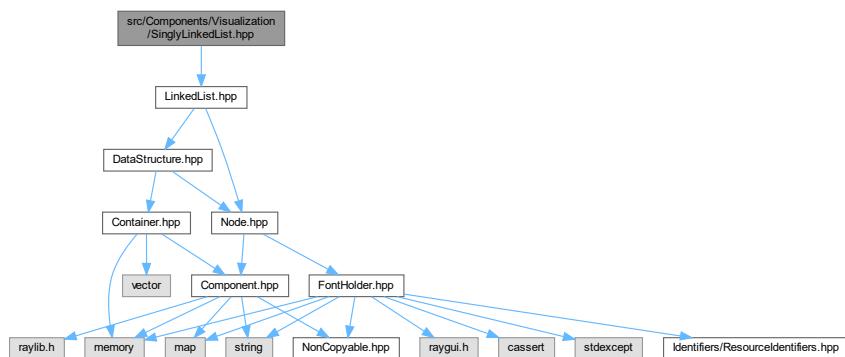
00031     void DeleteTail();
00032     void DeleteMiddle(int index);
00033
00034 public:
00035     void Update(int index, int value);
00036
00037 public:
00038     void Search(int value);
00039
00040 private:
00041     std::function< GUI::SinglyLinkedList(GUI::SinglyLinkedList, float,
00042                                         Vector2) >
00043     HighlightArrowFromCur(int index, bool drawVisualizer = true,
00044                           bool reverse = false);
00045
00046 private:
00047     void ResetVisualizer();
00048 };
00049 }; // namespace Algorithm
00050
00051 #endif // CORE_DATASTRUCTURES_LINKEDLIST_SINGLYLINKEDLIST_HPP

```

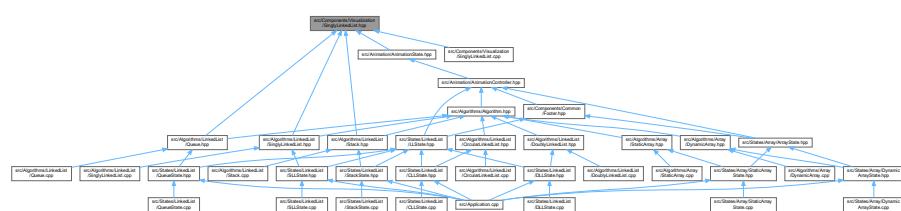
## 7.31 src/Components/Visualization/SinglyLinkedList.hpp File Reference

#include "LinkedList.hpp"

Include dependency graph for SinglyLinkedList.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::SinglyLinkedList](#)

## Namespaces

- namespace GUI

## 7.32 SinglyLinkedList.hpp

[Go to the documentation of this file.](#)

```

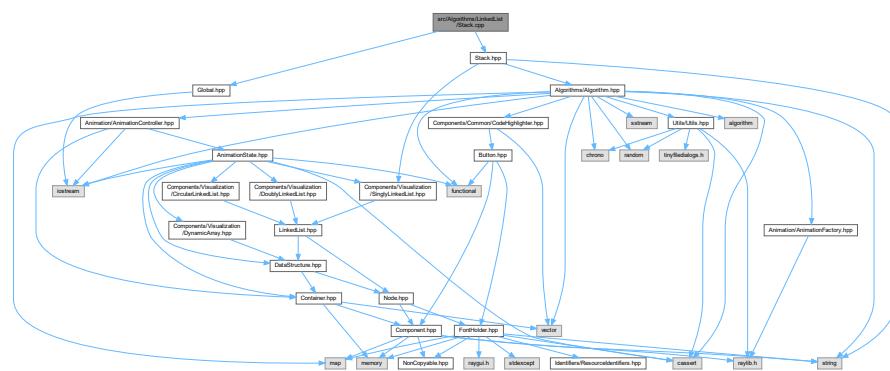
00001 #ifndef COMPONENTS_VISUALIZATION_SINGLYLINKEDLIST_HPP
00002 #define COMPONENTS_VISUALIZATION_SINGLYLINKEDLIST_HPP
00003
00004 #include "LinkedList.hpp"
00005
00006 namespace GUI {
00007     class SinglyLinkedList : public GUI::LinkedList {
00008     private:
00009         std::vector< ArrowType > arrowState;
00010
00011     public:
00012     public:
00013         SinglyLinkedList();
00014         SinglyLinkedList(FontHolder* fonts);
00015         ~SinglyLinkedList();
00016         bool isSelectable() const;
00017         void Draw(Vector2 base = {Vector2{0, 0}}, float t = 1.0f,
00018                 bool init = false);
00019
00020     public:
00021         void Import(std::vector< int > nodes);
00022         void InsertNode(std::size_t index, GUI::Node node,
00023                         bool rePosition = true);
00024         void DeleteNode(std::size_t index, bool rePosition = true);
00025
00026     public:
00027         void SetArrowType(std::size_t index, ArrowType type);
00028         ArrowType GetArrowType(std::size_t index);
00029         void ResetArrow();
00030
00031     private:
00032         void DrawArrow(Vector2 base, float t);
00033     };
00034 }; // namespace GUI
00035
00036 #endif // COMPONENTS_VISUALIZATION_SINGLYLINKEDLIST_HPP

```

## 7.33 src/Algorithms/LinkedList/Stack.cpp File Reference

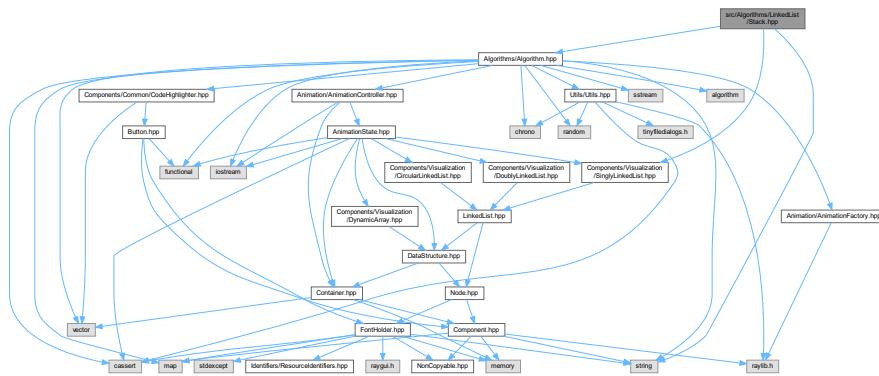
```
#include "Stack.hpp"
#include "Global.hpp"
```

Include dependency graph for Stack.cpp:

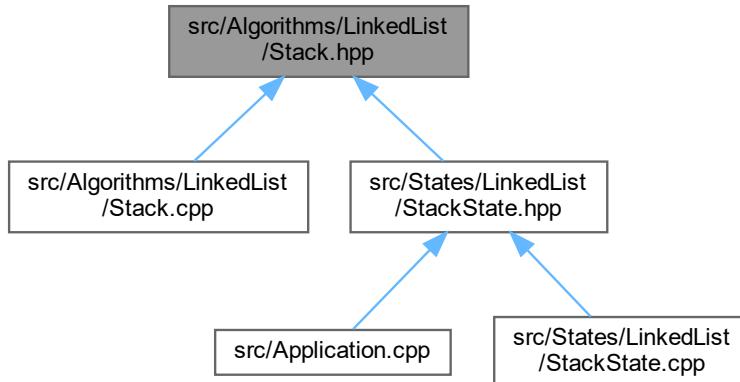


## 7.34 src/Algorithms/LinkedList/Stack.hpp File Reference

```
#include <string>
#include "Algorithms/Algorithm.hpp"
#include "Components/Visualization/SinglyLinkedList.hpp"
Include dependency graph for Stack.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Algorithm::Stack](#)

## Namespaces

- namespace [Algorithm](#)

## Typedefs

- using `ArrowType` = `GUI::SinglyLinkedList::ArrowType`
- using `Orientation` = `GUI::LinkedList::Orientation`

### 7.34.1 Typedef Documentation

#### 7.34.1.1 ArrowType

```
using ArrowType = GUI::SinglyLinkedList::ArrowType
```

#### 7.34.1.2 Orientation

```
using Orientation = GUI::LinkedList::Orientation
```

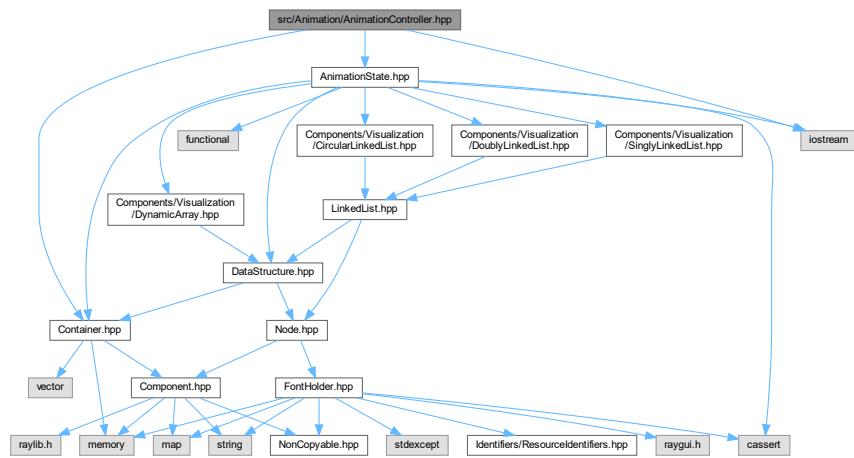
## 7.35 Stack.hpp

[Go to the documentation of this file.](#)

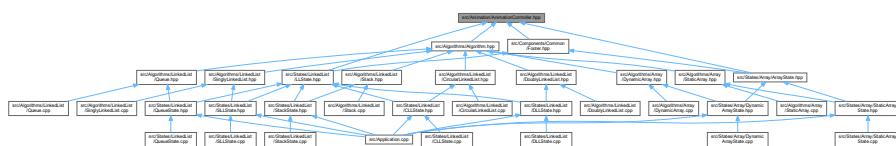
```
00001 #ifndef CORE_ALGORITHMS_LINKEDLIST_STACK_HPP
00002 #define CORE_ALGORITHMS_LINKEDLIST_STACK_HPP
00003
00004 #include <string>
00005
00006 // #include "BaseNode.hpp"
00007 #include "Algorithms/Algorithm.hpp"
00008 #include "Components/Visualization/SinglyLinkedList.hpp"
00009
00010 using ArrowType = GUI::SinglyLinkedList::ArrowType;
00011 using Orientation = GUI::LinkedList::Orientation;
00012
00013 namespace Algorithm {
00014     class Stack : public Algorithm<GUI::SinglyLinkedList, SLLAnimation> {
00015     public:
00016         static constexpr int maxN = 10;
00017         static constexpr Orientation mStackOrientation = Orientation::Vertical;
00018
00019     public:
00020         Stack(GUI::CodeHighlighter::Ptr codeHighlighter,
00021                 SLLAnimationController::Ptr animController, FontHolder* fonts);
00022         Stack();
00023         ~Stack();
00024         std::size_t size() const;
00025
00026     public:
00027         void Push(int value);
00028
00029     public:
00030         void Pop();
00031
00032     public:
00033         void Peek();
00034
00035     private:
00036         std::function<GUI::SinglyLinkedList(GUI::SinglyLinkedList, float,
00037                                         Vector2)>
00038         HighlightArrowFromCur(int index, bool drawVisualizer = true,
00039                               bool reverse = false);
00040     };
00041 }; // namespace Algorithm
00042
00043 #endif // CORE_ALGORITHMS_LINKEDLIST_STACK_HPP
```

## 7.36 src/Animation/AnimationController.hpp File Reference

```
#include "AnimationState.hpp"
#include "Container.hpp"
#include <iostream>
Include dependency graph for AnimationController.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Animation::AnimationController< T >](#)

## Namespaces

- namespace [Animation](#)

## Typedefs

- typedef [Animation::AnimationController< SLLAnimation > SLLAnimationController](#)
- typedef [Animation::AnimationController< DLLAnimation > DLLAnimationController](#)
- typedef [Animation::AnimationController< CLLAnimation > CLLAnimationController](#)
- typedef [Animation::AnimationController< DArrayAnimation > DArrayAnimationController](#)

### 7.36.1 Typedef Documentation

#### 7.36.1.1 CLLAnimationController

```
typedef Animation::AnimationController< CLLAnimation > CLLAnimationController
```

#### 7.36.1.2 DArrayAnimationController

```
typedef Animation::AnimationController< DArrayAnimation > DArrayAnimationController
```

#### 7.36.1.3 DLLAnimationController

```
typedef Animation::AnimationController< DLLAnimation > DLLAnimationController
```

#### 7.36.1.4 SLLAnimationController

```
typedef Animation::AnimationController< SLLAnimation > SLLAnimationController
```

## 7.37 AnimationController.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef CORE_ANIMATION_ANIMATIONCONTROLLER_HPP
00002 #define CORE_ANIMATION_ANIMATIONCONTROLLER_HPP
00003
00004 #include "AnimationState.hpp"
00005 #include "Container.hpp"
00006
00007 namespace Animation {
00008     template< typename T = SLLAnimation >
00009     class AnimationController {
00010     public:
00011         typedef std::shared_ptr< AnimationController > Ptr;
00012
00013     private:
00014         std::vector< T > animationGroup;
00015         static constexpr float defaultSpeed = 1;
00016
00017     private:
00018         std::size_t mCurrentAnimationIndex;
00019         float mSpeed;
00020         bool Playing;
00021         bool interactionLock;
00022
00023         static constexpr float stopDuration = 0.25;
00024         float currStopDuration;
00025
00026     public:
00027         AnimationController();
00028         ~AnimationController();
00029         void RunAll();
```

```
00030     void Reset();
00031     void ResetCurrent();
00032 
00033     void SetAnimation(std::size_t animationIndex);
00034     std::size_t CurrentAnimationIndex() const;
00035 
00036 public:
00037     void AddAnimation(T animation);
00038     void PopAnimation();
00039     void Clear();
00040 
00041 public:
00042     float GetAnimationDuration();
00043     float GetAnimateFrame(float dt) const;
00044     std::size_t GetNumAnimation() const;
00045     std::size_t GetAnimationIndex() const;
00046     bool Done() const;
00047     bool IsPlaying() const;
00048 
00049 public:
00050     void StepForward();
00051     void StepBackward();
00052     void Pause();
00053     void Continue();
00054     void InteractionLock();
00055     void InteractionAllow();
00056     bool IsInteractionAllow() const;
00057 
00058 public:
00059     void Update(float dt);
00060     void SetSpeed(float speed);
00061     float GetSpeed() const;
00062 
00063 public:
00064     T GetAnimation();
00065 
00066 private:
00067     float GetStopDuration();
00068 };
00069 }; // namespace Animation
00070 
00071 #include <iostream>
00072 
00073 template< typename T >
00074 Animation::AnimationController< T >::AnimationController()
00075     : mSpeed(defaultSpeed), animationGroup({}), mCurrentAnimationIndex(0),
00076     Playing(false), interactionLock(false), currStopDuration(0.0f) {}
00077 
00078 template< typename T >
00079 Animation::AnimationController< T >::~AnimationController() {}
00080 
00081 template< typename T >
00082 void Animation::AnimationController< T >::RunAll() {
00083     if (animationGroup.empty()) return;
00084     ResetCurrent();
00085     SetAnimation(animationGroup.size() - 1);
00086     animationGroup.back().PlayingAt(animationGroup.back().GetDuration());
00087 }
00088 
00089 template< typename T >
00090 void Animation::AnimationController< T >::Reset() {
00091     if (!IsInteractionAllow()) return;
00092     for (auto &animation : animationGroup) {
00093         animation.Reset();
00094     }
00095     SetAnimation(0);
00096     Continue();
00097 }
00098 
00099 template< typename T >
00100 void Animation::AnimationController< T >::StepForward() {
00101     if (mCurrentAnimationIndex == animationGroup.size() - 1) {
00102         RunAll();
00103         return;
00104     }
00105     ResetCurrent();
00106     SetAnimation(mCurrentAnimationIndex + 1);
00107     ResetCurrent();
00108 }
00109 
00110 template< typename T >
00111 void Animation::AnimationController< T >::StepBackward() {
00112     ResetCurrent();
00113     SetAnimation(mCurrentAnimationIndex - 1);
00114     ResetCurrent();
00115 }
```

```

00117 template< typename T >
00118 void Animation::AnimationController< T >::SetAnimation(std::size_t index) {
00119     if (!IsInteractionAllow()) return;
00120     if (0 <= index && index < GetNumAnimation()) {
00121         if (mCurrentAnimationIndex > index) {
00122             for (; mCurrentAnimationIndex > index; mCurrentAnimation--) {
00123                 ResetCurrent();
00124             }
00125         }
00126         mCurrentAnimationIndex = index;
00127     }
00128 }
00129
00130 template< typename T >
00131 std::size_t Animation::AnimationController< T >::CurrentAnimationIndex() const {
00132     return mCurrentAnimationIndex;
00133 }
00134
00135 template< typename T >
00136 void Animation::AnimationController< T >::AddAnimation(T animation) {
00137     animationGroup.emplace_back(animation);
00138 }
00139
00140 template< typename T >
00141 void Animation::AnimationController< T >::PopAnimation() {
00142     assert(GetNumAnimation() > 0);
00143     animationGroup.pop_back();
00144 }
00145
00146 template< typename T >
00147 inline void Animation::AnimationController< T >::Clear() {
00148     animationGroup.clear();
00149 }
00150
00151 template< typename T >
00152 void Animation::AnimationController< T >::Pause() {
00153     if (IsInteractionAllow()) Playing = false;
00154 }
00155
00156 template< typename T >
00157 void Animation::AnimationController< T >::ResetCurrent() {
00158     if (!IsInteractionAllow()) return;
00159     if (animationGroup.empty()) return;
00160     animationGroup[mCurrentAnimationIndex].Reset();
00161 }
00162
00163 template< typename T >
00164 void Animation::AnimationController< T >::Continue() {
00165     if (IsInteractionAllow()) Playing = true;
00166 }
00167
00168 template< typename T >
00169 inline void Animation::AnimationController< T >::InteractionLock() {
00170     interactionLock = true;
00171 }
00172
00173 template< typename T >
00174 inline void Animation::AnimationController< T >::InteractionAllow() {
00175     interactionLock = false;
00176 }
00177
00178 template< typename T >
00179 inline bool Animation::AnimationController< T >::IsInteractionAllow() const {
00180     return !interactionLock;
00181 }
00182
00183 template< typename T >
00184 float Animation::AnimationController< T >::GetAnimationDuration() {
00185     float totalDuration = 0.0f;
00186     for (auto &animation : animationGroup) {
00187         totalDuration += animation.GetDuration();
00188     }
00189     return totalDuration;
00190 }
00191
00192 template< typename T >
00193 void Animation::AnimationController< T >::Update(float dt) {
00194     if (animationGroup.empty()) return;
00195     dt = GetAnimateFrame(dt);
00196     animationGroup[mCurrentAnimationIndex].Update(dt);
00197     if (IsPlaying() && animationGroup[mCurrentAnimationIndex].Done()) {
00198         // std::cout << dt << " | " << mCurrentAnimationIndex << " - "
00199         // << GetStopDuration() << " " << stopDuration << std::endl;
00200         if (GetStopDuration() >= stopDuration) {
00201             SetAnimation(mCurrentAnimationIndex + 1);
00202             currStopDuration = 0.0f;
00203         } else {

```

```

00204         currStopDuration = currStopDuration + dt;
00205     }
00206 }
00207 if (Done()) Pause();
00208 }
00209
00210 template< typename T >
00211 void Animation::AnimationController< T >::SetSpeed(float speed) {
00212     mSpeed = speed;
00213 }
00214
00215 template< typename T >
00216 float Animation::AnimationController< T >::GetSpeed() const {
00217     return mSpeed;
00218 }
00219
00220 template< typename T >
00221 T Animation::AnimationController< T >::GetAnimation() {
00222     if (animationGroup.empty()) return T();
00223     return animationGroup[mCurrentAnimationIndex];
00224 }
00225
00226 template< typename T >
00227 inline float Animation::AnimationController< T >::GetStopDuration() {
00228     return currStopDuration;
00229 }
00230
00231 template< typename T >
00232 float Animation::AnimationController< T >::GetAnimateFrame(float dt) const {
00233     if (!IsPlaying() || animationGroup.empty() || Done()) return 0.0f;
00234     return dt * mSpeed;
00235 }
00236
00237 template< typename T >
00238 std::size_t Animation::AnimationController< T >::GetNumAnimation() const {
00239     return animationGroup.size();
00240 }
00241
00242 template< typename T >
00243 std::size_t Animation::AnimationController< T >::GetAnimationIndex() const {
00244     return mCurrentAnimationIndex;
00245 }
00246
00247 template< typename T >
00248 bool Animation::AnimationController< T >::Done() const {
00249     if (animationGroup.empty()) return true;
00250     return animationGroup.back().Done();
00251 }
00252
00253 template< typename T >
00254 bool Animation::AnimationController< T >::IsPlaying() const {
00255     return Playing;
00256 }
00257
00258 /* */
00259 typedef Animation::AnimationController< SLLAnimation > SLLAnimationController;
00260 typedef Animation::AnimationController< DLLAnimation > DLLAnimationController;
00261 typedef Animation::AnimationController< CLLAnimation > CLLAnimationController;
00262 typedef Animation::AnimationController< DArrayAnimation >
00263     DArrayAnimationController;
00264
00265 #endif // CORE_ANIMATION_ANIMATIONCONTROLLER_HPP

```

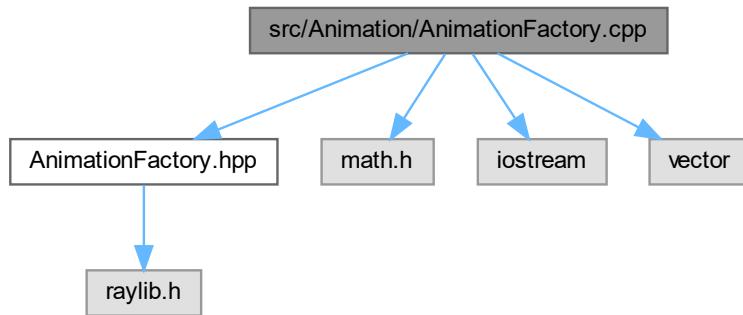
## 7.38 src/Animation/AnimationFactory.cpp File Reference

```

#include "AnimationFactory.hpp"
#include <math.h>
#include <iostream>
#include <vector>

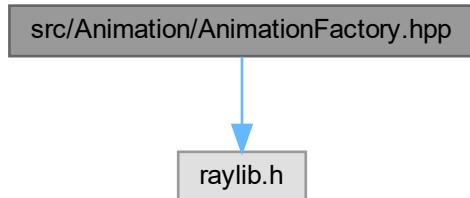
```

Include dependency graph for AnimationFactory.cpp:



## 7.39 src/Animation/AnimationFactory.hpp File Reference

```
#include "raylib.h"
Include dependency graph for AnimationFactory.hpp:
```



This graph shows which files directly or indirectly include this file:



## Namespaces

- namespace [AnimationFactory](#)

## Functions

- float `AnimationFactory::BounceOut` (float t)
- float `AnimationFactory::ElasticOut` (float t)
- void `AnimationFactory::DrawDirectionalArrow` (Vector2 start, Vector2 end, bool active, float t)
- void `AnimationFactory::DrawActiveArrow` (Vector2 start, Vector2 end, float t)
- void `AnimationFactory::DrawDoubleDirectionalArrow` (Vector2 start, Vector2 end, bool activeStart, bool activeEnd, float tStart, float tEnd)
- void `AnimationFactory::DrawDoubleActiveArrow` (Vector2 start, Vector2 end, float tStart, float tEnd)
- void `AnimationFactory::DrawCircularArrow` (Vector2 start, Vector2 end, bool active, float t)
- float `AnimationFactory::Dist` (Vector2 p1, Vector2 p2)
- Vector2 `AnimationFactory::InverseVector` (Vector2 vector)
- Vector2 `AnimationFactory::MoveNode` (Vector2 src, Vector2 dst, float t)
- Color `AnimationFactory::BlendColor` (Color src, Color dst, float t)
- void `AnimationFactory::ReCalculateEnds` (Vector2 &start, Vector2 &end, float radius, bool applyX=true, bool applyY=true)

## 7.40 AnimationFactory.hpp

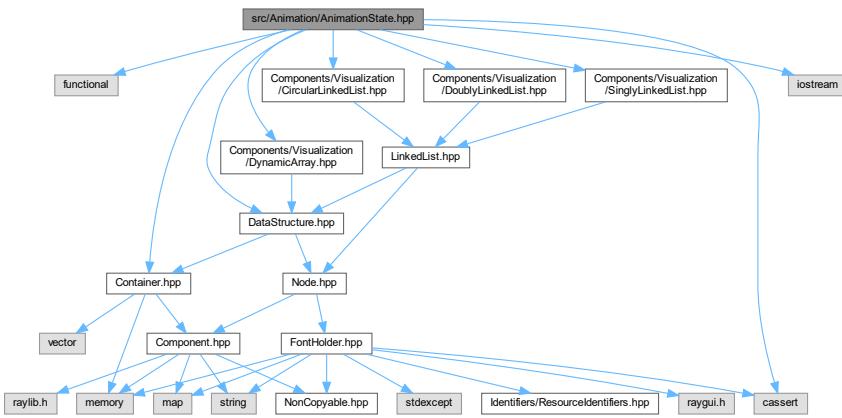
[Go to the documentation of this file.](#)

```
00001 #ifndef CORE_ANIMATION_ANIMATIONFACTORY_HPP
00002 #define CORE_ANIMATION_ANIMATIONFACTORY_HPP
00003
00004 #include "raylib.h"
00005
00006 namespace AnimationFactory {
00007     /* Pure Animation */
00008     float BounceOut(float t);
00009     float ElasticOut(float t);
00010     void DrawDirectionalArrow(Vector2 start, Vector2 end, bool active, float t);
00011     void DrawActiveArrow(Vector2 start, Vector2 end, float t);
00012
00013     void DrawDoubleDirectionalArrow(Vector2 start, Vector2 end,
00014                                     bool activeStart, bool activeEnd,
00015                                     float tStart, float tEnd);
00016
00017     void DrawDoubleActiveArrow(Vector2 start, Vector2 end, float tStart,
00018                               float tEnd);
00019
00020     void DrawCircularArrow(Vector2 start, Vector2 end, bool active, float t);
00021
00022     float Dist(Vector2 p1, Vector2 p2);
00023     Vector2 InverseVector(Vector2 vector);
00024     Vector2 MoveNode(Vector2 src, Vector2 dst, float t);
00025
00026     Color BlendColor(Color src, Color dst, float t);
00027
00028     void ReCalculateEnds(Vector2 &start, Vector2 &end, float radius,
00029                           bool applyX = true, bool applyY = true);
00030     /* Empty AnimationState */
00031 }; // namespace AnimationFactory
00032
00033 #endif // CORE_ANIMATION_ANIMATIONFACTORY_HPP
```

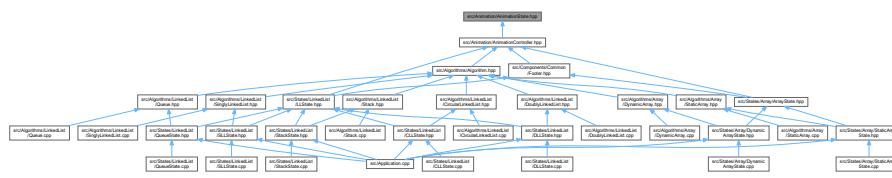
## 7.41 src/Animation/AnimationState.hpp File Reference

```
#include <functional>
#include "Components/Visualization/CircularLinkedList.hpp"
#include "Components/Visualization/DataStructure.hpp"
#include "Components/Visualization/DoublyLinkedList.hpp"
#include "Components/Visualization/DynamicArray.hpp"
#include "Components/Visualization/SinglyLinkedList.hpp"
#include "Container.hpp"
```

```
#include <cassert>
#include <iostream>
Include dependency graph for AnimationState.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Animation::AnimationState< T >](#)

## Namespaces

- namespace [Animation](#)

## Typedefs

- [typedef Animation::AnimationState< GUI::SinglyLinkedList > SLLAnimation](#)
- [typedef Animation::AnimationState< GUI::DoublyLinkedList > DLLAnimation](#)
- [typedef Animation::AnimationState< GUI::CircularLinkedList > CLLAnimation](#)
- [typedef Animation::AnimationState< GUI::DynamicArray > DArrayAnimation](#)

### 7.41.1 Typedef Documentation

### 7.41.1.1 CLLAnimation

```
typedef Animation::AnimationState< GUI::CircularLinkedList > CLLAnimation
```

### 7.41.1.2 DArrayAnimation

```
typedef Animation::AnimationState< GUI::DynamicArray > DArrayAnimation
```

### 7.41.1.3 DLLAnimation

```
typedef Animation::AnimationState< GUI::DoublyLinkedList > DLLAnimation
```

### 7.41.1.4 SLLAnimation

```
typedef Animation::AnimationState< GUI::SinglyLinkedList > SLLAnimation
```

## 7.42 AnimationState.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef CORE_ANIMATION_ANIMATIONSTATE_HPP
00002 #define CORE_ANIMATION_ANIMATIONSTATE_HPP
00003
00004 #include <functional>
00005
00006 #include "Components/Visualization/CircularLinkedList.hpp"
00007 #include "Components/Visualization/DataStructure.hpp"
00008 #include "Components/Visualization/DoublyLinkedList.hpp"
00009 #include "Components/Visualization/DynamicArray.hpp"
00010 #include "Components/Visualization/SinglyLinkedList.hpp"
00011 #include "Container.hpp"
00012
00013 namespace Animation {
00014     template< typename T >
00015     class AnimationState {
00016     public:
00017         typedef std::shared_ptr< AnimationState< T > > Ptr;
00018
00019     private:
00020         float mDuration;
00021         float mCurrentPlayingAt;
00022         int mHighlightedLine;
00023
00024         std::string actionDescription;
00025
00026     private:
00027         T mDataStructureBefore;
00028         std::function< T(T, float, Vector2) > mAnimation;
00029
00030     public:
00031         AnimationState();
00032         ~AnimationState();
00033
00034         void PlayingAt(float playingAt);
00035         float GetCurrentPlayingAt() const;
00036         void Draw(Vector2 base = (Vector2){0, 0});
00037         void Update(float dt);
00038         void Reset();
```

```

00039
00040     public:
00041         void SetDuration(float duration);
00042         float GetDuration() const;
00043         void SetAnimation(std::function< T(T, float, Vector2) > animation);
00044         void SetSourceDataStructure(T dataStructure);
00045         T GetDataStructure(float progress, Vector2 base = (Vector2){0, 0});
00046
00047     public:
00048         bool Done() const;
00049         void SetHighlightLine(int line);
00050         int GetHighlightedLine() const;
00051         void SetActionDescription(std::string description);
00052         std::string GetActionDescription() const;
00053     };
00054 }; // namespace Animation
00055
00056 #include <cassert>
00057 #include <iostream>
00058
00059 // #include "AnimationState.hpp"
00060
00061 template< typename T >
00062 Animation::AnimationState< T >::AnimationState()
00063     : mCurrentPlayingAt(0.0f), mDuration(0.0f), mHighlightedLine(-1),
00064       mDataStructureBefore(nullptr), actionDescription("") {
00065     mAnimation = [] (T srcDS, float playingAt, Vector2 base) {
00066         srcDS.Draw(base, playingAt);
00067         return srcDS;
00068     };
00069 }
00070
00071 template< typename T >
00072 Animation::AnimationState< T >::~AnimationState() {}
00073
00074 template< typename T >
00075 void Animation::AnimationState< T >::SetDuration(float duration) {
00076     mDuration = duration;
00077 }
00078
00079 template< typename T >
00080 float Animation::AnimationState< T >::GetDuration() const {
00081     return mDuration;
00082 }
00083
00084 template< typename T >
00085 void Animation::AnimationState< T >::SetAnimation(
00086     std::function< T(T, float, Vector2) > animation) {
00087     mAnimation = animation;
00088 }
00089
00090 template< typename T >
00091 void Animation::AnimationState< T >::PlayingAt(float playingAt) {
00092     mCurrentPlayingAt = playingAt;
00093 }
00094
00095 template< typename T >
00096 float Animation::AnimationState< T >::GetCurrentPlayingAt() const {
00097     return mCurrentPlayingAt;
00098 }
00099
00100 template< typename T >
00101 void Animation::AnimationState< T >::SetSourceDataStructure(T dataStructure) {
00102     mDataStructureBefore = dataStructure;
00103 }
00104
00105 template< typename T >
00106 T Animation::AnimationState< T >::GetDataStructure(float progress,
00107                                         Vector2 base) {
00108     assert(progress >= 0.0f && progress <= 1.0f);
00109     return mAnimation(mDataStructureBefore, progress, base);
00110 }
00111
00112 template< typename T >
00113 void Animation::AnimationState< T >::Update(float dt) {
00114     mCurrentPlayingAt += dt;
00115     if (mCurrentPlayingAt > mDuration + 1.0f)
00116         mCurrentPlayingAt = mDuration + 1.0f;
00117 }
00118
00119 template< typename T >
00120 void Animation::AnimationState< T >::Reset() {
00121     PlayingAt(0.0f);
00122 }
00123
00124 template< typename T >
00125 bool Animation::AnimationState< T >::Done() const {

```

```

00126     return mCurrentPlayingAt >= mDuration;
00127 }
00128
00129 template< typename T >
00130 void Animation::AnimationState< T >::SetHighlightLine(int line) {
00131     mHighlightedLine = line;
00132 }
00133
00134 template< typename T >
00135 int Animation::AnimationState< T >::GetHighlightedLine() const {
00136     return mHighlightedLine;
00137 }
00138
00139 template< typename T >
00140 inline void Animation::AnimationState< T >::SetActionDescription(
00141     std::string description) {
00142     actionDescription = description;
00143 }
00144
00145 template< typename T >
00146 inline std::string Animation::AnimationState< T >::GetActionDescription()
00147     const {
00148     return actionDescription;
00149 }
00150
00151 template< class T >
00152 void Animation::AnimationState< T >::Draw(Vector2 base) {
00153     float playingAt = std::min(
00154         1.0f, (mDuration == 0.0f ? 1.0f : mCurrentPlayingAt / mDuration));
00155     mAnimation(mDataStructureBefore, playingAt, base);
00156 }
00157
00158 typedef Animation::AnimationState< GUI::SinglyLinkedList > SLLAnimation;
00159 typedef Animation::AnimationState< GUI::DoublyLinkedList > DLLAnimation;
00160 typedef Animation::AnimationState< GUI::CircularLinkedList > CLLAnimation;
00161 typedef Animation::AnimationState< GUI::DynamicArray > DArrayAnimation;
00162
00163 #endif // CORE_ANIMATION_ANIMATIONSTATE_HPP

```

## 7.43 src/Application.cpp File Reference

```

#include "Application.hpp"
#include "raylib.h"
#include "Global.hpp"
#include "Settings.hpp"
#include "States/Array/DynamicArrayState.hpp"
#include "States/Array/StaticArrayState.hpp"
#include "States/HomepageState.hpp"
#include "States/LinkedList/CLLState.hpp"
#include "States/LinkedList/DLLState.hpp"
#include "States/LinkedList/QueueState.hpp"
#include "States/LinkedList/SLLState.hpp"
#include "States/LinkedList/StackState.hpp"
#include "States/SettingsState.hpp"

```

Include dependency graph for Application.cpp:



## Macros

- #define RAYGUI\_IMPLEMENTATION

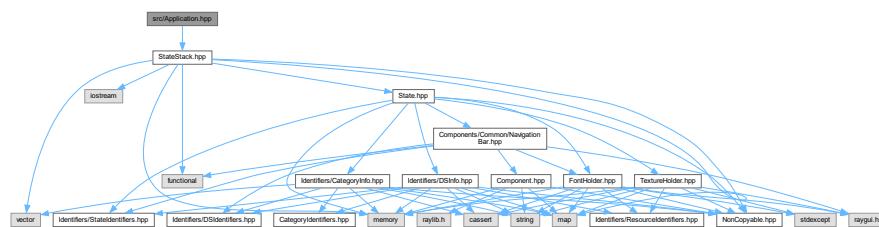
### 7.43.1 Macro Definition Documentation

#### 7.43.1.1 RAYGUI\_IMPLEMENTATION

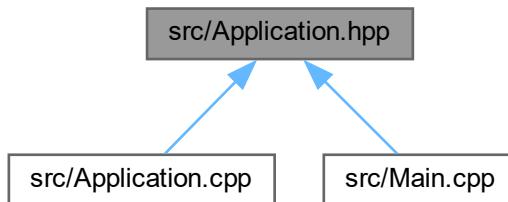
```
#define RAYGUI_IMPLEMENTATION
```

## 7.44 src/Application.hpp File Reference

#include "StateStack.hpp"  
 Include dependency graph for Application.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Application](#)

## 7.45 Application.hpp

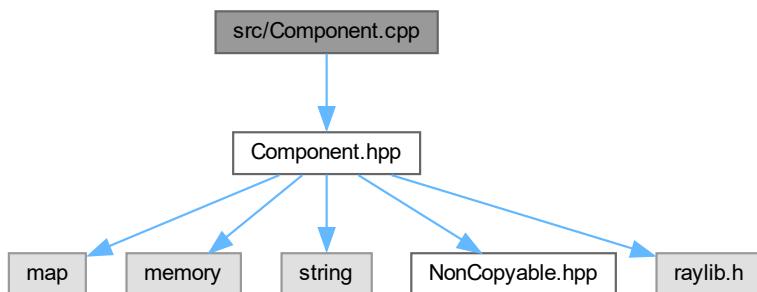
Go to the documentation of this file.

```
00001 #ifndef APPLICATION_HPP
00002 #define APPLICATION_HPP
00003
00004 #include "StateStack.hpp"
00005
00006 class Application {
00007 public:
00008     Application();
00009     ~Application();
00010     void Run();
00011     void Close();
00012     void Init();
00013     bool WindowClosed();
00014
00015 private:
00016     void Render();
00017     void RegisterStates();
00018     void LoadResources();
00019     void Update(float dt);
00020
00021 private:
00022     bool closed = false;
00023     // Image favicon;
00024
00025 private:
00026     StateStack mStack;
00027     FontHolder* fonts;
00028     TextureHolder* images;
00029     CategoryInfo* categories;
00030     DSInfo* dsInfo;
00031 };
00032
00033 #endif // APPLICATION_HPP
```

## 7.46 src/Component.cpp File Reference

```
#include "Component.hpp"
```

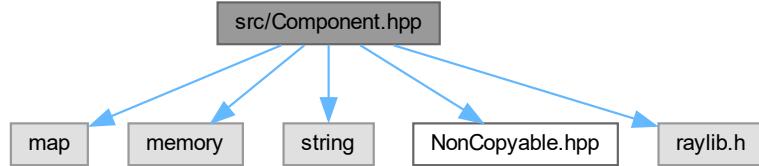
Include dependency graph for Component.cpp:



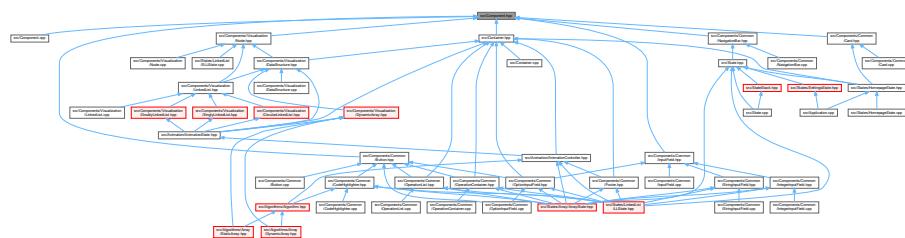
## 7.47 src/Component.hpp File Reference

```
#include <map>
#include <memory>
#include <string>
```

```
#include "NonCopyable.hpp"
#include "raylib.h"
Include dependency graph for Component.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::Component](#)

## Namespaces

- namespace [GUI](#)

## 7.48 Component.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef COMPONENT_HPP
00002 #define COMPONENT_HPP
00003
00004 #include <map>
00005 #include <memory>
00006 #include <string>
00007
00008 #include "NonCopyable.hpp"
00009 #include "raylib.h"
00010
00011 namespace GUI {
00012     class Component
00013     //: private NonCopyable< Component >
00014     {
00015     public:
00016         typedef std::shared_ptr< Component > Ptr;
00017
00018     public:
00019         Component();
```

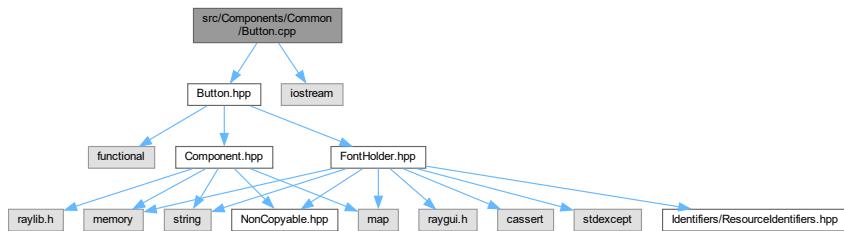
```

00020     virtual ~Component();
00021
00022     virtual void Draw(Vector2 base = (Vector2){0, 0});
00023
00024     virtual bool IsSelectable() const = 0;
00025     bool IsSelected() const;
00026     virtual void Select();
00027     virtual void Deselect();
00028
00029     virtual void ToggleVisible();
00030     virtual void SetVisible(bool visible);
00031     virtual bool GetVisible();
00032
00033 public:
00034     void SetPosition(float x, float y);
00035     void SetPosition(Vector2 position);
00036
00037     Vector2 GetPosition();
00038
00039     virtual Vector2 GetSize();
00040
00041 protected:
00042     Vector2 mPosition;
00043     bool mVisible;
00044
00045 private:
00046     bool mIsSelected;
00047
00048 protected:
00049     virtual void UpdateMouseCursorWhenHover(
00050         std::map< std::string, Rectangle > bounds, bool hover,
00051         bool noHover);
00052     virtual void UpdateMouseCursorWhenHover(Rectangle bound, bool hover,
00053                                             bool noHover);
00054     virtual bool GetHoverStatus(std::map< std::string, Rectangle > bounds,
00055                                 bool hover, bool noHover);
00056     virtual bool GetHoverStatus(Rectangle bound, bool hover, bool noHover);
00057 };
00058 }; // namespace GUI
00059
00060 #endif // COMPONENT_HPP

```

## 7.49 src/Components/Common/Button.cpp File Reference

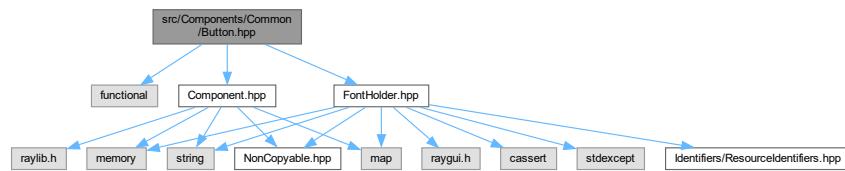
```
#include "Button.hpp"
#include <iostream>
Include dependency graph for Button.cpp:
```



## 7.50 src/Components/Common/Button.hpp File Reference

```
#include <functional>
#include "Component.hpp"
```

```
#include "FontHolder.hpp"
Include dependency graph for Button.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::Button](#)

## Namespaces

- namespace [GUI](#)

## 7.51 Button.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef COMPONENTS_GUIBUTTON_HPP
00002 #define COMPONENTS_GUIBUTTON_HPP
00003
00004 #include <functional>
00005
00006 #include "Component.hpp"
00007 #include "FontHolder.hpp"
00008
00009 namespace GUI {
00010     class Button : public GUI::Component {
00011     public:
00012         enum TextAlignment { Left, Center, Right, AlignmentCount };
00013         typedef std::shared_ptr< Button > Ptr;
00014         Button(std::string text, FontHolder* fonts);
00015         Button();
00016         ~Button();
00017         void Draw(Vector2 base = (Vector2){0, 0});
00018
00019         void SetAction(std::function< void() > clickedAction);
00020
00021         bool isSelectable() const;
00022
00023     public:
00024         void SetButtonHoverColor(Color color);
00025         void SetButtonColor(Color color);
00026         void SetTextColor(Color color);
00027         void SetSize(float width, float height);
```

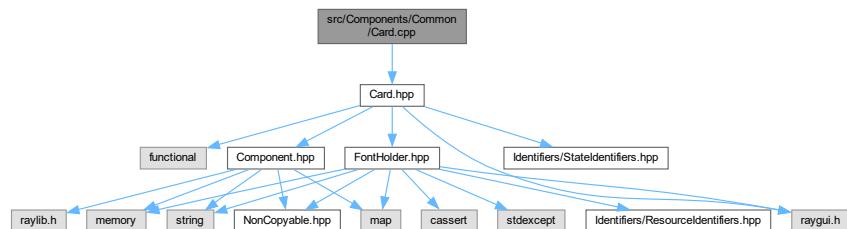
```

00028     void SetText(std::string text);
00029
00030     public:
00031         void SetFontSize(float textSize);
00032         float GetFontSize() const;
00033         void SetTextAlignment(TextAlignment textAlignment);
00034
00035         void EnableFitContent();
00036         void DisableFitContent();
00037
00038         Vector2 GetSize();
00039
00040         void SetActionOnHover(bool actionOnHover);
00041
00042     private:
00043         bool IsClicked();
00044         void DrawButtonText();
00045         Vector2 GetContentPos();
00046         Vector2 GetContentSize();
00047
00048         void FitContent();
00049
00050     private:
00051         FontHolder* fonts;
00052         std::string content;
00053         Color buttonColor;
00054         Color buttonHoverColor;
00055         Color textColor;
00056         TextAlignment alignment;
00057         float fontSize;
00058
00059     private:
00060         bool fitContent;
00061         bool mActionOnHover;
00062
00063     private:
00064         bool isHover;
00065         Rectangle bound;
00066         std::function< void() > action;
00067     };
00068 }; // namespace GUI
00069
00070 #endif // COMPONENTS_Guibutton_HPP

```

## 7.52 src/Components/Common/Card.cpp File Reference

#include "Card.hpp"  
Include dependency graph for Card.cpp:



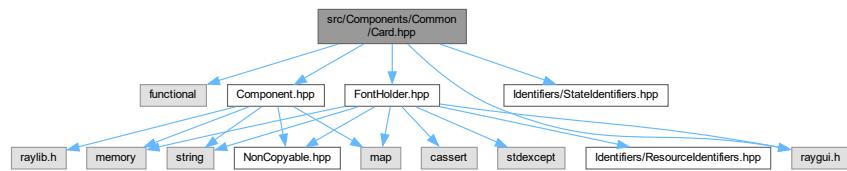
## 7.53 src/Components/Common/Card.hpp File Reference

```

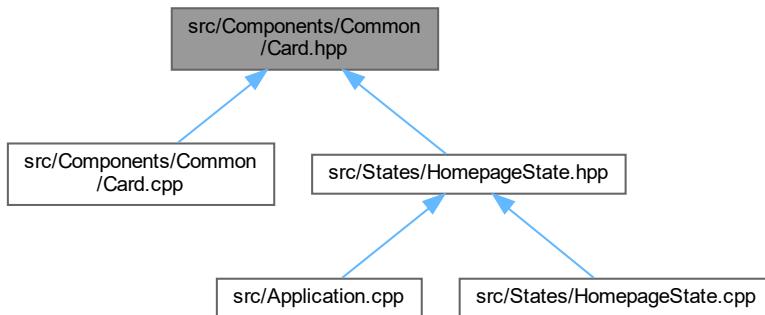
#include <functional>
#include "Component.hpp"
#include "FontHolder.hpp"
#include "Identifiers/StateIdentifiers.hpp"

```

```
#include "raygui.h"
Include dependency graph for Card.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::Card](#)

## Namespaces

- namespace [GUI](#)

## 7.54 Card.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef COMPONENTS_CARD_HPP
00002 #define COMPONENTS_CARD_HPP
00003
00004 #include <functional>
00005
00006 #include "Component.hpp"
00007 #include "FontHolder.hpp"
00008 #include "Identifiers/StateIdentifiers.hpp"
00009 #include "raygui.h"
00010
00011 /* Little note
00012     - Card height: 250
00013     - Card width: 200
00014     - Title: 24px
```

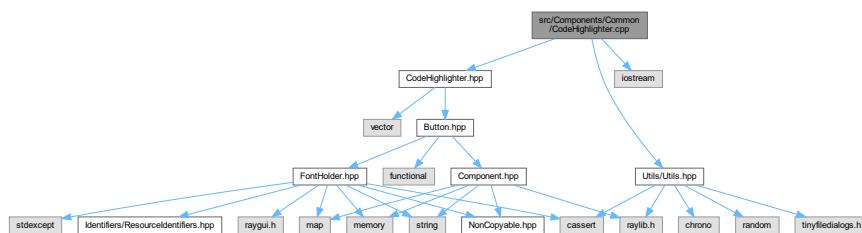
```

00015     - Image: 160x250
00016
00017 */
00018
00019 namespace GUI {
00020     class Card : public GUI::Component {
00021     public:
00022         Card(std::string text, Texture thumbnail, FontHolder* fonts);
00023         Card();
00024         ~Card();
00025         void SetLink(std::function< void(States::ID) > link);
00026         void SetStateID(States::ID stateID);
00027
00028         void SetText(std::string text);
00029
00030     public:
00031         bool IsSelectable() const;
00032         void Draw(Vector2 base = (Vector2){0, 0});
00033
00034     private:
00035         bool DrawImage(Vector2 base);
00036         bool DrawTitle(Vector2 base);
00037
00038     private:
00039         FontHolder* fonts;
00040         Texture thumbnail;
00041         std::string title;
00042         std::function< void(States::ID) > toLink;
00043         States::ID stateID;
00044
00045     private:
00046         std::map< std::string, Rectangle > hoverBounds;
00047         bool isHover;
00048     };
00049 }; // namespace GUI
00050
00051 #endif // COMPONENTS_CARD_HPP

```

## 7.55 src/Components/Common/CodeHighlighter.cpp File Reference

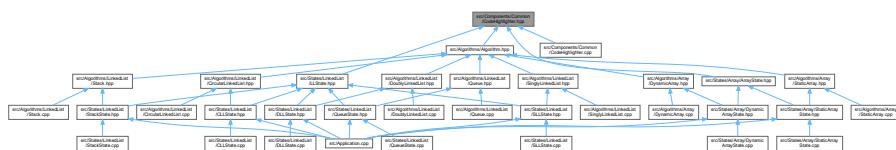
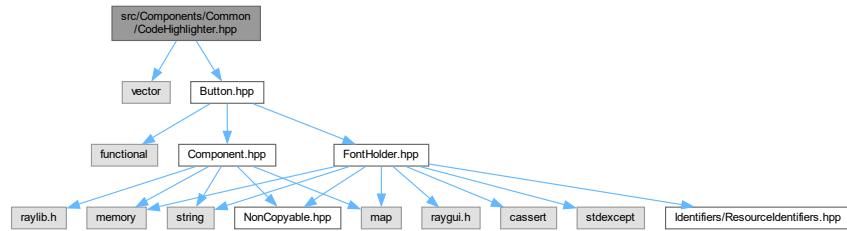
```
#include "CodeHighlighter.hpp"
#include <iostream>
#include "Utils/Utils.hpp"
Include dependency graph for CodeHighlighter.cpp:
```



## 7.56 src/Components/Common/CodeHighlighter.hpp File Reference

```
#include <vector>
#include "Button.hpp"
```

Include dependency graph for CodeHighlighter.hpp:



## Classes

- class [GUI::CodeHighlighter](#)

## Namespaces

- namespace [GUI](#)

## 7.57 CodeHighlighter.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef COMPONENTS_COMMON_CODEHIGHLIGHTER_HPP
00002 #define COMPONENTS_COMMON_CODEHIGHLIGHTER_HPP
00003
00004 #include <vector>
00005
00006 // #include "Component.hpp"
00007 #include "Button.hpp"
00008
00009 namespace GUI {
00010     class CodeHighlighter : public GUI::Component {
00011     public:
00012         typedef std::shared_ptr< CodeHighlighter > Ptr;
00013
00014     private:
00015         GUI::Button mButtonShowCode;
00016         bool mShowCode;
00017
00018         GUI::Button mButtonShowAction;
00019         bool mShowActionDescription;
00020
00021         FontHolder* fonts;
00022
00023     private:
00024         std::vector< std::string > mCode;
00025         int mHighlightedLine;
00026         std::string mActionDescription;
00027
  
```

```

00028     public:
00029         CodeHighlighter(FontHolder* fonts);
00030         ~CodeHighlighter();
00031         void Draw(Vector2 base = (Vector2){0, 0});
00032         bool IsSelectable() const;
00033         void InitButtons();
00034
00035     public:
00036         void AddCode(std::vector< std::string > code);
00037         void Highlight(int line);
00038         void ToggleShowCode();
00039         void ToggleShowAction();
00040         void SetShowCode(bool show);
00041         void SetShowAction(bool show);
00042
00043     public:
00044         void AddActionDescription(std::string description);
00045
00046     private:
00047         void DrawActionDescription(Vector2 base = (Vector2){0, 0});
00048         void DrawCodeHighlighter(Vector2 base = (Vector2){0, 0});
00049     };
00050 } // namespace GUI
00051
00052 #endif // COMPONENTS_COMMON_CODEHIGHLIGHTER_HPP

```

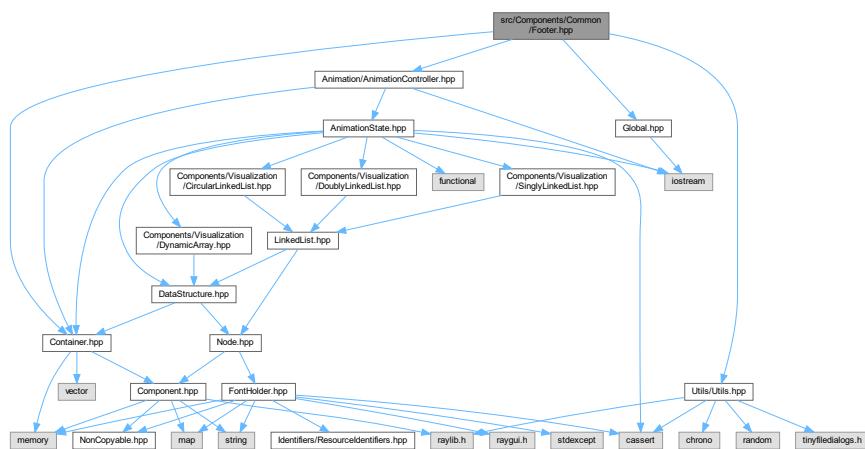
## 7.58 src/Components/Common/Footer.cpp File Reference

## 7.59 src/Components/Common/Footer.hpp File Reference

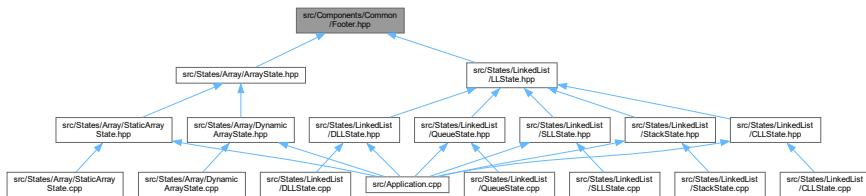
```

#include "Animation/AnimationController.hpp"
#include "Container.hpp"
#include "Global.hpp"
#include "Utils/Utils.hpp"
Include dependency graph for Footer.hpp:

```



This graph shows which files directly or indirectly include this file:



## Classes

- class **GUI::Footer< T >**

# Namespaces

- namespace **GUI**

## 7.60 Footer.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef COMPONENTS_COMMON_FOOTER_HPP
00002 #define COMPONENTS_COMMON_FOOTER_HPP
00003
00004 #include "Animation/AnimationController.hpp"
00005 #include "Container.hpp"
00006
00007 namespace GUI {
00008     template< typename T >
00009     class Footer : public GUI::Container {
00010     public:
00011         Footer();
00012         ~Footer();
00013         void Draw(T* animController, Vector2 base = (Vector2){0, 0});
00014     };
00015 }; // namespace GUI
00016
00017 #include "Global.hpp"
00018 #include "Utils/Utils.hpp"
00019 // #include "raygui.h"
00020
00021 template< typename T >
00022 GUI::Footer< T >::Footer() {}
00023
00024 template< typename T >
00025 GUI::Footer< T >::~Footer() {}
00026
00027 template< typename T >
00028 void GUI::Footer< T >::Draw(T* animController, Vector2 base) {
00029     base.x += mPosition.x;
00030     base.y += mPosition.y;
00031     DrawRectangleRec((Rectangle){base.x, base.y, global::SCREEN_WIDTH, 40},
00032                       BLACK);
00033
00034     animController->SetSpeed(
00035         int(GuiSlider((Rectangle){base.x + 80, base.y + 15, 140, 10}, nullptr,
00036                         TextFormat("%dx", int(animController->GetSpeed())),
00037                         animController->GetSpeed(), 1, 7)));
00038
00039     { // Draw Reset
00040         if (Utils::DrawIcon(129, base.x + 345, base.y + 12, 1, WHITE,
00041                            (Color){82, 188, 105, 255})) {
00042             animController->Reset();
00043         }
00044     }
00045 }
```

```

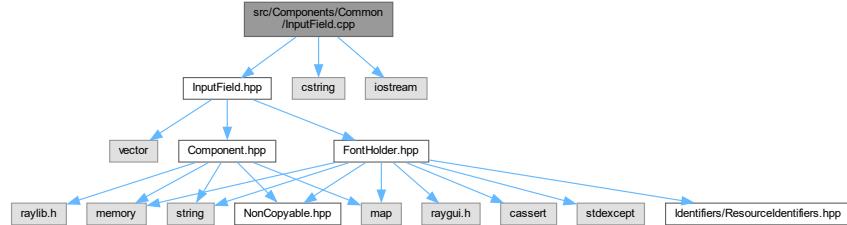
00046     { // Draw Step Backward
00047         if (Utils::DrawIcon(130, base.x + 365, base.y + 12, 1, WHITE,
00048                             (Color){82, 188, 105, 255})) {
00049             animController->StepBackward();
00050         }
00051     }
00052
00053     { // Draw play button
00054         if (animController->Done()) {
00055             if (Utils::DrawIcon(74, base.x + 385, base.y + 12, 1, WHITE,
00056                             (Color){82, 188, 105, 255})) {
00057                 animController->Reset();
00058             }
00059         } else if (animController->IsPlaying()) {
00060             if (Utils::DrawIcon(132, base.x + 385, base.y + 12, 1, WHITE,
00061                             (Color){82, 188, 105, 255})) {
00062                 animController->Pause();
00063             }
00064         } else if (Utils::DrawIcon(133, base.x + 385, base.y + 12, 1, WHITE,
00065                             (Color){82, 188, 105, 255})) {
00066             animController->Continue();
00067         }
00068     }
00069
00070     { // Draw Step Forward
00071         if (Utils::DrawIcon(131, base.x + 405, base.y + 12, 1, WHITE,
00072                             (Color){82, 188, 105, 255})) {
00073             animController->StepForward();
00074         }
00075     }
00076
00077     { // Draw Run All
00078         if (Utils::DrawIcon(134, base.x + 425, base.y + 12, 1, WHITE,
00079                             (Color){82, 188, 105, 255})) {
00080             animController->RunAll();
00081         }
00082     }
00083
00084     // progressValue =
00085     animController->SetAnimation(
00086         GuiSliderBar(Rectangle){base.x + 450, base.y + 15, 400, 10}, nullptr,
00087         nullptr, animController->GetAnimationIndex(), 0,
00088         std::max(0, int(animController->GetNumAnimation() - 1)));
00089 }
00090
00091 #endif // COMPONENTS_COMMON_FOOTER_HPP

```

## 7.61 src/Components/Common/InputField.cpp File Reference

```
#include "InputField.hpp"
#include <cstring>
#include <iostream>
```

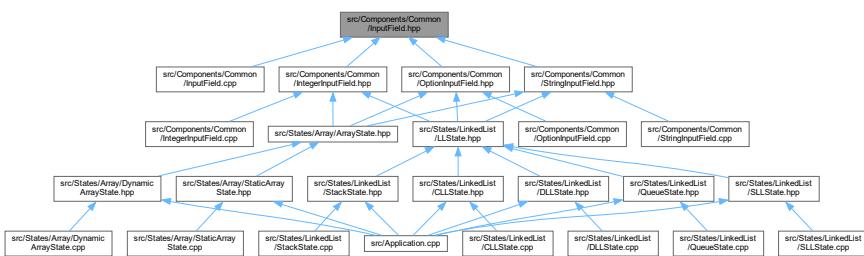
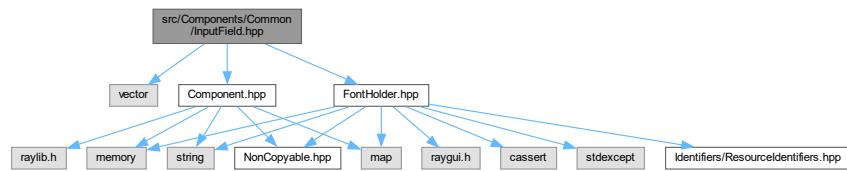
Include dependency graph for InputField.cpp:



## 7.62 src/Components/Common/InputField.hpp File Reference

```
#include <vector>
#include "Component.hpp"
```

```
#include "FontHolder.hpp"
Include dependency graph for InputField.hpp:
```



## Classes

- class [GUI::InputField](#)

## Namespaces

- namespace [GUI](#)

## 7.63 InputField.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef COMPONENTS_INPUTFIELD_HPP
00002 #define COMPONENTS_INPUTFIELD_HPP
00003
00004 #include <vector>
00005
00006 #include "Component.hpp"
00007 #include "FontHolder.hpp"
00008
00009 namespace GUI {
00010     class InputField : public GUI::Component {
00011     public:
00012         typedef std::shared_ptr< InputField > Ptr;
00013         InputField(FontHolder* fonts);
00014         ~InputField();
00015         virtual std::string ExtractValue() = 0;
00016         virtual void Draw(Vector2 base = (Vector2){0, 0});
00017         virtual void DrawField(Vector2 base = (Vector2){0, 0}) = 0;
00018         virtual bool isSelectable() const;
00019
00020     public:
00021         virtual void SetLabelSize(float fontSize);
00022         virtual void SetInputFieldSize(Vector2 size);
```

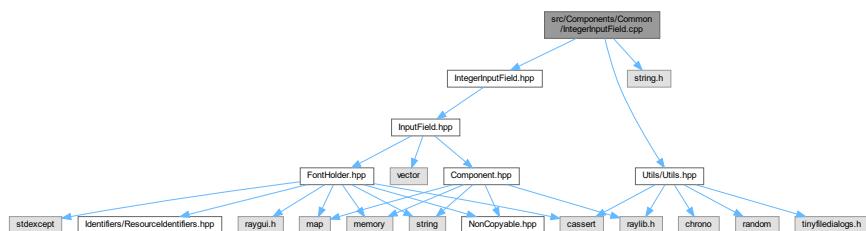
```

00023     virtual Vector2 GetSize();
00024     virtual void SetLabel(std::string labelContent);
00025     virtual std::string GetLabel() const;
00026
00027 protected:
00028     virtual bool IsClicked(Vector2 base = (Vector2){0, 0}) const;
00029     virtual void SetEditMode(bool canEdit);
00030     virtual bool GetEditMode() const;
00031     virtual void AllFieldDisableEdit();
00032
00033 public:
00034     virtual void Randomize();
00035
00036 protected:
00037     bool editMode;
00038     std::size_t mFieldIndex;
00039
00040 protected:
00041     float labelFontSize;
00042     Vector2 inputFieldSize;
00043
00044 protected:
00045     std::string label;
00046     FontHolder* fonts;
00047     std::string extractedValue;
00048
00049 public:
00050     static std::vector< bool > fields;
00051 };
00052 }; // namespace GUI
00053
00054 #endif // COMPONENTS_INPUTFIELD_HPP

```

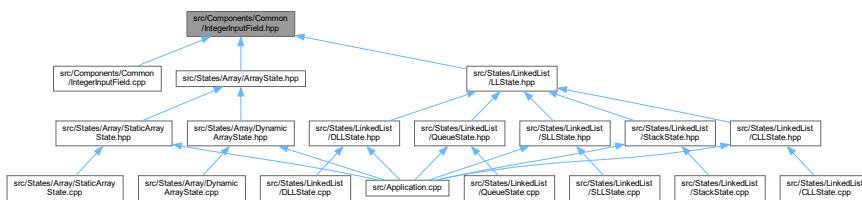
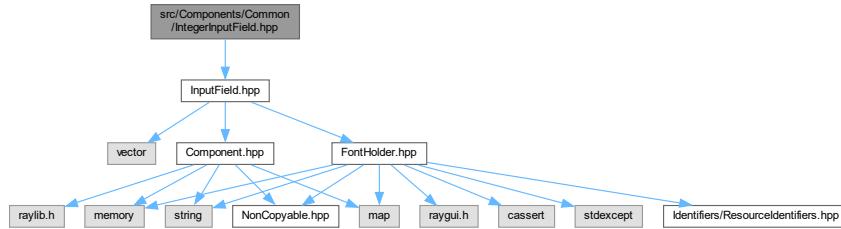
## 7.64 src/Components/Common/IntegerField.cpp File Reference

```
#include "IntegerField.hpp"
#include <string.h>
#include "Utils/Utils.hpp"
Include dependency graph for IntegerInputField.cpp:
```



## 7.65 src/Components/Common/IntegerField.hpp File Reference

```
#include "InputField.hpp"
Include dependency graph for IntegerInputField.hpp:
```



## Classes

- class [GUI::IntegerField](#)

## Namespaces

- namespace [GUI](#)

## 7.66 IntegerInputField.hpp

[Go to the documentation of this file.](#)

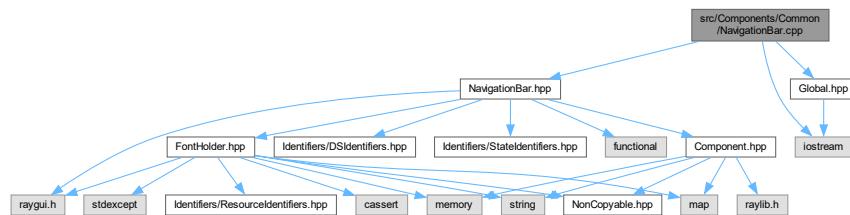
```

00001 #ifndef COMPONENTS_INTEGERINPUTFIELD_HPP
00002 #define COMPONENTS_INTEGERINPUTFIELD_HPP
00003
00004 #include "InputField.hpp"
00005
00006 namespace GUI {
00007     class IntegerInputField : public GUI::InputField {
00008     public:
00009         typedef std::shared_ptr< IntegerInputField > Ptr;
00010         IntegerInputField(FontHolder* fonts);
00011         ~IntegerInputField();
00012         std::string ExtractValue();
00013         void DrawField(Vector2 base = (Vector2){0, 0});
00014
00015         void SetConstraint(int minValue, int maxValue);
00016         bool IsSelectable() const;
00017
  
```

```
00018     public:
00019         void Randomize();
00020
00021     private:
00022         int mMinValue;
00023         int mMaxValue;
00024         int input;
00025     };
00026 }; // namespace GUI
00027
00028 #endif // COMPONENTS_INTEGERINPUTFIELD_HPP
```

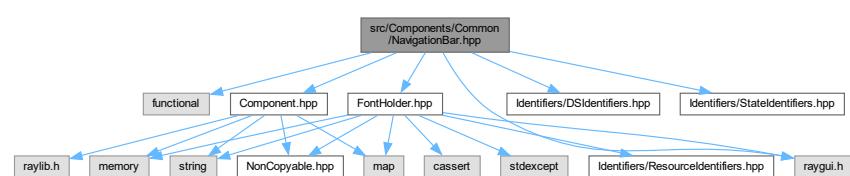
## 7.67 src/Components/Common/NavigationBar.cpp File Reference

```
#include "NavigationBar.hpp"
#include "Global.hpp"
#include <iostream>
Include dependency graph for NavigationBar.cpp
```

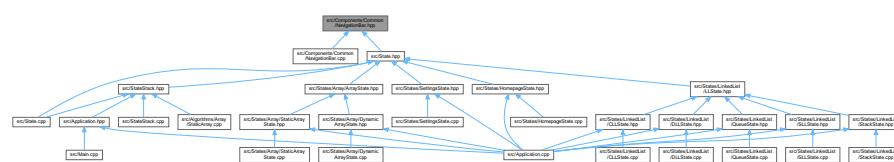


## 7.68 src/Components/Common/NavigationBar.hpp File Reference

```
#include <functional>
#include "Component.hpp"
#include "FontHolder.hpp"
#include "Identifiers/DSIIdentifiers.hpp"
#include "Identifiers/StateIdentifiers.hpp"
#include "raygui.h"
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::NavigationBar](#)
- struct [GUI::NavigationBar::TitleInfo](#)

## Namespaces

- namespace [GUI](#)

## 7.69 NavigationBar.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef COMPONENTS_NAVIGATIONBAR_HPP
00002 #define COMPONENTS_NAVIGATIONBAR_HPP
00003
00004 #include <functional>
00005
00006 #include "Component.hpp"
00007 #include "FontHolder.hpp"
00008 #include "Identifiers/DSIIdentifiers.hpp"
00009 #include "Identifiers/StateIdentifiers.hpp"
00010 #include "raygui.h"
00011
00012 namespace GUI {
00013     class NavigationBar : public GUI::Component {
00014     public:
00015         NavigationBar(FontHolder* fonts);
00016         NavigationBar();
00017         ~NavigationBar();
00018
00019         void SetHomepageID(States::ID id);
00020         void SetSettingsID(States::ID id);
00021         void SetDirectLink(std::function< void(States::ID) > link);
00022         void SetBackToPreviousLink(std::function< void() > link);
00023
00024         void AtSettings(bool settings);
00025
00026         void SetCategory(std::string category);
00027
00028         void InsertTitle(DataStructures::ID titleID, States::ID stateID,
00029                         std::string abbrTitle, std::string titleName);
00030         void SetActiveTitle(DataStructures::ID title);
00031         void ClearTitle();
00032
00033         void SetVisibleTitle(bool visible);
00034
00035     public:
00036         bool isSelectable() const;
00037         void Draw(Vector2 base = (Vector2){0, 0});
00038
00039     private:
00040         bool DrawSettings();
00041         bool DrawLogo();
00042         States::ID DrawTitles();
00043
00044     private:
00045         struct TitleInfo {
00046             States::ID stateID;
00047             std::string abbrTitle;
00048             std::string titleName;
00049         };
00050
00051         FontHolder* fonts;
00052         std::string currentCategory;
00053         std::map< DataStructures::ID, TitleInfo > mTitles;
00054         DataStructures::ID activeTitle;
00055         bool hasTitle;
00056
00057     private:
00058         std::function< void(States::ID) > toLink;
00059         std::function< void() > backToPrvState;
00060         States::ID homepageID;
00061         States::ID settingsID;
00062         bool atSettings;
00063
00064     private:

```

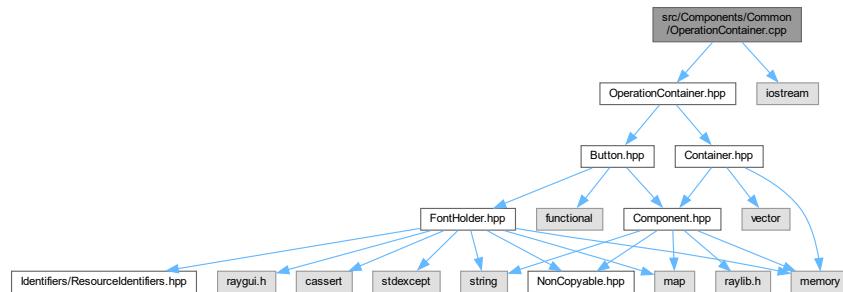
```

00065     std::map< std::string, Rectangle > hoverBounds;
00066     bool isHover;
00067 };
00068 }; // namespace GUI
00069
00070 #endif // COMPONENTS_NAVIGATIONBAR_HPP

```

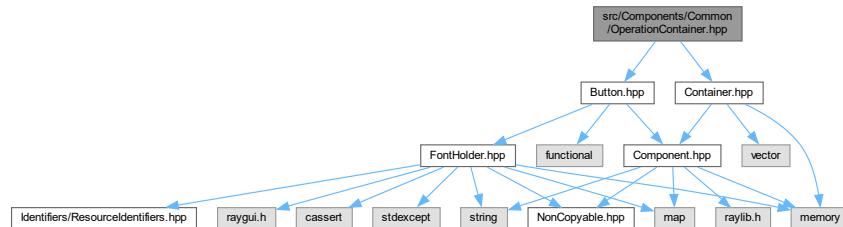
## 7.70 src/Components/Common/OperationContainer.cpp File Reference

```
#include "OperationContainer.hpp"
#include <iostream>
Include dependency graph for OperationContainer.cpp:
```

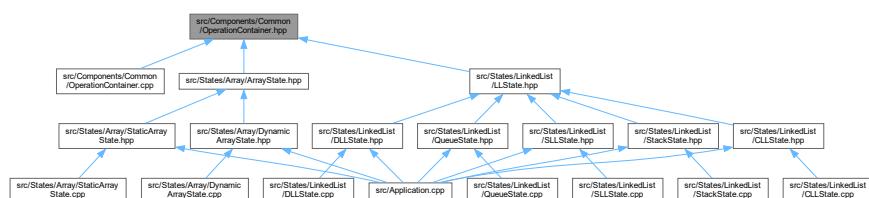


## 7.71 src/Components/Common/OperationContainer.hpp File Reference

```
#include "Button.hpp"
#include "Container.hpp"
Include dependency graph for OperationContainer.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::OperationContainer](#)

## Namespaces

- namespace [GUI](#)

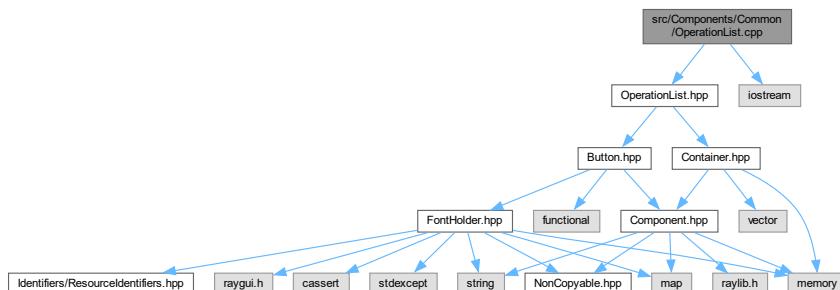
## 7.72 OperationContainer.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef COMPONENTS_OPERATIONCONTAINER_HPP
00002 #define COMPONENTS_OPERATIONCONTAINER_HPP
00003
00004 #include "Button.hpp"
00005 #include "Container.hpp"
00006
00007 namespace GUI {
00008     class OperationContainer : public GUI::Container {
00009     public:
00010         typedef std::shared_ptr< OperationContainer > Ptr;
00011         OperationContainer();
00012         ~OperationContainer();
00013
00014         void DrawCurrent(Vector2 base = (Vector2){0, 0});
00015
00016         void SetVisible(bool visible);
00017
00018     private:
00019         void UpdatePosition();
00020     };
00021 }; // namespace GUI
00022
00023 #endif // COMPONENTS_OPERATIONCONTAINER_HPP
```

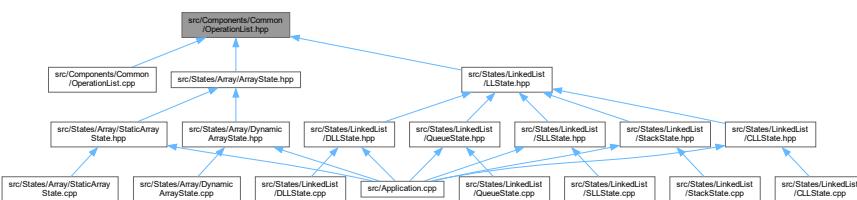
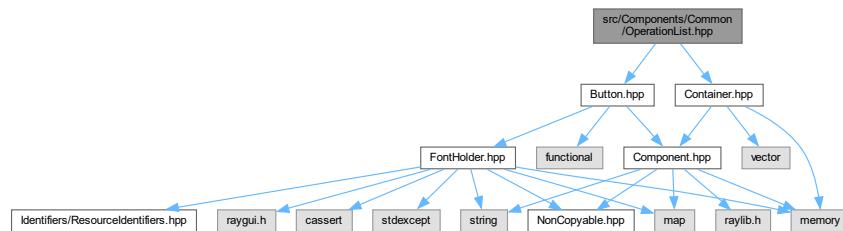
## 7.73 src/Components/Common/OperationList.cpp File Reference

```
#include "OperationList.hpp"
#include <iostream>
Include dependency graph for OperationList.cpp:
```



## 7.74 src/Components/Common/OperationList.hpp File Reference

```
#include "Button.hpp"
#include "Container.hpp"
Include dependency graph for OperationList.hpp:
```



### Classes

- class [GUI::OperationList](#)

### Namespaces

- namespace [GUI](#)

## 7.75 OperationList.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef COMPONENTS_OPERATIONLIST_HPP
00002 #define COMPONENTS_OPERATIONLIST_HPP
00003 #include "Button.hpp"
00004 #include "Container.hpp"
00005
00006 namespace GUI {
00007     class OperationList : public GUI::Container {
00008     private:
00009         GUI::Container buttons;
00010         GUI::Container optionContainers;
00011         bool isHide;
00012         GUI::Button toggleButton;
00013
00014     public:
00015         void ShowOptions(std::size_t index);
  
```

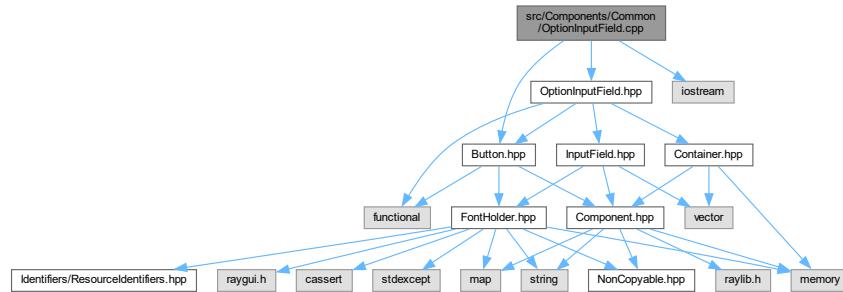
```

00016     void HideAllOptions();
00017     void ToggleOperations();
00018
00019 public:
00020     void Draw(Vector2 base = (Vector2){0, 0});
00021     OperationList();
00022     OperationList(FontHolder *fonts);
00023     ~OperationList();
00024     void AddOperation(GUI::Button::Ptr action,
00025                         GUI::Container::Ptr optionContainer);
00026     void InitActionBar();
00027
00028     Vector2 GetSize();
00029 };
00030 // namespace GUI
00031
00032 #endif // COMPONENTS_OPERATIONLIST_HPP

```

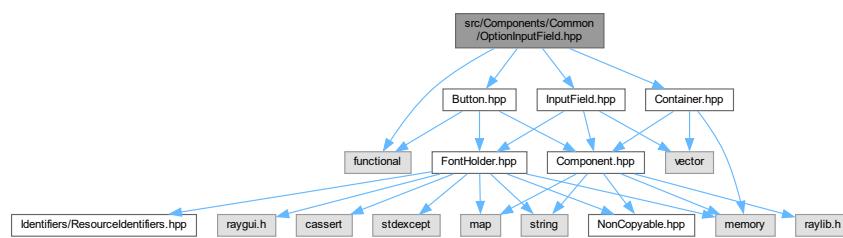
## 7.76 src/Components/Common/OptionInputField.cpp File Reference

```
#include "OptionInputField.hpp"
#include <iostream>
#include "Button.hpp"
Include dependency graph for OptionInputField.cpp:
```

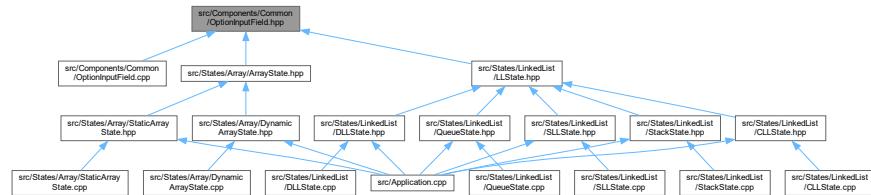


## 7.77 src/Components/Common/OptionInputField.hpp File Reference

```
#include <functional>
#include "Button.hpp"
#include "Container.hpp"
#include "InputField.hpp"
Include dependency graph for OptionInputField.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::OptionInputField](#)

## Namespaces

- namespace [GUI](#)

## 7.78 OptionInputField.hpp

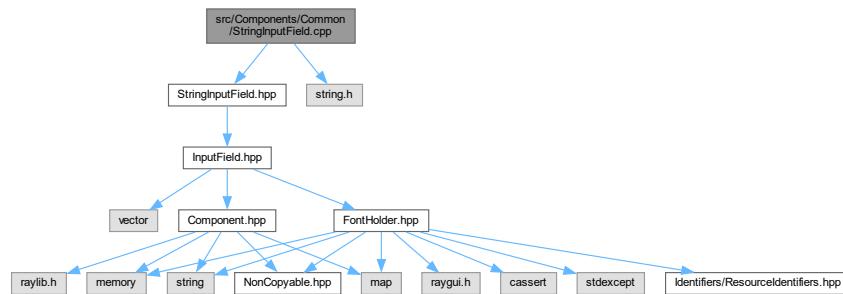
[Go to the documentation of this file.](#)

```

00001 #ifndef COMPONENTS_OPTIONINPUTFIELD_HPP
00002 #define COMPONENTS_OPTIONINPUTFIELD_HPP
00003
00004 #include <functional>
00005
00006 #include "Button.hpp"
00007 #include "Container.hpp"
00008 #include "InputField.hpp"
00009
00010 namespace GUI {
00011     class OptionInputField : public GUI::Container {
00012     public:
00013         typedef std::shared_ptr< OptionInputField > Ptr;
00014
00015     public:
00016         OptionInputField(FontHolder *fonts);
00017         ~OptionInputField();
00018         void SetOption(
00019             std::string content, std::vector< InputField::Ptr > fields,
00020             std::function< void(std::map< std::string, std::string >) > action);
00021         void SetNoFieldOption(std::string content,
00022                               std::function< void() > action);
00023
00024         void ToggleInputFields();
00025
00026         void DrawCurrent(Vector2 base = (Vector2){0, 0});
00027         void SetVisible(bool visible);
00028
00029     public:
00030         virtual Vector2 GetSize();
00031
00032     private:
00033         std::map< std::string, std::string > ExtractInput();
00034         void AddInputField(InputField::Ptr inputField);
00035         void AddSubmitField(
00036             std::function< void(std::map< std::string, std::string >) > action);
00037
00038     private:
00039         bool HasInputField();
00040         GUI::Container::Ptr mInputField;
00041         std::map< std::string, std::string > mInput;
00042
00043     private:
00044         FontHolder *fonts;
00045         bool hasInputField;
00046         // std::function< void(std::map< std::string, std::string >) > mAction;
00047     };
00048 }; // namespace GUI
00049
00050 #endif // COMPONENTS_OPTIONINPUTFIELD_HPP
  
```

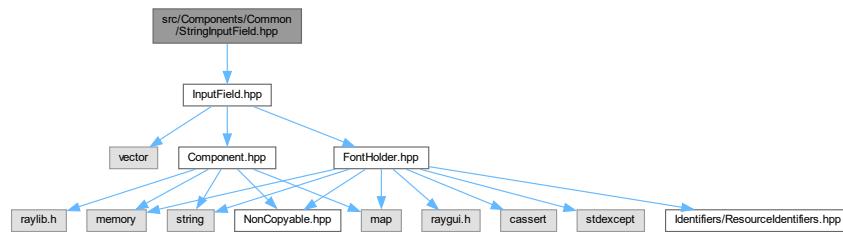
## 7.79 src/Components/Common/StringInputField.cpp File Reference

```
#include "StringInputField.hpp"
#include <string.h>
Include dependency graph for StringInputField.cpp:
```

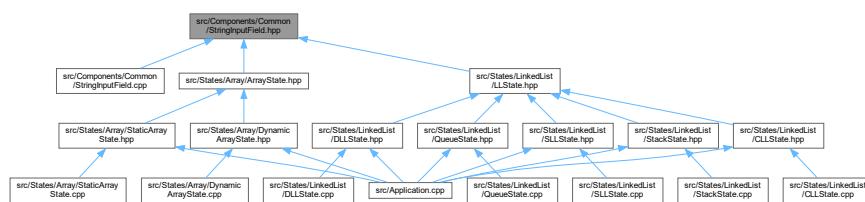


## 7.80 src/Components/Common/StringInputField.hpp File Reference

```
#include "InputField.hpp"
Include dependency graph for StringInputField.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::StringInputField](#)

## Namespaces

- namespace GUI

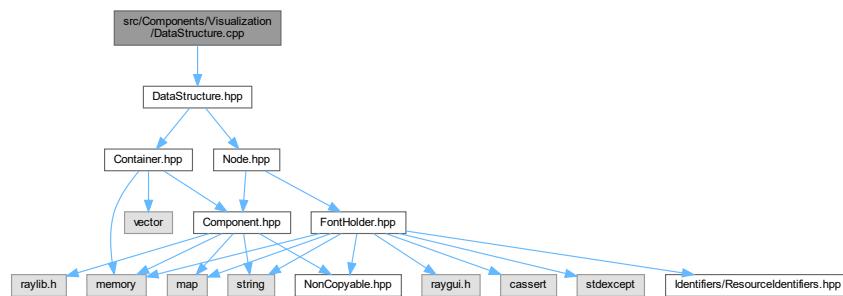
## 7.81 StringInputField.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef COMPONENTS_STRINGINPUTFIELD_HPP
00002 #define COMPONENTS_STRINGINPUTFIELD_HPP
00003
00004 #include "InputField.hpp"
00005
00006 namespace GUI {
00007     class StringInputField : public GUI::InputField {
00008     public:
00009         typedef std::shared_ptr< StringInputField > Ptr;
00010         StringInputField(FontHolder* fonts);
00011         ~StringInputField();
00012         std::string ExtractValue();
00013         void DrawField(Vector2 base = (Vector2){0, 0});
00014         bool IsSelectable() const;
00015
00016     private:
00017         std::size_t maxLength;
00018         std::string content;
00019     };
00020 } // namespace GUI
00021
00022 #endif // COMPONENTS_STRINGINPUTFIELD_HPP
```

## 7.82 src/Components/Visualization/DataStructure.cpp File Reference

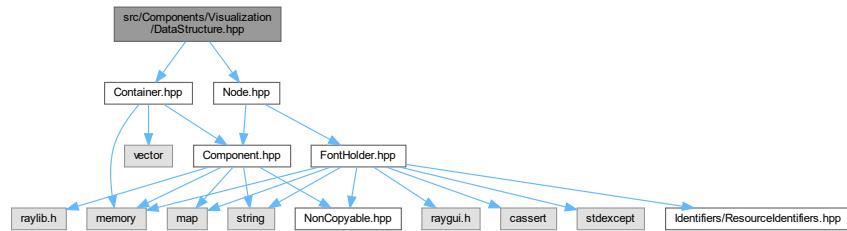
```
#include "DataStructure.hpp"
Include dependency graph for DataStructure.cpp:
```



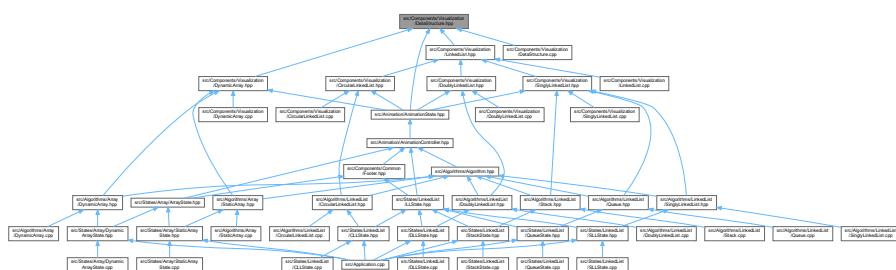
## 7.83 src/Components/Visualization/DataStructure.hpp File Reference

```
#include "Container.hpp"
#include "Node.hpp"
```

Include dependency graph for DataStructure.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::DataStructure](#)

## Namespaces

- namespace [GUI](#)

## 7.84 DataStructure.hpp

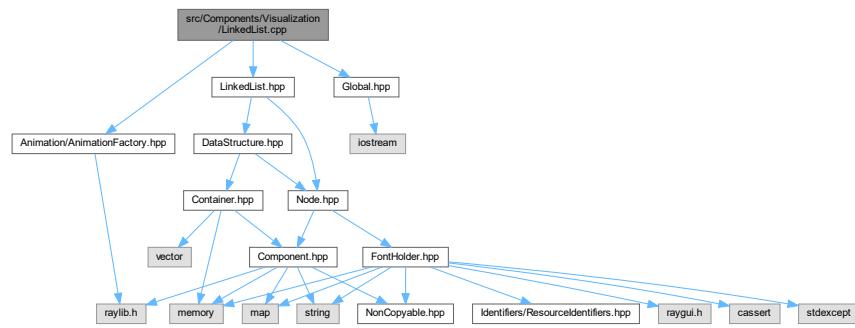
[Go to the documentation of this file.](#)

```

00001 #ifndef COMPONENTS_VISUALIZATION_DATASTRUCTURE_HPP
00002 #define COMPONENTS_VISUALIZATION_DATASTRUCTURE_HPP
00003
00004 #include "Container.hpp"
00005 #include "Node.hpp"
00006
00007 namespace GUI {
00008     class DataStructure : public GUI::Container {
00009     public:
00010         typedef std::shared_ptr< DataStructure > Ptr;
00011
00012     public:
00013         DataStructure();
00014         virtual ~DataStructure();
00015         virtual bool isSelectable() const;
00016         virtual void Draw(Vector2 base);
00017     };
00018 }; // namespace GUI
00019
00020 #endif // COMPONENTS_VISUALIZATION_DATASTRUCTURE_HPP
  
```

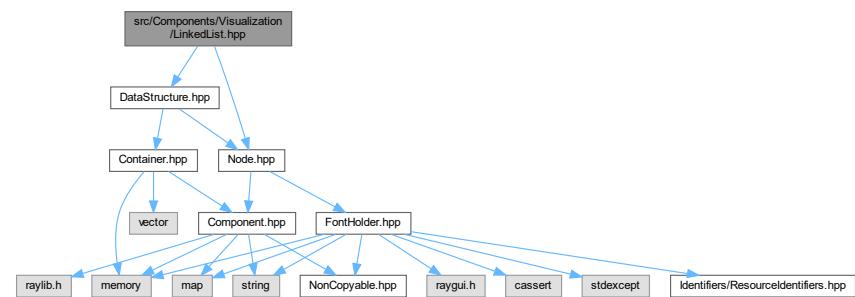
## 7.85 src/Components/Visualization/LinkedList.cpp File Reference

```
#include "LinkedList.hpp"
#include "Animation/AnimationFactory.hpp"
#include "Global.hpp"
Include dependency graph for LinkedList.cpp:
```

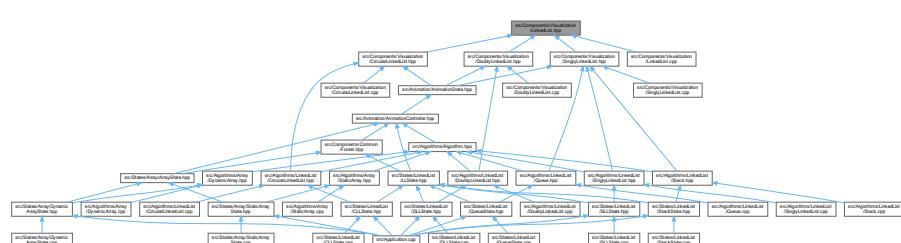


## 7.86 src/Components/Visualization/LinkedList.hpp File Reference

```
#include "DataStructure.hpp"
#include "Node.hpp"
Include dependency graph for LinkedList.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::LinkedList](#)

## Namespaces

- namespace [GUI](#)

## 7.87 LinkedList.hpp

[Go to the documentation of this file.](#)

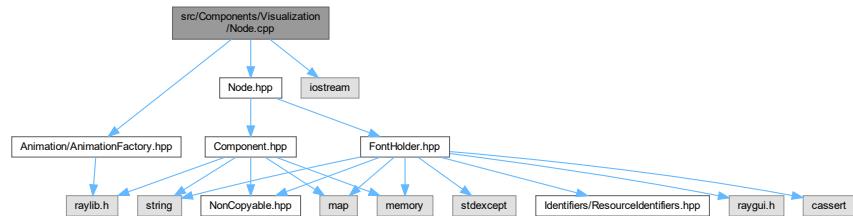
```

00001 #ifndef COMPONENTS_VISUALIZATION_LINKEDLIST_HPP
00002 #define COMPONENTS_VISUALIZATION_LINKEDLIST_HPP
00003
00004 #include "DataStructure.hpp"
00005 #include "Node.hpp"
00006
00007 namespace GUI {
00008     class LinkedList : public GUI::DataStructure {
00009     public:
00010         enum ArrowType {
00011             Default,
00012             Hidden,
00013             Active,
00014             Skip,
00015             ArrowTypeCount,
00016         };
00017         enum Orientation {
00018             Horizontal,
00019             Vertical,
00020             OrientationCount,
00021         };
00022     };
00023
00024     public:
00025         static constexpr float mNodeDistance = 20;
00026
00027     protected:
00028         FontHolder* fonts;
00029         std::vector<GUI::Node> list;
00030         bool mDisplayHeadAndTail;
00031
00032         Orientation mOrientation = Orientation::Horizontal;
00033         GUI::Node::Shape mShape = GUI::Node::Shape::Circle;
00034
00035     public:
00036         void SetShape(GUI::Node::Shape shape);
00037         GUI::Node::Shape GetShape() const;
00038
00039     public:
00040         LinkedList();
00041         LinkedList(FontHolder* fonts);
00042         ~LinkedList();
00043         bool isSelectable() const;
00044         virtual void Draw(Vector2 base = {Vector2{0, 0}, float t = 1.0f,
00045                                         bool init = false}) = 0;
00046         virtual std::size_t size() const;
00047
00048         virtual void SetShowHeadAndTail(bool show);
00049         virtual void SetOrientation(Orientation orientation);
00050
00051     public:
00052         virtual std::vector<GUI::Node>& GetList();
00053         virtual GUI::Node GenerateNode(int value);
00054         virtual void Import(std::vector<int> nodes);
00055         virtual void InsertNode(std::size_t index, GUI::Node node,
00056                                bool rePosition = true);
00057         virtual void DeleteNode(std::size_t index, bool rePosition = true);
00058         virtual void Relayout();
00059
00060     protected:
00061         Vector2 GetNodeDefaultPosition(std::size_t index);
00062     };
00063 } // namespace GUI
00064
00065 #endif // COMPONENTS_VISUALIZATION_LINKEDLIST_HPP

```

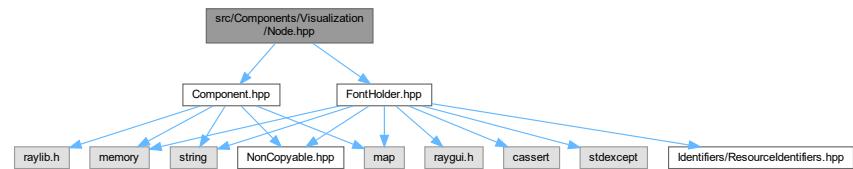
## 7.88 src/Components/Visualization/Node.cpp File Reference

```
#include "Node.hpp"
#include <iostream>
#include "Animation/AnimationFactory.hpp"
Include dependency graph for Node.cpp:
```

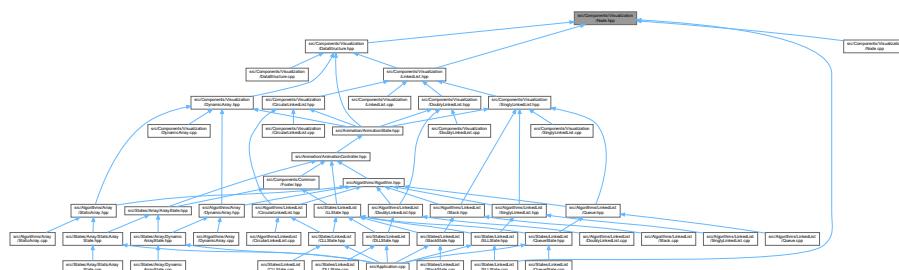


## 7.89 src/Components/Visualization/Node.hpp File Reference

```
#include "Component.hpp"
#include "FontHolder.hpp"
Include dependency graph for Node.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::Node](#)

## Namespaces

- namespace GUI

## 7.90 Node.hpp

[Go to the documentation of this file.](#)

```

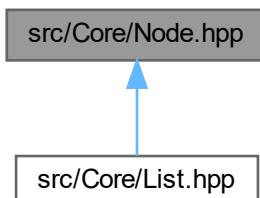
00001
00002 #ifndef COMPONENTS_VISUALIZATION_GUINODE_HPP
00003 #define COMPONENTS_VISUALIZATION_GUINODE_HPP
00004
00005 #include "Component.hpp"
00006 #include "FontHolder.hpp"
00007
00008 namespace GUI {
00009     class Node : public GUI::Component {
00010     public:
00011         enum State {
00012             Default,
00013             Active,
00014             ActiveBlue,
00015             ActiveGreen,
00016             ActiveRed,
00017             Iterated,
00018             Hide,
00019             StateCount,
00020         };
00021
00022         enum Shape {
00023             Circle,
00024             Square,
00025
00026             ShapeCount,
00027         };
00028
00029     private:
00030         Shape mShape = Shape::Circle;
00031
00032     public:
00033         void SetShape(Shape shape);
00034         Shape GetShape() const;
00035
00036     public:
00037         Node(int value, FontHolder* fonts);
00038         Node();
00039         ~Node();
00040         bool IsSelectable() const;
00041         void Draw(Vector2 base = (Vector2){0, 0}, float t = 1.0f);
00042         void SetActive(bool active);
00043         bool IsActive();
00044
00045     public:
00046         void SetValue(int value);
00047         int GetValue() const;
00048         void SetLabel(std::string label);
00049         void ClearLabel();
00050
00051     public:
00052         void AnimationOnNode(bool animate);
00053         void SetRadius(float radius);
00054
00055     private:
00056         void DrawLabel(Vector2 base = (Vector2){0, 0});
00057         void DrawNode(Vector2 base = (Vector2){0, 0}, float t = 1.0f);
00058
00059     public:
00060         void SetValueFontSize(int fontSize);
00061         void SetLabelFontSize(int fontSize);
00062
00063         void SetNodeState(State state);
00064         State GetNodeState() const;
00065
00066     private:
00067         Color GetOutlineColor(float t = 1.0f);
00068         Color GetBackgroundColor(float t = 1.0f);
00069         Color GetTextColor(float t = 1.0f);
00070         void AddColor();
00071
00072     public:
00073         void SetReachable(bool reachable);

```

```
00074     bool GetReachable() const;
00075
00076     private:
00077         std::map< State, std::pair< Color, Color > > mOutlineColor;
00078         std::map< State, std::pair< Color, Color > > mBackgroundColor;
00079         std::map< State, std::pair< Color, Color > > mTextColor;
00080
00081     private:
00082         int mValue;
00083         float mRadius;
00084         float valueFontSize;
00085         float labelFontSize;
00086
00087         std::string mLabel;
00088         State mNodeState;
00089         bool mReachable;
00090
00091     private:
00092         bool animateNode;
00093         bool mActive;
00094         FontHolder* fonts;
00095         Color mDefaultColor;
00096         Color mActiveColor;
00097         Color mBorderColor;
00098     };
00099 }; // namespace GUI
00100
00101 #endif // COMPONENTS_VISUALIZATION_GUINODE_HPP
```

## 7.91 src/Core/Node.hpp File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Core::Node< T >](#)

### Namespaces

- namespace [Core](#)

## 7.92 Node.hpp

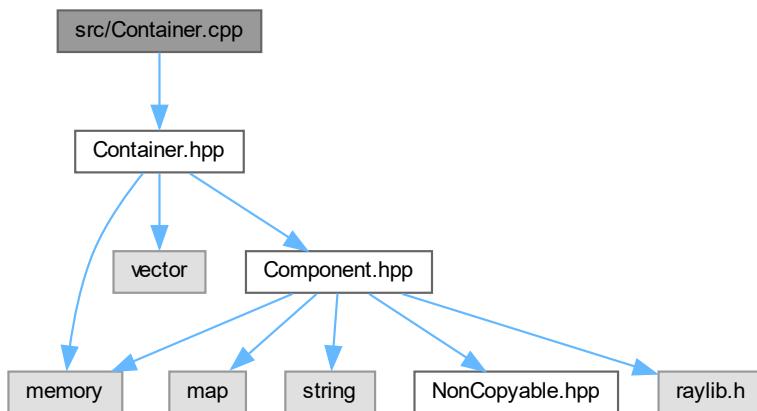
[Go to the documentation of this file.](#)

```
00001 #ifndef CORE_NODE_HPP
00002 #define CORE_NODE_HPP
00003
00004 namespace Core {
00005     template< typename T >
00006     class Node {
00007     public:
00008         T mValue;
00009         Node< T *>*mPrev, *mNext;
00010
00011     public:
00012         Node() : mValue(T()), mPrev(nullptr), mNext(nullptr) {}
00013         Node(const T& value) : mValue(value), mPrev(nullptr), mNext(nullptr) {}
00014         Node(const Node< T >& node)
00015             : mValue(node.mValue), mPrev(node.mPrev), mNext(node.mNext) {}
00016         Node(const T& value, Node< T *>* const prev, Node< T *>* const next)
00017             : mValue(value), mPrev(prev), mNext(next) {}
00018     };
00019 } // namespace Core
00020
00021 #endif // CORE_NODE_HPP
```

## 7.93 src/Container.cpp File Reference

```
#include "Container.hpp"
```

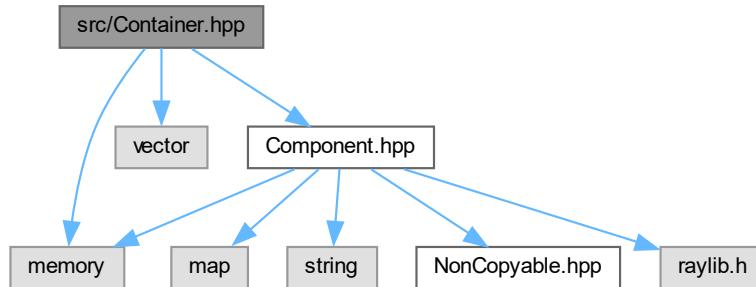
Include dependency graph for Container.cpp:



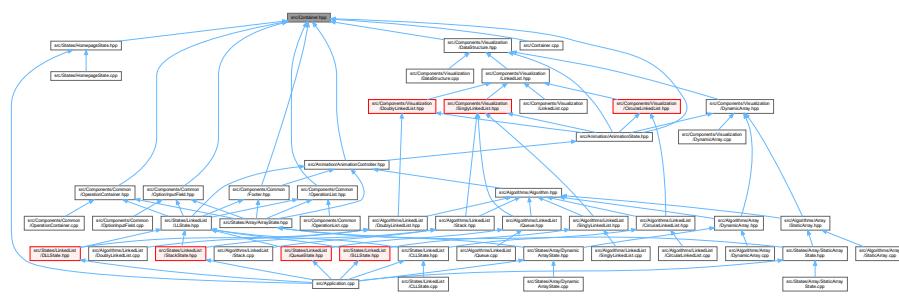
## 7.94 src/Container.hpp File Reference

```
#include <memory>
#include <vector>
```

```
#include "Component.hpp"
Include dependency graph for Container.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [GUI::Container](#)

## Namespaces

- namespace [GUI](#)

## 7.95 Container.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef CONTAINER_HPP
00002 #define CONTAINER_HPP
00003
00004 #include <memory>
00005 #include <vector>
00006
00007 #include "Component.hpp"
00008
00009 namespace GUI {
00010     class Container : public Component {
00011     public:
00012         typedef std::shared_ptr< Container > Ptr;
  
```

```

00013
00014     public:
00015         Container();
00016         void pack(Component::Ptr component);
00017         void UnpackAll();
00018         virtual void Draw(Vector2 base = (Vector2){0, 0});
00019         virtual bool IsSelectable() const;
00020         std::vector< Component::Ptr > GetChildren();
00021
00022         virtual void DrawCurrent(Vector2 base);
00023
00024         virtual Vector2 GetSize();
00025
00026     protected:
00027         std::vector< Component::Ptr > mChildren;
00028     };
00029 }; // namespace GUI
00030
00031 #endif // CONTAINER_HPP

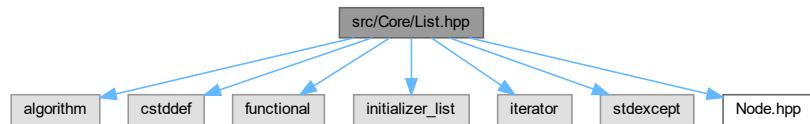
```

## 7.96 src/Core/List.hpp File Reference

```

#include <algorithm>
#include <cstddef>
#include <functional>
#include <initializer_list>
#include <iterator>
#include <stdexcept>
#include "Node.hpp"
Include dependency graph for List.hpp:

```



## Classes

- class [Core::List< T >](#)
- class [Core::List< T >::iterator](#)
- class [Core::List< T >::const\\_iterator](#)

## Namespaces

- namespace [Core](#)

## 7.97 List.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef CORE_LIST_HPP
00002 #define CORE_LIST_HPP
00003
00004 // #define _SILENCE_ALL_CXX17_DEPRECATED_WARNINGS
00005 // #define _SILENCE_CXX17_ITERATOR_BASE_CLASS_DEPRECATED_WARNING
00006
00007 #include <algorithm>
00008 #include <cstddef>
00009 #include <functional>
00010 #include <initializer_list>
00011 #include <iterator>
00012 #include <stdexcept>
00013
00014 #include "Node.hpp"
00015
00016 namespace Core {
00017     template< typename T >
00018     class List {
00019     public:
00020         class iterator;
00021         class const_iterator;
00022
00023     private:
00024         iterator mBegin, mEnd;
00025         std::size_t mSize;
00026
00027         void reset() {
00028             this->mBegin = this->mEnd;
00029             this->mEnd.ptr->mPrev = nullptr;
00030             this->mSize = 0;
00031         }
00032
00033         void insert_previous(const iterator& it, const iterator& it_prev) {
00034             Node< T *> node = it.ptr;
00035             Node< T *> node_prev = it_prev.ptr;
00036
00037             node_prev->mNext = node;
00038             node_prev->mPrev = node->mPrev;
00039             node->mPrev = node_prev;
00040             if (node_prev->mPrev != nullptr) {
00041                 node_prev->mPrev->mNext = node_prev;
00042             }
00043
00044             if (it == this->begin()) {
00045                 this->mBegin = node_prev;
00046             }
00047
00048             ++this->mSize;
00049         }
00050
00051         /* @brief Move nodes from a list of range ['first', 'last') to before
00052         * 'it' */
00053         /* @brief Return iterator pointing to the first inserted element, or
00054         * 'it' if 'first' == 'last'. */
00055         /* @exception: undefined behaviour: null pointer reference */
00056         iterator move_previous(const iterator& it, const iterator& first,
00057                                const iterator& last) {
00058             mSize += std::distance(first, last);
00059
00060             iterator prev = it;
00061             prev--;
00062
00063             if (first != last) {
00064                 Node< T *> node = it.ptr;
00065                 Node< T *> node_first = first.ptr;
00066                 Node< T *> node_last = last.ptr->mPrev;
00067
00068                 node_first->mPrev = node->mPrev;
00069                 if (node->mPrev != nullptr) {
00070                     node->mPrev->mNext = node_first;
00071                 }
00072
00073                 node_last->mNext = node;
00074                 node->mPrev = node_last;
00075
00076                 if (it == this->begin()) {
00077                     this->mBegin = node_first;
00078                 }
00079             }
00080
00081             return prev == nullptr ? this->begin() : prev++;
00082         }
00083     }
```

```

00083     iterator get_iterator(std::size_t index) {
00084         if (!(index >= 0 && index < this->size())) {
00085             throw std::out_of_range("Index out of range");
00086         }
00087
00088         iterator it;
00089         if (index < this->size() / 2) {
00090             it = this->begin();
00091             for (std::size_t i = 0; i < index; ++i) ++it;
00092         } else {
00093             it = this->end();
00094             for (std::size_t i = this->size() - 1; i > index; --i) --it;
00095         }
00096         return it;
00097     }
00098
00099
00100     public:
00101         List() : mSize(0) { this->mBegin = this->mEnd = new Node< T >(); }
00102         List(std::initializer_list< T > list) : mSize(0) {
00103             this->mBegin = this->mEnd = new Node< T >();
00104             for (const auto& element : list) {
00105                 this->push_back(element);
00106             }
00107         }
00108
00109         /* @brief Copy constructor */
00110         List(const List< T >& list) : mSize(0) {
00111             this->mBegin = this->mEnd = new Node< T >();
00112             for (const auto& element : list) {
00113                 this->push_back(element);
00114             }
00115         }
00116
00117         /* @brief Move constructor */
00118         List(List< T >&& list) : mSize(0) {
00119             this->mBegin = this->mEnd = new Node< T >();
00120             move_previous(this->end(), list.begin(), list.end());
00121             list.reset();
00122         }
00123         List< T >& operator=(const List< T >&& list) {
00124             if (this != &list) {
00125                 this->clear();
00126                 move_previous(this->end(), list.begin(), list.end());
00127                 list.reset();
00128             }
00129             return *this;
00130         }
00131
00132     public:
00133         iterator begin() { return mBegin; }
00134         const_iterator begin() const { return mBegin; }
00135         iterator end() { return mEnd; }
00136         const_iterator end() const { return mEnd; }
00137         bool empty() const { return this->size() == 0; }
00138         std::size_t size() const { return this->mSize; }
00139
00140
00141     public:
00142         /* Exception(s): undefined behavior: null pointer dereference */
00143         T& front() { return *this->begin(); }
00144         const T& front() const { return *this->begin(); }
00145         /* Exception(s): undefined */
00146         T& back() {
00147             auto it = this->end();
00148             --it;
00149             return *it;
00150         }
00151         const T& back() const {
00152             auto it = this->end();
00153             --it;
00154             return *it;
00155         }
00156         T& operator[](std::size_t index) { return *this->get_iterator(index); }
00157         const T& operator[](std::size_t index) const {
00158             return *this->get_iterator(index);
00159         }
00160
00161         T& at(std::size_t index) { return (*this)[index]; }
00162
00163         const T& at(std::size_t index) const { return (*this)[index]; }
00164
00165         void push_front(const T& value) {
00166             Node< T >* node = new Node< T >(value);
00167             this->insert_previous(this->begin(), iterator(node));
00168         }
00169

```

```

00170     void push_back(const T& value) {
00171         Node< T >* node = new Node< T >(value);
00172         this->insert_previous(this->end(), iterator(node));
00173     }
00174
00175     void pop_front() {
00176         auto it = this->begin();
00177         this->remove(it);
00178     }
00179
00180     void pop_back() {
00181         auto it = --this->end();
00182         this->remove(it);
00183     }
00184
00185 public:
00186     iterator remove(const iterator& it) {
00187         if (it == this->end())
00188             throw std::out_of_range("Iterator out of range");
00189     }
00190
00191     Node< T >* node = it.ptr;
00192     Node< T >* node_prev = node->mPrev;
00193     Node< T >* node_next = node->mNext;
00194
00195     if (node_prev != nullptr) node_prev->mNext = node_next;
00196
00197     if (node_next != nullptr) node_next->mPrev = node_prev;
00198
00199     if (it == this->begin()) {
00200         this->mBegin = node_next;
00201     }
00202
00203     delete node;
00204     --this->mSize;
00205     return node_next;
00206 }
00207
00208 /**
00209  * @brief Return resulting size
00210 */
00211 int remove(const T& value, const iterator& begin, const iterator& end) {
00212     return this->remove_if(
00213         [&value](const T& element) { return element == value; }, begin,
00214         end);
00215 }
00216
00217 /**
00218  * @brief Return resulting size
00219 */
00220 int remove(const T& value) {
00221     return this->remove_if(value, this->begin(), this->end());
00222 }
00223
00224 /**
00225  * @brief Removes all elements satisfying specific criteria from the
00226  * range [first, last) and returns a past-the-end iterator for the new
00227  * end of the range.
00228  * @return the resulting size
00229 */
00230 int remove_if(std::function< bool(const T&) > predicate,
00231               const iterator& begin, const iterator& end) {
00232     for (auto it = begin; it != end; ) {
00233         if (predicate(*it)) {
00234             it = this->remove(it);
00235         } else {
00236             ++it;
00237         }
00238     }
00239     return this->size();
00240 }
00241
00242 /**
00243  * @brief Removes all elements satisfying specific criteria from the
00244  * range [first, last) and returns a past-the-end iterator for the new
00245  * end of the range.
00246  * @return the resulting size
00247 */
00248 int remove_if(std::function< bool(const T&) > predicate) {
00249     return this->remove_if(predicate, this->begin(), this->end());
00250 }
00251
00252 public:
00253     void clear() {
00254         if (!this->empty()) {
00255             for (auto it = this->begin(); it != this->end(); ) {
00256                 Node< T >* node = it.ptr;
00257
00258
00259
00260
00261
00262
00263
00264
00265
00266
00267
00268
00269
00270
00271
00272
00273
00274
00275
00276
00277
00278
00279
00280
00281
00282
00283
00284
00285
00286
00287
00288
00289
00290
00291
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302
00303
00304
00305
00306
00307
00308
00309
00310
00311
00312
00313
00314
00315
00316
00317
00318
00319
00320
00321
00322
00323
00324
00325
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335
00336
00337
00338
00339
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350
00351
00352
00353
00354
00355
00356
00357
00358
00359
00360
00361
00362
00363
00364
00365
00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395
00396
00397
00398
00399
00400
00401
00402
00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467
00468
00469
00470
00471
00472
00473
00474
00475
00476
00477
00478
00479
00480
00481
00482
00483
00484
00485
00486
00487
00488
00489
00490
00491
00492
00493
00494
00495
00496
00497
00498
00499
00500
00501
00502
00503
00504
00505
00506
00507
00508
00509
00510
00511
00512
00513
00514
00515
00516
00517
00518
00519
00520
00521
00522
00523
00524
00525
00526
00527
00528
00529
00530
00531
00532
00533
00534
00535
00536
00537
00538
00539
00540
00541
00542
00543
00544
00545
00546
00547
00548
00549
00550
00551
00552
00553
00554
00555
00556
00557
00558
00559
00560
00561
00562
00563
00564
00565
00566
00567
00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00590
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600
00601
00602
00603
00604
00605
00606
00607
00608
00609
00610
00611
00612
00613
00614
00615
00616
00617
00618
00619
00620
00621
00622
00623
00624
00625
00626
00627
00628
00629
00630
00631
00632
00633
00634
00635
00636
00637
00638
00639
00640
00641
00642
00643
00644
00645
00646
00647
00648
00649
00650
00651
00652
00653
00654
00655
00656
00657
00658
00659
00660
00661
00662
00663
00664
00665
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681
00682
00683
00684
00685
00686
00687
00688
00689
00690
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705
00706
00707
00708
00709
00710
00711
00712
00713
00714
00715
00716
00717
00718
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728
00729
00730
00731
00732
00733
00734
00735
00736
00737
00738
00739
00740
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757
00758
00759
00760
00761
00762
00763
00764
00765
00766
00767
00768
00769
00770
00771
00772
00773
00774
00775
00776
00777
00778
00779
00780
00781
00782
00783
00784
00785
00786
00787
00788
00789
00790
00791
00792
00793
00794
00795
00796
00797
00798
00799
00800
00801
00802
00803
00804
00805
00806
00807
00808
00809
00810
00811
00812
00813
00814
00815
00816
00817
00818
00819
00820
00821
00822
00823
00824
00825
00826
00827
00828
00829
00830
00831
00832
00833
00834
00835
00836
00837
00838
00839
00840
00841
00842
00843
00844
00845
00846
00847
00848
00849
00850
00851
00852
00853
00854
00855
00856
00857
00858
00859
00860
00861
00862
00863
00864
00865
00866
00867
00868
00869
00870
00871
00872
00873
00874
00875
00876
00877
00878
00879
00880
00881
00882
00883
00884
00885
00886
00887
00888
00889
00890
00891
00892
00893
00894
00895
00896
00897
00898
00899
00900
00901
00902
00903
00904
00905
00906
00907
00908
00909
00910
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00930
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00990
00991
00992
00993
00994
00995
00996
00997
00998
00999
01000
01001
01002
01003
01004
01005
01006
01007
01008
01009
01010
01011
01012
01013
01014
01015
01016
01017
01018
01019
01020
01021
01022
01023
01024
01025
01026
01027
01028
01029
01030
01031
01032
01033
01034
01035
01036
01037
01038
01039
01040
01041
01042
01043
01044
01045
01046
01047
01048
01049
01050
01051
01052
01053
01054
01055
01056
01057
01058
01059
01060
01061
01062
01063
01064
01065
01066
01067
01068
01069
01070
01071
01072
01073
01074
01075
01076
01077
01078
01079
01080
01081
01082
01083
01084
01085
01086
01087
01088
01089
01090
01091
01092
01093
01094
01095
01096
01097
01098
01099
01100
01101
01102
01103
01104
01105
01106
01107
01108
01109
01110
01111
01112
01113
01114
01115
01116
01117
01118
01119
01120
01121
01122
01123
01124
01125
01126
01127
01128
01129
01130
01131
01132
01133
01134
01135
01136
01137
01138
01139
01140
01141
01142
01143
01144
01145
01146
01147
01148
01149
01150
01151
01152
01153
01154
01155
01156
01157
01158
01159
01160
01161
01162
01163
01164
01165
01166
01167
01168
01169
01170
01171
01172
01173
01174
01175
01176
01177
01178
01179
01180
01181
01182
01183
01184
01185
01186
01187
01188
01189
01190
01191
01192
01193
01194
01195
01196
01197
01198
01199
01200
01201
01202
01203
01204
01205
01206
01207
01208
01209
01210
01211
01212
01213
01214
01215
01216
01217
01218
01219
01220
01221
01222
01223
01224
01225
01226
01227
01228
01229
01230
01231
01232
01233
01234
01235
01236
01237
01238
01239
01240
01241
01242
01243
01244
01245
01246
01247
01248
01249
01250
01251
01252
01253
01254
01255
01256
01257
01258
01259
01260
01261
01262
01263
01264
01265
01266
01267
01268
01269
01270
01271
01272
01273
01274
01275
01276
01277
01278
01279
01280
01281
01282
01283
01284
01285
01286
01287
01288
01289
01290
01291
01292
01293
01294
01295
01296
01297
01298
01299
01300
01301
01302
01303
01304
01305
01306
01307
01308
01309
01310
01311
01312
01313
01314
01315
01316
01317
01318
01319
01320
01321
01322
01323
01324
01325
01326
01327
01328
01329
01330
01331
01332
01333
01334
01335
01336
01337
01338
01339
01340
01341
01342
01343
01344
01345
01346
01347
01348
01349
01350
01351
01352
01353
01354
01355
01356
01357
01358
01359
01360
01361
01362
01363
01364
01365
01366
01367
01368
01369
01370
01371
01372
01373
01374
01375
01376
01377
01378
01379
01380
01381
01382
01383
01384
01385
01386
01387
01388
01389
01390
01391
01392
01393
01394
01395
01396
01397
01398
01399
01400
01401
01402
01403
01404
01405
01406
01407
01408
01409
01410
01411
01412
01413
01414
01415
01416
01417
01418
01419
01420
01421
01422
01423
01424
01425
01426
01427
01428
01429
01430
01431
01432
01433
01434
01435
01436
01437
01438
01439
01440
01441
01442
01443
01444
01445
01446
01447
01448
01449
01450
01451
01452
01453
01454
01455
01456
01457
01458
01459
01460
01461
01462
01463
01464
01465
01466
01467
01468
01469
01470
01471
01472
01473
01474
01475
01476
01477
01478
01479
01480
01481
01482
01483
01484
01485
01486
01487
01488
01489
01490
01491
01492
01493
01494
01495
01496
01497
01498
01499
01500
01501
01502
01503
01504
01505
01506
01507
01508
01509
01510
01511
01512
01513
01514
01515
01516
01517
01518
01519
01520
01521
01522
01523
01524
01525
01526
01527
01528
01529
01530
01531
01532
01533
01534
01535
01536
01537
01538
01539
01540
01541
01542
01543
01544
01545
01546
01547
01548
01549
01550
01551
01552
01553
01554
01555
01556
01557
01558
01559
01560
01561
01562
01563
01564
01565
01566
01567
01568
01569
01570
01571
01572
01573
01574
01575
01576
01577
01578
01579
01580
01581
01582
01583
01584
01585
01586
01587
01588
01589
01590
01591
01592
01593
01594
01595
01596
01597
01598
01599
01600
01601
01602
01603
01604
01605
01606
01607
01608
01609
01610
01611
01612
01613
01614
01615
01616
01617
01618
01619
01620
01621
01622
01623
01624
01625
01626
01627
01628
01629
01630
01631
01632
01633
01634
01635
01636
01637
01638
01639
01640
01641
01642
01643
01644
01645
01646
01647
01648
01649
01650
01651
01652
01653
01654
01655
01656
01657
01658
01659
01660
01661
01662
01663
01664
01665
01666
01667
01668
01669
01670
01671
01672
01673
01674
01675
01676
01677
01678
01679
01680
01681
01682
01683
01684
01685
01686
01687
01688
01689
01690
01691
01692
01693
01694
01695
01696
01697
01698
01699
01700
01701
01702
01703
01704
01705
01706
01707
01708
01709
01710
01711
01712
01713
01714
01715
01716
01717
01718
01719
01720
01721
01722
01723
01724
01725
01726
01727
01728
01729
01730
01731
01732
01733
01734
01735
01736
01737
01738
01739
01740
01741
01742
01743
01744
01745
01746
01747
01748
01749
01750
01751
01752
01753
01754
01755
01756
01757
01758
01759
01760
01761
01762
01763
01764
01765
01766
01767
01768
01769
01770
01771
01772
01773
01774
01775
01776
01777
01778
01779
01780
01781
01782
01783
01784
01785
01786
01787
01788
01789
01790
01791
01792
01793
01794
01795
01796
01797
01798
01799
01800
01801
01802
01803
01804
01805
01806
01807
01808
01809
01810
01811
01812
01813
01814
01815
01816
01817
01818
01819
01820
01821
01822
01823
01824
01825
01826
01827
01828
01829
01830
01831
01832
01833
01834
01835
01836
01837
01838
01839
01840
01841
01842
01843
01844
01845
01846
01847
01848
01849
01850
01851
01852
01853
01854
01855
01856
01857
01858
01859
01860
01861
01862
01863
01864
01865
01866
01867
01868
01869
01870
01871
01872
01873
01874
01875
01876
01877
01878
01879
01880
01881
01882
01883
01884
01885
01886
01887
01888
01889
01890
01891
01892
01893
01894
01895
01896
01897
01898
01899
01900
01901
01902
01903
01904
01905
01906
01907
01908
01909
01910
01911
01912
01913
01914
01915
01916
01917
01918
01919
01920
01
```

```

00257             ++it;
00258             delete node;
00259         }
00260     }
00261     this->reset();
00262 }
00263 };
00264 } // namespace Core
00265
00266 namespace Core {
00267     template< typename T >
00268     class List< T >::iterator
00269     // : public std::iterator< std::bidirectional_iterator_tag, T,
00270     //                           std::ptrdiff_t, T*, T& >
00271     // deprecated since C++17
00272     {
00273         friend class List;
00274
00275     private:
00276         Node< T >* ptr;
00277
00278     public:
00279         using iterator_category = std::bidirectional_iterator_tag;
00280         using value_type = T;
00281         using difference_type = std::ptrdiff_t;
00282         using pointer = T*;
00283         using reference = T&;
00284
00285         iterator() : ptr{nullptr} {}
00286         iterator(Node< value_type >* const p) : ptr(p) {}
00287         reference operator*() const {
00288             reference value = ptr->mValue;
00289             return value;
00290         }
00291
00292         pointer operator->() const { return std::addressof(operator*()); }
00293         /* Prefix increment */
00294         iterator& operator++() {
00295             ptr = ptr->mNext;
00296             return *this;
00297         }
00298         /* Postfix increment */
00299         iterator operator++(int) {
00300             iterator tmp = *this;
00301             ++(*this);
00302             return tmp;
00303         }
00304         /* Prefix decrement */
00305         iterator& operator--() {
00306             ptr = ptr->mPrev;
00307             return *this;
00308         }
00309         /* Postfix decrement */
00310         iterator operator--(int) {
00311             iterator tmp = *this;
00312             --(*this);
00313             return tmp;
00314         }
00315         iterator& operator=(Node< value_type >* const& p) {
00316             ptr = p;
00317             return *this;
00318         }
00319         iterator operator+(const int& step) {
00320             iterator it = *this;
00321             if (step > 0) {
00322                 for (int i = 0; i < step; ++i) ++it;
00323             }
00324             if (step < 0) {
00325                 for (int i = 0; i < -step; ++i) --it;
00326             }
00327             return it;
00328         }
00329         iterator operator-(const int& step) {
00330             iterator it = *this;
00331             if (step > 0) {
00332                 for (int i = 0; i < step; ++i) --it;
00333             }
00334             if (step < 0) {
00335                 for (int i = 0; i < -step; ++i) ++it;
00336             }
00337             return it;
00338         }
00339         bool operator==(const iterator& it) const { return it.ptr == ptr; }
00340         bool operator!=(const iterator& it) const { return it.ptr != ptr; }
00341         void swap(iterator& other) { std::swap(this->ptr, other.ptr); }
00342     };
00343 }
```

```
00344     template< class T >
00345     class List< T >::const_iterator
00346     // : public std::iterator< std::bidirectional_iterator_tag, T,
00347     //                           std::ptrdiff_t, T*, T& >
00348     // deprecated since C++17
00349     {
00350         friend class List;
00351
00352     private:
00353         const Node< T *>* ptr;
00354
00355     public:
00356         using iterator_category = std::bidirectional_iterator_tag;
00357         using value_type = T;
00358         using difference_type = std::ptrdiff_t;
00359         using pointer = const T*;
00360         using reference = const T&;
00361
00362         const_iterator() : ptr(nullptr) {}
00363         const_iterator(Node< value_type *>* const& p) : ptr(p) {}
00364         const_iterator(const iterator& other) : ptr(other.ptr) {}
00365         const_iterator(const const_iterator& other) : ptr(other.ptr) {}
00366         reference operator*() const {
00367             reference value = ptr->value;
00368             return value;
00369         }
00370
00371         pointer operator->() const { return std::addressof(operator*()); }
00372
00373         /* Prefix increment */
00374         const_iterator& operator++() {
00375             ptr = ptr->next;
00376             return *this;
00377         }
00378
00379         /* Postfix increment */
00380         const_iterator operator++(int) {
00381             const_iterator tmp = *this;
00382             ++(*this);
00383             return tmp;
00384         }
00385
00386         /* Prefix decrement */
00387         const_iterator& operator--() {
00388             ptr = ptr->prev;
00389             return *this;
00390         }
00391
00392         /* Postfix decrement */
00393         const_iterator operator--(int) {
00394             const_iterator tmp = *this;
00395             --(*this);
00396             return tmp;
00397         }
00398         const_iterator operator+(const int& step) {
00399             const_iterator it = *this;
00400             if (step > 0) {
00401                 for (int i = 0; i < step; ++i) ++it;
00402             }
00403             if (step < 0) {
00404                 for (int i = 0; i < -step; ++i) --it;
00405             }
00406             return it;
00407         }
00408         const_iterator operator-(const int& step) {
00409             const_iterator it = *this;
00410             if (step > 0) {
00411                 for (int i = 0; i < step; ++i) --it;
00412             }
00413             if (step < 0) {
00414                 for (int i = 0; i < -step; ++i) ++it;
00415             }
00416             return it;
00417         }
00418         const_iterator& operator=(Node< value_type *>* const& p) {
00419             ptr = p;
00420             return *this;
00421         }
00422         const_iterator& operator=(const iterator& other) {
00423             ptr = other.ptr;
00424             return *this;
00425         }
00426         bool operator==(const const_iterator& it) const {
00427             return it.ptr == ptr;
00428         }
00429         bool operator!=(const const_iterator& it) const {
00430             return it.ptr != ptr;
```

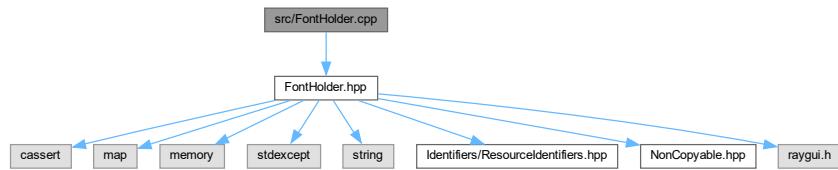
```

00431         }
00432         void swap(const_iterator& other) { std::swap(this->ptr, other.ptr); }
00433     };
00434 } // namespace Core
00435
00436 #endif // CORE_LIST_HPP

```

## 7.98 src/FontHolder.cpp File Reference

#include "FontHolder.hpp"  
Include dependency graph for FontHolder.cpp:



## 7.99 src/FontHolder.hpp File Reference

#include <cassert>
#include <map>
#include <memory>
#include <stdexcept>
#include <string>
#include "Identifiers/ResourceIdentifiers.hpp"
#include "NonCopyable.hpp"
#include "raygui.h"

Include dependency graph for FontHolder.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [FontHolder](#)

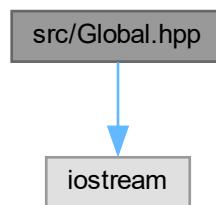
## 7.100 FontHolder.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef FONTHOLDER_HPP
00002 #define FONTHOLDER_HPP
00003
00004 #include <cassert>
00005 #include <map>
00006 #include <memory>
00007 #include <stdexcept>
00008 #include <string>
00009
00010 #include "Identifiers/ResourceIdentifiers.hpp"
00011 #include "NonCopyable.hpp"
00012 #include "raygui.h"
00013
00014 class FontHolder : private NonCopyable<FontHolder> {
00015 public:
00016     FontHolder();
00017     ~FontHolder();
00018     void Load(Fonts::ID id, const std::string& filename);
00019     Font& Get(Fonts::ID id);
00020     const Font& Get(Fonts::ID id) const;
00021
00022 private:
00023     void InsertResource(Fonts::ID id, std::unique_ptr<Font> font);
00024
00025 private:
00026     std::map<Fonts::ID, std::unique_ptr<Font>> mFontMap;
00027 };
00028 #endif // FONTHOLDER_HPP
```

## 7.101 src/Global.hpp File Reference

```
#include <iostream>
Include dependency graph for Global.hpp:
```



This graph shows which files directly or indirectly include this file:



## Namespaces

- namespace [global](#)

# Variables

- `constexpr int global::SCREEN_WIDTH = 1300`
  - `constexpr int global::SCREEN_HEIGHT = 800`
  - `const std::string global::kTitle = "CS162 - VisuAlgo Clone"`
  - `const std::string global::favicon = "assets/images/favicon.png"`
  - `const std::string global::defaultColorPath = "assets/default-theme.dat"`

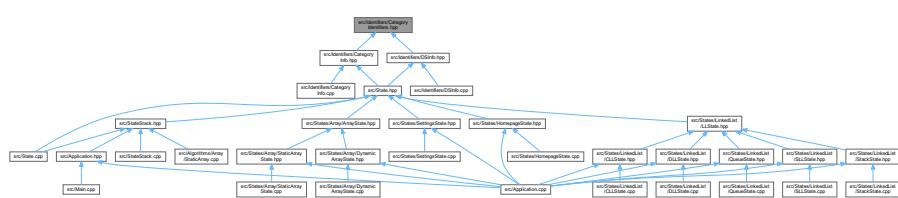
## 7.102 Global.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include <iostream>
00004
00005 // Global variables
00006 namespace global {
00007     constexpr int SCREEN_WIDTH = 1300;
00008     constexpr int SCREEN_HEIGHT = 800;
00009     // constexpr int kFramesPerSecond = 120;
00010     const std::string kTitle = "CS162 - VisuAlgo Clone";
00011
00012     const std::string favicon = "assets/images/favicon.png";
00013
00014     const std::string defaultColorPath = "assets/default-theme.dat";
00015 } // namespace global
```

## 7.103 src/Identifiers/CategoryIdentifiers.hpp File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- namespace Category

## Enumerations

- ```
• enum Category::ID { Category::None , Category::Array , Category::LinkedList , Category::Count }
```

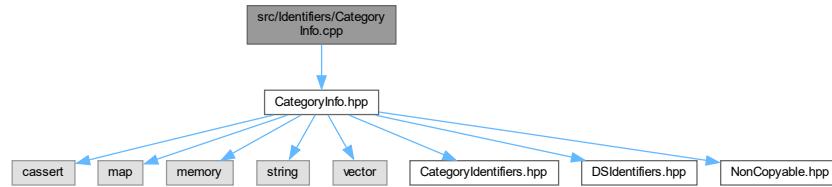
## 7.104 CategoryIdentifiers.hpp

[Go to the documentation of this file.](#)

```
Go to the documentation of this file.  
00001 #ifndef CATEGORYIDENTIFIERS_HPP  
00002 #define CATEGORYIDENTIFIERS_HPP  
00003  
00004 namespace Category {  
00005     enum ID { None, Array, LinkedList, Count };  
00006 };  
00007  
00008 #endif // CATEGORYIDENTIFIERS_HPP
```

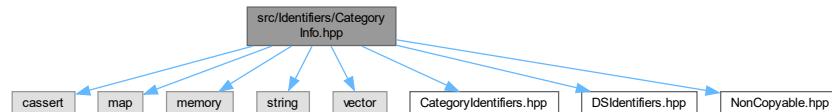
## 7.105 src/Identifiers/CategoryInfo.cpp File Reference

```
#include "CategoryInfo.hpp"
Include dependency graph for CategoryInfo.cpp:
```

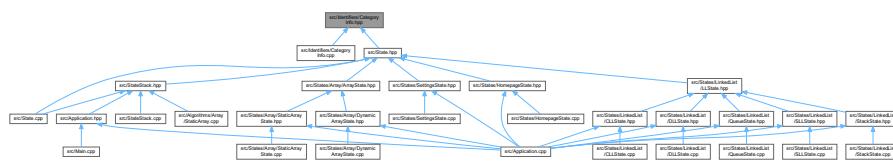


## 7.106 src/Identifiers/CategoryInfo.hpp File Reference

```
#include <cassert>
#include <map>
#include <memory>
#include <string>
#include <vector>
#include "CategoryIdentifiers.hpp"
#include "DSIdentifiers.hpp"
#include "NonCopyable.hpp"
Include dependency graph for CategoryInfo.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [CategoryInfo](#)
- struct [CategoryInfo::Info](#)

## 7.107 CategoryInfo.hpp

[Go to the documentation of this file.](#)

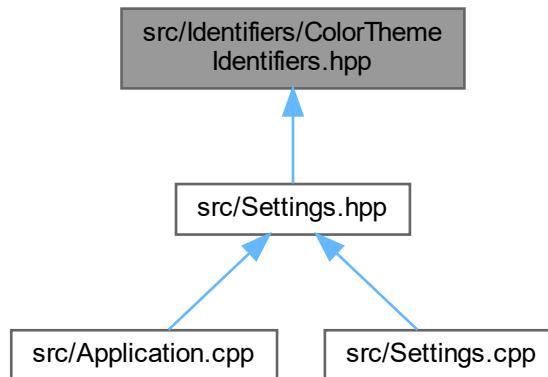
```

00001 #ifndef CATEGORYINFO_HPP
00002 #define CATEGORYINFO_HPP
00003
00004 #include <cassert>
00005 #include <map>
00006 #include <memory>
00007 #include <string>
00008 #include <vector>
00009
00010 #include "CategoryIdentifiers.hpp"
00011 #include "DSIIdentifiers.hpp"
00012 #include "NonCopyable.hpp"
00013
00014 class CategoryInfo : private NonCopyable< CategoryInfo > {
00015 private:
00016     struct Info {
00017         Info(Category::ID categoryId, std::vector< DataStructures::ID > mDS,
00018               std::string name);
00019         Category::ID categoryId;
00020         std::string categoryName;
00021         std::vector< DataStructures::ID > mDS;
00022     };
00023     std::map< Category::ID, Info > mFactories;
00024
00025 public:
00026     void Register(Category::ID, Info info);
00027     const Info& Get(Category::ID id) const;
00028 };
00029
00030 #endif // CATEGORYINFO_HPP

```

## 7.108 src/Identifiers/ColorThemeIdentifiers.hpp File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- namespace [ColorTheme](#)

## Enumerations

- enum ColorTheme::ID {
   
 ColorTheme::Background , ColorTheme::Logo1FirstPart , ColorTheme::Logo1SecondPart , ColorTheme::Logo2FirstPart
   
 ,
   
 ColorTheme::Logo2SecondPart , ColorTheme::NavigationBar\_SelectedTitle , ColorTheme::NavigationBar\_UnselectedTitle
   
 , ColorTheme::NavigationBar\_Background ,
   
 ColorTheme::Card\_Background , ColorTheme::Card\_Text , ColorTheme::ActionList\_Text , ColorTheme::ActionList\_Background
   
 ,
   
 ColorTheme::ActionList\_HoverBackground , ColorTheme::CodeHighlighter\_Background , ColorTheme::ActionDescription\_Background
   
 , ColorTheme::Visualizer\_Label ,
   
 ColorTheme::Visualizer\_Node\_Default\_Outline1 , ColorTheme::Visualizer\_Node\_Default\_Outline2 ,
   
 ColorTheme::Visualizer\_Node\_Default\_Background1 , ColorTheme::Visualizer\_Node\_Default\_Background2
   
 ,
   
 ColorTheme::Visualizer\_Node\_Default\_Text1 , ColorTheme::Visualizer\_Node\_Default\_Text2 , ColorTheme::Visualizer\_Node\_ActiveOutline1
   
 , ColorTheme::Visualizer\_Node\_ActiveOutline2 ,
   
 ColorTheme::Visualizer\_Node\_ActiveBackground1 , ColorTheme::Visualizer\_Node\_ActiveBackground2 ,
   
 ColorTheme::Visualizer\_Node\_ActiveText1 , ColorTheme::Visualizer\_Node\_ActiveText2 ,
   
 ColorTheme::Visualizer\_Node\_ActiveBlue\_Outline1 , ColorTheme::Visualizer\_Node\_ActiveBlue\_Outline2 ,
   
 ColorTheme::Visualizer\_Node\_ActiveBlue\_Background1 , ColorTheme::Visualizer\_Node\_ActiveBlue\_Background2
   
 ,
   
 ColorTheme::Visualizer\_Node\_ActiveBlue\_Text1 , ColorTheme::Visualizer\_Node\_ActiveBlue\_Text2 ,
   
 ColorTheme::Visualizer\_Node\_ActiveGreen\_Outline1 , ColorTheme::Visualizer\_Node\_ActiveGreen\_Outline2
   
 ,
   
 ColorTheme::Visualizer\_Node\_ActiveGreen\_Background1 , ColorTheme::Visualizer\_Node\_ActiveGreen\_Background2
   
 , ColorTheme::Visualizer\_Node\_ActiveGreen\_Text1 , ColorTheme::Visualizer\_Node\_ActiveGreen\_Text2 ,
   
 ColorTheme::Visualizer\_Node\_ActiveRed\_Outline1 , ColorTheme::Visualizer\_Node\_ActiveRed\_Outline2 ,
   
 ColorTheme::Visualizer\_Node\_ActiveRed\_Background1 , ColorTheme::Visualizer\_Node\_ActiveRed\_Background2
   
 ,
   
 ColorTheme::Visualizer\_Node\_ActiveRed\_Text1 , ColorTheme::Visualizer\_Node\_ActiveRed\_Text2 ,
   
 ColorTheme::Visualizer\_Node\_Iterated\_Outline1 , ColorTheme::Visualizer\_Node\_Iterated\_Outline2 ,
   
 ColorTheme::Visualizer\_Node\_Iterated\_Background1 , ColorTheme::Visualizer\_Node\_Iterated\_Background2
   
 , ColorTheme::Visualizer\_Node\_Iterated\_Text1 , ColorTheme::Visualizer\_Node\_Iterated\_Text2 ,
   
 ColorTheme::Visualizer\_Arrow\_Default , ColorTheme::Visualizer\_Arrow\_Active , ColorTheme::Count }

## 7.109 ColorThemeldentifiers.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef IDENTIFIERS_COLORIDENTIFIERS_HPP
00002 #define IDENTIFIERS_COLORIDENTIFIERS_HPP
00003
00004 namespace ColorTheme {
00005     enum ID {
00006         /* Background */
00007         Background,
00008         /* GUI */
00009         /* Logo color */
00010         Logo1FirstPart,
00011         Logo1SecondPart,
00012         Logo2FirstPart,
00013         Logo2SecondPart,
00014
00015         /* Navigation bar */
00016         NavigationBar_SelectedTitle,
00017         NavigationBar_UnselectedTitle,
00018         NavigationBar_Background,
00019
00020         /* Card (title) */
00021         Card_Background,
00022         Card_Text,
00023
00024         /* Action list */
00025         ActionList_Text,
00026         ActionList_Background,
00027         ActionList_HoverBackground,
```

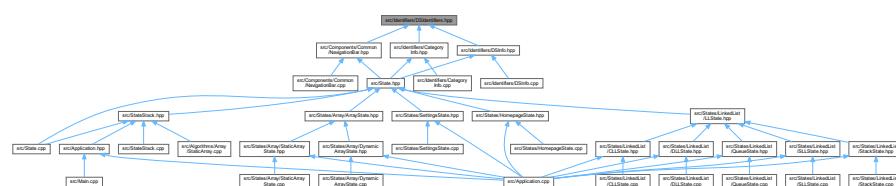
```

00028
00029     /* Code highlighter */
00030     CodeHighlighter_Background,
00031     ActionDescription_Background,
00032
00033     /* Visualizer */
00034     /* Node */
00035     Visualizer_Label,
00036     /* Default */
00037     Visualizer_Node_Default_Outline1,
00038     Visualizer_Node_Default_Outline2,
00039     Visualizer_Node_Default_Background1,
00040     Visualizer_Node_Default_Background2,
00041     Visualizer_Node_Default_Text1,
00042     Visualizer_Node_Default_Text2,
00043
00044     /* Active */
00045     Visualizer_Node_Active_Outline1,
00046     Visualizer_Node_Active_Outline2,
00047     Visualizer_Node_Active_Background1,
00048     Visualizer_Node_Active_Background2,
00049     Visualizer_Node_Active_Text1,
00050     Visualizer_Node_Active_Text2,
00051
00052     /* Active Blue */
00053     Visualizer_Node_ActiveBlue_Outline1,
00054     Visualizer_Node_ActiveBlue_Outline2,
00055     Visualizer_Node_ActiveBlue_Background1,
00056     Visualizer_Node_ActiveBlue_Background2,
00057     Visualizer_Node_ActiveBlue_Text1,
00058     Visualizer_Node_ActiveBlue_Text2,
00059
00060     /* Active Green */
00061     Visualizer_Node_ActiveGreen_Outline1,
00062     Visualizer_Node_ActiveGreen_Outline2,
00063     Visualizer_Node_ActiveGreen_Background1,
00064     Visualizer_Node_ActiveGreen_Background2,
00065     Visualizer_Node_ActiveGreen_Text1,
00066     Visualizer_Node_ActiveGreen_Text2,
00067
00068     /* Active Red */
00069     Visualizer_Node_ActiveRed_Outline1,
00070     Visualizer_Node_ActiveRed_Outline2,
00071     Visualizer_Node_ActiveRed_Background1,
00072     Visualizer_Node_ActiveRed_Background2,
00073     Visualizer_Node_ActiveRed_Text1,
00074     Visualizer_Node_ActiveRed_Text2,
00075
00076     /* Iterated */
00077     Visualizer_Node_Iterated_Outline1,
00078     Visualizer_Node_Iterated_Outline2,
00079     Visualizer_Node_Iterated_Background1,
00080     Visualizer_Node_Iterated_Background2,
00081     Visualizer_Node_Iterated_Text1,
00082     Visualizer_Node_Iterated_Text2,
00083
00084     /* Arrow */
00085     Visualizer_Arrow_Default,
00086     Visualizer_Arrow_Active,
00087
00088     Count,
00089 };
00090 };
00091
00092 #endif // IDENTIFIERS_COLORIDENTIFIERS_HPP

```

## 7.110 src/Identifiers/DSIdentifiers.hpp File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- namespace [DataStructures](#)

## Enumerations

- enum [DataStructures::ID](#) {
 [DataStructures::None](#) , [DataStructures::StaticArray](#) , [DataStructures::DynamicArray](#) , [DataStructures::SinglyLinkedList](#) ,
 [DataStructures::DoublyLinkedList](#) , [DataStructures::CircularLinkedList](#) , [DataStructures::Stack](#) , [DataStructures::Queue](#) ,
 [DataStructures::Count](#) }

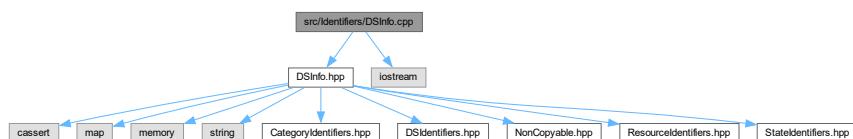
## 7.111 DSIdentifiers.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef DSIDENTIFIERS_HPP
00002 #define DSIDENTIFIERS_HPP
00003
00004 namespace DataStructures {
00005     enum ID {
00006         None,
00007         StaticArray,
00008         DynamicArray,
00009         SinglyLinkedList,
00010         DoublyLinkedList,
00011         CircularLinkedList,
00012         Stack,
00013         Queue,
00014         Count
00015     };
00016 };
00017
00018 #endif // DSIDENTIFIERS_HPP
```

## 7.112 src/Identifiers/DSInfo.cpp File Reference

```
#include "DSInfo.hpp"
#include <iostream>
Include dependency graph for DSInfo.cpp:
```



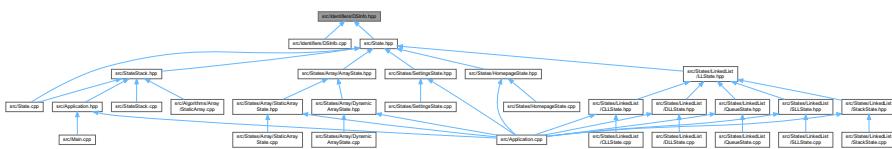
## 7.113 src/Identifiers/DSInfo.hpp File Reference

```
#include <cassert>
#include <map>
#include <memory>
#include <string>
#include "CategoryIdentifiers.hpp"
#include "DSIdentifiers.hpp"
#include "NonCopyable.hpp"
#include "ResourceIdentifiers.hpp"
#include "StateIdentifiers.hpp"
Include dependency graph for DSInfo.hpp:
```

Include dependency graph for DSpin8.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class DSInfo
  - struct DSInfo::Info

## 7.114 DSInfo.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef DSINFO_HPP
00002 #define DSINFO_HPP
00003
00004 #include <cassert>
00005 #include <map>
00006 #include <memory>
00007 #include <string>
00008
00009 #include "CategoryIdentifiers.hpp"
00010 #include "DSIdentifiers.hpp"
00011 #include "NonCopyable.hpp"
00012 #include "ResourceIdentifiers.hpp"
00013 #include "StateIdentifiers.hpp"
00014
00015 class DSInfo : private NonCopyable< DSInfo > {
00016 private:
00017     struct Info {
00018         Info(DataStructures::ID ID, States::ID stateID, Category::ID categoryID,
00019               Textures::ID thumbnail, std::string name, std::string abbr);
00020         DataStructures::ID ID;
00021         States::ID stateID;
00022         Category::ID categoryID;
00023         Textures::ID thumbnail;
```

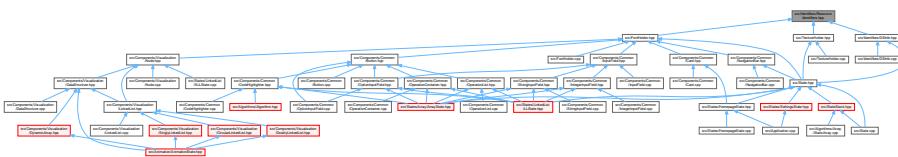
```

00024     std::string name;
00025     std::string abbr;
00026 };
00027 std::map< DataStructures::ID, Info > mFactories;
00028
00029 public:
00030     void Register(DataStructures::ID, Info info);
00031     Info Get(DataStructures::ID id) const;
00032 };
00033
00034 #endif // DSINFO_HPP

```

## 7.115 src/Identifiers/Resourcelidentifiers.hpp File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- namespace [Fonts](#)
- namespace [Textures](#)

## Enumerations

- enum [Fonts::ID](#) {
 [Fonts::Default](#) , [Fonts::Default\\_Italic](#) , [Fonts::Default\\_Bold](#) , [Fonts::Silkscreen](#) ,
 [Fonts::Consolas](#) , [Fonts::Courier](#) , [Fonts::Courier\\_Bold](#) , [Fonts::Count](#) }
- enum [Textures::ID](#) {
 [Textures::Blank](#) , [Textures::StaticArray](#) , [Textures::DynamicArray](#) , [Textures::SinglyLinkedList](#) ,
 [Textures::DoublyLinkedList](#) , [Textures::CircularLinkedList](#) , [Textures::Stack](#) , [Textures::Queue](#) ,
 [Textures::Favicon](#) , [Textures::Count](#) }

## 7.116 Resourcelidentifiers.hpp

[Go to the documentation of this file.](#)

```

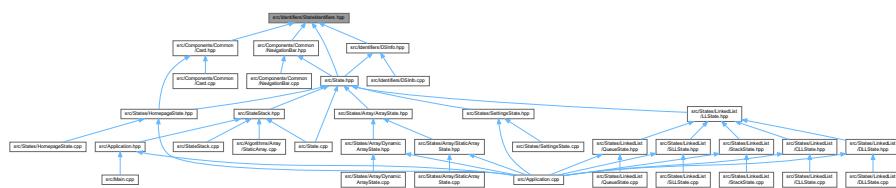
00001 #ifndef RESOURCEIDENTIFIER_HPP
00002 #define RESOURCEIDENTIFIER_HPP
00003
00004 namespace Fonts {
00005     enum ID {
00006         Default,
00007         Default_Italic,
00008         Default_Bold,
00009         Silkscreen,
00010         Consolas,
00011         Courier,
00012         Courier_Bold,
00013         Count
00014     };
00015 }
00016
00017 namespace Textures {
00018     enum ID {

```

```
00019     Blank,  
00020     StaticArray,  
00021     DynamicArray,  
00022     SinglyLinkedList,  
00023     DoublyLinkedList,  
00024     CircularLinkedList,  
00025     Stack,  
00026     Queue,  
00027     Favicon,  
00028     Count  
00029 };  
00030 };  
00031  
00032 #endif // RESOURCEIDENTIFIER_HPP
```

## 7.117 src/Identifiers/StatIdentifiers.hpp File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- namespace `States`

## Enumerations

- enum States::ID {  
States::None , States::Homepage , States::Settings , States::StaticArray ,  
States::DynamicArray , States::SinglyLinkedList , States::DoublyLinkedList , States::CircularLinkedList ,  
States::Stack , States::Queue , States::Count }

## 7.118 StatIdentifiers.hpp

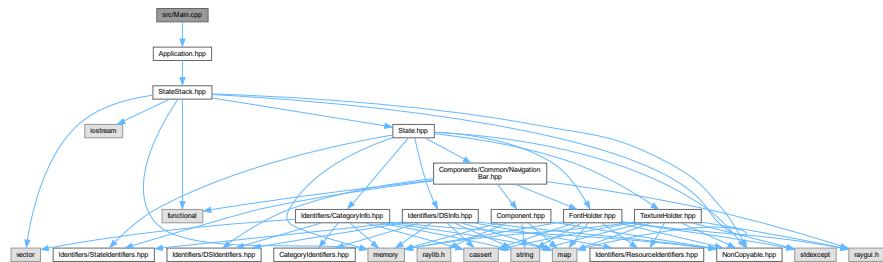
[Go to the documentation of this file.](#)

```
Go to the documentation of this file.
00001 #ifndef STATEIDENTIFIER_HPP
00002 #define STATEIDENTIFIER_HPP
00003
00004 namespace States {
00005     enum ID {
00006         None,
00007         Homepage,
00008         Settings,
00009         StaticArray,
00010         DynamicArray,
00011         SinglyLinkedList,
00012         DoublyLinkedList,
00013         CircularLinkedList,
00014         Stack,
00015         Queue,
00016         Count
00017     };
00018 };
00019
00020#endif // STATEIDENTIFIER_HPP
```

## 7.119 src/Main.cpp File Reference

```
#include "Application.hpp"
```

Include dependency graph for Main.cpp:



## Functions

- int `main ()`

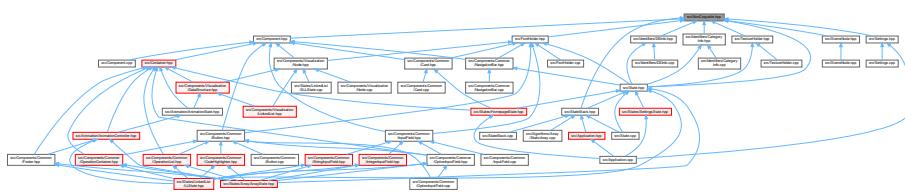
### 7.119.1 Function Documentation

#### 7.119.1.1 main()

```
int main ( )
```

## 7.120 src/NonCopyable.hpp File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class `NonCopyable< T >`

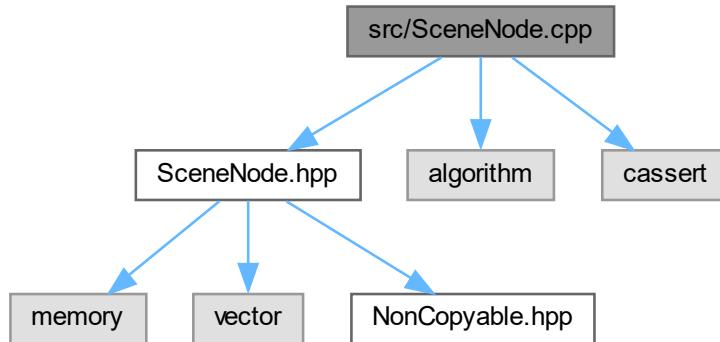
## 7.121 NonCopyable.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef NONCOPYABLE_HPP
00002 #define NONCOPYABLE_HPP
00003
00004 template< class T >
00005 class NonCopyable {
00006 public:
00007     NonCopyable(const NonCopyable &) = delete;
00008     T &operator=(const T &) = delete;
00009
00010 protected:
00011     NonCopyable() = default;
00012     ~NonCopyable() = default; // Protected non-virtual destructor
00013 };
00014
00015 #endif // NONCOPYABLE_HPP
```

## 7.122 src/SceneNode.cpp File Reference

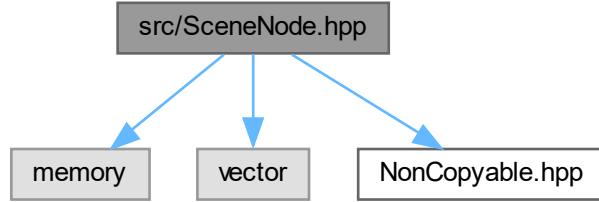
```
#include "SceneNode.hpp"
#include <algorithm>
#include <cassert>
Include dependency graph for SceneNode.cpp:
```



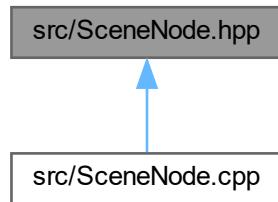
## 7.123 src/SceneNode.hpp File Reference

```
#include <memory>
#include <vector>
```

```
#include "NonCopyable.hpp"  
Include dependency graph for SceneNode.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [SceneNode](#)

## 7.124 SceneNode.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef SCENENODE_HPP  
00002 #define SCENENODE_HPP  
00003  
00004 #include <memory>  
00005 #include <vector>  
00006  
00007 #include "NonCopyable.hpp"  
00008  
00009 class SceneNode  
00010 // : public NonCopyable< SceneNode >  
00011 {  
00012 public:  
00013     typedef std::shared_ptr< SceneNode > Ptr;  
00014  
00015 public:  
00016     SceneNode();  
00017     virtual ~SceneNode(){};  
00018     void AttachChild(Ptr child);
```

```

00019     Ptr DetachChild(const SceneNode& node);
00020     virtual void Draw();
00021     virtual void DrawCurrent() = 0;
00022     // virtual void Update()
00023 private:
00024     std::vector<Ptr> mChildren;
00025     SceneNode* mParent;
00026 };
00027
00028 #endif // SCENENODE_HPP

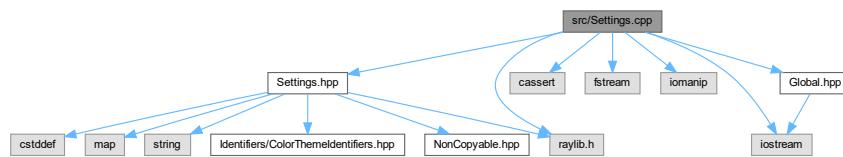
```

## 7.125 src/Settings.cpp File Reference

```

#include "Settings.hpp"
#include <cassert>
#include <fstream>
#include <iomanip>
#include <iostream>
#include "Global.hpp"
#include "raylib.h"
Include dependency graph for Settings.cpp:

```

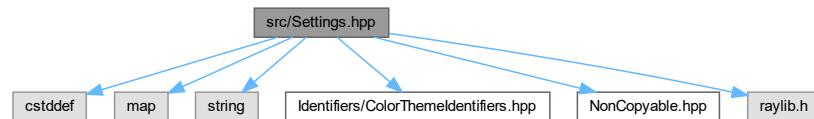


## 7.126 src/Settings.hpp File Reference

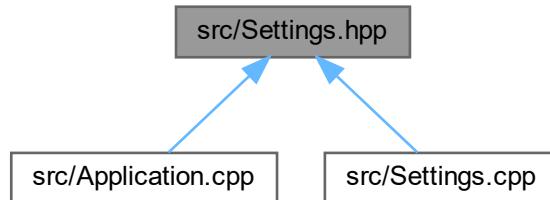
```

#include <cstddef>
#include <map>
#include <string>
#include "Identifiers/ColorThemeIdentifiers.hpp"
#include "NonCopyable.hpp"
#include "raylib.h"
Include dependency graph for Settings.hpp:

```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Settings](#)

## 7.127 Settings.hpp

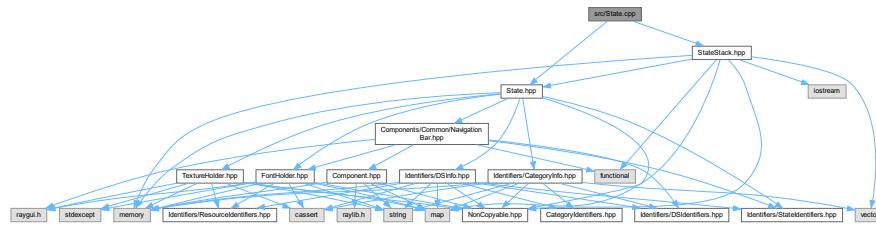
[Go to the documentation of this file.](#)

```
00001 #ifndef SETTINGS_HPP_
00002 #define SETTINGS_HPP_
00003
00004 #include <cstddef>
00005 #include <map>
00006 #include <string>
00007
00008 #include "Identifiers/ColorThemeIdentifiers.hpp"
00009 #include "NonCopyable.hpp"
00010 #include "raylib.h"
00011
00012 class Settings : private NonCopyable< Settings > {
00013     private:
00014         std::map< ColorTheme::ID, Color > mColors;
00015
00016     private:
00017         Settings() = default;
00018
00019     public:
00020         Settings(Settings&&) = delete;
00021         Settings& operator=(Settings&&) = delete;
00022         ~Settings();
00023
00024     public:
00025         static Settings& getInstance();
00026         void LoadDefaultColors();
00027
00028         Color& getColor(ColorTheme::ID id);
00029         Color getColor(ColorTheme::ID id) const;
00030
00031         void SaveToFile(const std::string& path);
00032         void LoadFromFile(const std::string& path);
00033     };
00034
00035 #endif // SETTINGS_HPP_
```

## 7.128 src/State.cpp File Reference

```
#include "State.hpp"
#include "StateStack.hpp"
```

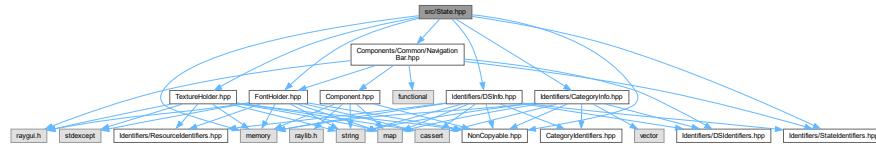
Include dependency graph for State.cpp:



## 7.129 src/State.hpp File Reference

```
#include <memory>
#include "Components/Common/NavigationBar.hpp"
#include "FontHolder.hpp"
#include "Identifiers/StateIdentifiers.hpp"
#include "TextureHolder.hpp"
#include "Identifiers/CategoryInfo.hpp"
#include "Identifiers/DSInfo.hpp"
#include "NonCopyable.hpp"
```

Include dependency graph for State.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [State](#)
- struct [State::Context](#)

## 7.130 State.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef STATE_HPP
00002 #define STATE_HPP
00003
00004 #include <memory>
00005
00006 #include "Components/Common/NavigationBar.hpp"
00007 #include "FontHolder.hpp"
00008 #include "Identifiers/StateIdentifiers.hpp"
00009 #include "TextureHolder.hpp"
00010
00011 // Categories and Data Structures identifiers
00012 #include "Identifiers/CategoryInfo.hpp"
00013 #include "Identifiers/DSInfo.hpp"
00014 #include "NonCopyable.hpp"
00015
00016 class StateStack;
00017
00018 class State : private NonCopyable< State > {
00019 public:
00020     typedef std::unique_ptr< State > Ptr;
00021     struct Context {
00022         Context();
00023         Context(FontHolder* fonts, TextureHolder* textures,
00024                  CategoryInfo* categories, DSInfo* dsInfo);
00025         FontHolder* fonts;
00026         TextureHolder* textures;
00027         CategoryInfo* categories;
00028         DSInfo* dsInfo;
00029     };
00030
00031     public:
00032         State(StateStack& stack, Context context);
00033         virtual ~State();
00034         virtual void Draw() = 0;
00035         virtual bool Update(float dt) = 0;
00036         // virtual bool handleEvent(Event event) = 0;
00037
00038 protected:
00039     void RequestStackPush(States::ID stateID);
00040     void RequestStackPop();
00041     void RequestStackClear();
00042     Context GetContext() const;
00043
00044     void InitNavigationBar();
00045
00046 private:
00047     StateStack* mStack;
00048     Context mContext;
00049
00050 protected:
00051     GUI::NavigationBar navigation;
00052 };
00053
00054 #endif // STATE_HPP

```

## 7.131 src/States/Array/ArrayState.hpp File Reference

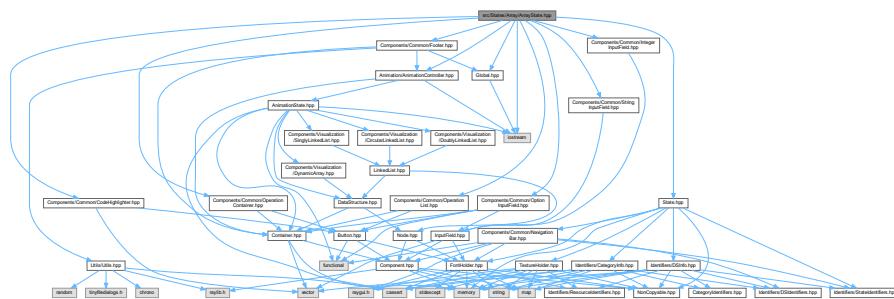
```

#include "Animation/AnimationController.hpp"
#include "Components/Common/CodeHighlighter.hpp"
#include "Components/Common/Footer.hpp"
#include "Components/Common/OperationContainer.hpp"
#include "Components/Common/OperationList.hpp"
#include "Components/Common/OptionInputField.hpp"
#include "State.hpp"
#include <iostream>
#include "Components/Common/IntegerInputField.hpp"
#include "Components/Common/StringInputField.hpp"

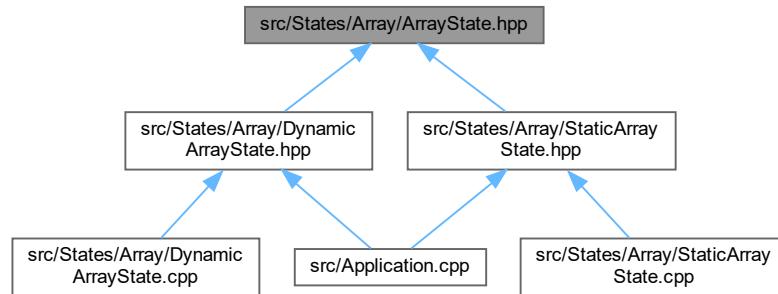
```

```
#include "Global.hpp"
```

Include dependency graph for `ArrayType.hpp`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `ArrayType< T >`
- struct `ArrayType< T >::IntegerInput`

## 7.132 ArrayType.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef STATES_LINKEDLIST_ARRAYSTATE_HPP
00002 #define STATES_LINKEDLIST_ARRAYSTATE_HPP
00003
00004 #include "Animation/AnimationController.hpp"
00005 #include "Components/Common/CodeHighlighter.hpp"
00006 #include "Components/Common/Footer.hpp"
00007 #include "Components/Common/OperationContainer.hpp"
00008 #include "Components/Common/OperationList.hpp"
00009 #include "Components/Common/OptionInputField.hpp"
00010 #include "State.hpp"
00011
00012 template< typename T >
00013 class ArrayType : public State {
00014 public:
00015     struct IntegerInput {
00016         std::string label;
00017         int width;
00018         int minValue;
  
```

```
00019         int maxValue;
00020     };
00021
00022 public:
00023     ArrayState(StateStack& stack, Context context, DataStructures::ID activeDS);
00024     ~ArrayState();
00025     virtual void Draw() = 0;
00026     virtual bool Update(float dt);
00027     virtual void SetCurrentAction(std::string action);
00028     virtual void SetCurrentError(std::string error);
00029     virtual void ClearError();
00030     virtual void ClearAction();
00031     virtual void Success();
00032
00033 protected:
00034     virtual void DrawCurrentActionText();
00035     virtual void DrawCurrentErrorText();
00036
00037 protected:
00038     void InitNavigationBar();
00039     Context mContext;
00040
00041 protected:
00042     GUI::CodeHighlighter::Ptr codeHighlighter;
00043     GUI::Footer< T > footer;
00044     std::string mCurrentAction;
00045     std::string mCurrentError;
00046
00047 protected:
00048     typename T::Ptr animController;
00049
00050 protected:
00051     GUI::OperationList operationList;
00052     virtual void AddOperations(); // DO NOT OVERRIDE THIS FUNCTION
00053     virtual void AddInitializeOperation();
00054     virtual void AddInsertOperation();
00055     virtual void AddDeleteOperation();
00056     virtual void AddUpdateOperation();
00057     virtual void AddSearchOperation();
00058     virtual void AddAccessOperation();
00059
00060 protected:
00061     virtual void AddNoFieldOperationOption(
00062         GUI::OperationContainer::Ptr container, std::string title,
00063         std::function< void() > action);
00064     virtual void AddIntFieldOperationOption(
00065         GUI::OperationContainer::Ptr container, std::string title,
00066         std::vector< IntegerInput > fields,
00067         std::function< void(std::map< std::string, std::string >) > action);
00068     virtual void AddStringFieldOption(
00069         GUI::OperationContainer::Ptr container, std::string title,
00070         std::string label,
00071         std::function< void(std::map< std::string, std::string >) > action);
00072
00073 private:
00074     DataStructures::ID activeDS;
00075 };
00076
00077 #include <iostream>
00078
00079 #include "Components/Common/IntegerField.hpp"
00080 #include "Components/Common/StringInputField.hpp"
00081 #include "Global.hpp"
00082
00083 template< typename T >
00084 ArrayState< T >::ArrayState(StateStack& stack, Context context,
00085     DataStructures::ID activeDS)
00086     : State(stack, context), activeDS(activeDS),
00087     codeHighlighter(new GUI::CodeHighlighter(context.fonts)),
00088     footer(GUI::Footer< T >()), animController(new T()) {
00089     InitNavigationBar();
00090     operationList = GUI::OperationList(context.fonts);
00091     codeHighlighter->SetPosition(global::SCREEN_WIDTH - 40,
00092                                 global::SCREEN_HEIGHT - 334);
00093     codeHighlighter->InitButtons();
00094
00095     footer.SetPosition(0, global::SCREEN_HEIGHT - 40);
00096 }
00097
00098 template< typename T >
00099 ArrayState< T >::~ArrayState() {}
00100
00101 template< typename T >
00102 bool ArrayState< T >::Update(float dt) {
00103     animController->Update(dt);
00104     codeHighlighter->Highlight(
00105         animController->GetAnimation().GetHighlightedLine());
00106 }
```

```
00106     codeHighlighter->AddActionDescription(
00107         animController->GetAnimation().GetActionDescription());
00108     return true;
00109 }
00110
00111 template< typename T >
00112     inline void ArrayState< T >::SetCurrentAction(std::string action) {
00113     mCurrentAction = action;
00114 }
00115
00116 template< typename T >
00117     inline void ArrayState< T >::SetCurrentError(std::string error) {
00118     mCurrentError = error;
00119 }
00120
00121 template< typename T >
00122     inline void ArrayState< T >::ClearError() {
00123     mCurrentError.clear();
00124 }
00125
00126 template< typename T >
00127     inline void ArrayState< T >::ClearAction() {
00128     mCurrentAction.clear();
00129 }
00130
00131 template< typename T >
00132     inline void ArrayState< T >::Success() {
00133     operationList.ToggleOperations();
00134     SetMouseCursor(MOUSE_CURSOR_DEFAULT);
00135     ClearError();
00136 }
00137
00138 template< typename T >
00139     inline void ArrayState< T >::DrawCurrentActionText() {
00140     float rightAlignment = global::SCREEN_WIDTH - 50;
00141     Font font = GetContext().fonts->Get(FONT::Default_Bold);
00142     float width = MeasureTextEx(font, mCurrentAction.c_str(), 32, 0).x;
00143     float x = rightAlignment - width;
00144
00145     DrawTextEx(font, mCurrentAction.c_str(),
00146                 (Vector2){x, global::SCREEN_HEIGHT - 376}, 32, 0, BLACK);
00147
00148 // codeHighlighter->SetPosition(), global::SCREEN_HEIGHT - 334);
00149 }
00150
00151 template< typename T >
00152     inline void ArrayState< T >::DrawCurrentErrorText() {
00153     Font font = GetContext().fonts->Get(FONT::Default_Bold);
00154     float width = MeasureTextEx(font, mCurrentError.c_str(), 24, 0).x;
00155     float x = 50;
00156
00157     DrawTextEx(
00158         font, mCurrentError.c_str(),
00159         (Vector2){x, global::SCREEN_HEIGHT - operationList.GetSize().y - 90},
00160         24, 0, RED);
00161 }
00162
00163 template< typename T >
00164     void ArrayState< T >::InitNavigationBar() {
00165     navigation.SetVisibleTitle(true);
00166     auto info = GetContext().categories->Get(CATEGORY::Array);
00167     navigation.SetCategory(info.categoryName);
00168     navigation.SetActiveTitle(activeDS);
00169     for (auto dsID : info.mDS) {
00170         auto dsInfo = GetContext().dsInfo->Get(dsID);
00171         navigation.InsertTitle(dsID, dsInfo.stateID, dsInfo.abbr, dsInfo.name);
00172     };
00173 }
00174
00175 template< typename T >
00176     void ArrayState< T >::AddOperations() {
00177     AddInitializeOperation();
00178     AddInsertOperation();
00179     AddDeleteOperation();
00180     AddAccessOperation();
00181     AddUpdateOperation();
00182     AddSearchOperation();
00183
00184     operationList.setPosition(
00185         0, global::SCREEN_HEIGHT - 60 - operationList.GetSize().y);
00186     operationList.InitActionBar();
00187 }
00188
00189 template< typename T >
00190     void ArrayState< T >::AddInitializeOperation() {}
00191
00192 template< typename T >
```

```

00193 void ArrayState< T >::AddInsertOperation() {}
00194
00195 template< typename T >
00196 void ArrayState< T >::AddDeleteOperation() {}
00197
00198 template< typename T >
00199 void ArrayState< T >::AddUpdateOperation() {}
00200
00201 template< typename T >
00202 void ArrayState< T >::AddSearchOperation() {}
00203
00204 template< typename T >
00205 inline void ArrayState< T >::AddAccessOperation() {}
00206
00207 template< typename T >
00208 void ArrayState< T >::AddNoFieldOperationOption(
00209     GUI::OperationContainer::Ptr container, std::string title,
00210     std::function< void() > action) {
00211     GUI::OptionInputField::Ptr button(
00212         new GUI::OptionInputField(getContext().fonts));
00213
00214     button.get() ->SetNoFieldOption(title, action);
00215
00216     container.get() ->pack(button);
00217 }
00218
00219 template< typename T >
00220 void ArrayState< T >::AddIntFieldOperationOption(
00221     GUI::OperationContainer::Ptr container, std::string title,
00222     std::vector< IntegerInput > fields,
00223     std::function< void(std::map< std::string, std::string >) > action) {
00224     GUI::OptionInputField::Ptr button(
00225         new GUI::OptionInputField(getContext().fonts));
00226     std::vector< GUI::InputField::Ptr > intFields;
00227     for (auto field : fields) {
00228         GUI::IntegerInputField::Ptr intField(
00229             new GUI::IntegerInputField(getContext().fonts));
00230         intField.get() ->SetLabel(field.label);
00231         intField.get() ->SetInputFieldSize((Vector2){(float)field.width, 30});
00232         intField.get() ->SetConstraint(field.minValue, field.maxValue);
00233         intField.get() ->Randomize();
00234         intFields.push_back(intField);
00235     }
00236
00237     button.get() ->SetOption(title, intFields, action);
00238
00239     container.get() ->pack(button);
00240 }
00241
00242 template< typename T >
00243 void ArrayState< T >::AddStringFieldOption(
00244     GUI::OperationContainer::Ptr container, std::string title,
00245     std::string label,
00246     std::function< void(std::map< std::string, std::string >) > action) {
00247     GUI::OptionInputField::Ptr button(
00248         new GUI::OptionInputField(getContext().fonts));
00249     GUI::StringInputField::Ptr strField(
00250         new GUI::StringInputField(getContext().fonts));
00251     strField.get() ->SetLabel(label);
00252     strField.get() ->SetInputFieldSize((Vector2){100, 30});
00253     button.get() ->SetOption(title, {strField}, action);
00254     container.get() ->pack(button);
00255 }
00256
00257 #endif // STATES_LINKEDLIST_ARRAYSTATE_HPP

```

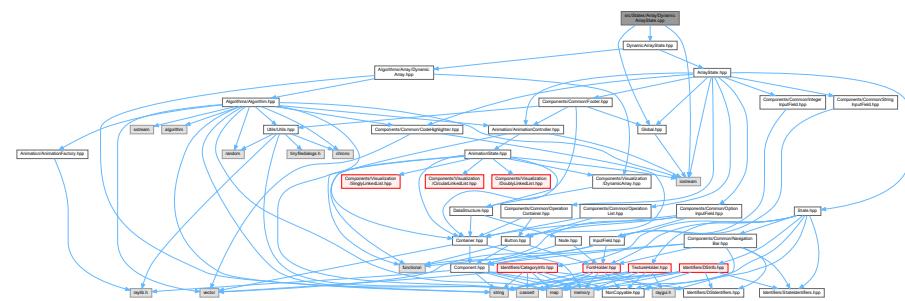
## 7.133 src/States/Array/DynamicArrayState.cpp File Reference

```

#include "DynamicArrayState.hpp"
#include <iostream>
#include "Global.hpp"

```

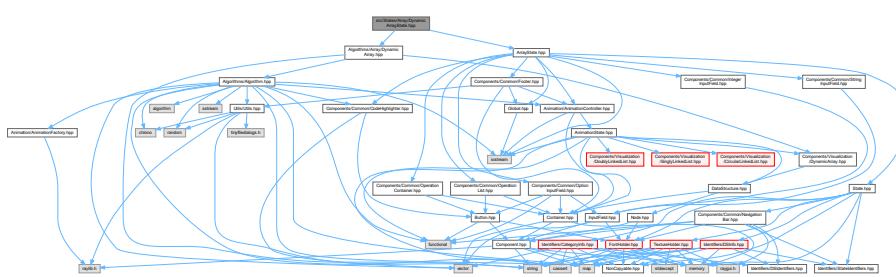
Include dependency graph for DynamicArrayState.cpp:



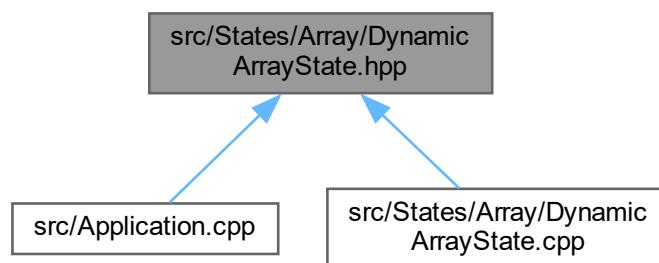
## 7.134 src/States/Array/DynamicArrayState.hpp File Reference

```
#include "Algorithms/Array/DynamicArray.hpp"
#include "ArrayState.hpp"
```

Include dependency graph for DynamicArrayState.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [DynamicArrayState](#)

## 7.135 DynamicArrayState.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef STATES_ARRAY_DYNAMICARRAYSTATE_HPP
00002 #define STATES_ARRAY_DYNAMICARRAYSTATE_HPP
00003
00004 #include "Algorithms/Array/DynamicArray.hpp"
00005 #include "ArrayState.hpp"
00006
00007 class DynamicArrayState : public ArrayState< DArrayAnimationController > {
00008 private:
00009     Algorithm::DynamicArray mDynamicArray;
00010
00011 private:
00012     void AddInsertOperation();
00013     void AddInitializeOperation();
00014     void AddUpdateOperation();
00015     void AddDeleteOperation();
00016     void AddSearchOperation();
00017     void AddAccessOperation();
00018
00019 public:
00020     DynamicArrayState(StateStack& stack, Context context);
00021     ~DynamicArrayState();
00022     void Draw();
00023 };
00024
00025 #endif // STATES_ARRAY_DYNAMICARRAYSTATE_HPP

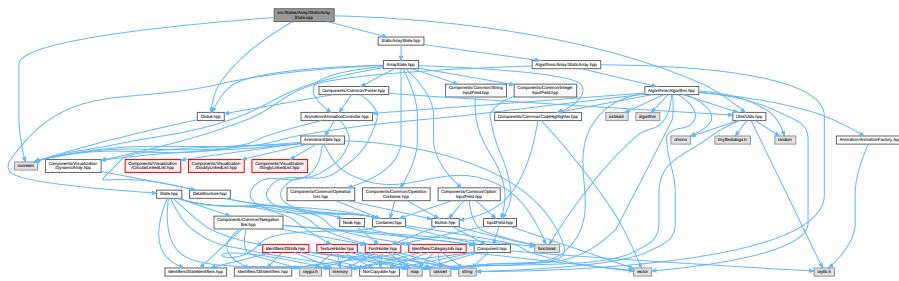
```

## 7.136 src/States/Array/StaticArrayState.cpp File Reference

```

#include "StaticArrayState.hpp"
#include <iostream>
#include "Global.hpp"
#include "Utils/Utils.hpp"
Include dependency graph for StaticArrayState.cpp:

```



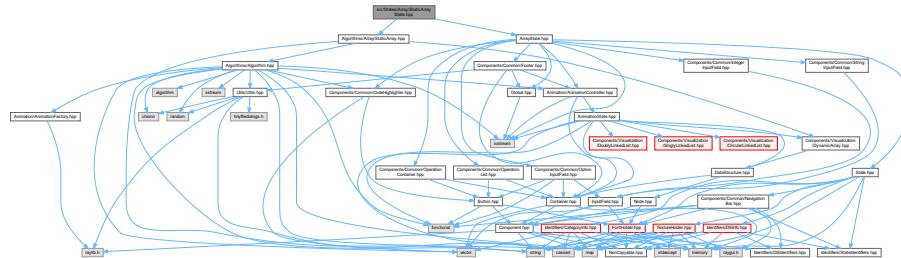
## 7.137 src/States/Array/StaticArrayState.hpp File Reference

```

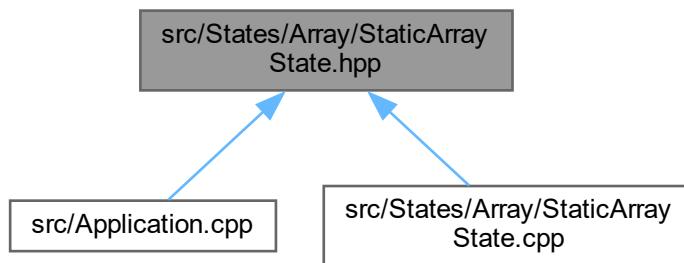
#include "Algorithms/Array/StaticArray.hpp"
#include "ArrayState.hpp"

```

Include dependency graph for StaticArrayState.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [StaticArrayState](#)

## 7.138 StaticArrayState.hpp

[Go to the documentation of this file.](#)

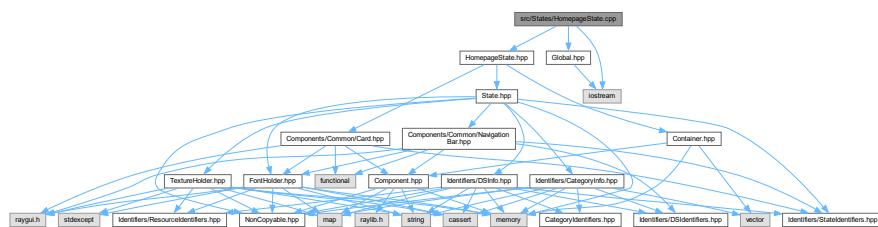
```

00001 #ifndef STATES_ARRAY_STATICARRAYSTATE_HPP
00002 #define STATES_ARRAY_STATICARRAYSTATE_HPP
00003
00004 #include "Algorithms/Array/StaticArray.hpp"
00005 #include "ArrayState.hpp"
00006 class StaticArrayState : public ArrayState< DArrayAnimationController > {
00007 private:
00008     Algorithm::StaticArray mStaticArray;
00009
00010 private:
00011     void AddInitializeOperation();
00012     void AddUpdateOperation();
00013     void AddSearchOperation();
00014     void AddAccessOperation();
00015
00016 public:
00017     StaticArrayState(StateStack& stack, Context context);
00018     ~StaticArrayState();
00019     void Draw();
00020 };
00021
00022 #endif // STATES_ARRAY_STATICARRAYSTATE_HPP

```

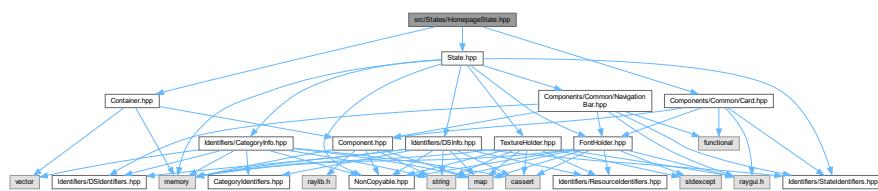
## 7.139 src/States/HomepageState.cpp File Reference

```
#include "HomepageState.hpp"
#include <iostream>
#include "Global.hpp"
Include dependency graph for HomepageState.cpp:
```

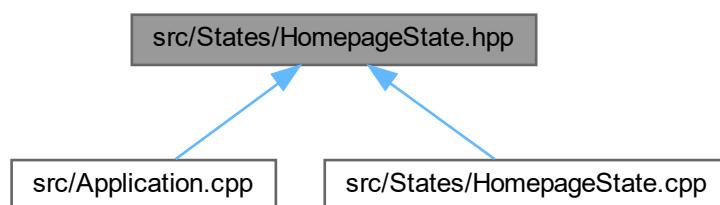


## 7.140 src/States/HomepageState.hpp File Reference

```
#include "Container.hpp"
#include "Components/Common/Card.hpp"
#include "State.hpp"
Include dependency graph for HomepageState.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [HomepageState](#)

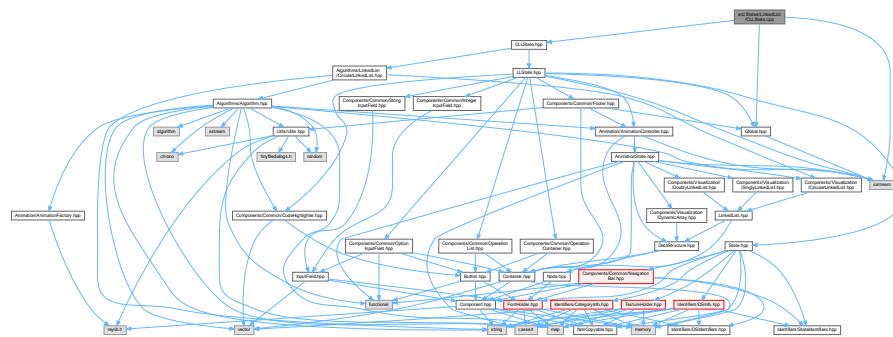
## 7.141 HomepageState.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef STATES_HOMEPAGESTATE_HPP
00002 #define STATES_HOMEPAGESTATE_HPP
00003
00004 #include "Container.hpp"
00005 // #include "../Core/Operations/Create/Create.hpp"
00006 #include "Components/Common/Card.hpp"
00007 #include "State.hpp"
00008
00009 class HomepageState : public State {
00010 private:
00011 public:
00012     HomepageState(StateStack& stack, Context context);
00013     ~HomepageState();
00014     void Draw();
00015     bool Update(float dt);
00016
00017 private:
00018     void DrawIntroduction();
00019     void InitCards();
00020     void CreateCard(States::ID stateID, std::string title,
00021                     Textures::ID textureID, int x, int y);
00022
00023 private:
00024     GUI::Container mCards;
00025     bool hasInitializeCard;
00026 };
00027
00028 #endif // STATES_HOMEPAGESTATE_HPP
```

## 7.142 src/States/LinkedList/CLLState.cpp File Reference

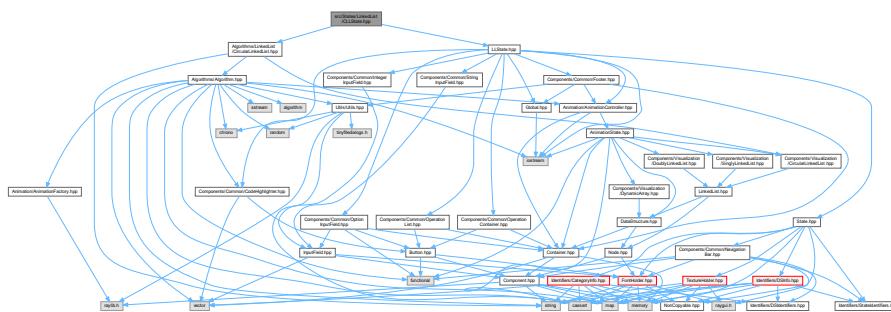
```
#include "CLLState.hpp"
#include <iostream>
#include "Global.hpp"
Include dependency graph for CLLState.cpp:
```



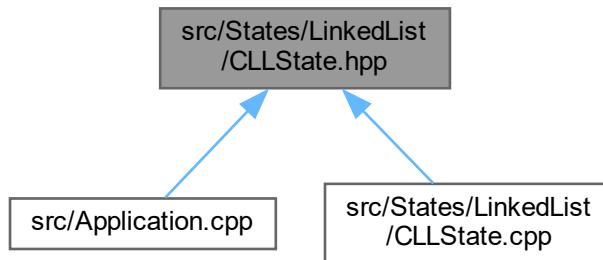
## 7.143 src/States/LinkedList/CLLState.hpp File Reference

```
#include "Algorithms/LinkedList/CircularLinkedList.hpp"
#include "LLState.hpp"
```

Include dependency graph for CLLState.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [CLLState](#)

## 7.144 CLLState.hpp

[Go to the documentation of this file.](#)

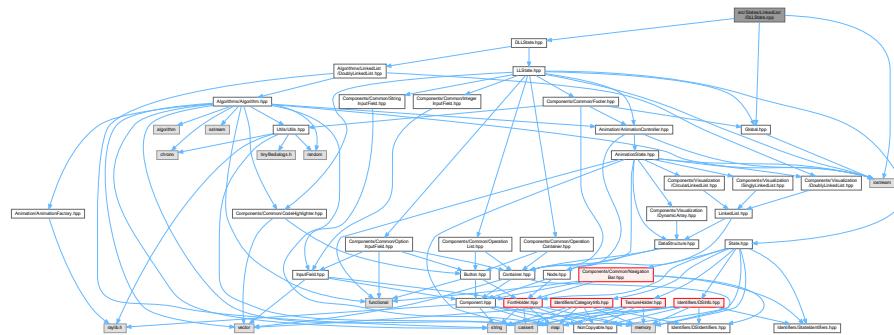
```

00001 #ifndef STATES_LINKEDLIST_CLLSTATE_HPP
00002 #define STATES_LINKEDLIST_CLLSTATE_HPP
00003
00004 #include "Algorithms/LinkedList/CircularLinkedList.hpp"
00005 #include "LLState.hpp"
00006
00007 class CLLState : public LLState< CLLAnimationController > {
00008 private:
00009     Algorithm::CircularLinkedList CLL;
00010
00011 public:
00012     void AddInsertOperation();
00013     void AddInitializeOperation();
00014     void AddUpdateOperation();
00015     void AddDeleteOperation();
00016     void AddSearchOperation();
00017
00018 public:
00019     CLLState(StateStack& stack, Context context);
00020     ~CLLState();
00021     void Draw();
00022 };
00023
00024 #endif // STATES_LINKEDLIST_CLLSTATE_HPP

```

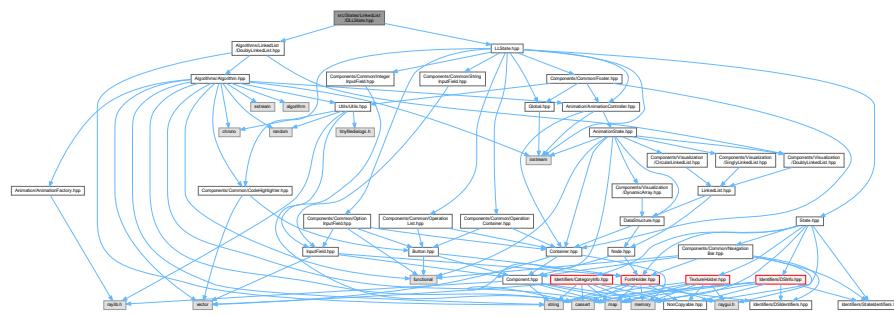
## 7.145 src/States/LinkedList/DLLState.cpp File Reference

```
#include "DLLState.hpp"
#include <iostream>
#include "Global.hpp"
Include dependency graph for DLLState.cpp:
```

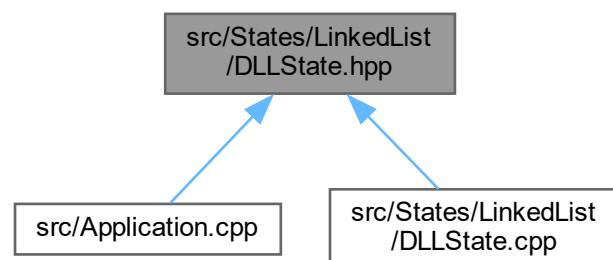


## 7.146 src/States/LinkedList/DLLState.hpp File Reference

```
#include "Algorithms/LinkedList/DoublyLinkedList.hpp"
#include "LLState.hpp"
Include dependency graph for DLLState.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class `DLLState`

## 7.147 DLLState.hpp

[Go to the documentation of this file.](#)

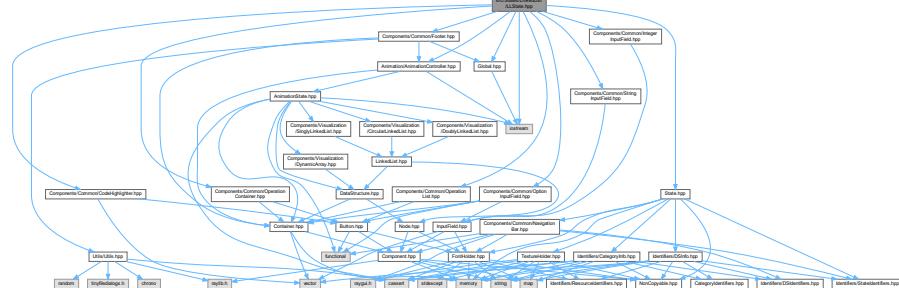
```
00001 #ifndef STATES_LINKEDLIST_DLLSTATE_HPP
00002 #define STATES_LINKEDLIST_DLLSTATE_HPP
00003
00004 #include "Algorithms/LinkedList/DoublyLinkedList.hpp"
00005 #include "LLState.hpp"
00006
00007 class DLLState : public LLState< DLLAnimationController > {
00008 private:
00009     Algorithm::DoublyLinkedList mDLL;
00010
00011 public:
00012     void AddInsertOperation();
00013     void AddInitializeOperation();
00014     void AddUpdateOperation();
00015     void AddDeleteOperation();
00016     void AddSearchOperation();
00017
00018 public:
00019     DLLState(StateStack& stack, Context context);
00020     ~DLLState();
00021     void Draw();
00022 };
00023
00024 #endif // STATES_LINKEDLIST_DLLSTATE_HPP
```

## 7.148 src/States/LinkedList/LLState.hpp File Reference

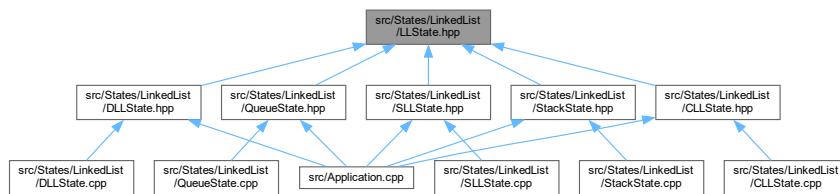
```
#include "Animation/AnimationController.hpp"
#include "Components/Common/CodeHighlighter.hpp"
#include "Components/Common/Footer.hpp"
#include "Components/Common/OperationContainer.hpp"
#include "Components/Common/OperationList.hpp"
#include "Components/Common/OptionInputField.hpp"
#include "State.hpp"
#include <iostream>
#include "Components/Common/IntegerField.hpp"
#include "Components/Common/StringInputField.hpp"
#include "Global.hpp"
```

Include dependency graph for LLState.hpp:

Include dependency graph for LLState.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [LLState< T >](#)
- struct [LLState< T >::IntegerInput](#)

## 7.149 LLState.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef STATES_LINKEDLIST_LLSTATE_HPP
00002 #define STATES_LINKEDLIST_LLSTATE_HPP
00003
00004 #include "Animation/AnimationController.hpp"
00005 #include "Components/Common/CodeHighlighter.hpp"
00006 #include "Components/Common/Footer.hpp"
00007 #include "Components/Common/OperationContainer.hpp"
00008 #include "Components/Common/OperationList.hpp"
00009 #include "Components/Common/OptionInputField.hpp"
00010 #include "State.hpp"
00011
00012 template< typename T >
00013 class LLState : public State {
00014 public:
00015     struct IntegerInput {
00016         std::string label;
00017         int width;
00018         int minValue;
00019         int maxValue;
00020     };
00021
00022 public:
00023     LLState(StateStack& stack, Context context, DataStructures::ID activeDS);
00024     ~LLState();
00025     virtual void Draw() = 0;
00026     virtual bool Update(float dt);
00027     virtual void SetCurrentAction(std::string action);
00028     virtual void SetCurrentError(std::string error);
00029     virtual void ClearError();
00030     virtual void ClearAction();
00031     virtual void Success();
00032
00033 protected:
00034     virtual void DrawCurrentActionText();
00035     virtual void DrawCurrentErrorText();
00036
00037 protected:
00038     void InitNavigationBar();
00039     Context mContext;
00040
00041 protected:
00042     GUI::CodeHighlighter::Ptr codeHighlighter;
00043     GUI::Footer< T > footer;
00044     std::string mCurrentAction;
00045     std::string mCurrentError;
00046
00047 protected:
00048     typename T::Ptr animController;
00049
00050 protected:

```

```
00051     GUI::OperationList operationList;
00052     virtual void AddOperations(); // DO NOT OVERRIDE THIS FUNCTION
00053     virtual void AddInitializeOperation();
00054     virtual void AddInsertOperation();
00055     virtual void AddDeleteOperation();
00056     virtual void AddUpdateOperation();
00057     virtual void AddSearchOperation();
00058
00059 protected:
00060     virtual void AddNoFieldOperationOption(
00061         GUI::OperationContainer::Ptr container, std::string title,
00062         std::function< void() > action);
00063     virtual void AddIntFieldOperationOption(
00064         GUI::OperationContainer::Ptr container, std::string title,
00065         std::vector< IntegerInput > fields,
00066         std::function< void(std::map< std::string, std::string >) > action);
00067     virtual void AddStringFieldOption(
00068         GUI::OperationContainer::Ptr container, std::string title,
00069         std::string label,
00070         std::function< void(std::map< std::string, std::string >) > action);
00071
00072 private:
00073     DataStructures::ID activeDS;
00074 };
00075
00076 #include <iostream>
00077
00078 #include "Components/Common/IntegerField.hpp"
00079 #include "Components/Common/StringField.hpp"
00080 #include "Global.hpp"
00081
00082 template< typename T >
00083 LLState< T >::LLState(StateStack& stack, Context context,
00084                         DataStructures::ID activeDS)
00085 : State(stack, context), activeDS(activeDS),
00086   codeHighlighter(new GUI::CodeHighlighter(context.fonts)),
00087   footer(GUI::Footer< T >()), animController(new T()) {
00088     InitNavigationBar();
00089     operationList = GUI::OperationList(context.fonts);
00090     codeHighlighter->SetPosition(global::SCREEN_WIDTH - 40,
00091                                 global::SCREEN_HEIGHT - 334);
00092     codeHighlighter->InitButtons();
00093
00094     footer.SetPosition(0, global::SCREEN_HEIGHT - 40);
00095 }
00096
00097 template< typename T >
00098 LLState< T >::~LLState() {}
00099
00100 template< typename T >
00101 bool LLState< T >::Update(float dt) {
00102     animController->Update(dt);
00103     codeHighlighter->Highlight(
00104         animController->GetAnimation().GetHighlightedLine());
00105     codeHighlighter->AddActionDescription(
00106         animController->GetAnimation().GetActionDescription());
00107     return true;
00108 }
00109
00110 template< typename T >
00111 inline void LLState< T >::SetCurrentAction(std::string action) {
00112     mCurrentAction = action;
00113 }
00114
00115 template< typename T >
00116 inline void LLState< T >::SetCurrentError(std::string error) {
00117     mCurrentError = error;
00118 }
00119
00120 template< typename T >
00121 inline void LLState< T >::ClearError() {
00122     mCurrentError.clear();
00123 }
00124
00125 template< typename T >
00126 inline void LLState< T >::ClearAction() {
00127     mCurrentAction.clear();
00128 }
00129
00130 template< typename T >
00131 inline void LLState< T >::Success() {
00132     operationList.ToggleOperations();
00133     SetMouseCursor(MOUSE_CURSOR_DEFAULT);
00134     ClearError();
00135 }
00136
00137 template< typename T >
```

```

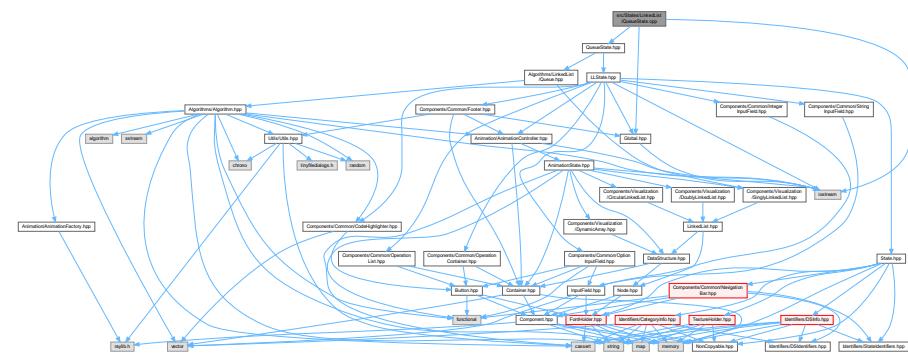
00138 inline void LLState< T >::DrawCurrentActionText() {
00139     float rightAlignment = global::SCREEN_WIDTH - 50;
00140     Font font = GetContext().fonts->Get(Fonts::Default_Bold);
00141     float width = MeasureTextEx(font, mCurrentAction.c_str(), 32, 0).x;
00142     float x = rightAlignment - width;
00143
00144     DrawTextEx(font, mCurrentAction.c_str(),
00145                 (Vector2){x, global::SCREEN_HEIGHT - 376}, 32, 0, BLACK);
00146
00147 // codeHighlighter->SetPosition(, global::SCREEN_HEIGHT - 334);
00148 }
00149
00150 template< typename T >
00151 inline void LLState< T >::DrawCurrentErrorText() {
00152     Font font = GetContext().fonts->Get(Fonts::Default_Bold);
00153     float width = MeasureTextEx(font, mCurrentError.c_str(), 24, 0).x;
00154     float x = 50;
00155
00156     DrawTextEx(
00157         font, mCurrentError.c_str(),
00158         (Vector2){x, global::SCREEN_HEIGHT - operationList.GetSize().y - 90},
00159         24, 0, RED);
00160 }
00161
00162 template< typename T >
00163 void LLState< T >::InitNavigationBar() {
00164     navigation.SetVisibleTitle(true);
00165     auto info = GetContext().categories->Get(Category::LinkedList);
00166     navigation.SetCategory(info.categoryName);
00167     navigation.SetActiveTitle(activeDS);
00168     for (auto dsID : info.mDS) {
00169         auto dsInfo = GetContext().dsInfo->Get(dsID);
00170         navigation.InsertTitle(dsID, dsInfo.stateID, dsInfo.abbr, dsInfo.name);
00171     };
00172 }
00173
00174 template< typename T >
00175 void LLState< T >::AddOperations() {
00176     AddInitializeOperation();
00177     AddInsertOperation();
00178     AddDeleteOperation();
00179     AddUpdateOperation();
00180     AddSearchOperation();
00181
00182     operationList.SetPosition(
00183         0, global::SCREEN_HEIGHT - 60 - operationList.GetSize().y);
00184     operationList.InitActionBar();
00185 }
00186
00187 template< typename T >
00188 void LLState< T >::AddInitializeOperation() {}
00189
00190 template< typename T >
00191 void LLState< T >::AddInsertOperation() {}
00192
00193 template< typename T >
00194 void LLState< T >::AddDeleteOperation() {}
00195
00196 template< typename T >
00197 void LLState< T >::AddUpdateOperation() {}
00198
00199 template< typename T >
00200 void LLState< T >::AddSearchOperation() {}
00201
00202 template< typename T >
00203 void LLState< T >::AddNoFieldOperationOption(
00204     GUI::OperationContainer::Ptr container, std::string title,
00205     std::function< void() > action) {
00206     GUI::OptionInputField::Ptr button(
00207         new GUI::OptionInputField(GetContext().fonts));
00208
00209     button.get()->SetNoFieldOption(title, action);
00210
00211     container.get()->pack(button);
00212 }
00213
00214 template< typename T >
00215 void LLState< T >::AddIntFieldOperationOption(
00216     GUI::OperationContainer::Ptr container, std::string title,
00217     std::vector< IntegerInput > fields,
00218     std::function< void(std::map< std::string, std::string >) > action) {
00219     GUI::OptionInputField::Ptr button(
00220         new GUI::OptionInputField(GetContext().fonts));
00221     std::vector< GUI::InputField::Ptr > intFields;
00222     for (auto field : fields) {
00223         GUI::IntegerInputField::Ptr intField(
00224             new GUI::IntegerInputField(GetContext().fonts));

```

```
00225     intField.get() -> SetLabel(field.label);
00226     intField.get() -> SetInputFieldSize((Vector2){(float)field.width, 30});
00227     intField.get() -> SetConstraint(field.minValue, field.maxValue);
00228     intField.get() -> Randomize();
00229     intFields.push_back(intField);
00230 }
00231
00232     button.get() -> SetOption(title, intFields, action);
00233
00234     container.get() -> pack(button);
00235 }
00236
00237 template< typename T >
00238 void LLState< T >::AddStringFieldOption(
00239     GUI::OperationContainer::Ptr container, std::string title,
00240     std::string label,
00241     std::function< void(std::map< std::string, std::string >) > action) {
00242     GUI::OptionInputField::Ptr button(
00243         new GUI::OptionInputField(getContext().fonts));
00244     GUI::StringInputField::Ptr strField(
00245         new GUI::StringInputField(getContext().fonts));
00246     strField.get() -> SetLabel(label);
00247     strField.get() -> SetInputFieldSize((Vector2){100, 30});
00248     button.get() -> SetOption(title, {strField}, action);
00249     container.get() -> pack(button);
00250 }
00251
00252 #endif // STATES_LINKEDLIST_LLSTATE_HPP
```

## 7.150 src/States/LinkedList/QueueState.cpp File Reference

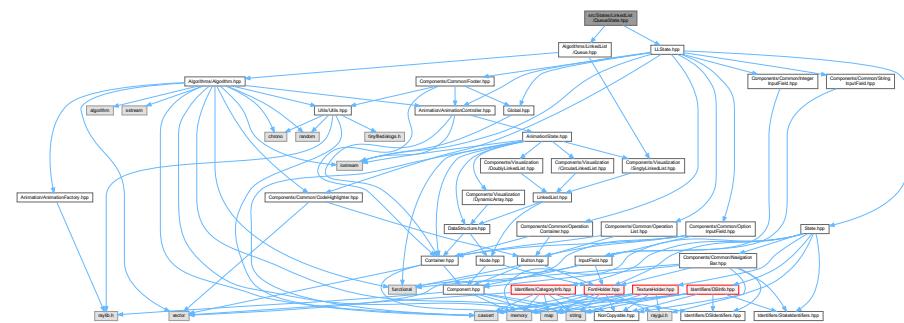
```
#include "QueueState.hpp"
#include <iostream>
#include "Global.hpp"
Include dependency graph for QueueState.cpp:
```



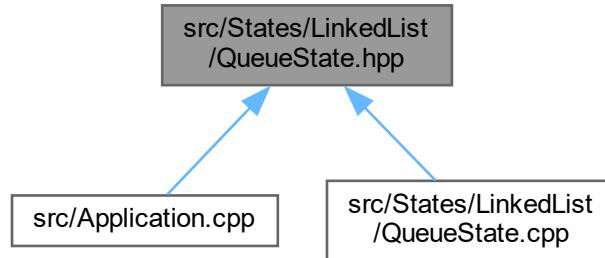
## 7.151 src/States/LinkedList/QueueState.hpp File Reference

```
#include "Algorithms/LinkedList/Queue.hpp"
#include "LLState.hpp"
```

Include dependency graph for QueueState.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class QueueState

## 7.152 QueueState.hpp

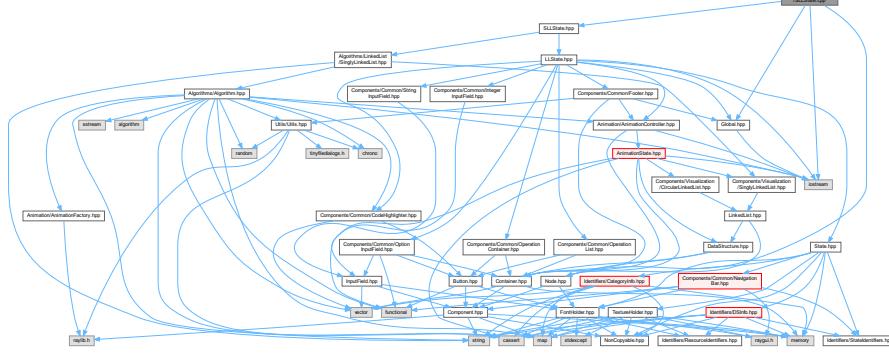
[Go to the documentation of this file.](#)

```
00001 #ifndef STATES_LINKEDLIST_QUEUESTATE_HPP
00002 #define STATES_LINKEDLIST_QUEUESTATE_HPP
00003
00004 #include "Algorithms/LinkedList/Queue.hpp"
00005 #include "LLState.hpp"
00006
00007 class QueueState : public LLState< SLLAnimationController > {
00008 private:
00009     Algorithm::Queue queue;
00010
00011 public:
00012     void AddInsertOperation();
00013     void AddInitializeOperation();
00014     void AddDeleteOperation();
00015     void AddSearchOperation();
00016
00017 public:
00018     QueueState(StateStack& stack, Context context);
00019     ~QueueState();
00020     void Draw();
00021 };
00022
00023 #endif // STATES_LINKEDLIST_QUEUESTATE_HPP
```

## 7.153 src/States/LinkedList/SLLState.cpp File Reference

```
#include "SLLState.hpp"
#include <iostream>
#include "Components/Visualization/Node.hpp"
#include "Global.hpp"
Include dependency graph for SLLState.cpp:
```

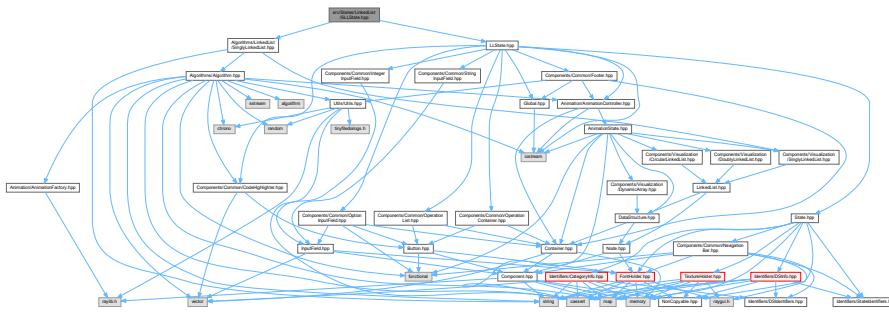
include dependencies given in the `setup.py`.



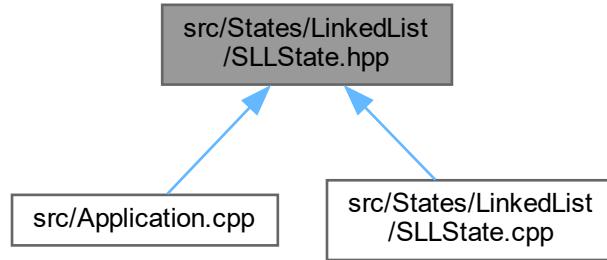
## 7.154 src/States/LinkedList/SLLState.hpp File Reference

```
#include "Algorithms/LinkedList/SinglyLinkedList.hpp"
#include "LLState.hpp"
```

Include dependency graph for SLLState.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [SLLState](#)

## 7.155 SLLState.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef STATES_LINKEDLIST_SLLSTATE_HPP
00002 #define STATES_LINKEDLIST_SLLSTATE_HPP
00003
00004 #include "Algorithms/LinkedList/SinglyLinkedList.hpp"
00005 #include "LLState.hpp"
00006
00007 class SLLState : public LLState< SLLAnimationController > {
00008 private:
00009     Algorithm::SinglyLinkedList SLL;
00010
00011 private:
00012     void AddInsertOperation();
00013     void AddInitializeOperation();
00014     void AddUpdateOperation();
00015     void AddDeleteOperation();
00016     void AddSearchOperation();
00017
00018 protected:
00019 public:
00020     SLLState(StateStack& stack, Context context);
00021     ~SLLState();
00022     void Draw();
00023 };
00024
00025 #endif // STATES_LINKEDLIST_SLLSTATE_HPP
  
```

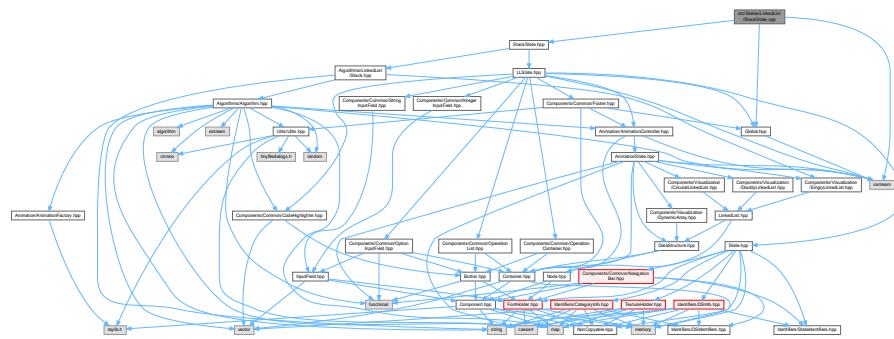
## 7.156 src/States/LinkedList/StackState.cpp File Reference

```

#include "StackState.hpp"
#include <iostream>
  
```

```
#include "Global.hpp"
```

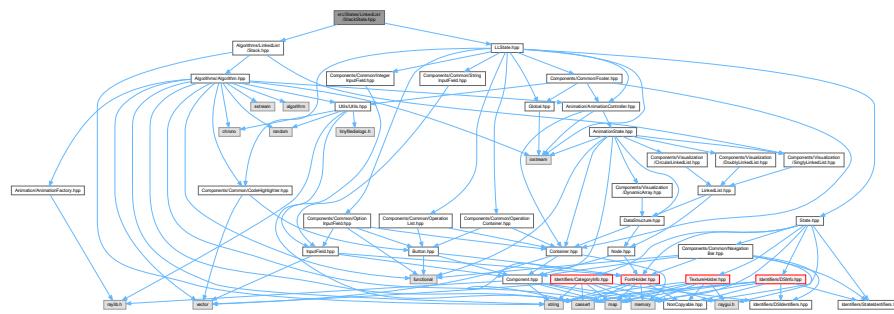
Include dependency graph for StackState.cpp:



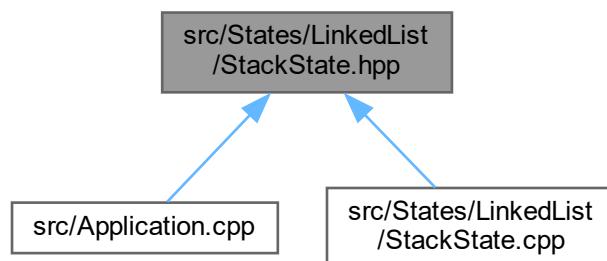
## 7.157 src/States/LinkedList/StackState.hpp File Reference

```
#include "Algorithms/LinkedList/Stack.hpp"
#include "LLState.hpp"
```

Include dependency graph for StackState.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [StackState](#)

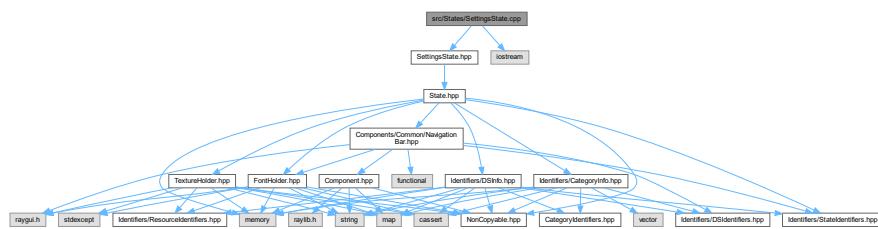
## 7.158 StackState.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef STATES_LINKEDLIST_STACKSTATE_HPP
00002 #define STATES_LINKEDLIST_STACKSTATE_HPP
00003
00004 #include "Algorithms/LinkedList/Stack.hpp"
00005 #include "LLState.hpp"
00006
00007 class StackState : public LLState< SLLAnimationController > {
00008 private:
00009     Algorithm::Stack mStackAlgorithm;
00010
00011 public:
00012     void AddInsertOperation();
00013     void AddInitializeOperation();
00014     void AddDeleteOperation();
00015     void AddSearchOperation();
00016
00017 public:
00018     StackState(StateStack& stack, Context context);
00019     ~StackState();
00020     void Draw();
00021 };
00022
00023 #endif // STATES_LINKEDLIST_STACKSTATE_HPP
```

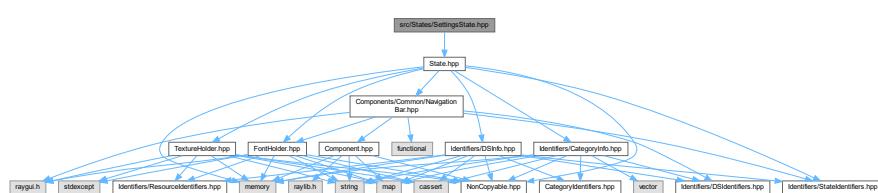
## 7.159 src/States/SettingsState.cpp File Reference

```
#include "SettingsState.hpp"
#include <iostream>
Include dependency graph for SettingsState.cpp:
```

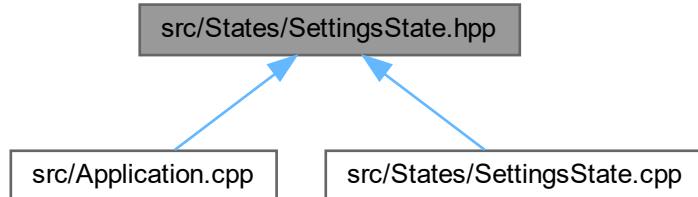


## 7.160 src/States/SettingsState.hpp File Reference

```
#include "State.hpp"
Include dependency graph for SettingsState.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [SettingsState](#)

## 7.161 SettingsState.hpp

[Go to the documentation of this file.](#)

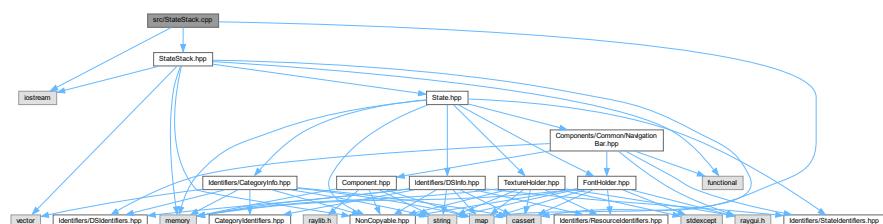
```

00001 #ifndef STATES_SETTINGSSTATE_HPP
00002 #define STATES_SETTINGSSTATE_HPP
00003
00004 #include "State.hpp"
00005
00006 class SettingsState : public State {
00007 private:
00008 public:
00009     SettingsState(StateStack& stack, Context context);
00010     ~SettingsState();
00011     void Draw();
00012     bool Update(float dt);
00013 };
00014
00015 #endif // STATES_SETTINGSSTATE_HPP
  
```

## 7.162 src/StateStack.cpp File Reference

```

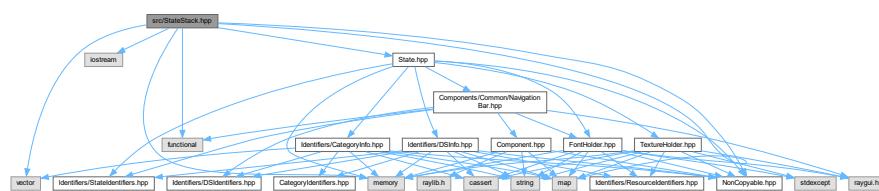
#include "StateStack.hpp"
#include <cassert>
#include <iostream>
Include dependency graph for StateStack.cpp:
  
```



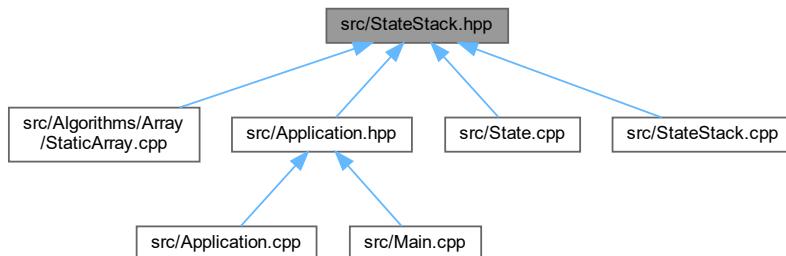
## 7.163 src/StateStack.hpp File Reference

```
#include <functional>
#include <iostream>
#include <map>
#include <memory>
#include <vector>
#include "NonCopyable.hpp"
#include "State.hpp"
```

Include dependency graph for StateStack.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class [StateStack](#)
- struct [StateStack::PendingChange](#)

## 7.164 StateStack.hpp

[Go to the documentation of this file.](#)

```
00001 #ifndef STATESTACK_HPP
00002 #define STATESTACK_HPP
00003
00004 #include <functional>
00005 #include <iostream>
00006 #include <map>
00007 #include <memory>
00008 #include <vector>
00009
00010 #include "NonCopyable.hpp"
00011 #include "State.hpp"
00012
```

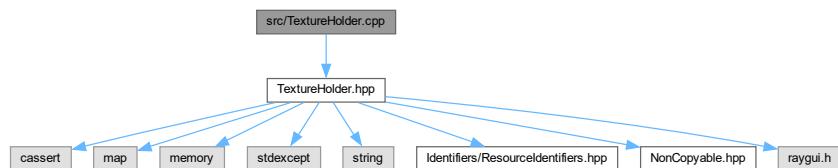
```

00013 class StateStack : private NonCopyable< StateStack > {
00014 public:
00015     enum class Action {
00016         Push,
00017         Pop,
00018         Clear,
00019     };
00020
00021 public:
00022     explicit StateStack(State::Context context);
00023
00024     template< class T >
00025     void RegisterState(States::ID stateID);
00026     void Update(float dt);
00027     void Draw();
00028     // void handleEvent(Event event);
00029     void PushState(States::ID stateID);
00030     void PopState();
00031     void ClearStates();
00032     bool IsEmpty() const;
00033
00034 private:
00035     State::Ptr createState(States::ID stateID);
00036     void ApplyPendingChanges();
00037
00038 private:
00039     struct PendingChange {
00040         explicit PendingChange(Action action,
00041             States::ID stateID = States::None);
00042         Action action;
00043         States::ID stateID;
00044     };
00045
00046 private:
00047     std::vector< State::Ptr > mStack;
00048     std::vector< PendingChange > mPendingList;
00049     State::Context mContext;
00050     std::map< States::ID, std::function< State::Ptr() > > mFactories;
00051 };
00052
00053 template< class T >
00054 void StateStack::RegisterState(States::ID stateID) {
00055     mFactories[stateID] = [this]() {
00056         return State::Ptr(new T(*this, mContext));
00057     };
00058 }
00059
00060 #endif // STATESTACK_HPP

```

## 7.165 src/TextureHolder.cpp File Reference

#include "TextureHolder.hpp"  
Include dependency graph for TextureHolder.cpp:



## 7.166 src/TextureHolder.hpp File Reference

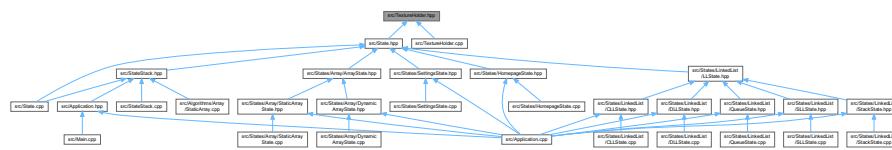
```
#include <cassert>
#include <map>
```

```
#include <memory>
#include <stdexcept>
#include <string>
#include "Identifiers/ResourceIdentifiers.hpp"
#include "NonCopyable.hpp"
#include "raygui.h"

Include dependency graph for TextureHolder.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [TextureHolder](#)

## 7.167 TextureHolder.hpp

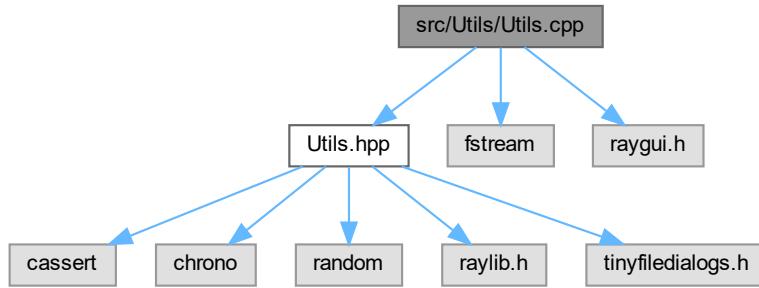
[Go to the documentation of this file.](#)

```

00001 #ifndef IMAGEHOLDER_HPP
00002 #define IMAGEHOLDER_HPP
00003
00004 #include <cassert>
00005 #include <map>
00006 #include <memory>
00007 #include <stdexcept>
00008 #include <string>
00009
00010 #include "Identifiers/ResourceIdentifiers.hpp"
00011 #include "NonCopyable.hpp"
00012 #include "raygui.h"
00013
00014 class TextureHolder : private NonCopyable< TextureHolder > {
00015 public:
00016     TextureHolder();
00017     ~TextureHolder();
00018     void Load(Textures::ID id, const std::string& filename);
00019     void Load(Textures::ID id, const std::string& filename, int width,
00020               int height, bool cropCenter = false);
00021     void LoadFromImage(Textures::ID id, Image& image);
00022     void LoadFromImage(Textures::ID id, Image& image, int width, int height);
00023     Texture& Get(Textures::ID id);
00024     const Texture& Get(Textures::ID id) const;
00025
00026 private:
00027     void InsertResource(Textures::ID id, std::unique_ptr< Texture > texture);
00028
00029 private:
00030     std::map< Textures::ID, std::unique_ptr< Texture > > mTextureMap;
00031 };
00032
00033 #endif // IMAGEHOLDER_HPP
  
```

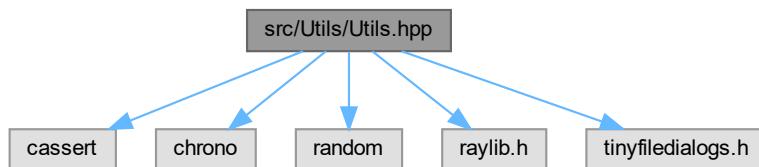
## 7.168 src/Utils/Utils.cpp File Reference

```
#include "Utils.hpp"
#include <fstream>
#include "raygui.h"
Include dependency graph for Utils.cpp:
```

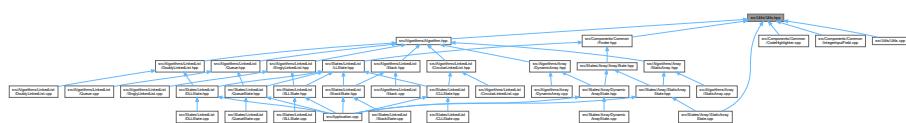


## 7.169 src/Utils/Utils.hpp File Reference

```
#include <cassert>
#include <chrono>
#include <random>
#include "raylib.h"
#include "tinyfiledialogs.h"
Include dependency graph for Utils.hpp:
```



This graph shows which files directly or indirectly include this file:



## Namespaces

- namespace [Utils](#)

## Functions

- std::string [Utils::OpenFileDialog](#) (std::string title, std::string description, std::vector< std::string > filters, std::string defaultPath, bool allowMultipleSelect)
- std::string [Utils::ReadInputFromFile](#) (std::string path)
- int [Utils::Rand](#) (int lower, int upper)
- void [Utils::DrawTextBoxed](#) (Font font, const char \*text, Rectangle rec, float fontSize, float spacing, bool wordWrap, Color tint)
- void [Utils::DrawTextBoxedSelectable](#) (Font font, const char \*text, Rectangle rec, float fontSize, float spacing, bool wordWrap, Color tint, int selectStart, int selectLength, Color selectTint, Color selectBackTint)
- bool [Utils::DrawIcon](#) (int iconID, int x, int y, int pixelSize, Color color, Color hoverColor)

## 7.170 Utils.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef UTILS_HPP
00002 #define UTILS_HPP
00003
00004 #include <cassert>
00005 #include <chrono>
00006 #include <random>
00007
00008 #include "raylib.h"
00009 #include "tinyfiledialogs.h"
00010
00011 namespace Utils {
00012     std::string OpenFileDialog(std::string title, std::string description,
00013                             std::vector< std::string > filters,
00014                             std::string defaultPath,
00015                             bool allowMultipleSelect);
00016
00017     std::string ReadInputFromFile(std::string path);
00018
00019     // Returns a random integer in the range [lower, upper]
00020     int Rand(int lower, int upper);
00021
00022     // Draw text using font inside rectangle limits
00023     void DrawTextBoxed(Font font, const char* text, Rectangle rec,
00024                         float fontSize, float spacing, bool wordWrap,
00025                         Color tint);
00026
00027     // Draw text using font inside rectangle limits with support for text
00028     // selection};
00029     void DrawTextBoxedSelectable(Font font, const char* text, Rectangle rec,
00030                                 float fontSize, float spacing, bool wordWrap,
00031                                 Color tint, int selectStart, int selectLength,
00032                                 Color selectTint, Color selectBackTint);
00033     bool DrawIcon(int iconID, int x, int y, int pixelSize, Color color,
00034                   Color hoverColor);
00035 }; // namespace Utils
00036
00037 #endif // UTILS_HPP

```

# Index

~Algorithm  
    Algorithm::Algorithm< GUIAlgorithm, AnimationState >, [24](#)  
~AnimationController  
    Animation::AnimationController< T >, [29](#)  
~AnimationState  
    Animation::AnimationState< T >, [35](#)  
~Application  
    Application, [40](#)  
~ArrayState  
    ArrayState< T >, [45](#)  
~Button  
    GUI::Button, [55](#)  
~CLLState  
    CLLState, [94](#)  
~Card  
    GUI::Card, [65](#)  
~CircularLinkedList  
    Algorithm::CircularLinkedList, [74](#)  
    GUI::CircularLinkedList, [83](#)  
~CodeHighlighter  
    GUI::CodeHighlighter, [102](#)  
~Component  
    GUI::Component, [110](#)  
~DLLState  
    DLLState, [135](#)  
~DataStructure  
    GUI::DataStructure, [128](#)  
~DoublyLinkedList  
    Algorithm::DoublyLinkedList, [143](#)  
    GUI::DoublyLinkedList, [152](#)  
~DynamicArray  
    Algorithm::DynamicArray, [164](#)  
    GUI::DynamicArray, [171](#)  
~DynamicArrayState  
    DynamicArrayState, [182](#)  
~FontHolder  
    FontHolder, [189](#)  
~Footer  
    GUI::Footer< T >, [193](#)  
~HomepageState  
    HomepageState, [200](#)  
~InputField  
    GUI::InputField, [208](#)  
~IntegerInputField  
    GUI::IntegerInputField, [219](#)  
~LLState  
    LLState< T >, [251](#)  
~LinkedList  
    GUI::LinkedList, [234](#)  
~NavigationBar  
    GUI::NavigationBar, [259](#)  
~Node  
    GUI::Node, [272](#)  
~NonCopyable  
    NonCopyable< T >, [282](#)  
~OperationContainer  
    GUI::OperationContainer, [285](#)  
~OperationList  
    GUI::OperationList, [292](#)  
~OptionInputField  
    GUI::OptionInputField, [300](#)  
~Queue  
    Algorithm::Queue, [310](#)  
~QueueState  
    QueueState, [317](#)  
~SLLState  
    SLLState, [353](#)  
~SceneNode  
    SceneNode, [323](#)  
~Settings  
    Settings, [326](#)  
~SettingsState  
    SettingsState, [329](#)  
~SinglyLinkedList  
    Algorithm::SinglyLinkedList, [334](#)  
    GUI::SinglyLinkedList, [342](#)  
~Stack  
    Algorithm::Stack, [361](#)  
~StackState  
    StackState, [368](#)  
~State  
    State, [375](#)  
~StaticArray  
    Algorithm::StaticArray, [383](#)  
~StaticArrayState  
    StaticArrayState, [389](#)  
~StringInputField  
    GUI::StringInputField, [398](#)  
~TextureHolder  
    TextureHolder, [406](#)  
abbr  
    DSInfo::Info, [204](#)  
abbrTitle  
    GUI::NavigationBar::TitleInfo, [408](#)  
Access  
    Algorithm::DynamicArray, [164](#)  
    Algorithm::StaticArray, [383](#)

Action  
     StateStack, 378

action  
     GUI::Button, 60  
     StateStack::PendingChange, 306

actionDescription  
     Animation::AnimationState< T >, 38

ActionDescription\_Background  
     ColorTheme, 15

ActionList\_Background  
     ColorTheme, 15

ActionList\_HoverBackground  
     ColorTheme, 15

ActionList\_Text  
     ColorTheme, 15

Active  
     GUI::CircularLinkedList, 82  
     GUI::DoublyLinkedList, 151  
     GUI::LinkedList, 233  
     GUI::Node, 272  
     GUI::SinglyLinkedList, 341

ActiveBlue  
     GUI::Node, 272

activeDS  
     ArrayState< T >, 49  
     CLLState, 98  
     DLLState, 139  
     DynamicArrayList, 186  
     LLState< T >, 255  
     QueueState, 321  
     SLLState, 357  
     StackState, 372  
     StaticArrayList, 393

ActiveGreen  
     GUI::Node, 272

ActiveRed  
     GUI::Node, 272

activeTitle  
     GUI::NavigationBar, 264

AddAccessOperation  
     ArrayState< T >, 45  
     DynamicArrayList, 182  
     StaticArrayList, 389

AddActionDescription  
     GUI::CodeHighlighter, 102

AddAnimation  
     Animation::AnimationController< T >, 29

AddCode  
     GUI::CodeHighlighter, 102

AddColor  
     GUI::Node, 273

AddDeleteOperation  
     ArrayState< T >, 45  
     CLLState, 94  
     DLLState, 135  
     DynamicArrayList, 182  
     LLState< T >, 251  
     QueueState, 317

AddInitializeOperation  
     ArrayState< T >, 46  
     CLLState, 94  
     DLLState, 135  
     DynamicArrayList, 182  
     LLState< T >, 251  
     QueueState, 317  
     SLLState, 353  
     StackState, 368  
     StaticArrayList, 390

AddInputField  
     GUI::OptionInputField, 300

AddInsertOperation  
     ArrayState< T >, 46  
     CLLState, 94  
     DLLState, 135  
     DynamicArrayList, 183  
     LLState< T >, 251  
     QueueState, 317  
     SLLState, 353  
     StackState, 368  
     StaticArrayList, 390

AddIntegerFieldOperationOption  
     ArrayState< T >, 46  
     CLLState, 95  
     DLLState, 135  
     DynamicArrayList, 183  
     LLState< T >, 252  
     QueueState, 317  
     SLLState, 354  
     StackState, 368  
     StaticArrayList, 390

AddNoFieldOperationOption  
     ArrayState< T >, 46  
     CLLState, 95  
     DLLState, 136  
     DynamicArrayList, 183  
     LLState< T >, 252  
     QueueState, 318  
     SLLState, 354  
     StackState, 369  
     StaticArrayList, 390

AddOperation  
     GUI::OperationList, 292

AddOperations  
     ArrayState< T >, 46  
     CLLState, 95  
     DLLState, 136  
     DynamicArrayList, 183  
     LLState< T >, 252  
     QueueState, 318  
     SLLState, 354  
     StackState, 369  
     StaticArrayList, 390

AddSearchOperation

ArrayState< T >, 47  
    CLLState, 95  
    DLLState, 136  
    DynamicArrayList, 183  
    LLState< T >, 252  
    QueueState, 318  
    SLLState, 354  
    StackState, 369  
    StaticArrayList, 390  
AddStringFieldOption  
    ArrayState< T >, 47  
    CLLState, 95  
    DLLState, 136  
    DynamicArrayList, 183  
    LLState< T >, 252  
    QueueState, 318  
    SLLState, 354  
    StackState, 369  
    StaticArrayList, 391  
AddSubmitField  
    GUI::OptionInputField, 300  
AddUpdateOperation  
    ArrayState< T >, 47  
    CLLState, 95  
    DLLState, 136  
    DynamicArrayList, 184  
    LLState< T >, 253  
    QueueState, 318  
    SLLState, 354  
    StackState, 369  
    StaticArrayList, 391  
Algorithm, 11  
    Algorithm::Algorithm< GUIAlgorithm, Animation-  
        State >, 24  
Algorithm::Algorithm< GUIAlgorithm, AnimationState  
    >, 23  
    ~Algorithm, 24  
    Algorithm, 24  
    animController, 27  
    ApplyInput, 24  
    codeHighlighter, 27  
    Empty, 24  
    EmptyGenerator, 24  
    GenerateAnimation, 25  
    GenerateRelayoutAnimation, 25  
    InitAction, 25  
    Random, 25  
    RandomFixedSize, 25  
    RandomFixedSizeGenerator, 25  
    RandomGenerator, 26  
    ReadFromExternalFile, 26  
    ReadFromFileGenerator, 26  
    UserDefined, 26  
    UserDefinedGenerator, 26  
    visualizer, 27  
Algorithm::CircularLinkedList, 72  
    ~CircularLinkedList, 74  
    animController, 78  
    ApplyInput, 74  
    CircularLinkedList, 74  
    codeHighlighter, 78  
    DeleteHead, 74  
    DeleteMiddle, 74  
    DeleteTail, 74  
    Empty, 75  
    EmptyGenerator, 75  
    GenerateAnimation, 75  
    GenerateRelayoutAnimation, 75  
    HighlightArrowFromCur, 75  
    HighlightCircularArrow, 75  
    InitAction, 76  
    InsertAfterTail, 76  
    InsertHead, 76  
    InsertMiddle, 76  
    maxN, 78  
    Random, 76  
    RandomFixedSize, 76  
    RandomFixedSizeGenerator, 76  
    RandomGenerator, 77  
    ReadFromExternalFile, 77  
    ReadFromFileGenerator, 77  
    ResetVisualizer, 77  
    Search, 77  
    size, 77  
    Update, 77  
    UserDefined, 78  
    UserDefinedGenerator, 78  
    visualizer, 78  
Algorithm::DoublyLinkedList, 140  
    ~DoublyLinkedList, 143  
    animController, 147  
    ApplyInput, 143  
    codeHighlighter, 147  
    DeleteHead, 143  
    DeleteMiddle, 143  
    DeleteTail, 143  
    DoublyLinkedList, 142, 143  
    Empty, 144  
    EmptyGenerator, 144  
    GenerateAnimation, 144  
    GenerateRelayoutAnimation, 144  
    HighlightArrowBoth, 144  
    HighlightArrowNext, 144  
    HighlightArrowPrev, 145  
    InitAction, 145  
    InsertAfterTail, 145  
    InsertHead, 145  
    InsertMiddle, 145  
    maxN, 148  
    Random, 145  
    RandomFixedSize, 146  
    RandomFixedSizeGenerator, 146  
    RandomGenerator, 146  
    ReadFromExternalFile, 146  
    ReadFromFileGenerator, 146  
    ResetVisualizer, 146

Search, 146  
 size, 147  
 Update, 147  
 UserDefined, 147  
 UserDefinedGenerator, 147  
 visualizer, 148  
**Algorithm::DynamicArray**, 162  
 ~DynamicArray, 164  
 Access, 164  
 animController, 167  
 ApplyInput, 164  
 codeHighlighter, 167  
 DynamicArray, 163  
 Empty, 164  
 EmptyGenerator, 164  
 GenerateAnimation, 164  
 GenerateRelayoutAnimation, 165  
 InitAction, 165  
 maxN, 167  
 PopBack, 165  
 PushBack, 165  
 Random, 165  
 RandomFixedSize, 165  
 RandomFixedSizeGenerator, 165  
 RandomGenerator, 166  
 ReadFromExternalFile, 166  
 ReadFromFileGenerator, 166  
 ResetVisualizer, 166  
 Search, 166  
 size, 166  
 Update, 166  
 UserDefined, 167  
 UserDefinedGenerator, 167  
 visualizer, 167  
**Algorithm::Queue**, 307  
 ~Queue, 310  
 animController, 313  
 ApplyInput, 310  
 codeHighlighter, 313  
 Dequeue, 310  
 Empty, 310  
 EmptyGenerator, 310  
 Enqueue, 310  
 EnqueueEmpty, 310  
 GenerateAnimation, 311  
 GenerateRelayoutAnimation, 311  
 HighlightArrowFromCur, 311  
 InitAction, 311  
 maxN, 313  
 PeekBack, 311  
 PeekFront, 311  
 Queue, 309  
 Random, 312  
 RandomFixedSize, 312  
 RandomFixedSizeGenerator, 312  
 RandomGenerator, 312  
 ReadFromExternalFile, 312  
 ReadFromFileGenerator, 312  
 size, 312  
 UserDefined, 313  
 UserDefinedGenerator, 313  
 visualizer, 313  
**Algorithm::SinglyLinkedList**, 332  
 ~SinglyLinkedList, 334  
 animController, 338  
 ApplyInput, 334  
 codeHighlighter, 338  
 DeleteHead, 334  
 DeleteMiddle, 334  
 DeleteTail, 334  
 Empty, 335  
 EmptyGenerator, 335  
 GenerateAnimation, 335  
 GenerateRelayoutAnimation, 335  
 HighlightArrowFromCur, 335  
 InitAction, 335  
 InsertAfterTail, 336  
 InsertHead, 336  
 InsertMiddle, 336  
 maxN, 338  
 Random, 336  
 RandomFixedSize, 336  
 RandomFixedSizeGenerator, 336  
 RandomGenerator, 336  
 ReadFromExternalFile, 337  
 ReadFromFileGenerator, 337  
 ResetVisualizer, 337  
 Search, 337  
 SinglyLinkedList, 333, 334  
 size, 337  
 Update, 337  
 UserDefined, 337  
 UserDefinedGenerator, 338  
 visualizer, 338  
**Algorithm::Stack**, 359  
 ~Stack, 361  
 animController, 364  
 ApplyInput, 361  
 codeHighlighter, 364  
 Empty, 361  
 EmptyGenerator, 361  
 GenerateAnimation, 361  
 GenerateRelayoutAnimation, 362  
 HighlightArrowFromCur, 362  
 InitAction, 362  
 maxN, 364  
 mStackOrientation, 364  
 Peek, 362  
 Pop, 362  
 Push, 362  
 Random, 363  
 RandomFixedSize, 363  
 RandomFixedSizeGenerator, 363  
 RandomGenerator, 363  
 ReadFromExternalFile, 363  
 ReadFromFileGenerator, 363

size, 363  
Stack, 360, 361  
UserDefined, 364  
UserDefinedGenerator, 364  
visualizer, 365  
Algorithm::StaticArray, 381  
  ~StaticArray, 383  
  Access, 383  
  animController, 386  
  ApplyInput, 383  
  codeHighlighter, 386  
  Empty, 383  
  EmptyGenerator, 383  
  GenerateAnimation, 383  
  GenerateRelayoutAnimation, 384  
  InitAction, 384  
  maxN, 386  
  Random, 384  
  RandomFixedSize, 384  
  RandomFixedSizeGenerator, 384  
  RandomGenerator, 384  
  ReadFromExternalFile, 385  
  ReadFromFileGenerator, 385  
  ResetVisualizer, 385  
  Search, 385  
  size, 385  
  StaticArray, 382  
  Update, 385  
  UserDefined, 385  
  UserDefinedGenerator, 386  
  visualizer, 386  
alignment  
  GUI::Button, 60  
AlignmentCount  
  GUI::Button, 54  
AllFieldDisableEdit  
  GUI::InputField, 208  
  GUI::IntegerField, 219  
  GUI::StringInputField, 398  
animateNode  
  GUI::Node, 278  
Animation, 11  
Animation::AnimationController< T >, 27  
  ~AnimationController, 29  
  AddAnimation, 29  
  AnimationController, 29  
  animationGroup, 33  
  Clear, 29  
  Continue, 29  
  CurrentAnimationIndex, 29  
  currStopDuration, 33  
  defaultSpeed, 33  
  Done, 30  
  GetAnimateFrame, 30  
  GetAnimation, 30  
  GetAnimationDuration, 30  
  GetAnimationIndex, 30  
  GetNumAnimation, 30  
  GetSpeed, 30  
  GetStopDuration, 31  
  InteractionAllow, 31  
  InteractionLock, 31  
  interactionLock, 33  
  IsInteractionAllow, 31  
  IsPlaying, 31  
  mCurrentAnimationIndex, 33  
  mSpeed, 34  
  Pause, 31  
  Playing, 34  
  PopAnimation, 31  
  Ptr, 28  
  Reset, 32  
  ResetCurrent, 32  
  RunAll, 32  
  SetAnimation, 32  
  SetSpeed, 32  
  StepBackward, 32  
  StepForward, 32  
  stopDuration, 34  
  Update, 33  
  Animation::AnimationState< T >, 34  
    ~AnimationState, 35  
    actionDescription, 38  
    AnimationState, 35  
    Done, 36  
    Draw, 36  
    GetActionDescription, 36  
    GetCurrentPlayingAt, 36  
    GetDataStructure, 36  
    GetDuration, 36  
    GetHighlightedLine, 37  
    mAnimation, 38  
    mCurrentPlayingAt, 38  
    mDataStructureBefore, 39  
    mDuration, 39  
    mHighlightedLine, 39  
    PlayingAt, 37  
    Ptr, 35  
    Reset, 37  
    SetActionDescription, 37  
    SetAnimation, 37  
    SetDuration, 37  
    SetHighlightLine, 38  
    SetSourceDataStructure, 38  
    Update, 38  
  AnimationController  
    Animation::AnimationController< T >, 29  
  AnimationController.hpp  
    CLLAnimationController, 438  
    DArrayAnimationController, 438  
    DLLAnimationController, 438  
    SLLAnimationController, 438  
  AnimationFactory, 11  
    BlendColor, 12  
    BounceOut, 12  
    Dist, 12

DrawActiveArrow, 12  
 DrawCircularArrow, 12  
 DrawDirectionalArrow, 12  
 DrawDoubleActiveArrow, 13  
 DrawDoubleDirectionalArrow, 13  
 ElasticOut, 13  
 InverseVector, 13  
 MoveNode, 13  
 ReCalculateEnds, 14  
 animationGroup  
     Animation::AnimationController< T >, 33  
 AnimationOnNode  
     GUI::Node, 273  
 AnimationState  
     Animation::AnimationState< T >, 35  
 AnimationState.hpp  
     CLLAnimation, 444  
     DArrayAnimation, 445  
     DLLAnimation, 445  
     SLLAnimation, 445  
 animController  
     Algorithm::Algorithm< GUIAlgorithm, Animation-  
         State >, 27  
     Algorithm::CircularLinkedList, 78  
     Algorithm::DoublyLinkedList, 147  
     Algorithm::DynamicArray, 167  
     Algorithm::Queue, 313  
     Algorithm::SinglyLinkedList, 338  
     Algorithm::Stack, 364  
     Algorithm::StaticArray, 386  
     ArrayState< T >, 49  
     CLLState, 98  
     DLLState, 139  
     DynamicArrayList, 186  
     LLState< T >, 255  
     QueueState, 321  
     SLLState, 357  
     StackState, 372  
     StaticArrayList, 393  
 Application, 39  
     ~Application, 40  
     Application, 40  
     categories, 41  
     Close, 40  
     closed, 41  
     dsInfo, 42  
     fonts, 42  
     images, 42  
     Init, 40  
     LoadResources, 40  
     mStack, 42  
     RegisterStates, 41  
     Render, 41  
     Run, 41  
     Update, 41  
     WindowClosed, 41  
 Application.cpp  
     RAYGUI\_IMPLEMENTATION, 448

ApplyInput  
 Algorithm::Algorithm< GUIAlgorithm, Animation-  
     State >, 24  
 Algorithm::CircularLinkedList, 74  
 Algorithm::DoublyLinkedList, 143  
 Algorithm::DynamicArray, 164  
 Algorithm::Queue, 310  
 Algorithm::SinglyLinkedList, 334  
 Algorithm::Stack, 361  
 Algorithm::StaticArray, 383  
 ApplyPendingChanges  
     StateStack, 379  
 Array  
     Category, 14  
 ArrayState  
     ArrayState< T >, 45  
 ArrayState< T >, 43  
     ~ArrayState, 45  
     activeDS, 49  
     AddAccessOperation, 45  
     AddDeleteOperation, 45  
     AddInitializeOperation, 46  
     AddInsertOperation, 46  
     AddIntFieldOperationOption, 46  
     AddNoFieldOperationOption, 46  
     AddOperations, 46  
     AddSearchOperation, 47  
     AddStringFieldOption, 47  
     AddUpdateOperation, 47  
     animController, 49  
     ArrayState, 45  
     ClearAction, 47  
     ClearError, 47  
     codeHighlighter, 50  
     Draw, 47  
     DrawCurrentActionText, 48  
     DrawCurrentErrorText, 48  
     footer, 50  
     GetContext, 48  
     InitNavigationBar, 48  
     mContext, 50  
     mCurrentAction, 50  
     mCurrentError, 50  
     mStack, 50  
     navigation, 50  
     operationList, 51  
     Ptr, 45  
     RequestStackClear, 48  
     RequestStackPop, 48  
     RequestStackPush, 48  
     SetCurrentAction, 49  
     SetCurrentError, 49  
     Success, 49  
     Update, 49  
 ArrayState< T >::IntegerInput, 214  
     label, 214  
     maxValue, 214  
     minValue, 214

width, 214  
arrowState  
    GUI::CircularLinkedList, 90  
    GUI::SinglyLinkedList, 348  
arrowStateNext  
    GUI::DoublyLinkedList, 158  
arrowStatePrev  
    GUI::DoublyLinkedList, 159  
ArrowType  
    CircularLinkedList.cpp, 419  
    DoublyLinkedList.cpp, 424  
    GUI::CircularLinkedList, 82  
    GUI::DoublyLinkedList, 151  
    GUI::LinkedList, 233  
    GUI::SinglyLinkedList, 341  
    Queue.cpp, 428  
    SinglyLinkedList.cpp, 431  
    Stack.hpp, 436  
ArrowTypeCount  
    GUI::CircularLinkedList, 82  
    GUI::DoublyLinkedList, 151  
    GUI::LinkedList, 233  
    GUI::SinglyLinkedList, 341  
at  
    Core::List< T >, 243  
AtSettings  
    GUI::NavigationBar, 260  
atSettings  
    GUI::NavigationBar, 264  
AttachChild  
    SceneNode, 323  
  
back  
    Core::List< T >, 243  
Background  
    ColorTheme, 15  
backToPrvState  
    GUI::NavigationBar, 264  
begin  
    Core::List< T >, 243  
Blank  
    Textures, 20  
BlendColor  
    AnimationFactory, 12  
BounceOut  
    AnimationFactory, 12  
bound  
    GUI::Button, 60  
Button  
    GUI::Button, 54, 55  
buttonColor  
    GUI::Button, 60  
buttonHoverColor  
    GUI::Button, 61  
buttons  
    GUI::OperationList, 296  
  
capacity  
    GUI::DynamicArray, 177  
  
Card  
    GUI::Card, 65  
Card\_Background  
    ColorTheme, 15  
Card\_Text  
    ColorTheme, 15  
categories  
    Application, 41  
    State::Context, 125  
Category, 14  
    Array, 14  
    Count, 14  
    ID, 14  
    LinkedList, 14  
    None, 14  
categoryID  
    CategoryInfo::Info, 203  
    DSInfo::Info, 204  
CategoryInfo, 70  
    Get, 71  
    mFactories, 71  
    Register, 71  
CategoryInfo::Info, 202  
    categoryID, 203  
    categoryName, 203  
    Info, 202  
    mDS, 203  
categoryName  
    CategoryInfo::Info, 203  
Center  
    GUI::Button, 54  
Circle  
    GUI::Node, 272  
circularArrowState  
    GUI::CircularLinkedList, 90  
CircularLinkedList  
    Algorithm::CircularLinkedList, 74  
    DataStructures, 17  
    GUI::CircularLinkedList, 83  
    States, 20  
    Textures, 20  
CircularLinkedList.cpp  
    ArrowType, 419  
Clear  
    Animation::AnimationController< T >, 29  
    GUI::DynamicArray, 171  
    StateStack, 378  
clear  
    Core::List< T >, 243  
ClearAction  
    ArrayState< T >, 47  
    CLLState, 96  
    DLLState, 136  
    DynamicArrayList, 184  
    LLState< T >, 253  
    QueueState, 318  
    SLLState, 355  
    StackState, 369

StaticArrayState, 391  
 ClearError  
     ArrayState< T >, 47  
     CLLState, 96  
     DLLState, 137  
     DynamicArrayState, 184  
     LLState< T >, 253  
     QueueState, 319  
     SLLState, 355  
     StackState, 370  
     StaticArrayState, 391  
 ClearLabel  
     GUI::Node, 273  
 ClearStates  
     StateStack, 379  
 ClearTitle  
     GUI::NavigationBar, 260  
 CLL  
     CLLState, 98  
 CLLAnimation  
     AnimationState.hpp, 444  
 CLLAnimationController  
     AnimationController.hpp, 438  
 CLLState, 92  
     ~CLLState, 94  
     activeDS, 98  
     AddDeleteOperation, 94  
     AddInitializeOperation, 94  
     AddInsertOperation, 94  
     AddIntFieldOperationOption, 95  
     AddNoFieldOperationOption, 95  
     AddOperations, 95  
     AddSearchOperation, 95  
     AddStringFieldOption, 95  
     AddUpdateOperation, 95  
     animController, 98  
     ClearAction, 96  
     ClearError, 96  
     CLL, 98  
     CLLState, 94  
     codeHighlighter, 98  
     Draw, 96  
     DrawCurrentActionText, 96  
     DrawCurrentErrorText, 96  
     footer, 98  
     GetContext, 96  
     InitNavigationBar, 96  
     mContext, 98  
     mCurrentAction, 99  
     mCurrentError, 99  
     mStack, 99  
     navigation, 99  
     operationList, 99  
     Ptr, 94  
     RequestStackClear, 97  
     RequestStackPop, 97  
     RequestStackPush, 97  
     SetCurrentAction, 97  
     SetCurrentError, 97  
     Success, 97  
     Update, 97  
 Close  
     Application, 40  
 closed  
     Application, 41  
 CodeHighlighter  
     GUI::CodeHighlighter, 102  
 codeHighlighter  
     Algorithm::Algorithm< GUIAlgorithm, AnimationState >, 27  
     Algorithm::CircularLinkedList, 78  
     Algorithm::DoublyLinkedList, 147  
     Algorithm::DynamicArray, 167  
     Algorithm::Queue, 313  
     Algorithm::SinglyLinkedList, 338  
     Algorithm::Stack, 364  
     Algorithm::StaticArray, 386  
     ArrayState< T >, 50  
     CLLState, 98  
     DLLState, 139  
     DynamicArrayState, 186  
     LLState< T >, 255  
     QueueState, 321  
     SLLState, 357  
     StackState, 372  
     StaticArrayState, 393  
 CodeHighlighter\_Background  
     ColorTheme, 15  
 ColorTheme, 14  
     ActionDescription\_Background, 15  
     ActionList\_Background, 15  
     ActionList\_HoverBackground, 15  
     ActionList\_Text, 15  
     Background, 15  
     Card\_Background, 15  
     Card\_Text, 15  
     CodeHighlighter\_Background, 15  
     Count, 16  
     ID, 15  
     Logo1FirstPart, 15  
     Logo1SecondPart, 15  
     Logo2FirstPart, 15  
     Logo2SecondPart, 15  
     NavigationBar\_Background, 15  
     NavigationBar\_SelectedTitle, 15  
     NavigationBar\_UnselectedTitle, 15  
     Visualizer\_Arrow\_Active, 16  
     Visualizer\_Arrow\_Default, 16  
     Visualizer\_Label, 15  
     Visualizer\_Node\_Active\_Background1, 16  
     Visualizer\_Node\_Active\_Background2, 16  
     Visualizer\_Node\_Active\_Outline1, 16  
     Visualizer\_Node\_Active\_Outline2, 16  
     Visualizer\_Node\_Active\_Text1, 16  
     Visualizer\_Node\_Active\_Text2, 16  
     Visualizer\_Node\_ActiveBlue\_Background1, 16

Visualizer\_Node\_ActiveBlue\_Background2, 16  
Visualizer\_Node\_ActiveBlue\_Outline1, 16  
Visualizer\_Node\_ActiveBlue\_Outline2, 16  
Visualizer\_Node\_ActiveBlue\_Text1, 16  
Visualizer\_Node\_ActiveBlue\_Text2, 16  
Visualizer\_Node\_ActiveGreen\_Background1, 16  
Visualizer\_Node\_ActiveGreen\_Background2, 16  
Visualizer\_Node\_ActiveGreen\_Outline1, 16  
Visualizer\_Node\_ActiveGreen\_Outline2, 16  
Visualizer\_Node\_ActiveGreen\_Text1, 16  
Visualizer\_Node\_ActiveGreen\_Text2, 16  
Visualizer\_Node\_ActiveRed\_Background1, 16  
Visualizer\_Node\_ActiveRed\_Background2, 16  
Visualizer\_Node\_ActiveRed\_Outline1, 16  
Visualizer\_Node\_ActiveRed\_Outline2, 16  
Visualizer\_Node\_ActiveRed\_Text1, 16  
Visualizer\_Node\_ActiveRed\_Text2, 16  
Visualizer\_Node\_Default\_Background1, 16  
Visualizer\_Node\_Default\_Background2, 16  
Visualizer\_Node\_Default\_Outline1, 15  
Visualizer\_Node\_Default\_Outline2, 16  
Visualizer\_Node\_Default\_Text1, 16  
Visualizer\_Node\_Default\_Text2, 16  
Visualizer\_Node\_Iterated\_Background1, 16  
Visualizer\_Node\_Iterated\_Background2, 16  
Visualizer\_Node\_Iterated\_Outline1, 16  
Visualizer\_Node\_Iterated\_Outline2, 16  
Visualizer\_Node\_Iterated\_Text1, 16  
Visualizer\_Node\_Iterated\_Text2, 16

Component  
  GUI::Component, 109

Consolas  
  Fonts, 18

const\_iterator  
  Core::List< T >::const\_iterator, 115

Container  
  GUI::Container, 120

content  
  GUI::Button, 61  
  GUI::StringInputField, 402

Context  
  State::Context, 124

Continue  
  Animation::AnimationController< T >, 29

Core, 16

Core::List< T >, 241  
  at, 243  
  back, 243  
  begin, 243  
  clear, 243  
  empty, 244  
  end, 244  
  front, 244  
  get\_iterator, 244  
  insert\_previous, 244  
  List, 242  
  mBegin, 247  
  mEnd, 247

move\_previous, 245  
mSize, 247  
operator=, 245  
operator[], 245  
pop\_back, 245  
pop\_front, 245  
push\_back, 246  
push\_front, 246  
remove, 246  
remove\_if, 246, 247  
reset, 247  
size, 247

Core::List< T >::const\_iterator, 113  
  const\_iterator, 115  
  difference\_type, 114  
  iterator\_category, 114  
  List, 117  
  operator!=, 115  
  operator\*, 115  
  operator+, 116  
  operator++, 116  
  operator-, 116  
  operator->, 117  
  operator--, 116  
  operator=, 117  
  operator==, 117  
  pointer, 114  
  ptr, 118  
  reference, 114  
  swap, 117  
  value\_type, 114

Core::List< T >::iterator, 226  
  difference\_type, 226  
  iterator, 227  
  iterator\_category, 226  
  List, 229  
  operator!=, 227  
  operator\*, 228  
  operator+, 228  
  operator++, 228  
  operator-, 228  
  operator->, 229  
  operator--, 228, 229  
  operator=, 229  
  operator==, 229  
  pointer, 227  
  ptr, 230  
  reference, 227  
  swap, 229  
  value\_type, 227

Core::Node< T >, 266  
  mNext, 267  
  mPrev, 268  
  mValue, 268  
  Node, 267

Count  
  Category, 14  
  ColorTheme, 16

DataStructures, 17  
 Fonts, 18  
 States, 20  
 Textures, 20  
 Courier  
   Fonts, 18  
 Courier\_Bold  
   Fonts, 18  
 CreateCard  
   HomepageState, 200  
 createState  
   StateStack, 379  
 CurrentAnimationIndex  
   Animation::AnimationController< T >, 29  
 currentCategory  
   GUI::NavigationBar, 265  
 currStopDuration  
   Animation::AnimationController< T >, 33  
 DArrayAnimation  
   AnimationState.hpp, 445  
 DArrayAnimationController  
   AnimationController.hpp, 438  
 DataStructure  
   GUI::DataStructure, 128  
 DataStructures, 17  
   CircularLinkedList, 17  
   Count, 17  
   DoublyLinkedList, 17  
   DynamicArray, 17  
   ID, 17  
   None, 17  
   Queue, 17  
   SinglyLinkedList, 17  
   Stack, 17  
   StaticArray, 17  
 Default  
   Fonts, 18  
   GUI::CircularLinkedList, 82  
   GUI::DoublyLinkedList, 151  
   GUI::LinkedList, 233  
   GUI::Node, 272  
   GUI::SinglyLinkedList, 341  
 Default\_Bold  
   Fonts, 18  
 Default\_Italic  
   Fonts, 18  
 defaultColorPath  
   global, 18  
 defaultSpeed  
   Animation::AnimationController< T >, 33  
 DeleteHead  
   Algorithm::CircularLinkedList, 74  
   Algorithm::DoublyLinkedList, 143  
   Algorithm::SinglyLinkedList, 334  
 DeleteMiddle  
   Algorithm::CircularLinkedList, 74  
   Algorithm::DoublyLinkedList, 143  
   Algorithm::SinglyLinkedList, 334  
 DeleteNode  
   GUI::CircularLinkedList, 83  
   GUI::DoublyLinkedList, 152  
   GUI::DynamicArray, 172  
   GUI::LinkedList, 234  
   GUI::SinglyLinkedList, 342  
 DeleteTail  
   Algorithm::CircularLinkedList, 74  
   Algorithm::DoublyLinkedList, 143  
   Algorithm::SinglyLinkedList, 334  
 Dequeue  
   Algorithm::Queue, 310  
 deselect  
   GUI::Button, 55  
   GUI::Card, 65  
   GUI::CircularLinkedList, 83  
   GUI::CodeHighlighter, 102  
   GUI::Component, 110  
   GUI::Container, 120  
   GUI::DataStructure, 128  
   GUI::DoublyLinkedList, 152  
   GUI::DynamicArray, 172  
   GUI::Footer< T >, 193  
   GUI::InputField, 208  
   GUI::IntegerInputField, 219  
   GUI::LinkedList, 234  
   GUI::NavigationBar, 260  
   GUI::Node, 273  
   GUI::OperationContainer, 285  
   GUI::OperationList, 292  
   GUI::OptionInputField, 300  
   GUI::SinglyLinkedList, 342  
   GUI::StringInputField, 398  
 DetachChild  
   SceneNode, 323  
 difference\_type  
   Core::List< T >::const\_iterator, 114  
   Core::List< T >::iterator, 226  
 DisableFitContent  
   GUI::Button, 55  
 Dist  
   AnimationFactory, 12  
 DLLAnimation  
   AnimationState.hpp, 445  
 DLLAnimationController  
   AnimationController.hpp, 438  
 DLLState, 132  
   ~DLLState, 135  
   activeDS, 139  
   AddDeleteOperation, 135  
   AddInitializeOperation, 135  
   AddInsertOperation, 135  
   AddIntFieldOperationOption, 135  
   AddNoFieldOperationOption, 136  
   AddOperations, 136  
   AddSearchOperation, 136  
   AddStringFieldOption, 136  
   AddUpdateOperation, 136

animController, 139  
ClearAction, 136  
ClearError, 137  
codeHighlighter, 139  
DLLState, 135  
Draw, 137  
DrawCurrentActionText, 137  
DrawCurrentErrorText, 137  
footer, 139  
GetContext, 137  
InitNavigationBar, 137  
mContext, 139  
mCurrentAction, 139  
mCurrentError, 139  
mDLL, 139  
mStack, 140  
navigation, 140  
operationList, 140  
Ptr, 134  
RequestStackClear, 137  
RequestStackPop, 138  
RequestStackPush, 138  
SetCurrentAction, 138  
SetCurrentError, 138  
Success, 138  
Update, 138

Done  
    Animation::AnimationController< T >, 30  
    Animation::AnimationState< T >, 36

DoublyLinkedList  
    Algorithm::DoublyLinkedList, 142, 143  
    DataStructures, 17  
    GUI::DoublyLinkedList, 152  
    States, 20  
    Textures, 20

DoublyLinkedList.cpp  
    ArrowType, 424

Draw  
    Animation::AnimationState< T >, 36  
    ArrayState< T >, 47  
    CLLState, 96  
    DLLState, 137  
    DynamicArrayState, 184  
    GUI::Button, 55  
    GUI::Card, 65  
    GUI::CircularLinkedList, 84  
    GUI::CodeHighlighter, 103  
    GUI::Component, 110  
    GUI::Container, 120  
    GUI::DataStructure, 128  
    GUI::DoublyLinkedList, 153  
    GUI::DynamicArray, 172  
    GUI::Footer< T >, 193  
    GUI::InputField, 208  
    GUI::IntegerInputField, 219  
    GUI::LinkedList, 234, 235  
    GUI::NavigationBar, 260  
    GUI::Node, 273

    GUI::OperationContainer, 285  
    GUI::OperationList, 292  
    GUI::OptionInputField, 301  
    GUI::SinglyLinkedList, 343  
    GUI::StringInputField, 398  
    HomepageState, 200  
    LLState< T >, 253  
    QueueState, 319  
    SceneNode, 323  
    SettingsState, 330  
    SLLState, 355  
    StackState, 370  
    State, 375  
    StateStack, 379  
    StaticArrayState, 391

    DrawActionDescription  
        GUI::CodeHighlighter, 103

DrawActiveArrow  
    AnimationFactory, 12

DrawArrow  
    GUI::CircularLinkedList, 84  
    GUI::DoublyLinkedList, 153  
    GUI::SinglyLinkedList, 343

DrawButtonText  
    GUI::Button, 55

DrawCircularArrow  
    AnimationFactory, 12

DrawCodeHighlighter  
    GUI::CodeHighlighter, 103

DrawCurrent  
    GUI::CircularLinkedList, 84  
    GUI::Container, 120  
    GUI::DataStructure, 128  
    GUI::DoublyLinkedList, 153  
    GUI::DynamicArray, 172  
    GUI::Footer< T >, 193  
    GUI::LinkedList, 235  
    GUI::OperationContainer, 285  
    GUI::OperationList, 293  
    GUI::OptionInputField, 301  
    GUI::SinglyLinkedList, 343  
    SceneNode, 324

    DrawCurrentActionText  
        ArrayState< T >, 48  
        CLLState, 96  
        DLLState, 137  
        DynamicArrayState, 184  
        LLState< T >, 253  
        QueueState, 319  
        SLLState, 355  
        StackState, 370  
        StaticArrayState, 391

    DrawCurrentErrorText  
        ArrayState< T >, 48  
        CLLState, 96  
        DLLState, 137  
        DynamicArrayState, 184  
        LLState< T >, 253

QueueState, 319  
 SLLState, 355  
 StackState, 370  
 StaticArrayState, 392  
**DrawDirectionalArrow**  
 AnimationFactory, 12  
**DrawDoubleActiveArrow**  
 AnimationFactory, 13  
**DrawDoubleDirectionalArrow**  
 AnimationFactory, 13  
**DrawField**  
 GUI::InputField, 208  
 GUI::IntegerField, 220  
 GUI::StringInputField, 399  
**DrawIcon**  
 Utils, 21  
**DrawImage**  
 GUI::Card, 65  
**DrawIntroduction**  
 HomepageState, 200  
**DrawLabel**  
 GUI::Node, 273  
**DrawLogo**  
 GUI::NavigationBar, 260  
**DrawNode**  
 GUI::Node, 274  
**DrawSettings**  
 GUI::NavigationBar, 260  
**DrawTextBoxed**  
 Utils, 21  
**DrawTextBoxedSelectable**  
 Utils, 21  
**DrawTitle**  
 GUI::Card, 66  
**DrawTitles**  
 GUI::NavigationBar, 260  
**DSInfo**, 160  
 Get, 161  
 mFactories, 161  
 Register, 161  
**dslInfo**  
 Application, 42  
 State::Context, 125  
**DSInfo::Info**, 203  
 abbr, 204  
 categoryID, 204  
 ID, 204  
 Info, 204  
 name, 204  
 stateID, 204  
 thumbnail, 204  
**DynamicArray**  
 Algorithm::DynamicArray, 163  
 DataStructures, 17  
 GUI::DynamicArray, 171  
 States, 20  
 Textures, 20  
**DynamicArrayState**, 179  
 ~DynamicArrayState, 182  
 activeDS, 186  
 AddAccessOperation, 182  
 AddDeleteOperation, 182  
 AddInitializeOperation, 182  
 AddInsertOperation, 183  
 AddIntFieldOperationOption, 183  
 AddNoFieldOperationOption, 183  
 AddOperations, 183  
 AddSearchOperation, 183  
 AddStringFieldOption, 183  
 AddUpdateOperation, 184  
 animController, 186  
 ClearAction, 184  
 ClearError, 184  
 codeHighlighter, 186  
 Draw, 184  
 DrawCurrentActionText, 184  
 DrawCurrentErrorText, 184  
 DynamicArrayState, 182  
 footer, 186  
 GetContext, 185  
 InitNavigationBar, 185  
 mContext, 187  
 mCurrentAction, 187  
 mCurrentError, 187  
 mDynamicArray, 187  
 mStack, 187  
 navigation, 187  
 operationList, 187  
 Ptr, 182  
 RequestStackClear, 185  
 RequestStackPop, 185  
 RequestStackPush, 185  
 SetCurrentAction, 185  
 SetCurrentError, 185  
 Success, 186  
 Update, 186  
**editMode**  
 GUI::InputField, 212  
 GUI::IntegerField, 224  
 GUI::StringInputField, 402  
**ElasticOut**  
 AnimationFactory, 13  
**Empty**  
 Algorithm::Algorithm< GUIAlgorithm, Animation-  
     State >, 24  
 Algorithm::CircularLinkedList, 75  
 Algorithm::DoublyLinkedList, 144  
 Algorithm::DynamicArray, 164  
 Algorithm::Queue, 310  
 Algorithm::SinglyLinkedList, 335  
 Algorithm::Stack, 361  
 Algorithm::StaticArray, 383  
**empty**  
 Core::List< T >, 244  
**EmptyGenerator**

Algorithm::Algorithm< GUIAlgorithm, AnimationState >, 24  
Algorithm::CircularLinkedList, 75  
Algorithm::DoublyLinkedList, 144  
Algorithm::DynamicArray, 164  
Algorithm::Queue, 310  
Algorithm::SinglyLinkedList, 335  
Algorithm::Stack, 361  
Algorithm::StaticArray, 383  
EnableFitContent  
    GUI::Button, 56  
end  
    Core::List< T >, 244  
Enqueue  
    Algorithm::Queue, 310  
EnqueueEmpty  
    Algorithm::Queue, 310  
extractedValue  
    GUI::InputField, 212  
    GUI::IntegerField, 224  
    GUI::StringInputField, 403  
ExtractInput  
    GUI::OptionInputField, 301  
ExtractValue  
    GUI::InputField, 209  
    GUI::IntegerField, 220  
    GUI::StringInputField, 399  
  
Favicon  
    Textures, 20  
favicon  
    global, 18  
fields  
    GUI::InputField, 212  
    GUI::IntegerField, 224  
    GUI::StringInputField, 403  
FitContent  
    GUI::Button, 56  
fitContent  
    GUI::Button, 61  
FontHolder, 188  
    ~FontHolder, 189  
    FontHolder, 189  
    Get, 189  
    InsertResource, 189  
    Load, 189  
    mFontMap, 190  
Fonts, 17  
    Consolas, 18  
    Count, 18  
    Courier, 18  
    Courier\_Bold, 18  
    Default, 18  
    Default\_Bold, 18  
    Default\_Italic, 18  
    ID, 17  
    Silkscreen, 18  
fonts  
    Application, 42  
    GUI::Button, 61  
    GUI::Card, 68  
    GUI::CircularLinkedList, 90  
    GUI::CodeHighlighter, 106  
    GUI::DoublyLinkedList, 159  
    GUI::DynamicArray, 177  
    GUI::InputField, 213  
    GUI::IntegerField, 224  
    GUI::LinkedList, 239  
    GUI::NavigationBar, 265  
    GUI::Node, 279  
    GUI::OptionInputField, 304  
    GUI::SinglyLinkedList, 348  
    GUI::StringInputField, 403  
    State::Context, 125  
    fontSize  
        GUI::Button, 61  
Footer  
    GUI::Footer< T >, 192  
footer  
    ArrayState< T >, 50  
    CLLState, 98  
    DLLState, 139  
    DynamicArrayList, 186  
    LLState< T >, 255  
    QueueState, 321  
    SLLState, 357  
    StackState, 372  
    StaticArrayList, 394  
front  
    Core::List< T >, 244  
GenerateAnimation  
    Algorithm::Algorithm< GUIAlgorithm, AnimationState >, 25  
    Algorithm::CircularLinkedList, 75  
    Algorithm::DoublyLinkedList, 144  
    Algorithm::DynamicArray, 164  
    Algorithm::Queue, 311  
    Algorithm::SinglyLinkedList, 335  
    Algorithm::Stack, 361  
    Algorithm::StaticArray, 383  
GenerateNode  
    GUI::CircularLinkedList, 84  
    GUI::DoublyLinkedList, 153  
    GUI::DynamicArray, 172  
    GUI::LinkedList, 235  
    GUI::SinglyLinkedList, 343  
GenerateRelayoutAnimation  
    Algorithm::Algorithm< GUIAlgorithm, AnimationState >, 25  
    Algorithm::CircularLinkedList, 75  
    Algorithm::DoublyLinkedList, 144  
    Algorithm::DynamicArray, 165  
    Algorithm::Queue, 311  
    Algorithm::SinglyLinkedList, 335  
    Algorithm::Stack, 362  
    Algorithm::StaticArray, 384  
Get

CategoryInfo, 71  
 DSInfo, 161  
 FontHolder, 189  
 TextureHolder, 406  
**get\_iterator**  
 Core::List< T >, 244  
**GetActionDescription**  
 Animation::AnimationState< T >, 36  
**GetAnimateFrame**  
 Animation::AnimationController< T >, 30  
**GetAnimation**  
 Animation::AnimationController< T >, 30  
**GetAnimationDuration**  
 Animation::AnimationController< T >, 30  
**GetAnimationIndex**  
 Animation::AnimationController< T >, 30  
**GetArrowType**  
 GUI::CircularLinkedList, 84  
 GUI::SinglyLinkedList, 343  
**GetArrowTypeNext**  
 GUI::DoublyLinkedList, 153  
**GetArrowTypePrev**  
 GUI::DoublyLinkedList, 154  
**GetBackgroundColor**  
 GUI::Node, 274  
**GetCapacity**  
 GUI::DynamicArray, 173  
**GetCapacityFromLength**  
 GUI::DynamicArray, 173  
**GetChildren**  
 GUI::CircularLinkedList, 85  
 GUI::Container, 120  
 GUI::DataStructure, 128  
 GUI::DoublyLinkedList, 154  
 GUI::DynamicArray, 173  
 GUI::Footer< T >, 193  
 GUI::LinkedList, 235  
 GUI::OperationContainer, 285  
 GUI::OperationList, 293  
 GUI::OptionInputField, 301  
 GUI::SinglyLinkedList, 344  
**GetCircularArrowType**  
 GUI::CircularLinkedList, 85  
**GetCircularEnds**  
 GUI::CircularLinkedList, 85  
**getColor**  
 Settings, 326  
**GetContentPos**  
 GUI::Button, 56  
**GetContentSize**  
 GUI::Button, 56  
**GetContext**  
 ArrayState< T >, 48  
 CLLState, 96  
 DLLState, 137  
 DynamicArrayState, 185  
 HomepageState, 200  
 LLState< T >, 254  
 QueueState, 319  
 SettingsState, 330  
 SLLState, 355  
 StackState, 370  
 State, 375  
 StaticArrayState, 392  
**GetCurrentPlayingAt**  
 Animation::AnimationState< T >, 36  
**GetDataStructure**  
 Animation::AnimationState< T >, 36  
**GetDuration**  
 Animation::AnimationState< T >, 36  
**GetEditMode**  
 GUI::InputField, 209  
 GUI::IntegerInputField, 220  
 GUI::StringInputField, 399  
**GetFontSize**  
 GUI::Button, 56  
**GetHighlightedLine**  
 Animation::AnimationState< T >, 37  
**GetHoverStatus**  
 GUI::Button, 56  
 GUI::Card, 66  
 GUI::CircularLinkedList, 85  
 GUI::CodeHighlighter, 103  
 GUI::Component, 110  
 GUI::Container, 120, 121  
 GUI::DataStructure, 128, 129  
 GUI::DoublyLinkedList, 154  
 GUI::DynamicArray, 173  
 GUI::Footer< T >, 194  
 GUI::InputField, 209  
 GUI::IntegerInputField, 220  
 GUI::LinkedList, 235  
 GUI::NavigationBar, 261  
 GUI::Node, 274  
 GUI::OperationContainer, 286  
 GUI::OperationList, 293  
 GUI::OptionInputField, 301  
 GUI::SinglyLinkedList, 344  
 GUI::StringInputField, 399  
**getInstance**  
 Settings, 326  
**GetLabel**  
 GUI::InputField, 209  
 GUI::IntegerInputField, 220  
 GUI::StringInputField, 399  
**GetLength**  
 GUI::DynamicArray, 173  
**GetList**  
 GUI::CircularLinkedList, 85  
 GUI::DoublyLinkedList, 154  
 GUI::DynamicArray, 173  
 GUI::LinkedList, 236  
 GUI::SinglyLinkedList, 344  
**GetNodeDefaultPosition**  
 GUI::CircularLinkedList, 85  
 GUI::DoublyLinkedList, 154

GUI::DynamicArray, 174  
    GUI::LinkedList, 236  
    GUI::SinglyLinkedList, 344  
GetNodeState  
    GUI::Node, 274  
GetNumAnimation  
    Animation::AnimationController< T >, 30  
GetOutlineColor  
    GUI::Node, 274  
GetPosition  
    GUI::Button, 57  
    GUI::Card, 66  
    GUI::CircularLinkedList, 86  
    GUI::CodeHighlighter, 103  
    GUI::Component, 110  
    GUI::Container, 121  
    GUI::DataStructure, 129  
    GUI::DoublyLinkedList, 154  
    GUI::DynamicArray, 174  
    GUI::Footer< T >, 194  
    GUI::InputField, 209  
    GUI::IntegerInputField, 221  
    GUI::LinkedList, 236  
    GUI::NavigationBar, 261  
    GUI::Node, 275  
    GUI::OperationContainer, 286  
    GUI::OperationList, 293  
    GUI::OptionInputField, 302  
    GUI::SinglyLinkedList, 344  
    GUI::StringInputField, 400  
GetReachable  
    GUI::Node, 275  
GetShape  
    GUI::CircularLinkedList, 86  
    GUI::DoublyLinkedList, 155  
    GUI::DynamicArray, 174  
    GUI::LinkedList, 236  
    GUI::Node, 275  
    GUI::SinglyLinkedList, 344  
GetSize  
    GUI::Button, 57  
    GUI::Card, 66  
    GUI::CircularLinkedList, 86  
    GUI::CodeHighlighter, 104  
    GUI::Component, 111  
    GUI::Container, 121  
    GUI::DataStructure, 129  
    GUI::DoublyLinkedList, 155  
    GUI::DynamicArray, 174  
    GUI::Footer< T >, 194  
    GUI::InputField, 210  
    GUI::IntegerInputField, 221  
    GUI::LinkedList, 236  
    GUI::NavigationBar, 261  
    GUI::Node, 275  
    GUI::OperationContainer, 286  
    GUI::OperationList, 293  
    GUI::OptionInputField, 302  
    GUI::SinglyLinkedList, 345  
    GUI::StringInputField, 400  
GetSpeed  
    Animation::AnimationController< T >, 30  
GetStopDuration  
    Animation::AnimationController< T >, 31  
GetTextColor  
    GUI::Node, 275  
GetValue  
    GUI::Node, 275  
GetVisible  
    GUI::Button, 57  
    GUI::Card, 66  
    GUI::CircularLinkedList, 86  
    GUI::CodeHighlighter, 104  
    GUI::Component, 111  
    GUI::Container, 121  
    GUI::DataStructure, 129  
    GUI::DoublyLinkedList, 155  
    GUI::DynamicArray, 174  
    GUI::Footer< T >, 194  
    GUI::InputField, 210  
    GUI::IntegerInputField, 221  
    GUI::LinkedList, 236  
    GUI::NavigationBar, 261  
    GUI::Node, 275  
    GUI::OperationContainer, 286  
    GUI::OperationList, 294  
    GUI::OptionInputField, 302  
    GUI::SinglyLinkedList, 345  
    GUI::StringInputField, 400  
global, 18  
    defaultColorPath, 18  
    favicon, 18  
    kTitle, 18  
    SCREEN\_HEIGHT, 18  
    SCREEN\_WIDTH, 19  
GUI, 19  
    GUI::Button, 51  
        ~Button, 55  
        action, 60  
        alignment, 60  
        AlignmentCount, 54  
        bound, 60  
        Button, 54, 55  
        buttonColor, 60  
        buttonHoverColor, 61  
        Center, 54  
        content, 61  
        deselect, 55  
        DisableFitContent, 55  
        Draw, 55  
        DrawButtonText, 55  
        EnableFitContent, 56  
        FitContent, 56  
        fitContent, 61  
        fonts, 61  
        fontSize, 61

GetContentPos, 56  
GetContentSize, 56  
GetFontSize, 56  
GetHoverStatus, 56  
GetPosition, 57  
GetSize, 57  
GetVisible, 57  
IsClicked, 57  
isHover, 61  
isSelectable, 57  
isSelected, 57  
Left, 54  
mActionOnHover, 61  
mIsSelected, 61  
mPosition, 62  
mVisible, 62  
Ptr, 54  
Right, 54  
select, 57  
SetAction, 58  
SetActionOnHover, 58  
SetButtonColor, 58  
SetButtonHoverColor, 58  
SetFontSize, 58  
SetPosition, 58  
SetSize, 59  
SetText, 59  
SetTextAlignment, 59  
SetTextColor, 59  
SetVisible, 59  
TextAlignment, 54  
textColor, 62  
ToggleVisible, 59  
UpdateMouseCursorWhenHover, 60  
GUI::Card, 62  
~Card, 65  
Card, 65  
deselect, 65  
Draw, 65  
DrawImage, 65  
DrawTitle, 66  
fonts, 68  
GetHoverStatus, 66  
GetPosition, 66  
GetSize, 66  
GetVisible, 66  
hoverBounds, 69  
isHover, 69  
isSelectable, 67  
isSelected, 67  
mIsSelected, 69  
mPosition, 69  
mVisible, 69  
Ptr, 64  
select, 67  
SetLink, 67  
SetPosition, 67  
SetStateID, 67  
SetText, 68  
SetVisible, 68  
stateID, 69  
thumbnail, 69  
title, 69  
ToggleVisible, 68  
toLink, 70  
UpdateMouseCursorWhenHover, 68  
GUI::CircularLinkedList, 79  
~CircularLinkedList, 83  
Active, 82  
arrowState, 90  
ArrowType, 82  
ArrowTypeCount, 82  
circularArrowState, 90  
CircularLinkedList, 83  
Default, 82  
DeleteNode, 83  
deselect, 83  
Draw, 84  
DrawArrow, 84  
DrawCurrent, 84  
fonts, 90  
GenerateNode, 84  
GetArrowType, 84  
GetChildren, 85  
GetCircularArrowType, 85  
GetCircularEnds, 85  
GetHoverStatus, 85  
GetList, 85  
GetNodeDefaultPosition, 85  
GetPosition, 86  
GetShape, 86  
GetSize, 86  
GetVisible, 86  
Hidden, 82  
Horizontal, 83  
Import, 86  
InsertNode, 86  
isSelectable, 87  
isSelected, 87  
list, 90  
mChildren, 90  
mCircularEnds, 90  
mDisplayHeadAndTail, 90  
mIsSelected, 91  
mNodeDistance, 91  
mOrientation, 91  
mPosition, 91  
mShape, 91  
mVisible, 91  
Orientation, 83  
OrientationCount, 83  
pack, 87  
Ptr, 82  
Relayout, 87  
ResetArrow, 87  
select, 87

SetArrowType, 87  
SetCircularArrowType, 88  
SetCircularEnds, 88  
SetOrientation, 88  
SetPosition, 88  
SetShape, 88  
SetShowHeadAndTail, 89  
SetVisible, 89  
size, 89  
Skip, 82  
ToggleVisible, 89  
UnpackAll, 89  
UpdateMouseCursorWhenHover, 89  
Vertical, 83  
GUI::CodeHighlighter, 100  
~CodeHighlighter, 102  
AddActionDescription, 102  
AddCode, 102  
CodeHighlighter, 102  
deselect, 102  
Draw, 103  
DrawActionDescription, 103  
DrawCodeHighlighter, 103  
fonts, 106  
GetHoverStatus, 103  
GetPosition, 103  
GetSize, 104  
GetVisible, 104  
Highlight, 104  
InitButtons, 104  
isSelectable, 104  
isSelected, 104  
mActionDescription, 106  
mButtonShowAction, 106  
mButtonShowCode, 107  
mCode, 107  
mHighlightedLine, 107  
mIsSelected, 107  
mPosition, 107  
mShowActionDescription, 107  
mShowCode, 107  
mVisible, 107  
Ptr, 102  
select, 104  
SetPosition, 105  
SetShowAction, 105  
SetShowCode, 105  
SetVisible, 105  
ToggleShowAction, 105  
ToggleShowCode, 105  
ToggleVisible, 106  
UpdateMouseCursorWhenHover, 106  
GUI::Component, 108  
~Component, 110  
Component, 109  
deselect, 110  
Draw, 110  
GetHoverStatus, 110  
GetPosition, 110  
GetSize, 111  
GetVisible, 111  
isSelectable, 111  
isSelected, 111  
mIsSelected, 112  
mPosition, 112  
mVisible, 113  
Ptr, 109  
select, 111  
SetPosition, 111  
SetVisible, 112  
ToggleVisible, 112  
UpdateMouseCursorWhenHover, 112  
GUI::Container, 118  
Container, 120  
deselect, 120  
Draw, 120  
DrawCurrent, 120  
GetChildren, 120  
GetHoverStatus, 120, 121  
GetPosition, 121  
GetSize, 121  
GetVisible, 121  
isSelectable, 121  
isSelected, 121  
mChildren, 123  
mIsSelected, 123  
mPosition, 123  
mVisible, 123  
pack, 122  
Ptr, 119  
select, 122  
SetPosition, 122  
SetVisible, 122  
ToggleVisible, 122  
UnpackAll, 122  
UpdateMouseCursorWhenHover, 123  
GUI::DataStructure, 126  
~DataStructure, 128  
DataStructure, 128  
deselect, 128  
Draw, 128  
DrawCurrent, 128  
GetChildren, 128  
GetHoverStatus, 128, 129  
GetPosition, 129  
GetSize, 129  
GetVisible, 129  
isSelectable, 129  
isSelected, 129  
mChildren, 131  
mIsSelected, 131  
mPosition, 131  
mVisible, 131  
pack, 130  
Ptr, 127  
select, 130

SetPosition, 130  
 SetVisible, 130  
 ToggleVisible, 130  
 UnpackAll, 130  
 UpdateMouseCursorWhenHover, 131  
**GUI::DoublyLinkedList**, 148  
 ~DoublyLinkedList, 152  
 Active, 151  
 arrowStateNext, 158  
 arrowStatePrev, 159  
 ArrowType, 151  
 ArrowTypeCount, 151  
 Default, 151  
 DeleteNode, 152  
 deselect, 152  
 DoublyLinkedList, 152  
 Draw, 153  
 DrawArrow, 153  
 DrawCurrent, 153  
 fonts, 159  
 GenerateNode, 153  
 GetArrowTypeNext, 153  
 GetArrowTypePrev, 154  
 GetChildren, 154  
 GetHoverStatus, 154  
 GetList, 154  
 GetNodeDefaultPosition, 154  
 GetPosition, 154  
 GetShape, 155  
 GetSize, 155  
 GetVisible, 155  
 Hidden, 151  
 Horizontal, 152  
 Import, 155  
 InsertNode, 155  
 isSelectable, 155  
 isSelected, 156  
 list, 159  
 mChildren, 159  
 mDisplayHeadAndTail, 159  
 mIsSelected, 159  
 mNodeDistance, 159  
 mOrientation, 159  
 mPosition, 160  
 mShape, 160  
 mVisible, 160  
 Orientation, 152  
 OrientationCount, 152  
 pack, 156  
 Ptr, 151  
 Relayout, 156  
 ResetArrow, 156  
 select, 156  
 SetArrowTypeNext, 156  
 SetArrowTypePrev, 156  
 SetOrientation, 157  
 SetPosition, 157  
 SetShape, 157  
 SetShowHeadAndTail, 157  
 SetVisible, 157  
 size, 158  
 Skip, 151  
 ToggleVisible, 158  
 UnpackAll, 158  
 UpdateMouseCursorWhenHover, 158  
 Vertical, 152  
**GUI::DynamicArray**, 168  
 ~DynamicArray, 171  
 capacity, 177  
 Clear, 171  
 DeleteNode, 172  
 deselect, 172  
 Draw, 172  
 DrawCurrent, 172  
 DynamicArray, 171  
 fonts, 177  
 GenerateNode, 172  
 GetCapacity, 173  
 GetCapacityFromLength, 173  
 GetChildren, 173  
 GetHoverStatus, 173  
 GetLength, 173  
 GetList, 173  
 GetNodeDefaultPosition, 174  
 GetPosition, 174  
 GetShape, 174  
 GetSize, 174  
 GetVisible, 174  
 Import, 174  
 InsertNode, 174  
 isSelectable, 175  
 isSelected, 175  
 length, 177  
 list, 178  
 mChildren, 178  
 mIsSelected, 178  
 mNodeDistance, 178  
 mPosition, 178  
 mShape, 178  
 mVisible, 178  
 operator[], 175  
 pack, 175  
 Ptr, 171  
 Relayout, 175  
 Reserve, 175  
 Resize, 176  
 select, 176  
 SetPosition, 176  
 SetShape, 176  
 SetVisible, 176  
 ToggleVisible, 176  
 UnpackAll, 177  
 UpdateMouseCursorWhenHover, 177  
**GUI::Footer< T >**, 190  
 ~Footer, 193  
 deselect, 193

Draw, 193  
DrawCurrent, 193  
Footer, 192  
GetChildren, 193  
GetHoverStatus, 194  
GetPosition, 194  
GetSize, 194  
GetVisible, 194  
isSelectable, 194  
isSelected, 195  
mChildren, 196  
mIsSelected, 196  
mPosition, 197  
mVisible, 197  
pack, 195  
Ptr, 192  
select, 195  
SetPosition, 195  
SetVisible, 195  
ToggleVisible, 196  
UnpackAll, 196  
UpdateMouseCursorWhenHover, 196  
GUI::InputField, 205  
~InputField, 208  
AllFieldDisableEdit, 208  
deselect, 208  
Draw, 208  
DrawField, 208  
editMode, 212  
extractedValue, 212  
ExtractValue, 209  
fields, 212  
fonts, 213  
GetEditMode, 209  
GetHoverStatus, 209  
GetLabel, 209  
GetPosition, 209  
GetSize, 210  
GetVisible, 210  
InputField, 208  
inputFontSize, 213  
IsClicked, 210  
isSelectable, 210  
isSelected, 210  
label, 213  
labelFontSize, 213  
mFieldIndex, 213  
mIsSelected, 213  
mPosition, 213  
mVisible, 213  
Ptr, 208  
Randomize, 210  
select, 210  
SetEditMode, 211  
SetInputFontSize, 211  
SetLabel, 211  
SetLabelSize, 211  
SetPosition, 211  
SetVisible, 211  
ToggleVisible, 212  
UpdateMouseCursorWhenHover, 212  
GUI::IntegerField, 216  
~IntegerField, 219  
AllFieldDisableEdit, 219  
deselect, 219  
Draw, 219  
DrawField, 220  
editMode, 224  
extractedValue, 224  
ExtractValue, 220  
fields, 224  
fonts, 224  
GetEditMode, 220  
GetHoverStatus, 220  
GetLabel, 220  
GetPosition, 221  
GetSize, 221  
GetVisible, 221  
input, 224  
inputFontSize, 224  
IntegerField, 219  
IsClicked, 221  
isSelectable, 221  
isSelected, 221  
label, 224  
labelFontSize, 224  
mFieldIndex, 225  
mIsSelected, 225  
mMaxValue, 225  
mMinValue, 225  
mPosition, 225  
mVisible, 225  
Ptr, 219  
Randomize, 221  
select, 222  
SetConstraint, 222  
SetEditMode, 222  
SetInputFontSize, 222  
SetLabel, 222  
SetLabelSize, 222  
SetPosition, 222, 223  
SetVisible, 223  
ToggleVisible, 223  
UpdateMouseCursorWhenHover, 223  
GUI::LinkedList, 230  
~LinkedList, 234  
Active, 233  
ArrowType, 233  
ArrowTypeCount, 233  
Default, 233  
DeleteNode, 234  
deselect, 234  
Draw, 234, 235  
DrawCurrent, 235  
fonts, 239  
GenerateNode, 235

GetChildren, 235  
GetHoverStatus, 235  
GetList, 236  
GetNodeDefaultPosition, 236  
GetPosition, 236  
GetShape, 236  
GetSize, 236  
GetVisible, 236  
Hidden, 233  
Horizontal, 234  
Import, 236  
InsertNode, 237  
isSelectable, 237  
isSelected, 237  
LinkedList, 234  
list, 239  
mChildren, 240  
mDisplayHeadAndTail, 240  
mIsSelected, 240  
mNodeDistance, 240  
mOrientation, 240  
mPosition, 240  
mShape, 240  
mVisible, 240  
Orientation, 233  
OrientationCount, 234  
pack, 237  
Ptr, 233  
Relayout, 237  
select, 237  
SetOrientation, 238  
SetPosition, 238  
SetShape, 238  
SetShowHeadAndTail, 238  
SetVisible, 238  
size, 238  
Skip, 233  
ToggleVisible, 239  
UnpackAll, 239  
UpdateMouseCursorWhenHover, 239  
Vertical, 234  
GUI::NavigationBar, 257  
~NavigationBar, 259  
activeTitle, 264  
AtSettings, 260  
atSettings, 264  
backToPrvState, 264  
ClearTitle, 260  
currentCategory, 265  
deselect, 260  
Draw, 260  
DrawLogo, 260  
DrawSettings, 260  
DrawTitles, 260  
fonts, 265  
GetHoverStatus, 261  
GetPosition, 261  
GetSize, 261  
GetVisible, 261  
hasTitle, 265  
homepageID, 265  
hoverBounds, 265  
InsertTitle, 261  
isHover, 265  
isSelectable, 262  
isSelected, 262  
mIsSelected, 265  
mPosition, 265  
mTitles, 266  
mVisible, 266  
NavigationBar, 259  
Ptr, 259  
select, 262  
SetActiveTitle, 262  
SetBackToPreviousLink, 262  
SetCategory, 262  
SetDirectLink, 263  
SetHomepageID, 263  
SetPosition, 263  
SetSettingsID, 263  
settingsID, 266  
SetVisibleTitle, 263  
SetVisible, 263  
ToggleVisible, 264  
toLink, 266  
UpdateMouseCursorWhenHover, 264  
GUI::NavigationBar::TitleInfo, 407  
abbrTitle, 408  
stateID, 408  
titleName, 408  
GUI::Node, 268  
~Node, 272  
Active, 272  
ActiveBlue, 272  
ActiveGreen, 272  
ActiveRed, 272  
AddColor, 273  
animateNode, 278  
AnimationOnNode, 273  
Circle, 272  
ClearLabel, 273  
Default, 272  
deselect, 273  
Draw, 273  
DrawLabel, 273  
DrawNode, 274  
fonts, 279  
GetBackgroundColor, 274  
GetHoverStatus, 274  
GetNodeState, 274  
GetOutlineColor, 274  
GetPosition, 275  
GetReachable, 275  
GetShape, 275  
GetSize, 275  
GetTextColor, 275

GetValue, 275  
    GetVisible, 275  
    Hide, 272  
    IsActive, 276  
    isSelectable, 276  
    isSelected, 276  
    Iterated, 272  
    labelFontSize, 279  
    mActive, 279  
    mActiveColor, 279  
    mBackgroundColor, 279  
    mBorderColor, 279  
    mDefaultColor, 279  
    mIsSelected, 279  
    mLabel, 280  
    mNodeState, 280  
    mOutlineColor, 280  
    mPosition, 280  
    mRadius, 280  
    mReachable, 280  
    mShape, 280  
    mTextColor, 280  
    mValue, 281  
    mVisible, 281  
    Node, 272  
    Ptr, 271  
    select, 276  
    SetActive, 276  
    SetLabel, 276  
    SetLabelFontSize, 276  
    SetNodeState, 277  
    SetPosition, 277  
    SetRadius, 277  
    SetReachable, 277  
    SetShape, 277  
    SetValue, 277  
    SetValueFontSize, 278  
    SetVisible, 278  
    Shape, 271  
    ShapeCount, 272  
    Square, 272  
    State, 272  
    StateCount, 272  
    ToggleVisible, 278  
    UpdateMouseCursorWhenHover, 278  
    valueFontSize, 281

    GUI::OperationContainer, 283  
        ~OperationContainer, 285  
        deselect, 285  
        Draw, 285  
        DrawCurrent, 285  
        GetChildren, 285  
        GetHoverStatus, 286  
        GetPosition, 286  
        GetSize, 286  
        GetVisible, 286  
        isSelectable, 286  
        isSelected, 287

    mChildren, 288  
    mIsSelected, 289  
    mPosition, 289  
    mVisible, 289  
    OperationContainer, 285  
        pack, 287  
        Ptr, 284  
        select, 287  
        SetPosition, 287  
        SetVisible, 287  
        ToggleVisible, 288  
        UnpackAll, 288  
        UpdateMouseCursorWhenHover, 288  
        UpdatePosition, 288

    GUI::OperationList, 289  
        ~OperationList, 292  
        AddOperation, 292  
        buttons, 296  
        deselect, 292  
        Draw, 292  
        DrawCurrent, 293  
        GetChildren, 293  
        GetHoverStatus, 293  
        GetPosition, 293  
        GetSize, 293  
        GetVisible, 294  
        HideAllOptions, 294  
        InitActionBar, 294  
        isHide, 296  
        isSelectable, 294  
        isSelected, 294  
        mChildren, 296  
        mIsSelected, 296  
        mPosition, 297  
        mVisible, 297  
        OperationList, 292  
        optionContainers, 297  
        pack, 294  
        Ptr, 291  
        select, 294  
        GetPosition, 295  
        SetVisible, 295  
        ShowOptions, 295  
        toggleButton, 297  
        ToggleOperations, 295  
        ToggleVisible, 295  
        UnpackAll, 295  
        UpdateMouseCursorWhenHover, 296

    GUI::OptionInputField, 298  
        ~OptionInputField, 300  
        AddInputField, 300  
        AddSubmitField, 300  
        deselect, 300  
        Draw, 301  
        DrawCurrent, 301  
        ExtractInput, 301  
        fonts, 304  
        GetChildren, 301

GetHoverStatus, 301  
 GetPosition, 302  
 GetSize, 302  
 GetVisible, 302  
 HasInputField, 302  
 hasInputField, 305  
 isSelectable, 302  
 isSelected, 302  
 mChildren, 305  
 mInput, 305  
 mInputField, 305  
 mIsSelected, 305  
 mPosition, 305  
 mVisible, 305  
 OptionInputField, 300  
 pack, 303  
 Ptr, 300  
 select, 303  
 SetNoFieldOption, 303  
 SetOption, 303  
 SetPosition, 303  
 SetVisible, 303  
 ToggleInputFields, 304  
 ToggleVisible, 304  
 UnpackAll, 304  
 UpdateMouseCursorWhenHover, 304  
**GUI::SinglyLinkedList**, 339  
 ~SinglyLinkedList, 342  
 Active, 341  
 arrowState, 348  
 ArrowType, 341  
 ArrowTypeCount, 341  
 Default, 341  
 DeleteNode, 342  
 deselect, 342  
 Draw, 343  
 DrawArrow, 343  
 DrawCurrent, 343  
 fonts, 348  
 GenerateNode, 343  
 GetArrowType, 343  
 GetChildren, 344  
 GetHoverStatus, 344  
 GetList, 344  
 GetNodeDefaultPosition, 344  
 GetPosition, 344  
 GetShape, 344  
 GetSize, 345  
 GetVisible, 345  
 Hidden, 341  
 Horizontal, 342  
 Import, 345  
 InsertNode, 345  
 isSelectable, 345  
 isSelected, 345  
 list, 348  
 mChildren, 348  
 mDisplayHeadAndTail, 349  
 mIsSelected, 349  
 mNodeDistance, 349  
 mOrientation, 349  
 mPosition, 349  
 mShape, 349  
 mVisible, 349  
 Orientation, 341  
 OrientationCount, 342  
 pack, 346  
 Ptr, 341  
 Relayout, 346  
 ResetArrow, 346  
 select, 346  
 SetArrowType, 346  
 SetOrientation, 346  
 SetPosition, 346, 347  
 SetShape, 347  
 SetShowHeadAndTail, 347  
 SetVisible, 347  
 SinglyLinkedList, 342  
 size, 347  
 Skip, 341  
 ToggleVisible, 347  
 UnpackAll, 347  
 UpdateMouseCursorWhenHover, 348  
 Vertical, 342  
**GUI::StringInputField**, 395  
 ~StringInputField, 398  
 AllFieldDisableEdit, 398  
 content, 402  
 deselect, 398  
 Draw, 398  
 DrawField, 399  
 editMode, 402  
 extractedValue, 403  
 ExtractValue, 399  
 fields, 403  
 fonts, 403  
 GetEditMode, 399  
 GetHoverStatus, 399  
 GetLabel, 399  
 GetPosition, 400  
 GetSize, 400  
 GetVisible, 400  
 inputFontSize, 403  
 IsClicked, 400  
 isSelectable, 400  
 isSelected, 400  
 label, 403  
 labelFontSize, 403  
 mFieldIndex, 403  
 mIsSelected, 403  
 mMaxLength, 404  
 mPosition, 404  
 mVisible, 404  
 Ptr, 398  
 Randomize, 400  
 select, 401

SetEditMode, 401  
SetInputFieldSize, 401  
SetLabel, 401  
SetLabelSize, 401  
SetPosition, 401  
SetVisible, 402  
StringInputField, 398  
ToggleVisible, 402  
UpdateMouseCursorWhenHover, 402

hasInitializeCard  
    HomepageState, 201

HasInputField  
    GUI::OptionInputField, 302

hasInputField  
    GUI::OptionInputField, 305

hasTitle  
    GUI::NavigationBar, 265

Hidden  
    GUI::CircularLinkedList, 82  
    GUI::DoublyLinkedList, 151  
    GUI::LinkedList, 233  
    GUI::SinglyLinkedList, 341

Hide  
    GUI::Node, 272

HideAllOptions  
    GUI::OperationList, 294

Highlight  
    GUI::CodeHighlighter, 104

HighlightArrowBoth  
    Algorithm::DoublyLinkedList, 144

HighlightArrowFromCur  
    Algorithm::CircularLinkedList, 75  
    Algorithm::Queue, 311  
    Algorithm::SinglyLinkedList, 335  
    Algorithm::Stack, 362

HighlightArrowNext  
    Algorithm::DoublyLinkedList, 144

HighlightArrowPrev  
    Algorithm::DoublyLinkedList, 145

HighlightCircularArrow  
    Algorithm::CircularLinkedList, 75

Homepage  
    States, 20

homepageID  
    GUI::NavigationBar, 265

HomepageState, 197  
    ~HomepageState, 200  
    CreateCard, 200  
    Draw, 200  
    DrawIntroduction, 200  
    GetContext, 200  
    hasInitializeCard, 201  
    HomepageState, 199  
    InitCards, 200  
    InitNavigationBar, 201  
    mCards, 201  
    mContext, 202  
    mStack, 202

navigation, 202  
Ptr, 199  
RequestStackClear, 201  
RequestStackPop, 201  
RequestStackPush, 201  
Update, 201

Horizontal  
    GUI::CircularLinkedList, 83  
    GUI::DoublyLinkedList, 152  
    GUI::LinkedList, 234  
    GUI::SinglyLinkedList, 342

hoverBounds  
    GUI::Card, 69  
    GUI::NavigationBar, 265

ID  
    Category, 14  
    ColorTheme, 15  
    DataStructures, 17  
    DSInfo::Info, 204  
    Fonts, 17  
    States, 19  
    Textures, 20

images  
    Application, 42

Import  
    GUI::CircularLinkedList, 86  
    GUI::DoublyLinkedList, 155  
    GUI::DynamicArray, 174  
    GUI::LinkedList, 236  
    GUI::SinglyLinkedList, 345

Info  
    CategoryInfo::Info, 202  
    DSInfo::Info, 204

Init  
    Application, 40

InitAction  
    Algorithm::Algorithm< GUIAlgorithm, AnimationState >, 25  
    Algorithm::CircularLinkedList, 76  
    Algorithm::DoublyLinkedList, 145  
    Algorithm::DynamicArray, 165  
    Algorithm::Queue, 311  
    Algorithm::SinglyLinkedList, 335  
    Algorithm::Stack, 362  
    Algorithm::StaticArray, 384

InitActionBar  
    GUI::OperationList, 294

InitButtons  
    GUI::CodeHighlighter, 104

InitCards  
    HomepageState, 200

InitNavigationBar  
    ArrayState< T >, 48  
    CLLState, 96  
    DLLState, 137  
    DynamicArrayState, 185  
    HomepageState, 201  
    LLState< T >, 254

QueueState, 319  
 SettingsState, 330  
 SLLState, 355  
 StackState, 370  
 State, 375  
 StaticArrayState, 392  
**input**  
     GUI::IntegerInputField, 224  
**InputField**  
     GUI::InputField, 208  
**inputFontSize**  
     GUI::InputField, 213  
     GUI::IntegerInputField, 224  
     GUI::StringInputField, 403  
**insert\_previous**  
     Core::List< T >, 244  
**InsertAfterTail**  
     Algorithm::CircularLinkedList, 76  
     Algorithm::DoublyLinkedList, 145  
     Algorithm::SinglyLinkedList, 336  
**InsertHead**  
     Algorithm::CircularLinkedList, 76  
     Algorithm::DoublyLinkedList, 145  
     Algorithm::SinglyLinkedList, 336  
**InsertMiddle**  
     Algorithm::CircularLinkedList, 76  
     Algorithm::DoublyLinkedList, 145  
     Algorithm::SinglyLinkedList, 336  
**InsertNode**  
     GUI::CircularLinkedList, 86  
     GUI::DoublyLinkedList, 155  
     GUI::DynamicArray, 174  
     GUI::LinkedList, 237  
     GUI::SinglyLinkedList, 345  
**InsertResource**  
     FontHolder, 189  
     TextureHolder, 406  
**InsertTitle**  
     GUI::NavigationBar, 261  
**IntegerInputField**  
     GUI::IntegerInputField, 219  
**InteractionAllow**  
     Animation::AnimationController< T >, 31  
**InteractionLock**  
     Animation::AnimationController< T >, 31  
**interactionLock**  
     Animation::AnimationController< T >, 33  
**InverseVector**  
     AnimationFactory, 13  
**IsActive**  
     GUI::Node, 276  
**IsClicked**  
     GUI::Button, 57  
     GUI::InputField, 210  
     GUI::IntegerInputField, 221  
     GUI::StringInputField, 400  
**IsEmpty**  
     StateStack, 379  
**isHide**  
     GUI::OperationList, 296  
**isHover**  
     GUI::Button, 61  
     GUI::Card, 69  
     GUI::NavigationBar, 265  
**IsInteractionAllow**  
     Animation::AnimationController< T >, 31  
**IsPlaying**  
     Animation::AnimationController< T >, 31  
**isSelectable**  
     GUI::Button, 57  
     GUI::Card, 67  
     GUI::CircularLinkedList, 87  
     GUI::CodeHighlighter, 104  
     GUI::Component, 111  
     GUI::Container, 121  
     GUI::DataStructure, 129  
     GUI::DoublyLinkedList, 155  
     GUI::DynamicArray, 175  
     GUI::Footer< T >, 194  
     GUI::InputField, 210  
     GUI::IntegerInputField, 221  
     GUI::LinkedList, 237  
     GUI::NavigationBar, 262  
     GUI::Node, 276  
     GUI::OperationContainer, 286  
     GUI::OperationList, 294  
     GUI::OptionInputField, 302  
     GUI::SinglyLinkedList, 345  
     GUI::StringInputField, 400  
**isSelected**  
     GUI::Button, 57  
     GUI::Card, 67  
     GUI::CircularLinkedList, 87  
     GUI::CodeHighlighter, 104  
     GUI::Component, 111  
     GUI::Container, 121  
     GUI::DataStructure, 129  
     GUI::DoublyLinkedList, 156  
     GUI::DynamicArray, 175  
     GUI::Footer< T >, 195  
     GUI::InputField, 210  
     GUI::IntegerInputField, 221  
     GUI::LinkedList, 237  
     GUI::NavigationBar, 262  
     GUI::Node, 276  
     GUI::OperationContainer, 287  
     GUI::OperationList, 294  
     GUI::OptionInputField, 302  
     GUI::SinglyLinkedList, 345  
     GUI::StringInputField, 400  
**Iterated**  
     GUI::Node, 272  
**iterator**  
     Core::List< T >::iterator, 227  
**iterator\_category**  
     Core::List< T >::const\_iterator, 114

Core::List< T >::iterator, 226  
kTitle  
    global, 18  
label  
    ArrayState< T >::IntegerInput, 214  
    GUI::InputField, 213  
    GUI::IntegerField, 224  
    GUI::StringInputField, 403  
    LLState< T >::IntegerInput, 215  
labelFontSize  
    GUI::InputField, 213  
    GUI::IntegerField, 224  
    GUI::Node, 279  
    GUI::StringInputField, 403  
Left  
    GUI::Button, 54  
length  
    GUI::DynamicArray, 177  
LinkedList  
    Category, 14  
    GUI::LinkedList, 234  
List  
    Core::List< T >, 242  
    Core::List< T >::const\_iterator, 117  
    Core::List< T >::iterator, 229  
list  
    GUI::CircularLinkedList, 90  
    GUI::DoublyLinkedList, 159  
    GUI::DynamicArray, 178  
    GUI::LinkedList, 239  
    GUI::SinglyLinkedList, 348  
LLState  
    LLState< T >, 251  
LLState< T >, 248  
    ~LLState, 251  
activeDS, 255  
AddDeleteOperation, 251  
AddInitializeOperation, 251  
AddInsertOperation, 251  
AddIntFieldOperationOption, 252  
AddNoFieldOperationOption, 252  
AddOperations, 252  
AddSearchOperation, 252  
AddStringFieldOption, 252  
AddUpdateOperation, 253  
animController, 255  
ClearAction, 253  
ClearError, 253  
codeHighlighter, 255  
Draw, 253  
DrawCurrentActionText, 253  
DrawCurrentErrorText, 253  
footer, 255  
GetContext, 254  
InitNavigationBar, 254  
LLState, 251  
mContext, 256  
mCurrentAction, 256  
mCurrentError, 256  
mStack, 256  
navigation, 256  
operationList, 256  
Ptr, 251  
RequestStackClear, 254  
RequestStackPop, 254  
RequestStackPush, 254  
SetCurrentAction, 254  
SetCurrentError, 254  
Success, 255  
Update, 255  
LLState< T >::IntegerInput, 215  
    label, 215  
    maxValue, 215  
    minValue, 215  
    width, 215  
Load  
    FontHolder, 189  
    TextureHolder, 406  
LoadDefaultColors  
    Settings, 327  
LoadFromFile  
    Settings, 327  
LoadFromImage  
    TextureHolder, 407  
LoadResources  
    Application, 40  
Logo1FirstPart  
    ColorTheme, 15  
Logo1SecondPart  
    ColorTheme, 15  
Logo2FirstPart  
    ColorTheme, 15  
Logo2SecondPart  
    ColorTheme, 15  
mActionDescription  
    GUI::CodeHighlighter, 106  
mActionOnHover  
    GUI::Button, 61  
mActive  
    GUI::Node, 279  
mActiveColor  
    GUI::Node, 279  
main  
    Main.cpp, 497  
Main.cpp  
    main, 497  
mAnimation  
    Animation::AnimationState< T >, 38  
maxN  
    Algorithm::CircularLinkedList, 78  
    Algorithm::DoublyLinkedList, 148  
    Algorithm::DynamicArray, 167  
    Algorithm::Queue, 313  
    Algorithm::SinglyLinkedList, 338  
    Algorithm::Stack, 364

Algorithm::StaticArray, 386  
 maxValue  
     ArrayState< T >::IntegerInput, 214  
     LLState< T >::IntegerInput, 215  
 mBackgroundColor  
     GUI::Node, 279  
 mBegin  
     Core::List< T >, 247  
 mBorderColor  
     GUI::Node, 279  
 mButtonShowAction  
     GUI::CodeHighlighter, 106  
 mButtonShowCode  
     GUI::CodeHighlighter, 107  
 mCards  
     HomepageState, 201  
 mChildren  
     GUI::CircularLinkedList, 90  
     GUI::Container, 123  
     GUI::DataStructure, 131  
     GUI::DoublyLinkedList, 159  
     GUI::DynamicArray, 178  
     GUI::Footer< T >, 196  
     GUI::LinkedList, 240  
     GUI::OperationContainer, 288  
     GUI::OperationList, 296  
     GUI::OptionInputField, 305  
     GUI::SinglyLinkedList, 348  
     SceneNode, 324  
 mCircularEnds  
     GUI::CircularLinkedList, 90  
 mCode  
     GUI::CodeHighlighter, 107  
 mColors  
     Settings, 327  
 mContext  
     ArrayState< T >, 50  
     CLLState, 98  
     DLLState, 139  
     DynamicArrayState, 187  
     HomepageState, 202  
     LLState< T >, 256  
     QueueState, 321  
     SettingsState, 331  
     SLLState, 357  
     StackState, 372  
     State, 376  
     StateStack, 380  
     StaticArrayState, 394  
 mCurrentAction  
     ArrayState< T >, 50  
     CLLState, 99  
     DLLState, 139  
     DynamicArrayState, 187  
     LLState< T >, 256  
     QueueState, 321  
     SLLState, 357  
     StackState, 372  
 mcurrentState, 394  
 mCurrentAnimationIndex  
     Animation::AnimationController< T >, 33  
 mCurrentError  
     ArrayState< T >, 50  
     CLLState, 99  
     DLLState, 139  
     DynamicArrayState, 187  
     LLState< T >, 256  
     QueueState, 321  
     SLLState, 358  
     StackState, 372  
     StaticArrayState, 394  
 mCurrentPlayingAt  
     Animation::AnimationState< T >, 38  
 mDataStructureBefore  
     Animation::AnimationState< T >, 39  
 mDefaultColor  
     GUI::Node, 279  
 mDisplayHeadAndTail  
     GUI::CircularLinkedList, 90  
     GUI::DoublyLinkedList, 159  
     GUI::LinkedList, 240  
     GUI::SinglyLinkedList, 349  
 mDLL  
     DLLState, 139  
 mDS  
     CategoryInfo::Info, 203  
 mDuration  
     Animation::AnimationState< T >, 39  
 mDynamicArray  
     DynamicArrayState, 187  
 mEnd  
     Core::List< T >, 247  
 mFactories  
     CategoryInfo, 71  
     DSInfo, 161  
     StateStack, 380  
 mFieldIndex  
     GUI::InputField, 213  
     GUI::IntegerInputField, 225  
     GUI::StringInputField, 403  
 mFontMap  
     FontHolder, 190  
 mHighlightedLine  
     Animation::AnimationState< T >, 39  
     GUI::CodeHighlighter, 107  
 mInput  
     GUI::OptionInputField, 305  
 mInputField  
     GUI::OptionInputField, 305  
 minValue  
     ArrayState< T >::IntegerInput, 214  
     LLState< T >::IntegerInput, 215  
 mIsSelected  
     GUI::Button, 61  
     GUI::Card, 69  
     GUI::CircularLinkedList, 91

GUI::CodeHighlighter, 107  
GUI::Component, 112  
GUI::Container, 123  
GUI::DataStructure, 131  
GUI::DoublyLinkedList, 159  
GUI::DynamicArray, 178  
GUI::Footer< T >, 197  
GUI::InputField, 213  
GUI::IntegerField, 225  
GUI::LinkedList, 240  
GUI::NavigationBar, 265  
GUI::Node, 279  
GUI::OperationContainer, 289  
GUI::OperationList, 296  
GUI::OptionInputField, 305  
GUI::SinglyLinkedList, 349  
GUI::StringInputField, 403  
mLabel  
    GUI::Node, 280  
mMaxLength  
    GUI::StringInputField, 404  
mMaxValue  
    GUI::IntegerField, 225  
mMinValue  
    GUI::IntegerField, 225  
mNext  
    Core::Node< T >, 267  
mNodeDistance  
    GUI::CircularLinkedList, 91  
    GUI::DoublyLinkedList, 159  
    GUI::DynamicArray, 178  
    GUI::LinkedList, 240  
    GUI::SinglyLinkedList, 349  
mNodeState  
    GUI::Node, 280  
mOrientation  
    GUI::CircularLinkedList, 91  
    GUI::DoublyLinkedList, 159  
    GUI::LinkedList, 240  
    GUI::SinglyLinkedList, 349  
mOutlineColor  
    GUI::Node, 280  
move\_previous  
    Core::List< T >, 245  
MoveNode  
    AnimationFactory, 13  
mParent  
    SceneNode, 324  
mPendingList  
    StateStack, 380  
mPosition  
    GUI::Button, 62  
    GUI::Card, 69  
    GUI::CircularLinkedList, 91  
    GUI::CodeHighlighter, 107  
    GUI::Component, 112  
    GUI::Container, 123  
    GUI::DataStructure, 131  
    GUI::DoublyLinkedList, 160  
    GUI::DynamicArray, 178  
    GUI::Footer< T >, 197  
    GUI::InputField, 213  
    GUI::IntegerField, 225  
    GUI::LinkedList, 240  
    GUI::NavigationBar, 265  
    GUI::Node, 280  
    GUI::OperationContainer, 289  
    GUI::OperationList, 297  
    GUI::OptionInputField, 305  
    GUI::SinglyLinkedList, 349  
    GUI::StringInputField, 404  
mPrev  
    Core::Node< T >, 268  
mRadius  
    GUI::Node, 280  
mReachable  
    GUI::Node, 280  
mShape  
    GUI::CircularLinkedList, 91  
    GUI::DoublyLinkedList, 160  
    GUI::DynamicArray, 178  
    GUI::LinkedList, 240  
    GUI::Node, 280  
    GUI::SinglyLinkedList, 349  
mShowActionDescription  
    GUI::CodeHighlighter, 107  
mShowCode  
    GUI::CodeHighlighter, 107  
mSize  
    Core::List< T >, 247  
mSpeed  
    Animation::AnimationController< T >, 34  
mStack  
    Application, 42  
    ArrayState< T >, 50  
    CLLState, 99  
    DLLState, 140  
    DynamicArrayState, 187  
    HomepageState, 202  
    LLState< T >, 256  
    QueueState, 321  
    SettingsState, 331  
    SLLState, 358  
    StackState, 372  
    State, 376  
    StateStack, 380  
    StaticArrayState, 394  
mStackAlgorithm  
    StackState, 373  
mStackOrientation  
    Algorithm::Stack, 364  
mStaticArray  
    StaticArrayState, 394  
mTextColor  
    GUI::Node, 280  
mTextureMap

TextureHolder, 407  
 mTitles  
   GUI::NavigationBar, 266  
 mValue  
   Core::Node< T >, 268  
   GUI::Node, 281  
 mVisible  
   GUI::Button, 62  
   GUI::Card, 69  
   GUI::CircularLinkedList, 91  
   GUI::CodeHighlighter, 107  
   GUI::Component, 113  
   GUI::Container, 123  
   GUI::DataStructure, 131  
   GUI::DoublyLinkedList, 160  
   GUI::DynamicArray, 178  
   GUI::Footer< T >, 197  
   GUI::InputField, 213  
   GUI::IntegerInputField, 225  
   GUI::LinkedList, 240  
   GUI::NavigationBar, 266  
   GUI::Node, 281  
   GUI::OperationContainer, 289  
   GUI::OperationList, 297  
   GUI::OptionInputField, 305  
   GUI::SinglyLinkedList, 349  
   GUI::StringInputField, 404  
  
 name  
   DSInfo::Info, 204  
  
 navigation  
   ArrayState< T >, 50  
   CLLState, 99  
   DLLState, 140  
   DynamicArrayList, 187  
   HomepageState, 202  
   LLState< T >, 256  
   QueueState, 322  
   SettingsState, 331  
   SLLState, 358  
   StackState, 373  
   State, 376  
   StaticArrayList, 394  
  
 NavigationBar  
   GUI::NavigationBar, 259  
  
 NavigationBar\_Background  
   ColorTheme, 15  
  
 NavigationBar\_SelectedTitle  
   ColorTheme, 15  
  
 NavigationBar\_UnselectedTitle  
   ColorTheme, 15  
  
 Node  
   Core::Node< T >, 267  
   GUI::Node, 272  
  
 NonCopyable  
   NonCopyable< T >, 281, 282  
  
 NonCopyable< T >, 281  
   ~NonCopyable, 282  
   NonCopyable, 281, 282  
  
 operator=, 282  
  
 None  
   Category, 14  
   DataStructures, 17  
   States, 20  
  
 OpenFileDialog  
   Utils, 22  
  
 OperationContainer  
   GUI::OperationContainer, 285  
  
 OperationList  
   GUI::OperationList, 292  
  
 operationList  
   ArrayList< T >, 51  
   CLLState, 99  
   DLLState, 140  
   DynamicArrayList, 187  
   LLState< T >, 256  
   QueueState, 322  
   SLLState, 358  
   StackState, 373  
   StaticArrayList, 395  
  
 operator!=  
   Core::List< T >::const\_iterator, 115  
   Core::List< T >::iterator, 227  
  
 operator\*  
   Core::List< T >::const\_iterator, 115  
   Core::List< T >::iterator, 228  
  
 operator+  
   Core::List< T >::const\_iterator, 116  
   Core::List< T >::iterator, 228  
  
 operator++  
   Core::List< T >::const\_iterator, 116  
   Core::List< T >::iterator, 228  
  
 operator-  
   Core::List< T >::const\_iterator, 116  
   Core::List< T >::iterator, 228  
  
 operator->  
   Core::List< T >::const\_iterator, 117  
   Core::List< T >::iterator, 229  
  
 operator--  
   Core::List< T >::const\_iterator, 116  
   Core::List< T >::iterator, 228, 229  
  
 operator=  
   Core::List< T >, 245  
   Core::List< T >::const\_iterator, 117  
   Core::List< T >::iterator, 229  
   NonCopyable< T >, 282  
   Settings, 327  
  
 operator==  
   Core::List< T >::const\_iterator, 117  
   Core::List< T >::iterator, 229  
  
 operator[]  
   Core::List< T >, 245  
   GUI::DynamicArray, 175  
  
 optionContainers  
   GUI::OperationList, 297  
  
 OptionInputField  
   GUI::OptionInputField, 300

Orientation  
  GUI::CircularLinkedList, 83  
  GUI::DoublyLinkedList, 152  
  GUI::LinkedList, 233  
  GUI::SinglyLinkedList, 341  
  Stack.hpp, 436

OrientationCount  
  GUI::CircularLinkedList, 83  
  GUI::DoublyLinkedList, 152  
  GUI::LinkedList, 234  
  GUI::SinglyLinkedList, 342

pack  
  GUI::CircularLinkedList, 87  
  GUI::Container, 122  
  GUI::DataStructure, 130  
  GUI::DoublyLinkedList, 156  
  GUI::DynamicArray, 175  
  GUI::Footer< T >, 195  
  GUI::LinkedList, 237  
  GUI::OperationContainer, 287  
  GUI::OperationList, 294  
  GUI::OptionInputField, 303  
  GUI::SinglyLinkedList, 346

Pause  
  Animation::AnimationController< T >, 31

Peek  
  Algorithm::Stack, 362

PeekBack  
  Algorithm::Queue, 311

PeekFront  
  Algorithm::Queue, 311

PendingChange  
  StateStack::PendingChange, 306

Playing  
  Animation::AnimationController< T >, 34

PlayingAt  
  Animation::AnimationState< T >, 37

pointer  
  Core::List< T >::const\_iterator, 114  
  Core::List< T >::iterator, 227

Pop  
  Algorithm::Stack, 362  
  StateStack, 378

pop\_back  
  Core::List< T >, 245

pop\_front  
  Core::List< T >, 245

PopAnimation  
  Animation::AnimationController< T >, 31

PopBack  
  Algorithm::DynamicArray, 165

PopState  
  StateStack, 379

Ptr  
  Animation::AnimationController< T >, 28  
  Animation::AnimationState< T >, 35  
  ArrayState< T >, 45  
  CLLState, 94

DLLState, 134  
DynamicArrayList, 182  
GUI::Button, 54  
GUI::Card, 64  
GUI::CircularLinkedList, 82  
GUI::CodeHighlighter, 102  
GUI::Component, 109  
GUI::Container, 119  
GUI::DataStructure, 127  
GUI::DoublyLinkedList, 151  
GUI::DynamicArray, 171  
GUI::Footer< T >, 192  
GUI::InputField, 208  
GUI::IntegerInputField, 219  
GUI::LinkedList, 233  
GUI::NavigationBar, 259  
GUI::Node, 271  
GUI::OperationContainer, 284  
GUI::OperationList, 291  
GUI::OptionInputField, 300  
GUI::SinglyLinkedList, 341  
GUI::StringInputField, 398  
HomepageState, 199  
LLState< T >, 251  
QueueState, 316  
SceneNode, 323  
SettingsState, 329  
SLLState, 353  
StackState, 367  
State, 374  
StaticArrayList, 389

ptr  
  Core::List< T >::const\_iterator, 118  
  Core::List< T >::iterator, 230

Push  
  Algorithm::Stack, 362  
  StateStack, 378

push\_back  
  Core::List< T >, 246

push\_front  
  Core::List< T >, 246

PushBack  
  Algorithm::DynamicArray, 165

PushState  
  StateStack, 379

Queue  
  Algorithm::Queue, 309  
  DataStructures, 17  
  States, 20  
  Textures, 20

queue  
  QueueState, 322

Queue.cpp  
  ArrowType, 428

QueueState, 314  
  ~QueueState, 317

activeDS, 321  
AddDeleteOperation, 317

AddInitializeOperation, 317  
 AddInsertOperation, 317  
 AddIntFieldOperationOption, 317  
 AddNoFieldOperationOption, 318  
 AddOperations, 318  
 AddSearchOperation, 318  
 AddStringFieldOption, 318  
 AddUpdateOperation, 318  
 animController, 321  
 ClearAction, 318  
 ClearError, 319  
 codeHighlighter, 321  
 Draw, 319  
 DrawCurrentActionText, 319  
 DrawCurrentErrorText, 319  
 footer, 321  
 GetContext, 319  
 InitNavigationBar, 319  
 mContext, 321  
 mCurrentAction, 321  
 mCurrentError, 321  
 mStack, 321  
 navigation, 322  
 operationList, 322  
 Ptr, 316  
 queue, 322  
 QueueState, 317  
 RequestStackClear, 319  
 RequestStackPop, 320  
 RequestStackPush, 320  
 SetCurrentAction, 320  
 SetCurrentError, 320  
 Success, 320  
 Update, 320

Rand  
 Utils, 22

Random  
 Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 25  
 Algorithm::CircularLinkedList, 76  
 Algorithm::DoublyLinkedList, 145  
 Algorithm::DynamicArray, 165  
 Algorithm::Queue, 312  
 Algorithm::SinglyLinkedList, 336  
 Algorithm::Stack, 363  
 Algorithm::StaticArray, 384

RandomFixedSize  
 Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 25  
 Algorithm::CircularLinkedList, 76  
 Algorithm::DoublyLinkedList, 146  
 Algorithm::DynamicArray, 165  
 Algorithm::Queue, 312  
 Algorithm::SinglyLinkedList, 336  
 Algorithm::Stack, 363  
 Algorithm::StaticArray, 384

RandomFixedSizeGenerator

Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 25  
 Algorithm::CircularLinkedList, 76  
 Algorithm::DoublyLinkedList, 146  
 Algorithm::DynamicArray, 165  
 Algorithm::Queue, 312  
 Algorithm::SinglyLinkedList, 336  
 Algorithm::Stack, 363  
 Algorithm::StaticArray, 384

RandomGenerator  
 Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 26  
 Algorithm::CircularLinkedList, 77  
 Algorithm::DoublyLinkedList, 146  
 Algorithm::DynamicArray, 166  
 Algorithm::Queue, 312  
 Algorithm::SinglyLinkedList, 336  
 Algorithm::Stack, 363  
 Algorithm::StaticArray, 384

Randomize  
 GUI::InputField, 210  
 GUI::IntegerField, 221  
 GUI::StringInputField, 400

RAYGUI\_IMPLEMENTATION  
 Application.cpp, 448

ReadFromExternalFile  
 Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 26  
 Algorithm::CircularLinkedList, 77  
 Algorithm::DoublyLinkedList, 146  
 Algorithm::DynamicArray, 166  
 Algorithm::Queue, 312  
 Algorithm::SinglyLinkedList, 337  
 Algorithm::Stack, 363  
 Algorithm::StaticArray, 385

ReadFromFileGenerator  
 Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 26  
 Algorithm::CircularLinkedList, 77  
 Algorithm::DoublyLinkedList, 146  
 Algorithm::DynamicArray, 166  
 Algorithm::Queue, 312  
 Algorithm::SinglyLinkedList, 337  
 Algorithm::Stack, 363  
 Algorithm::StaticArray, 385

ReadInputFromFile  
 Utils, 22

ReCalculateEnds  
 AnimationFactory, 14

reference  
 Core::List< T >::const\_iterator, 114  
 Core::List< T >::iterator, 227

Register  
 CategoryInfo, 71  
 DSInfo, 161

RegisterState  
 StateStack, 379

RegisterStates

Application, 41  
Relayout  
  GUI::CircularLinkedList, 87  
  GUI::DoublyLinkedList, 156  
  GUI::DynamicArray, 175  
  GUI::LinkedList, 237  
  GUI::SinglyLinkedList, 346  
remove  
  Core::List< T >, 246  
remove\_if  
  Core::List< T >, 246, 247  
Render  
  Application, 41  
RequestStackClear  
  AppState< T >, 48  
  CLLState, 97  
  DLLState, 137  
  DynamicArrayState, 185  
  HomepageState, 201  
  LLState< T >, 254  
  QueueState, 319  
  SettingsState, 330  
  SLLState, 356  
  StackState, 370  
  State, 375  
  StaticArrayState, 392  
RequestStackPop  
  AppState< T >, 48  
  CLLState, 97  
  DLLState, 138  
  DynamicArrayState, 185  
  HomepageState, 201  
  LLState< T >, 254  
  QueueState, 320  
  SettingsState, 330  
  SLLState, 356  
  StackState, 371  
  State, 375  
  StaticArrayState, 392  
RequestStackPush  
  AppState< T >, 48  
  CLLState, 97  
  DLLState, 138  
  DynamicArrayState, 185  
  HomepageState, 201  
  LLState< T >, 254  
  QueueState, 320  
  SettingsState, 330  
  SLLState, 356  
  StackState, 371  
  State, 376  
  StaticArrayState, 392  
Reserve  
  GUI::DynamicArray, 175  
Reset  
  Animation::AnimationController< T >, 32  
  Animation::AnimationState< T >, 37  
reset  
  Core::List< T >, 247  
ResetArrow  
  GUI::CircularLinkedList, 87  
  GUI::DoublyLinkedList, 156  
  GUI::SinglyLinkedList, 346  
ResetCurrent  
  Animation::AnimationController< T >, 32  
ResetVisualizer  
  Algorithm::CircularLinkedList, 77  
  Algorithm::DoublyLinkedList, 146  
  Algorithm::DynamicArray, 166  
  Algorithm::SinglyLinkedList, 337  
  Algorithm::StaticArray, 385  
Resize  
  GUI::DynamicArray, 176  
Right  
  GUI::Button, 54  
Run  
  Application, 41  
RunAll  
  Animation::AnimationController< T >, 32  
SaveToFile  
  Settings, 327  
SceneNode, 322  
  ~SceneNode, 323  
  AttachChild, 323  
  DetachChild, 323  
  Draw, 323  
  DrawCurrent, 324  
  mChildren, 324  
  mParent, 324  
  Ptr, 323  
  SceneNode, 323  
SCREEN\_HEIGHT  
  global, 18  
SCREEN\_WIDTH  
  global, 19  
Search  
  Algorithm::CircularLinkedList, 77  
  Algorithm::DoublyLinkedList, 146  
  Algorithm::DynamicArray, 166  
  Algorithm::SinglyLinkedList, 337  
  Algorithm::StaticArray, 385  
select  
  GUI::Button, 57  
  GUI::Card, 67  
  GUI::CircularLinkedList, 87  
  GUI::CodeHighlighter, 104  
  GUI::Component, 111  
  GUI::Container, 122  
  GUI::DataStructure, 130  
  GUI::DoublyLinkedList, 156  
  GUI::DynamicArray, 176  
  GUI::Footer< T >, 195  
  GUI::InputField, 210  
  GUI::IntegerInputField, 222  
  GUI::LinkedList, 237  
  GUI::NavigationBar, 262

GUI::Node, 276  
 GUI::OperationContainer, 287  
 GUI::OperationList, 294  
 GUI::OptionInputField, 303  
 GUI::SinglyLinkedList, 346  
 GUI::StringInputField, 401  
**SetAction**  
     GUI::Button, 58  
**SetActionDescription**  
     Animation::AnimationState< T >, 37  
**SetActionOnHover**  
     GUI::Button, 58  
**SetActive**  
     GUI::Node, 276  
**SetActiveTitle**  
     GUI::NavigationBar, 262  
**SetAnimation**  
     Animation::AnimationController< T >, 32  
     Animation::AnimationState< T >, 37  
**SetArrowType**  
     GUI::CircularLinkedList, 87  
     GUI::SinglyLinkedList, 346  
**SetArrowTypeNext**  
     GUI::DoublyLinkedList, 156  
**SetArrowTypePrev**  
     GUI::DoublyLinkedList, 156  
**SetBackToPreviousLink**  
     GUI::NavigationBar, 262  
**SetButtonColor**  
     GUI::Button, 58  
**SetButtonHoverColor**  
     GUI::Button, 58  
**SetCategory**  
     GUI::NavigationBar, 262  
**SetCircularArrowType**  
     GUI::CircularLinkedList, 88  
**SetCircularEnds**  
     GUI::CircularLinkedList, 88  
**SetConstraint**  
     GUI::IntegerInputField, 222  
**SetCurrentAction**  
     ArrayState< T >, 49  
     CLLState, 97  
     DLLState, 138  
     DynamicArrayState, 185  
     LLState< T >, 254  
     QueueState, 320  
     SLLState, 356  
     StackState, 371  
     StaticArrayState, 392  
**SetCurrentError**  
     ArrayState< T >, 49  
     CLLState, 97  
     DLLState, 138  
     DynamicArrayState, 185  
     LLState< T >, 254  
     QueueState, 320  
     SLLState, 356  
**SetDirectLink**  
     GUI::NavigationBar, 263  
**SetDuration**  
     Animation::AnimationState< T >, 37  
**SetEditMode**  
     GUI::InputField, 211  
     GUI::IntegerInputField, 222  
     GUI::StringInputField, 401  
**SetFontSize**  
     GUI::Button, 58  
**SetHighlightLine**  
     Animation::AnimationState< T >, 38  
**SetHomePageID**  
     GUI::NavigationBar, 263  
**SetInputFieldSize**  
     GUI::InputField, 211  
     GUI::IntegerInputField, 222  
     GUI::StringInputField, 401  
**SetLabel**  
     GUI::InputField, 211  
     GUI::IntegerInputField, 222  
     GUI::Node, 276  
     GUI::StringInputField, 401  
**SetLabelFontSize**  
     GUI::Node, 276  
**SetLabelSize**  
     GUI::InputField, 211  
     GUI::IntegerInputField, 222  
     GUI::StringInputField, 401  
**SetLink**  
     GUI::Card, 67  
**SetNodeState**  
     GUI::Node, 277  
**SetNoFieldOption**  
     GUI::OptionInputField, 303  
**SetOption**  
     GUI::OptionInputField, 303  
**SetOrientation**  
     GUI::CircularLinkedList, 88  
     GUI::DoublyLinkedList, 157  
     GUI::LinkedList, 238  
     GUI::SinglyLinkedList, 346  
**SetPosition**  
     GUI::Button, 58  
     GUI::Card, 67  
     GUI::CircularLinkedList, 88  
     GUI::CodeHighlighter, 105  
     GUI::Component, 111  
     GUI::Container, 122  
     GUI::DataStructure, 130  
     GUI::DoublyLinkedList, 157  
     GUI::DynamicArray, 176  
     GUI::Footer< T >, 195  
     GUI::InputField, 211  
     GUI::IntegerInputField, 222, 223  
     GUI::LinkedList, 238

GUI::NavigationBar, 263  
    GUI::Node, 277  
    GUI::OperationContainer, 287  
    GUI::OperationList, 295  
    GUI::OptionInputField, 303  
    GUI::SinglyLinkedList, 346, 347  
    GUI::StringInputField, 401  
**SetRadius**  
    GUI::Node, 277  
**SetReachable**  
    GUI::Node, 277  
**SetSettingsID**  
    GUI::NavigationBar, 263  
**SetShape**  
    GUI::CircularLinkedList, 88  
    GUI::DoublyLinkedList, 157  
    GUI::DynamicArray, 176  
    GUI::LinkedList, 238  
    GUI::Node, 277  
    GUI::SinglyLinkedList, 347  
**SetShowAction**  
    GUI::CodeHighlighter, 105  
**SetShowCode**  
    GUI::CodeHighlighter, 105  
**SetShowHeadAndTail**  
    GUI::CircularLinkedList, 89  
    GUI::DoublyLinkedList, 157  
    GUI::LinkedList, 238  
    GUI::SinglyLinkedList, 347  
**SetSize**  
    GUI::Button, 59  
**SetSourceDataStructure**  
    Animation::AnimationState< T >, 38  
**SetSpeed**  
    Animation::AnimationController< T >, 32  
**SetStateID**  
    GUI::Card, 67  
**SetText**  
    GUI::Button, 59  
    GUI::Card, 68  
**SetTextAlignment**  
    GUI::Button, 59  
**SetTextColor**  
    GUI::Button, 59  
**Settings, 324**  
    ~Settings, 326  
    getColor, 326  
    getInstance, 326  
    LoadDefaultColors, 327  
    LoadFromFile, 327  
    mColors, 327  
    operator=, 327  
    SaveToFile, 327  
    Settings, 326  
    States, 20  
**settingsID**  
    GUI::NavigationBar, 266  
**SettingsState, 328**

    ~SettingsState, 329  
    Draw, 330  
    GetContext, 330  
    InitNavigationBar, 330  
    mContext, 331  
    mStack, 331  
    navigation, 331  
    Ptr, 329  
    RequestStackClear, 330  
    RequestStackPop, 330  
    RequestStackPush, 330  
    SettingsState, 329  
    Update, 330  
**SetValue**  
    GUI::Node, 277  
**SetValueFontSize**  
    GUI::Node, 278  
**SetVisibleTitle**  
    GUI::NavigationBar, 263  
**SetVisible**  
    GUI::Button, 59  
    GUI::Card, 68  
    GUI::CircularLinkedList, 89  
    GUI::CodeHighlighter, 105  
    GUI::Component, 112  
    GUI::Container, 122  
    GUI::DataStructure, 130  
    GUI::DoublyLinkedList, 157  
    GUI::DynamicArray, 176  
    GUI::Footer< T >, 195  
    GUI::InputField, 211  
    GUI::IntegerInputField, 223  
    GUI::LinkedList, 238  
    GUI::NavigationBar, 263  
    GUI::Node, 278  
    GUI::OperationContainer, 287  
    GUI::OperationList, 295  
    GUI::OptionInputField, 303  
    GUI::SinglyLinkedList, 347  
    GUI::StringInputField, 402  
**Shape**  
    GUI::Node, 271  
**ShapeCount**  
    GUI::Node, 272  
**ShowOptions**  
    GUI::OperationList, 295  
**Silkscreen**  
    Fonts, 18  
**SinglyLinkedList**  
    Algorithm::SinglyLinkedList, 333, 334  
    DataStructures, 17  
    GUI::SinglyLinkedList, 342  
    States, 20  
    Textures, 20  
**SinglyLinkedList.cpp**  
    ArrowType, 431  
**size**  
    Algorithm::CircularLinkedList, 77

Algorithm::DoublyLinkedList, 147  
 Algorithm::DynamicArray, 166  
 Algorithm::Queue, 312  
 Algorithm::SinglyLinkedList, 337  
 Algorithm::Stack, 363  
 Algorithm::StaticArray, 385  
 Core::List< T >, 247  
 GUI::CircularLinkedList, 89  
 GUI::DoublyLinkedList, 158  
 GUI::LinkedList, 238  
 GUI::SinglyLinkedList, 347  
 Skip  
     GUI::CircularLinkedList, 82  
     GUI::DoublyLinkedList, 151  
     GUI::LinkedList, 233  
     GUI::SinglyLinkedList, 341  
 SLL  
     SLLState, 358  
 SLLAnimation  
     AnimationState.hpp, 445  
 SLLAnimationController  
     AnimationController.hpp, 438  
 SLLState, 350  
     ~SLLState, 353  
     activeDS, 357  
     AddDeleteOperation, 353  
     AddInitializeOperation, 353  
     AddInsertOperation, 353  
     AddIntFieldOperationOption, 354  
     AddNoFieldOperationOption, 354  
     AddOperations, 354  
     AddSearchOperation, 354  
     AddStringFieldOption, 354  
     AddUpdateOperation, 354  
     animController, 357  
     ClearAction, 355  
     ClearError, 355  
     codeHighlighter, 357  
     Draw, 355  
     DrawCurrentActionText, 355  
     DrawCurrentErrorText, 355  
     footer, 357  
     GetContext, 355  
     InitNavigationBar, 355  
     mContext, 357  
     mCurrentAction, 357  
     mCurrentError, 358  
     mStack, 358  
     navigation, 358  
     operationList, 358  
     Ptr, 353  
     RequestStackClear, 356  
     RequestStackPop, 356  
     RequestStackPush, 356  
     SetCurrentAction, 356  
     SetCurrentError, 356  
     SLL, 358  
     SLLState, 353  
 Success, 356  
 Update, 356  
 Square  
     GUI::Node, 272  
 src/Algorithms/Algorithm.hpp, 409, 410  
 src/Algorithms/Array/DynamicArray.cpp, 413  
 src/Algorithms/Array/DynamicArray.hpp, 414, 415  
 src/Algorithms/Array/StaticArray.cpp, 417  
 src/Algorithms/Array/StaticArray.hpp, 417, 418  
 src/Algorithms/LinkedList/CircularLinkedList.cpp, 419  
 src/Algorithms/LinkedList/CircularLinkedList.hpp, 420, 421  
 src/Algorithms/LinkedList/DoublyLinkedList.cpp, 423  
 src/Algorithms/LinkedList/DoublyLinkedList.hpp, 425, 426  
 src/Algorithms/LinkedList/Queue.cpp, 428  
 src/Algorithms/LinkedList/Queue.hpp, 429, 430  
 src/Algorithms/LinkedList/SinglyLinkedList.cpp, 430  
 src/Algorithms/LinkedList/SinglyLinkedList.hpp, 431, 432  
 src/Algorithms/LinkedList/Stack.cpp, 434  
 src/Algorithms/LinkedList/Stack.hpp, 435, 436  
 src/Animation/AnimationController.hpp, 437, 438  
 src/Animation/AnimationFactory.cpp, 441  
 src/Animation/AnimationFactory.hpp, 442, 443  
 src/Animation/AnimationState.hpp, 443, 445  
 src/Application.cpp, 447  
 src/Application.hpp, 448, 449  
 src/Component.cpp, 449  
 src/Component.hpp, 449, 450  
 src/Components/Common/Button.cpp, 451  
 src/Components/Common/Button.hpp, 451, 452  
 src/Components/Common/Card.cpp, 453  
 src/Components/Common/Card.hpp, 453, 454  
 src/Components/Common/CodeHighlighter.cpp, 455  
 src/Components/Common/CodeHighlighter.hpp, 455, 456  
 src/Components/Common/Footer.cpp, 457  
 src/Components/Common/Footer.hpp, 457, 458  
 src/Components/Common/InputField.cpp, 459  
 src/Components/Common/InputField.hpp, 459, 460  
 src/Components/Common/IntegerField.cpp, 461  
 src/Components/Common/IntegerField.hpp, 462  
 src/Components/Common/NavigationBar.cpp, 463  
 src/Components/Common/NavigationBar.hpp, 463, 464  
 src/Components/Common/OperationContainer.cpp, 465  
 src/Components/Common/OperationContainer.hpp, 465, 466  
 src/Components/Common/OperationList.cpp, 466  
 src/Components/Common/OperationList.hpp, 467  
 src/Components/Common/OptionInputField.cpp, 468  
 src/Components/Common/OptionInputField.hpp, 468, 469  
 src/Components/Common/StringInputField.cpp, 470  
 src/Components/Common/StringInputField.hpp, 470, 471  
 src/Components/Visualization/CircularLinkedList.cpp, 420

src/Components/Visualization/CircularLinkedList.hpp, 422, 423  
src/Components/Visualization/DataStructure.cpp, 471  
src/Components/Visualization/DataStructure.hpp, 471, 472  
src/Components/Visualization/DoublyLinkedList.cpp, 424  
src/Components/Visualization/DoublyLinkedList.hpp, 427  
src/Components/Visualization/DynamicArray.cpp, 413  
src/Components/Visualization/DynamicArray.hpp, 415, 416  
src/Components/Visualization/LinkedList.cpp, 473  
src/Components/Visualization/LinkedList.hpp, 473, 474  
src/Components/Visualization/Node.cpp, 475  
src/Components/Visualization/Node.hpp, 475, 476  
src/Components/Visualization/SinglyLinkedList.cpp, 431  
src/Components/Visualization/SinglyLinkedList.hpp, 433, 434  
src/Container.cpp, 478  
src/Container.hpp, 478, 479  
src/Core/List.hpp, 480, 481  
src/Core/Node.hpp, 477, 478  
src/FontHolder.cpp, 486  
src/FontHolder.hpp, 486, 487  
src/Global.hpp, 487, 488  
src/Identifiers/CategoryIdentifiers.hpp, 488  
src/Identifiers/CategoryInfo.cpp, 489  
src/Identifiers/CategoryInfo.hpp, 489, 490  
src/Identifiers/ColorThemelIdentifiers.hpp, 490, 491  
src/Identifiers/DSIdentifiers.hpp, 492, 493  
src/Identifiers/DSInfo.cpp, 493  
src/Identifiers/DSInfo.hpp, 494  
src/Identifiers/ResourceIdentifiers.hpp, 495  
src/Identifiers/StatIdentifiers.hpp, 496  
src/Main.cpp, 497  
src/NonCopyable.hpp, 497, 498  
src/SceneNode.cpp, 498  
src/SceneNode.hpp, 498, 499  
src/Settings.cpp, 500  
src/Settings.hpp, 500, 501  
src/State.cpp, 501  
src/State.hpp, 502, 503  
src/States/Array/ArrayType.hpp, 503, 504  
src/States/Array/DynamicArrayType.hpp, 507  
src/States/Array/DynamicArrayType.hpp, 508, 509  
src/States/Array/StaticArrayType.hpp, 509  
src/States/Array/StaticArrayType.hpp, 509, 510  
src/States/HomepageState.cpp, 511  
src/States/HomepageState.hpp, 511, 512  
src/States/LinkedList/CLLState.cpp, 512  
src/States/LinkedList/CLLState.hpp, 512, 513  
src/States/LinkedList/DLLState.cpp, 514  
src/States/LinkedList/DLLState.hpp, 514, 515  
src/States/LinkedList/LLState.hpp, 515, 516  
src/States/LinkedList/QueueState.cpp, 519  
src/States/LinkedList/QueueState.hpp, 519, 520  
src/States/LinkedList/SLLState.hpp, 521, 522  
src/States/LinkedList/StackState.cpp, 522  
src/States/LinkedList/StackState.hpp, 523, 524  
src/States/SettingsState.cpp, 524  
src/States/SettingsState.hpp, 524, 525  
src/StateStack.cpp, 525  
src/StateStack.hpp, 526  
src/TextureHolder.cpp, 527  
src/TextureHolder.hpp, 527, 528  
src/Utils/Utils.cpp, 529  
src/Utils/Utils.hpp, 529, 530  
Stack  
    Algorithm::Stack, 360, 361  
    DataStructures, 17  
    States, 20  
    Textures, 20  
Stack.hpp  
    ArrowType, 436  
    Orientation, 436  
StackState, 365  
    ~StackState, 368  
    activeDS, 372  
    AddDeleteOperation, 368  
    AddInitializeOperation, 368  
    AddInsertOperation, 368  
    AddIntFieldOperationOption, 368  
    AddNoFieldOperationOption, 369  
    AddOperations, 369  
    AddSearchOperation, 369  
    AddStringFieldOption, 369  
    AddUpdateOperation, 369  
    animController, 372  
    ClearAction, 369  
    ClearError, 370  
    codeHighlighter, 372  
    Draw, 370  
    DrawCurrentActionText, 370  
    DrawCurrentErrorText, 370  
    footer, 372  
    GetContext, 370  
    InitNavigationBar, 370  
    mContext, 372  
    mCurrentAction, 372  
    mCurrentError, 372  
    mStack, 372  
    mStackAlgorithm, 373  
    navigation, 373  
    operationList, 373  
    Ptr, 367  
    RequestStackClear, 370  
    RequestStackPop, 371  
    RequestStackPush, 371  
    SetCurrentAction, 371  
    SetCurrentError, 371  
    StackState, 368  
    Success, 371  
    Update, 371  
    State, 373

~State, 375  
 Draw, 375  
 GetContext, 375  
 GUI::Node, 272  
 InitNavigationBar, 375  
 mContext, 376  
 mStack, 376  
 navigation, 376  
 Ptr, 374  
 RequestStackClear, 375  
 RequestStackPop, 375  
 RequestStackPush, 376  
 State, 374  
 Update, 376  
 State::Context, 124  
 categories, 125  
 Context, 124  
 dsInfo, 125  
 fonts, 125  
 textures, 125  
 StateCount  
 GUI::Node, 272  
 stateID  
 DSInfo::Info, 204  
 GUI::Card, 69  
 GUI::NavigationBar::TitleInfo, 408  
 StateStack::PendingChange, 306  
 States, 19  
 CircularLinkedList, 20  
 Count, 20  
 DoublyLinkedList, 20  
 DynamicArray, 20  
 Homepage, 20  
 ID, 19  
 None, 20  
 Queue, 20  
 Settings, 20  
 SinglyLinkedList, 20  
 Stack, 20  
 StaticArray, 20  
 StateStack, 377  
 Action, 378  
 ApplyPendingChanges, 379  
 Clear, 378  
 ClearStates, 379  
 createState, 379  
 Draw, 379  
 IsEmpty, 379  
 mContext, 380  
 mFactories, 380  
 mPendingList, 380  
 mStack, 380  
 Pop, 378  
 PopState, 379  
 Push, 378  
 PushState, 379  
 RegisterState, 379  
 StateStack, 378  
 Update, 380  
 StateStack::PendingChange, 306  
 action, 306  
 PendingChange, 306  
 stateID, 306  
 StaticArray  
 Algorithm::StaticArray, 382  
 DataStructures, 17  
 States, 20  
 Textures, 20  
 StaticArrayState, 387  
 ~StaticArrayState, 389  
 activeDS, 393  
 AddAccessOperation, 389  
 AddDeleteOperation, 389  
 AddInitializeOperation, 390  
 AddInsertOperation, 390  
 AddIntFieldOperationOption, 390  
 AddNoFieldOperationOption, 390  
 AddOperations, 390  
 AddSearchOperation, 390  
 AddStringFieldOption, 391  
 AddUpdateOperation, 391  
 animController, 393  
 ClearAction, 391  
 ClearError, 391  
 codeHighlighter, 393  
 Draw, 391  
 DrawCurrentActionText, 391  
 DrawCurrentErrorText, 392  
 footer, 394  
 GetContext, 392  
 InitNavigationBar, 392  
 mContext, 394  
 mCurrentAction, 394  
 mCurrentError, 394  
 mStack, 394  
 mStaticArray, 394  
 navigation, 394  
 operationList, 395  
 Ptr, 389  
 RequestStackClear, 392  
 RequestStackPop, 392  
 RequestStackPush, 392  
 SetCurrentAction, 392  
 SetCurrentError, 393  
 StaticArrayState, 389  
 Success, 393  
 Update, 393  
 StepBackward  
 Animation::AnimationController< T >, 32  
 StepForward  
 Animation::AnimationController< T >, 32  
 stopDuration  
 Animation::AnimationController< T >, 34  
 StringInputField  
 GUI::StringInputField, 398  
 Success

ArrayState< T >, 49  
    CLLState, 97  
    DLLState, 138  
    DynamicArrayList, 186  
    LLState< T >, 255  
    QueueState, 320  
    SLLState, 356  
    StackState, 371  
    StaticArrayList, 393  
**swap**  
    Core::List< T >::const\_iterator, 117  
    Core::List< T >::iterator, 229  
  
    **TextAlignment**  
        GUI::Button, 54  
**textColor**  
    GUI::Button, 62  
**TextureHolder**, 404  
    ~TextureHolder, 406  
    Get, 406  
    InsertResource, 406  
    Load, 406  
    LoadFromImage, 407  
    mTextureMap, 407  
    TextureHolder, 406  
**Textures**, 20  
    Blank, 20  
    CircularLinkedList, 20  
    Count, 20  
    DoublyLinkedList, 20  
    DynamicArray, 20  
    Favicon, 20  
    ID, 20  
    Queue, 20  
    SinglyLinkedList, 20  
    Stack, 20  
    StaticArray, 20  
**textures**  
    State::Context, 125  
**thumbnail**  
    DSInfo::Info, 204  
    GUI::Card, 69  
**title**  
    GUI::Card, 69  
**titleName**  
    GUI::NavigationBar::TitleInfo, 408  
**toggleButton**  
    GUI::OperationList, 297  
**ToggleInputFields**  
    GUI::OptionInputField, 304  
**ToggleOperations**  
    GUI::OperationList, 295  
**ToggleShowAction**  
    GUI::CodeHighlighter, 105  
**ToggleShowCode**  
    GUI::CodeHighlighter, 105  
**ToggleVisible**  
    GUI::Button, 59  
    GUI::Card, 68  
  
    **GUI::CircularLinkedList**, 89  
    **GUI::CodeHighlighter**, 106  
    **GUI::Component**, 112  
    **GUI::Container**, 122  
    **GUI::DataStructure**, 130  
    **GUI::DoublyLinkedList**, 158  
    **GUI::DynamicArray**, 176  
    **GUI::Footer**< T >, 196  
    **GUI::InputField**, 212  
    **GUI::IntegerInputField**, 223  
    **GUI::LinkedList**, 239  
    **GUI::NavigationBar**, 264  
    **GUI::Node**, 278  
    **GUI::OperationContainer**, 288  
    **GUI::OperationList**, 295  
    **GUI::OptionInputField**, 304  
    **GUI::SinglyLinkedList**, 347  
    **GUI::StringInputField**, 402  
**toLink**  
    GUI::Card, 70  
    GUI::NavigationBar, 266  
  
    **UnpackAll**  
        GUI::CircularLinkedList, 89  
        GUI::Container, 122  
        GUI::DataStructure, 130  
        GUI::DoublyLinkedList, 158  
        GUI::DynamicArray, 177  
        GUI::Footer< T >, 196  
        GUI::LinkedList, 239  
        GUI::OperationContainer, 288  
        GUI::OperationList, 295  
        GUI::OptionInputField, 304  
        GUI::SinglyLinkedList, 347  
**Update**  
    Algorithm::CircularLinkedList, 77  
    Algorithm::DoublyLinkedList, 147  
    Algorithm::DynamicArray, 166  
    Algorithm::SinglyLinkedList, 337  
    Algorithm::StaticArray, 385  
    Animation::AnimationController< T >, 33  
    Animation::AnimationState< T >, 38  
    Application, 41  
    ArrayState< T >, 49  
    CLLState, 97  
    DLLState, 138  
    DynamicArrayList, 186  
    HomepageState, 201  
    LLState< T >, 255  
    QueueState, 320  
    SettingsState, 330  
    SLLState, 356  
    StackState, 371  
    State, 376  
    StateStack, 380  
    StaticArrayList, 393  
**UpdateMouseCursorWhenHover**  
    GUI::Button, 60  
    GUI::Card, 68

GUI::CircularLinkedList, 89  
 GUI::CodeHighlighter, 106  
 GUI::Component, 112  
 GUI::Container, 123  
 GUI::DataStructure, 131  
 GUI::DoublyLinkedList, 158  
 GUI::DynamicArray, 177  
 GUI::Footer< T >, 196  
 GUI::InputField, 212  
 GUI::IntegerField, 223  
 GUI::LinkedList, 239  
 GUI::NavigationBar, 264  
 GUI::Node, 278  
 GUI::OperationContainer, 288  
 GUI::OperationList, 296  
 GUI::OptionInputField, 304  
 GUI::SinglyLinkedList, 348  
 GUI::StringInputField, 402  
**UpdatePosition**  
 GUI::OperationContainer, 288  
**UserDefined**  
 Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 26  
 Algorithm::CircularLinkedList, 78  
 Algorithm::DoublyLinkedList, 147  
 Algorithm::DynamicArray, 167  
 Algorithm::Queue, 313  
 Algorithm::SinglyLinkedList, 337  
 Algorithm::Stack, 364  
 Algorithm::StaticArray, 385  
**UserDefinedGenerator**  
 Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 26  
 Algorithm::CircularLinkedList, 78  
 Algorithm::DoublyLinkedList, 147  
 Algorithm::DynamicArray, 167  
 Algorithm::Queue, 313  
 Algorithm::SinglyLinkedList, 338  
 Algorithm::Stack, 364  
 Algorithm::StaticArray, 386  
**Utils, 21**  
 DrawIcon, 21  
 DrawTextBoxed, 21  
 DrawTextBoxedSelectable, 21  
 OpenFileDialog, 22  
 Rand, 22  
 ReadInputFromFile, 22  
**value\_type**  
 Core::List< T >::const\_iterator, 114  
 Core::List< T >::iterator, 227  
**valueFontSize**  
 GUI::Node, 281  
**Vertical**  
 GUI::CircularLinkedList, 83  
 GUI::DoublyLinkedList, 152  
 GUI::LinkedList, 234  
 GUI::SinglyLinkedList, 342  
**visualizer**

Algorithm::Algorithm< GUIAlgorithm, Animation-State >, 27  
 Algorithm::CircularLinkedList, 78  
 Algorithm::DoublyLinkedList, 148  
 Algorithm::DynamicArray, 167  
 Algorithm::Queue, 313  
 Algorithm::SinglyLinkedList, 338  
 Algorithm::Stack, 365  
 Algorithm::StaticArray, 386  
**Visualizer\_Arrow\_Active**  
 ColorTheme, 16  
**Visualizer\_Arrow\_Default**  
 ColorTheme, 16  
**Visualizer\_Label**  
 ColorTheme, 15  
**Visualizer\_Node\_Active\_Background1**  
 ColorTheme, 16  
**Visualizer\_Node\_Active\_Background2**  
 ColorTheme, 16  
**Visualizer\_Node\_Active\_Outline1**  
 ColorTheme, 16  
**Visualizer\_Node\_Active\_Outline2**  
 ColorTheme, 16  
**Visualizer\_Node\_Active\_Text1**  
 ColorTheme, 16  
**Visualizer\_Node\_Active\_Text2**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveBlue\_Background1**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveBlue\_Background2**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveBlue\_Outline1**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveBlue\_Outline2**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveBlue\_Text1**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveBlue\_Text2**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveGreen\_Background1**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveGreen\_Background2**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveGreen\_Outline1**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveGreen\_Outline2**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveGreen\_Text1**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveGreen\_Text2**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveRed\_Background1**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveRed\_Background2**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveRed\_Outline1**  
 ColorTheme, 16  
**Visualizer\_Node\_ActiveRed\_Outline2**

ColorTheme, 16  
Visualizer\_Node\_ActiveRed\_Text1  
    ColorTheme, 16  
Visualizer\_Node\_ActiveRed\_Text2  
    ColorTheme, 16  
Visualizer\_Node\_Default\_Background1  
    ColorTheme, 16  
Visualizer\_Node\_Default\_Background2  
    ColorTheme, 16  
Visualizer\_Node\_Default\_Outline1  
    ColorTheme, 15  
Visualizer\_Node\_Default\_Outline2  
    ColorTheme, 16  
Visualizer\_Node\_Default\_Text1  
    ColorTheme, 16  
Visualizer\_Node\_Default\_Text2  
    ColorTheme, 16  
Visualizer\_Node\_Iterated\_Background1  
    ColorTheme, 16  
Visualizer\_Node\_Iterated\_Background2  
    ColorTheme, 16  
Visualizer\_Node\_Iterated\_Outline1  
    ColorTheme, 16  
Visualizer\_Node\_Iterated\_Outline2  
    ColorTheme, 16  
Visualizer\_Node\_Iterated\_Text1  
    ColorTheme, 16  
Visualizer\_Node\_Iterated\_Text2  
    ColorTheme, 16

width  
    ArrayState< T >::IntegerInput, 214  
    LLState< T >::IntegerInput, 215

WindowClosed  
    Application, 41