

Arcade

Generated by Doxygen 1.9.1

Wed Apr 5 2023 19:31:50

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Data Structure Index	3
2.1 Data Structures	3
3 File Index	5
3.1 File List	5
4 Data Structure Documentation	7
4.1 ArcadeError Class Reference	7
4.1.1 Constructor & Destructor Documentation	8
4.1.1.1 ArcadeError()	8
4.1.1.2 ~ArcadeError()	8
4.1.2 Member Function Documentation	9
4.1.2.1 what()	9
4.1.3 Field Documentation	9
4.1.3.1 msg_	9
4.2 color Struct Reference	9
4.2.1 Field Documentation	10
4.2.1.1 a	10
4.2.1.2 b	10
4.2.1.3 g	10
4.2.1.4 r	10
4.3 coord Struct Reference	10
4.3.1 Field Documentation	11
4.3.1.1 x	11
4.3.1.2 y	11
4.4 Core Class Reference	11
4.4.1 Constructor & Destructor Documentation	13
4.4.1.1 Core()	13
4.4.1.2 ~Core()	13
4.4.2 Member Function Documentation	14
4.4.2.1 coreStateHandler()	14
4.4.2.2 findPathIndex()	14
4.4.2.3 gameLoopHandler()	14
4.4.2.4 getAllLib()	14
4.4.2.5 getCoreState()	14
4.4.2.6 getCurrentGame()	14
4.4.2.7 getCurrentGraph()	14
4.4.2.8 getGamePaths()	15
4.4.2.9 getGraphPaths()	15
4.4.2.10 handleEvent()	15

4.4.2.11 isLibGraphical()	15
4.4.2.12 loadNextGame()	15
4.4.2.13 loadNextGraph()	15
4.4.2.14 loadSpecificGame()	15
4.4.2.15 loadSpecificGraph()	16
4.4.2.16 pushLib()	16
4.4.2.17 quitArcade()	16
4.4.2.18 restartGame()	16
4.4.2.19 setCoreState()	16
4.4.2.20 setCurrentGame()	16
4.4.2.21 setCurrentGraph()	16
4.4.3 Field Documentation	17
4.4.3.1 currentGame_	17
4.4.3.2 currentGraph_	17
4.4.3.3 gameLib_	17
4.4.3.4 gameLoader_	17
4.4.3.5 gamePaths_	17
4.4.3.6 gameState_	17
4.4.3.7 graphLib_	17
4.4.3.8 graphLoader_	18
4.4.3.9 graphPaths_	18
4.4.3.10 menu_	18
4.5 DLLoader< T > Class Template Reference	18
4.5.1 Constructor & Destructor Documentation	19
4.5.1.1 DLLoader()	19
4.5.1.2 ~DLLoader()	19
4.5.2 Member Function Documentation	19
4.5.2.1 closeLib()	20
4.5.2.2 getInstance()	20
4.5.3 Field Documentation	20
4.5.3.1 actualLib_	20
4.6 elemSize Struct Reference	20
4.6.1 Field Documentation	21
4.6.1.1 height	21
4.6.1.2 width	21
4.7 IGame Class Reference	21
4.7.1 Constructor & Destructor Documentation	23
4.7.1.1 ~IGame()	23
4.7.2 Member Function Documentation	23
4.7.2.1 display()	23
4.7.2.2 getDisplaySize()	24
4.7.2.3 getSpritePath()	24

4.7.2.4 init()	24
4.7.2.5 reset()	24
4.7.2.6 updateGame()	24
4.8 IGraphic Class Reference	25
4.8.1 Constructor & Destructor Documentation	26
4.8.1.1 ~IGraphic()	26
4.8.2 Member Function Documentation	26
4.8.2.1 clearWindow()	27
4.8.2.2 createWindow()	27
4.8.2.3 destroyWindow()	27
4.8.2.4 displayShape()	27
4.8.2.5 displaySprite()	27
4.8.2.6 displayText()	27
4.8.2.7 displayWindow()	28
4.8.2.8 getEvent()	28
4.8.2.9 isOpenWindow()	28
4.9 Menu Class Reference	28
4.9.1 Constructor & Destructor Documentation	30
4.9.1.1 Menu()	31
4.9.1.2 ~Menu()	31
4.9.2 Member Function Documentation	31
4.9.2.1 applyChanges()	31
4.9.2.2 chooseGame()	31
4.9.2.3 chooseLib()	31
4.9.2.4 createGuiTextMenu()	31
4.9.2.5 createNewUser()	31
4.9.2.6 createTitleMenu()	32
4.9.2.7 deleteChar()	32
4.9.2.8 getTop3Scores()	32
4.9.2.9 getUsername()	32
4.9.2.10 handleEvent()	32
4.9.2.11 handleUserInput()	32
4.9.2.12 highlightSelected()	32
4.9.2.13 highlightTitle()	33
4.9.2.14 isUserTyping()	33
4.9.2.15 loopTitle()	33
4.9.2.16 menuLoopHandler()	33
4.9.2.17 moveDown()	33
4.9.2.18 moveUp()	33
4.9.2.19 saveUserName()	33
4.9.2.20 setAvailableLibText()	34
4.9.2.21 setCursorsMenu()	34

4.9.2.22 setGameLibText()	34
4.9.2.23 setGraphLibText()	34
4.9.2.24 setHighScoreText()	34
4.9.2.25 setLibNameMenu()	34
4.9.2.26 setScoreboardTitle()	34
4.9.2.27 setUsernameText()	34
4.9.3 Field Documentation	35
4.9.3.1 counter_	35
4.9.3.2 gamePaths_	35
4.9.3.3 gameTextMenu_	35
4.9.3.4 graphPaths_	35
4.9.3.5 guiTextMenu_	35
4.9.3.6 incrGame_	35
4.9.3.7 incrLib_	35
4.9.3.8 isGameSelected_	36
4.9.3.9 isUserTyping_	36
4.9.3.10 keyMap_	36
4.9.3.11 lastUpdateTime_	36
4.9.3.12 libTextMenu_	36
4.9.3.13 scoreText_	36
4.9.3.14 titleMenu_	36
4.9.3.15 userName_	37
4.10 NcursesGraphic Class Reference	37
4.10.1 Constructor & Destructor Documentation	39
4.10.1.1 NcursesGraphic()	39
4.10.1.2 ~NcursesGraphic()	39
4.10.2 Member Function Documentation	39
4.10.2.1 clearWindow()	39
4.10.2.2 createWindow()	39
4.10.2.3 destroyWindow()	40
4.10.2.4 displayShape()	40
4.10.2.5 displaySprite()	40
4.10.2.6 displayText()	40
4.10.2.7 displayWindow()	40
4.10.2.8 drawRectangle()	40
4.10.2.9 getEvent()	41
4.10.2.10 isOpenWindow()	41
4.10.3 Field Documentation	41
4.10.3.1 isOpen_	41
4.10.3.2 window_	41
4.11 Nibbler Class Reference	42
4.11.1 Constructor & Destructor Documentation	44

4.11.1.1 Nibbler()	44
4.11.1.2 ~Nibbler()	45
4.11.2 Member Function Documentation	45
4.11.2.1 addFood()	45
4.11.2.2 addWall()	45
4.11.2.3 chooseDirection()	45
4.11.2.4 display()	45
4.11.2.5 foodHandler()	45
4.11.2.6 getDisplaySize()	46
4.11.2.7 getSpritePath()	46
4.11.2.8 init()	46
4.11.2.9 initNibbler()	46
4.11.2.10 initText()	46
4.11.2.11 isCollided()	46
4.11.2.12 isNibblerInCell()	47
4.11.2.13 loadMap()	47
4.11.2.14 moveSnake()	47
4.11.2.15 reset()	47
4.11.2.16 resetLevel()	47
4.11.2.17 restartEvent()	47
4.11.2.18 saveUserScore()	47
4.11.2.19 updateDirection()	48
4.11.2.20 updateGame()	48
4.11.3 Field Documentation	48
4.11.3.1 cellHeight_	48
4.11.3.2 cellWidth_	48
4.11.3.3 dir_	48
4.11.3.4 food_	48
4.11.3.5 lastUpdateTime_	48
4.11.3.6 mapIndex_	49
4.11.3.7 nibbler_	49
4.11.3.8 nibblerSize_	49
4.11.3.9 remainingFood_	49
4.11.3.10 score_	49
4.11.3.11 state	49
4.11.3.12 texts_	49
4.11.3.13 timer_	49
4.11.3.14 walls_	50
4.12 SDLGraphic Class Reference	50
4.12.1 Constructor & Destructor Documentation	53
4.12.1.1 SDLGraphic()	53
4.12.1.2 ~SDLGraphic()	53

4.12.2 Member Function Documentation	53
4.12.2.1 clearWindow()	53
4.12.2.2 createWindow()	54
4.12.2.3 destroyWindow()	54
4.12.2.4 displayShape()	54
4.12.2.5 displaySprite()	54
4.12.2.6 displayText()	54
4.12.2.7 displayWindow()	55
4.12.2.8 drawCircle()	55
4.12.2.9 drawRectangle()	55
4.12.2.10 getEvent()	55
4.12.2.11 isOpenWindow()	55
4.12.3 Field Documentation	55
4.12.3.1 fonts_	55
4.12.3.2 isOpen_	56
4.12.3.3 renderer_	56
4.12.3.4 spriteSurface_	56
4.12.3.5 spriteTexture_	56
4.12.3.6 textPath_	56
4.12.3.7 window_	56
4.13 SFMLGraphic Class Reference	57
4.13.1 Constructor & Destructor Documentation	59
4.13.1.1 SFMLGraphic()	59
4.13.1.2 ~SFMLGraphic()	59
4.13.2 Member Function Documentation	59
4.13.2.1 clearWindow()	60
4.13.2.2 createWindow()	60
4.13.2.3 destroyWindow()	60
4.13.2.4 displayShape()	60
4.13.2.5 displaySprite()	60
4.13.2.6 displayText()	60
4.13.2.7 displayWindow()	61
4.13.2.8 drawCircle()	61
4.13.2.9 drawRectangle()	61
4.13.2.10 getEvent()	61
4.13.2.11 isOpenWindow()	61
4.13.3 Field Documentation	61
4.13.3.1 circleList	61
4.13.3.2 fonts_	62
4.13.3.3 isLoadingTexture_	62
4.13.3.4 rectList	62
4.13.3.5 spriteList_	62

4.13.3.6 textList	62
4.13.3.7 textPath_	62
4.13.3.8 texture_	62
4.13.3.9 window_	63
4.14 shape Struct Reference	63
4.14.1 Field Documentation	64
4.14.1.1 m_color	64
4.14.1.2 pos	64
4.14.1.3 replacementChar	64
4.14.1.4 size	64
4.14.1.5 text	64
4.14.1.6 type	64
4.15 Snake Class Reference	65
4.15.1 Constructor & Destructor Documentation	67
4.15.1.1 Snake()	67
4.15.1.2 ~Snake()	68
4.15.2 Member Function Documentation	68
4.15.2.1 addFood()	68
4.15.2.2 addWall()	68
4.15.2.3 display()	68
4.15.2.4 foodHandler()	68
4.15.2.5 gameOver()	68
4.15.2.6 getDisplaySize()	69
4.15.2.7 getSpritePath()	69
4.15.2.8 init()	69
4.15.2.9 initSnake()	69
4.15.2.10 initText()	69
4.15.2.11 isCollided()	69
4.15.2.12 isSnakeInCell()	70
4.15.2.13 loadMap()	70
4.15.2.14 moveSnake()	70
4.15.2.15 reset()	70
4.15.2.16 resetLevel()	70
4.15.2.17 restartEvent()	70
4.15.2.18 saveUserScore()	70
4.15.2.19 updateDirection()	71
4.15.2.20 updateGame()	71
4.15.3 Field Documentation	71
4.15.3.1 cellHeight_	71
4.15.3.2 cellWidth_	71
4.15.3.3 dir_	71
4.15.3.4 food_	71

4.15.3.5 lastUpdateTime_	71
4.15.3.6 mapIndex_	72
4.15.3.7 score_	72
4.15.3.8 snake_	72
4.15.3.9 snakeSize_	72
4.15.3.10 state	72
4.15.3.11 texts_	72
4.15.3.12 timer_	72
4.15.3.13 walls_	73
4.16 sprite Struct Reference	73
4.16.1 Field Documentation	74
4.16.1.1 m_color	74
4.16.1.2 m_texture	74
4.16.1.3 pos	74
4.16.1.4 replacementChar	74
4.16.1.5 size	74
4.17 text Struct Reference	75
4.17.1 Field Documentation	75
4.17.1.1 fontPath	75
4.17.1.2 fontSize	76
4.17.1.3 m_color	76
4.17.1.4 pos	76
4.17.1.5 text	76
4.18 texture Struct Reference	76
4.18.1 Field Documentation	77
4.18.1.1 path	77
4.18.1.2 pos	77
4.18.1.3 size	77
5 File Documentation	79
5.1 include/core/Core.hpp File Reference	79
5.1.1 Enumeration Type Documentation	80
5.1.1.1 GState	80
5.1.1.2 libType	80
5.1.2 Variable Documentation	80
5.1.2.1 validLibs	80
5.2 include/core/DLLoader.hpp File Reference	81
5.3 include/core/Menu.hpp File Reference	82
5.4 include/core/Parser.hpp File Reference	83
5.4.1 Function Documentation	83
5.4.1.1 checkArgs()	83
5.4.1.2 checkEnv()	83

5.4.1.3 checkLibrary()	83
5.4.1.4 displayUsage()	84
5.5 include/error/Error.hpp File Reference	84
5.6 include/games/IGame.hpp File Reference	84
5.7 include/games/nibbler/Nibbler.hpp File Reference	85
5.7.1 Enumeration Type Documentation	87
5.7.1.1 direction	87
5.7.1.2 playerState	87
5.7.2 Variable Documentation	87
5.7.2.1 allMaps	87
5.8 include/games/snake/Snake.hpp File Reference	88
5.8.1 Enumeration Type Documentation	89
5.8.1.1 direction	89
5.8.1.2 playerState	89
5.8.2 Variable Documentation	89
5.8.2.1 allMaps	89
5.9 include/graphics/IGraphic.hpp File Reference	90
5.9.1 Enumeration Type Documentation	91
5.9.1.1 eventKey	91
5.9.1.2 shapeType	92
5.10 include/graphics/ncurses/NcursesEventKey.hpp File Reference	92
5.10.1 Variable Documentation	93
5.10.1.1 keyEvent	94
5.11 include/graphics/ncurses/NcursesGraphic.hpp File Reference	94
5.12 include/graphics/sdl/SDLEventKey.hpp File Reference	95
5.12.1 Variable Documentation	96
5.12.1.1 keyEvent	96
5.13 include/graphics/sdl/SDLGraphic.hpp File Reference	96
5.14 include/graphics/sfml/SFMLEventKey.hpp File Reference	97
5.14.1 Variable Documentation	98
5.14.1.1 keyEvent	98
5.15 include/graphics/sfml/SFMLGraphic.hpp File Reference	99
5.16 src/core/ActionsMenu.cpp File Reference	99
5.17 src/core/ArgsHandler.cpp File Reference	100
5.17.1 Function Documentation	100
5.17.1.1 checkArgs()	100
5.17.1.2 checkEnv()	100
5.17.1.3 checkLibrary()	101
5.17.1.4 displayUsage()	101
5.18 src/core/Core.cpp File Reference	101
5.19 src/core/Menu.cpp File Reference	101
5.20 src/core/SetMenuText.cpp File Reference	102

5.21 src/core/UserNameMenu.cpp File Reference	102
5.21.1 Function Documentation	102
5.21.1.1 random()	102
5.22 src/error/Error.cpp File Reference	103
5.23 src/games/nibbler/Compute.cpp File Reference	103
5.24 src/games/snake/Compute.cpp File Reference	104
5.24.1 Function Documentation	104
5.24.1.1 random()	104
5.25 src/games/nibbler/Display.cpp File Reference	105
5.26 src/games/snake/Display.cpp File Reference	105
5.27 src/games/nibbler/Nibbler.cpp File Reference	107
5.28 src/games/nibbler/NibblerLib.cpp File Reference	108
5.28.1 Function Documentation	108
5.28.1.1 entryPoint()	108
5.29 src/games/snake/Snake.cpp File Reference	109
5.30 src/games/snake/SnakeLib.cpp File Reference	110
5.30.1 Function Documentation	110
5.30.1.1 entryPoint()	110
5.31 src/graphics/ncurses/NcursesEvent.cpp File Reference	111
5.32 src/graphics/ncurses/NcursesGraphic.cpp File Reference	111
5.33 src/graphics/ncurses/NcursesGraphicLib.cpp File Reference	112
5.33.1 Function Documentation	113
5.33.1.1 entryPoint()	113
5.34 src/graphics/sdl/SDLEvent.cpp File Reference	113
5.35 src/graphics/sdl/SDLGraphic.cpp File Reference	114
5.36 src/graphics/sdl/SDLGraphicLib.cpp File Reference	114
5.36.1 Function Documentation	115
5.36.1.1 entryPoint()	115
5.37 src/graphics/sfml/SFMLEvent.cpp File Reference	115
5.38 src/graphics/sfml/SFMLGraphic.cpp File Reference	116
5.39 src/graphics/sfml/SFMLGraphicLib.cpp File Reference	116
5.39.1 Function Documentation	116
5.39.1.1 entryPoint()	116
5.40 src/Main.cpp File Reference	117
5.40.1 Function Documentation	117
5.40.1.1 main()	117
5.41 tests/tests_args.cpp File Reference	117
5.41.1 Function Documentation	118
5.41.1.1 Test() [1/9]	118
5.41.1.2 Test() [2/9]	118
5.41.1.3 Test() [3/9]	118
5.41.1.4 Test() [4/9]	118

5.41.1.5 Test() [5/9]	119
5.41.1.6 Test() [6/9]	119
5.41.1.7 Test() [7/9]	119
5.41.1.8 Test() [8/9]	119
5.41.1.9 Test() [9/9]	119
5.42 tests/tests_menu.cpp File Reference	119
5.42.1 Function Documentation	120
5.42.1.1 Test() [1/4]	120
5.42.1.2 Test() [2/4]	120
5.42.1.3 Test() [3/4]	120
5.42.1.4 Test() [4/4]	120
Index	121

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

color	9
coord	10
Core	11
DLLoader< T >	18
DLLoader< IGame >	18
DLLoader< IGraphic >	18
elemSize	20
std::exception	
ArcadeError	7
IGame	21
Nibbler	42
Snake	65
IGraphic	25
NcursesGraphic	37
SDLGraphic	50
SFMLGraphic	57
Menu	28
shape	63
sprite	73
text	75
texture	76

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

ArcadeError	7
color	9
coord	10
Core	11
DLLoader< T >	18
elemSize	20
IGame	21
IGraphic	25
Menu	28
NcursesGraphic	37
Nibbler	42
SDLGraphic	50
SFMLGraphic	57
shape	63
Snake	65
sprite	73
text	75
texture	76

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

include/core/Core.hpp	79
include/core/DLLoader.hpp	81
include/core/Menu.hpp	82
include/core/Parser.hpp	83
include/error/Error.hpp	84
include/games/IGame.hpp	84
include/games/nibbler/Nibbler.hpp	85
include/games/snake/Snake.hpp	88
include/graphics/IGraphic.hpp	90
include/graphics/ncurses/NcursesEventKey.hpp	92
include/graphics/ncurses/NcursesGraphic.hpp	94
include/graphics/sdl/SDLEventKey.hpp	95
include/graphics/sdl/SDLGraphic.hpp	96
include/graphics/sfml/SFMLEventKey.hpp	97
include/graphics/sfml/SFMLGraphic.hpp	99
src/Main.cpp	117
src/core/ActionsMenu.cpp	99
src/core/ArgsHandler.cpp	100
src/core/Core.cpp	101
src/core/Menu.cpp	101
src/core/SetMenuText.cpp	102
src/core/UserNameMenu.cpp	102
src/error/Error.cpp	103
src/games/nibbler/Compute.cpp	103
src/games/nibbler/Display.cpp	105
src/games/nibbler/Nibbler.cpp	107
src/games/nibbler/NibblerLib.cpp	108
src/games/snake/Compute.cpp	104
src/games/snake/Display.cpp	105
src/games/snake/Snake.cpp	109
src/games/snake/SnakeLib.cpp	110
src/graphics/ncurses/NcursesEvent.cpp	111
src/graphics/ncurses/NcursesGraphic.cpp	111
src/graphics/ncurses/NcursesGraphicLib.cpp	112
src/graphics/sdl/SDLEvent.cpp	113

src/graphics/sdl/SDLGraphic.cpp	114
src/graphics/sdl/SDLGraphicLib.cpp	114
src/graphics/sfml/SFMLEvent.cpp	115
src/graphics/sfml/SFMLGraphic.cpp	116
src/graphics/sfml/SFMLGraphicLib.cpp	116
tests/tests_args.cpp	117
tests/tests_menu.cpp	119

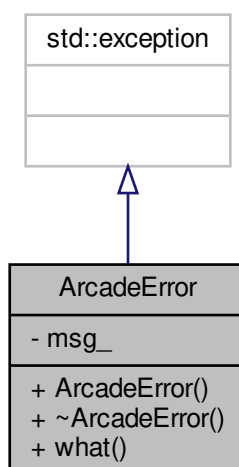
Chapter 4

Data Structure Documentation

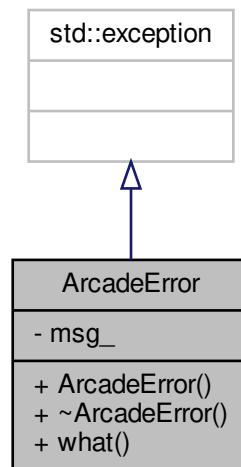
4.1 ArcadeError Class Reference

```
#include <Error.hpp>
```

Inheritance diagram for ArcadeError:



Collaboration diagram for `ArcadeError`:



Public Member Functions

- [ArcadeError](#) (const char *msg)
- [~ArcadeError](#) () override=default
- const char * [what](#) () const noexcept override

Private Attributes

- const char * [msg_](#)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 ArcadeError()

```
ArcadeError::ArcadeError (
    const char * msg )
```

4.1.1.2 ~ArcadeError()

```
ArcadeError::~~ArcadeError ( ) [override], [default]
```

4.1.2 Member Function Documentation

4.1.2.1 what()

```
const char * ArcadeError::what ( ) const [override], [noexcept]
```

4.1.3 Field Documentation

4.1.3.1 msg_

```
const char* ArcadeError::msg_ [private]
```

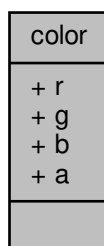
The documentation for this class was generated from the following files:

- include/error/[Error.hpp](#)
- src/error/[Error.cpp](#)

4.2 color Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for color:



Data Fields

- unsigned short [r](#)
- unsigned short [g](#)
- unsigned short [b](#)
- unsigned short [a](#)

4.2.1 Field Documentation

4.2.1.1 a

```
unsigned short color::a
```

4.2.1.2 b

```
unsigned short color::b
```

4.2.1.3 g

```
unsigned short color::g
```

4.2.1.4 r

```
unsigned short color::r
```

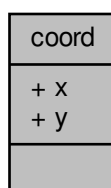
The documentation for this struct was generated from the following file:

- [include/graphics/IGraphic.hpp](#)

4.3 coord Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for coord:



Data Fields

- int [x](#)
- int [y](#)

4.3.1 Field Documentation

4.3.1.1 x

```
int coord::x
```

4.3.1.2 y

```
int coord::y
```

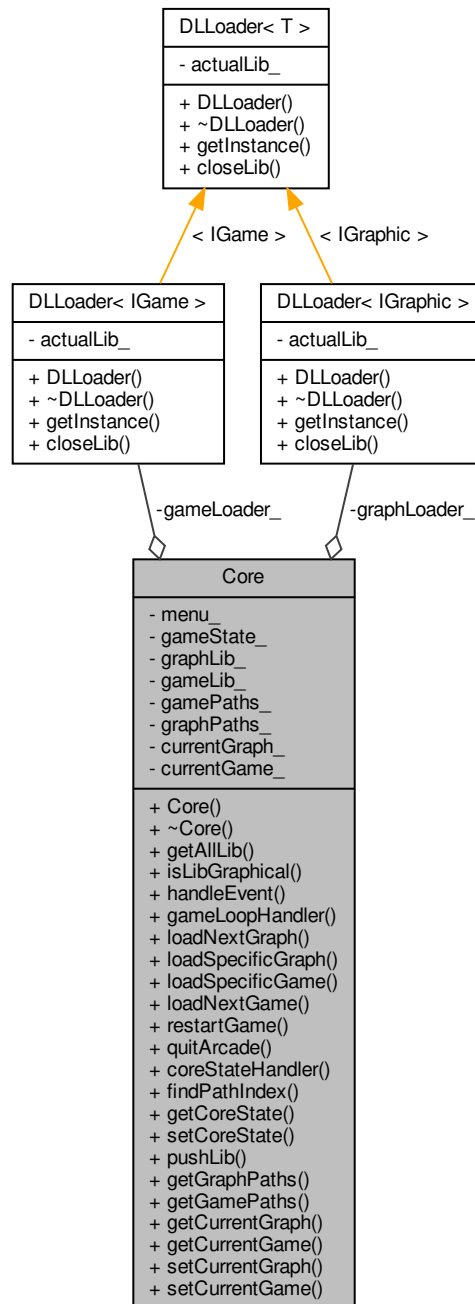
The documentation for this struct was generated from the following file:

- [include/graphics/IGraphic.hpp](#)

4.4 Core Class Reference

```
#include <Core.hpp>
```

Collaboration diagram for Core:



Public Member Functions

- [Core](#) (const char *libName)
- [~Core](#) ()=default
- void [getAllLib](#) ()
- void [isLibGraphical](#) (const std::string &libName)
- void [handleEvent](#) ()

- void [gameLoopHandler](#) ()
- void [loadNextGraph](#) ()
- void [loadSpecificGraph](#) (std::string path)
- void [loadSpecificGame](#) (std::string path)
- void [loadNextGame](#) ()
- void [restartGame](#) ()
- void [quitArcade](#) ()
- void [coreStateHandler](#) ()
- int [findPathIndex](#) (const std::string &path, const std::vector< std::string > &vec) const
- [GState](#) [getCoreState](#) () const
- void [setCoreState](#) (const [GState](#) &state)
- void [pushLib](#) (const std::string &path, std::vector< std::string > &container)
- std::vector< std::string > [getGraphPaths](#) ()
- std::vector< std::string > [getGamePaths](#) ()
- std::string [getCurrentGraph](#) () const
- std::string [getCurrentGame](#) () const
- void [setCurrentGraph](#) (const std::string ¤tGraph)
- void [setCurrentGame](#) (const std::string ¤tGame)

Private Attributes

- std::unique_ptr< [Menu](#) > [menu_](#)
- [GState](#) [gameState_](#)
- std::unique_ptr< [IGraphic](#) > [graphLib_](#)
- std::unique_ptr< [IGame](#) > [gameLib_](#)
- [DLLoader](#)< [IGraphic](#) > [graphLoader_](#) { [DLLoader](#)<[IGraphic](#)>() }
- [DLLoader](#)< [IGame](#) > [gameLoader_](#) { [DLLoader](#)<[IGame](#)>() }
- std::vector< std::string > [gamePaths_](#)
- std::vector< std::string > [graphPaths_](#)
- std::string [currentGraph_](#)
- std::string [currentGame_](#)

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Core()

```
Core::Core (
    const char * libName )
```

4.4.1.2 ~Core()

```
Core::~Core ( ) [default]
```

4.4.2 Member Function Documentation

4.4.2.1 coreStateHandler()

```
void Core::coreStateHandler ( )
```

4.4.2.2 findPathIndex()

```
int Core::findPathIndex (
    const std::string & path,
    const std::vector< std::string > & vec ) const
```

4.4.2.3 gameLoopHandler()

```
void Core::gameLoopHandler ( )
```

4.4.2.4 getAllLib()

```
void Core::getAllLib ( )
```

4.4.2.5 getCoreState()

```
GState Core::getCoreState ( ) const
```

4.4.2.6 getCurrentGame()

```
std::string Core::getCurrentGame ( ) const
```

4.4.2.7 getCurrentGraph()

```
std::string Core::getCurrentGraph ( ) const
```

4.4.2.8 getGamePaths()

```
std::vector< std::string > Core::getGamePaths ( )
```

4.4.2.9 getGraphPaths()

```
std::vector< std::string > Core::getGraphPaths ( )
```

4.4.2.10 handleEvent()

```
void Core::handleEvent ( )
```

4.4.2.11 isLibGraphical()

```
void Core::isLibGraphical (
    const std::string & libName )
```

4.4.2.12 loadNextGame()

```
void Core::loadNextGame ( )
```

4.4.2.13 loadNextGraph()

```
void Core::loadNextGraph ( )
```

4.4.2.14 loadSpecificGame()

```
void Core::loadSpecificGame (
    std::string path )
```

4.4.2.15 loadSpecificGraph()

```
void Core::loadSpecificGraph (
    std::string path )
```

4.4.2.16 pushLib()

```
void Core::pushLib (
    const std::string & path,
    std::vector< std::string > & container )
```

4.4.2.17 quitArcade()

```
void Core::quitArcade ( )
```

4.4.2.18 restartGame()

```
void Core::restartGame ( )
```

4.4.2.19 setCoreState()

```
void Core::setCoreState (
    const GState & state )
```

4.4.2.20 setCurrentGame()

```
void Core::setCurrentGame (
    const std::string & currentGame )
```

4.4.2.21 setCurrentGraph()

```
void Core::setCurrentGraph (
    const std::string & currentGraph )
```

4.4.3 Field Documentation

4.4.3.1 currentGame_

```
std::string Core::currentGame_ [private]
```

4.4.3.2 currentGraph_

```
std::string Core::currentGraph_ [private]
```

4.4.3.3 gameLib_

```
std::unique_ptr<IGame> Core::gameLib_ [private]
```

4.4.3.4 gameLoader_

```
DLLoader<IGame> Core::gameLoader_ { DLLoader<IGame>() } [private]
```

4.4.3.5 gamePaths_

```
std::vector<std::string> Core::gamePaths_ [private]
```

4.4.3.6 gameState_

```
GState Core::gameState_ [private]
```

4.4.3.7 graphLib_

```
std::unique_ptr<IGraphic> Core::graphLib_ [private]
```

4.4.3.8 graphLoader_

```
DLLoader<IGraphic> Core::graphLoader_ { DLoader<IGraphic>() } [private]
```

4.4.3.9 graphPaths_

```
std::vector<std::string> Core::graphPaths_ [private]
```

4.4.3.10 menu_

```
std::unique_ptr<Menu> Core::menu_ [private]
```

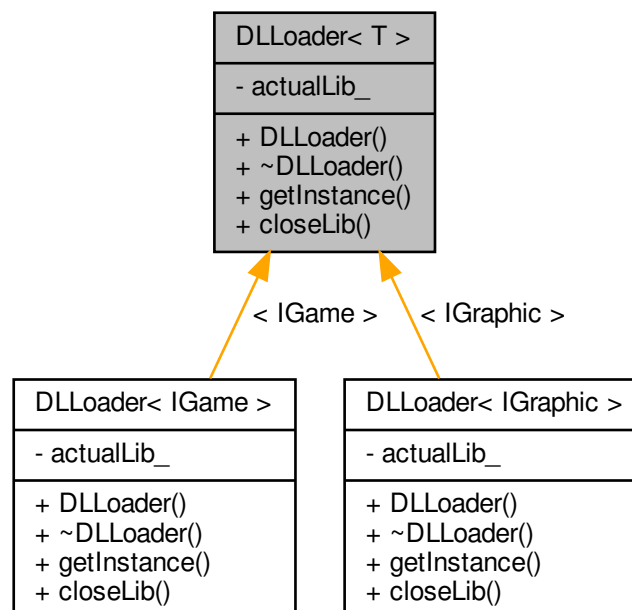
The documentation for this class was generated from the following files:

- include/core/Core.hpp
- src/core/Core.cpp

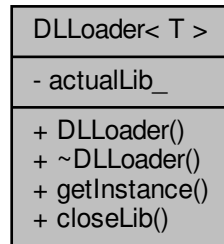
4.5 DLoader< T > Class Template Reference

```
#include <DLoader.hpp>
```

Inheritance diagram for DLoader< T >:



Collaboration diagram for DLoader< T >:



Public Member Functions

- [DLoader](#) ()=default
- [~DLoader](#) ()=default
- std::unique_ptr< T > [getInstance](#) (std::string filename)
- void [closeLib](#) (void)

Private Attributes

- void * [actualLib_](#) { nullptr }

4.5.1 Constructor & Destructor Documentation

4.5.1.1 DLoader()

```
template<typename T >
DLoader< T >::DLoader ( ) [default]
```

4.5.1.2 ~DLoader()

```
template<typename T >
DLoader< T >::~~DLoader ( ) [default]
```

4.5.2 Member Function Documentation

4.5.2.1 closeLib()

```
template<typename T >
void DLLoader< T >::closeLib (
    void ) [inline]
```

4.5.2.2 getInstance()

```
template<typename T >
std::unique_ptr<T> DLLoader< T >::getInstance (
    std::string filename ) [inline]
```

4.5.3 Field Documentation

4.5.3.1 actualLib_

```
template<typename T >
void* DLLoader< T >::actualLib_ { nullptr } [private]
```

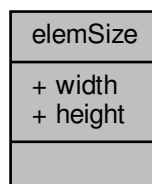
The documentation for this class was generated from the following file:

- [include/core/DLLoader.hpp](#)

4.6 elemSize Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for elemSize:



Data Fields

- int [width](#)
- int [height](#)

4.6.1 Field Documentation

4.6.1.1 height

```
int elemSize::height
```

4.6.1.2 width

```
int elemSize::width
```

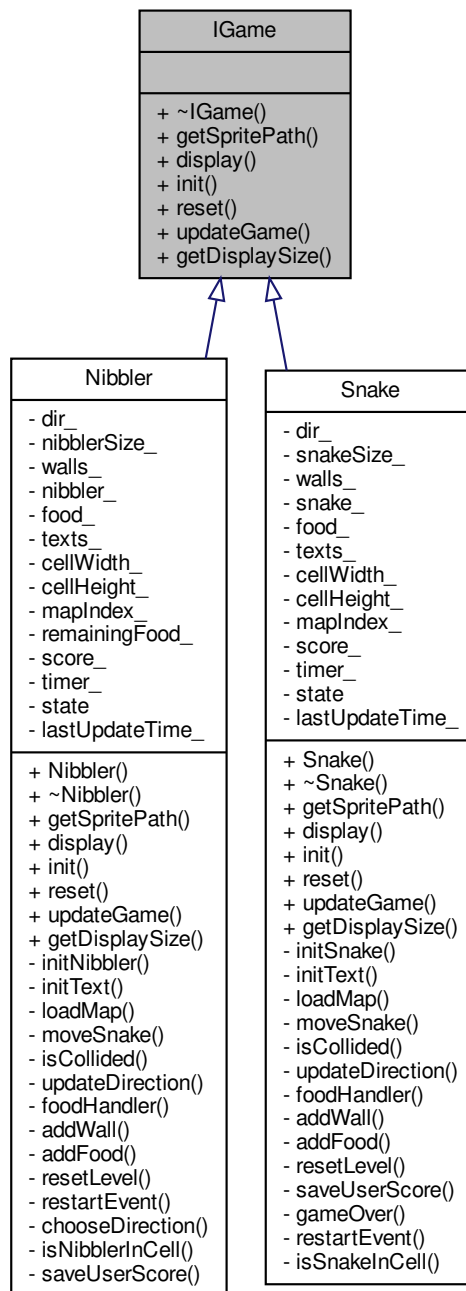
The documentation for this struct was generated from the following file:

- [include/graphics/IGraphic.hpp](#)

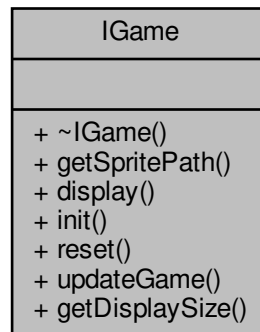
4.7 IGame Class Reference

```
#include <IGame.hpp>
```

Inheritance diagram for IGame:



Collaboration diagram for IGame:



Public Member Functions

- virtual [~IGame](#) () noexcept=default
- virtual std::string [getSpritePath](#) ()=0
- virtual void [display](#) (IGraphic &)=0
- virtual int [init](#) ()=0
- virtual void [reset](#) ()=0
- virtual int [updateGame](#) (eventKey evtKey)=0
- virtual elemSize [getDisplaySize](#) ()=0

4.7.1 Constructor & Destructor Documentation

4.7.1.1 ~IGame()

```
virtual IGame::~~IGame ( ) [virtual], [default], [noexcept]
```

4.7.2 Member Function Documentation

4.7.2.1 display()

```
virtual void IGame::display (
    IGraphic & ) [pure virtual]
```

Implemented in [Snake](#), and [Nibbler](#).

4.7.2.2 `getDisplaySize()`

```
virtual elemSize IGame::getDisplaySize ( ) [pure virtual]
```

Implemented in [Snake](#), and [Nibbler](#).

4.7.2.3 `getSpritePath()`

```
virtual std::string IGame::getSpritePath ( ) [pure virtual]
```

Implemented in [Snake](#), and [Nibbler](#).

4.7.2.4 `init()`

```
virtual int IGame::init ( ) [pure virtual]
```

Implemented in [Snake](#), and [Nibbler](#).

4.7.2.5 `reset()`

```
virtual void IGame::reset ( ) [pure virtual]
```

Implemented in [Snake](#), and [Nibbler](#).

4.7.2.6 `updateGame()`

```
virtual int IGame::updateGame (
    eventKey evtKey ) [pure virtual]
```

Implemented in [Snake](#), and [Nibbler](#).

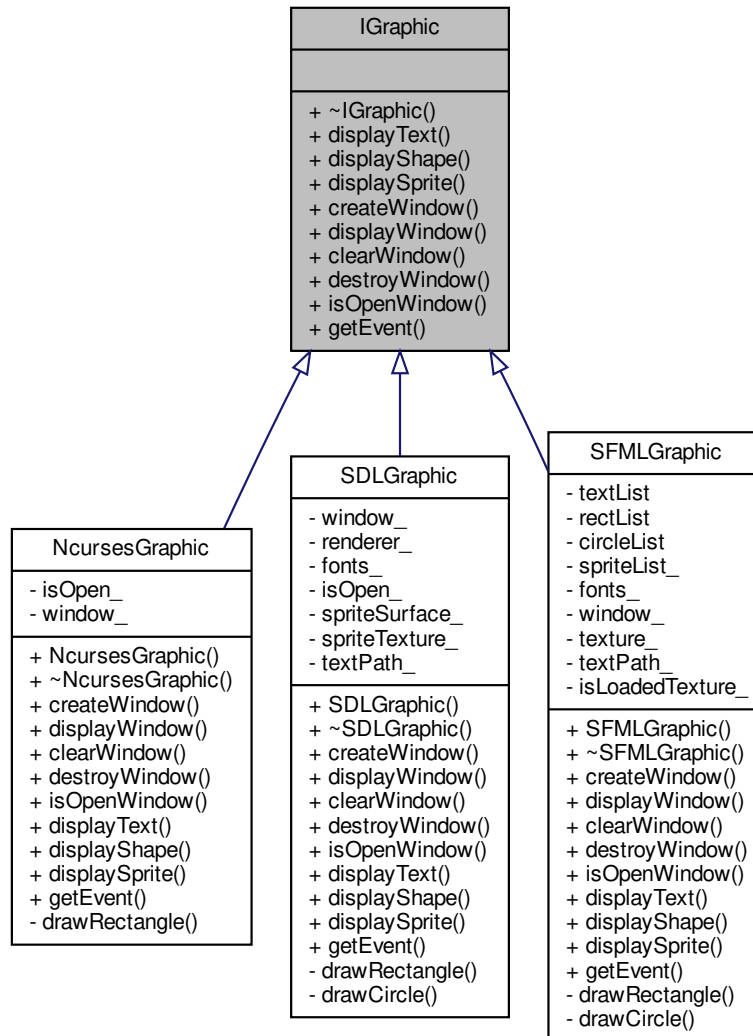
The documentation for this class was generated from the following file:

- [include/games/IGame.hpp](#)

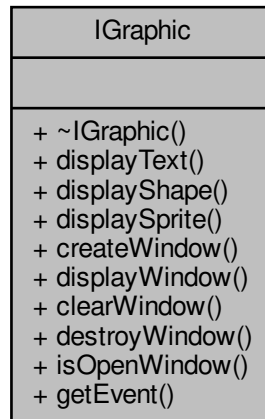
4.8 IGraphic Class Reference

```
#include <IGraphic.hpp>
```

Inheritance diagram for IGraphic:



Collaboration diagram for IGraphic:



Public Member Functions

- virtual [~IGraphic](#) () noexcept=default
- virtual void [displayText](#) (const [text](#) &text)=0
- virtual void [displayShape](#) (const [shape](#) &shape)=0
- virtual void [displaySprite](#) (const [sprite](#) &sprite)=0
- virtual void [createWindow](#) (std::string title, int width, int height)=0
- virtual void [displayWindow](#) ()=0
- virtual void [clearWindow](#) ()=0
- virtual void [destroyWindow](#) ()=0
- virtual bool [isOpenWindow](#) ()=0
- virtual [eventKey](#) [getEvent](#) ()=0

4.8.1 Constructor & Destructor Documentation

4.8.1.1 ~IGraphic()

```
virtual IGraphic::~~IGraphic ( ) [virtual], [default], [noexcept]
```

4.8.2 Member Function Documentation

4.8.2.1 clearWindow()

```
virtual void IGraphic::clearWindow ( ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

4.8.2.2 createWindow()

```
virtual void IGraphic::createWindow (
    std::string title,
    int width,
    int height ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

4.8.2.3 destroyWindow()

```
virtual void IGraphic::destroyWindow ( ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

4.8.2.4 displayShape()

```
virtual void IGraphic::displayShape (
    const shape & shape ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

4.8.2.5 displaySprite()

```
virtual void IGraphic::displaySprite (
    const sprite & sprite ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

4.8.2.6 displayText()

```
virtual void IGraphic::displayText (
    const text & text ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

4.8.2.7 displayWindow()

```
virtual void IGraphic::displayWindow ( ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

4.8.2.8 getEvent()

```
virtual eventKey IGraphic::getEvent ( ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

4.8.2.9 isOpenWindow()

```
virtual bool IGraphic::isOpenWindow ( ) [pure virtual]
```

Implemented in [SFMLGraphic](#), [SDLGraphic](#), and [NcursesGraphic](#).

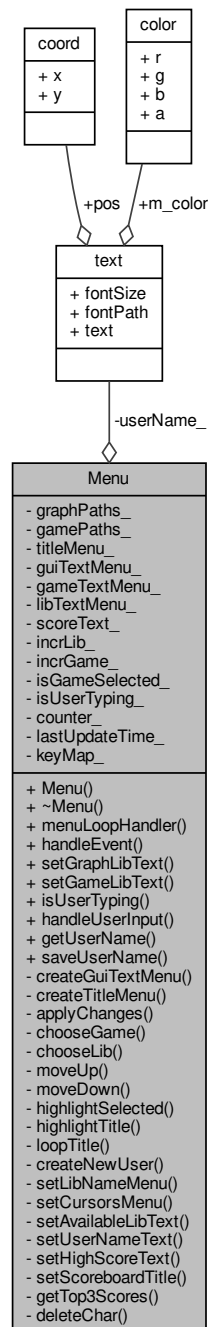
The documentation for this class was generated from the following file:

- [include/graphics/IGraphic.hpp](#)

4.9 Menu Class Reference

```
#include <Menu.hpp>
```

Collaboration diagram for Menu:



Public Member Functions

- [Menu](#) ()
- [~Menu](#) ()=default
- void [menuLoopHandler](#) (IGraphic &graphLib, [Core](#) &core)
- void [handleEvent](#) (eventKey evt, [Core](#) &core)
- void [setGraphLibText](#) ()

- void [setGameLibText](#) ()
- bool [isUserTyping](#) () const
- void [handleUserInput](#) ([eventKey](#) evt, [Core](#) &core)
- std::string [getUserName](#) () const
- void [saveUserName](#) ()

Private Member Functions

- void [createGuiTextMenu](#) ()
- void [createTitleMenu](#) ()
- void [applyChanges](#) ([Core](#) &core)
- void [chooseGame](#) ()
- void [chooseLib](#) ()
- void [moveUp](#) ([Core](#) &core)
- void [moveDown](#) ([Core](#) &core)
- void [highlightSelected](#) ([Core](#) &core)
- void [highlightTitle](#) (const int count)
- void [loopTitle](#) ()
- void [createNewUser](#) ()
- void [setLibNameMenu](#) ()
- void [setCursorsMenu](#) ()
- void [setAvailableLibText](#) ()
- void [setUserNameText](#) ()
- void [setHighScoreText](#) ()
- void [setScoreboardTitle](#) ()
- std::vector< std::string > [getTop3Scores](#) ()
- void [deleteChar](#) ()

Private Attributes

- std::vector< std::string > [graphPaths_](#)
- std::vector< std::string > [gamePaths_](#)
- std::vector< [text](#) > [titleMenu_](#)
- std::vector< [text](#) > [guiTextMenu_](#)
- std::vector< [text](#) > [gameTextMenu_](#)
- std::vector< [text](#) > [libTextMenu_](#)
- std::vector< [text](#) > [scoreText_](#)
- [text](#) [userName_](#)
- int [incrLib_](#) {0}
- int [incrGame_](#) {0}
- bool [isGameSelected_](#) {false}
- bool [isUserTyping_](#) {true}
- int [counter_](#) {0}
- std::chrono::steady_clock::time_point [lastUpdateTime_](#)
- std::unordered_map< [eventKey](#), char > [keyMap_](#)

4.9.1 Constructor & Destructor Documentation

4.9.1.1 Menu()

```
Menu::Menu ( )
```

4.9.1.2 ~Menu()

```
Menu::~Menu ( ) [default]
```

4.9.2 Member Function Documentation

4.9.2.1 applyChanges()

```
void Menu::applyChanges (
    Core & core ) [private]
```

4.9.2.2 chooseGame()

```
void Menu::chooseGame ( ) [private]
```

4.9.2.3 chooseLib()

```
void Menu::chooseLib ( ) [private]
```

4.9.2.4 createGuiTextMenu()

```
void Menu::createGuiTextMenu ( ) [private]
```

4.9.2.5 createNewUser()

```
void Menu::createNewUser ( ) [private]
```

4.9.2.6 createTitleMenu()

```
void Menu::createTitleMenu ( ) [private]
```

4.9.2.7 deleteChar()

```
void Menu::deleteChar ( ) [private]
```

4.9.2.8 getTop3Scores()

```
std::vector< std::string > Menu::getTop3Scores ( ) [private]
```

4.9.2.9 getUsername()

```
std::string Menu::getUserName ( ) const
```

4.9.2.10 handleEvent()

```
void Menu::handleEvent (
    eventKey evt,
    Core & core )
```

4.9.2.11 handleUserInput()

```
void Menu::handleUserInput (
    eventKey evt,
    Core & core )
```

4.9.2.12 highlightSelected()

```
void Menu::highlightSelected (
    Core & core ) [private]
```

4.9.2.13 highlightTitle()

```
void Menu::highlightTitle (
    const int count ) [private]
```

4.9.2.14 isUserTyping()

```
bool Menu::isUserTyping ( ) const
```

4.9.2.15 loopTitle()

```
void Menu::loopTitle ( ) [private]
```

4.9.2.16 menuLoopHandler()

```
void Menu::menuLoopHandler (
    IGraphic & graphLib,
    Core & core )
```

4.9.2.17 moveDown()

```
void Menu::moveDown (
    Core & core ) [private]
```

4.9.2.18 moveUp()

```
void Menu::moveUp (
    Core & core ) [private]
```

4.9.2.19 saveUserName()

```
void Menu::saveUserName ( )
```

4.9.2.20 setAvailableLibText()

```
void Menu::setAvailableLibText ( ) [private]
```

4.9.2.21 setCursorsMenu()

```
void Menu::setCursorsMenu ( ) [private]
```

4.9.2.22 setGameLibText()

```
void Menu::setGameLibText ( )
```

4.9.2.23 setGraphLibText()

```
void Menu::setGraphLibText ( )
```

4.9.2.24 setHighScoreText()

```
void Menu::setHighScoreText ( ) [private]
```

4.9.2.25 setLibNameMenu()

```
void Menu::setLibNameMenu ( ) [private]
```

4.9.2.26 setScoreboardTitle()

```
void Menu::setScoreboardTitle ( ) [private]
```

4.9.2.27 setUsernameText()

```
void Menu::setUsernameText ( ) [private]
```


4.9.3 Field Documentation

4.9.3.1 counter_

```
int Menu::counter_ {0} [private]
```

4.9.3.2 gamePaths_

```
std::vector<std::string> Menu::gamePaths_ [private]
```

4.9.3.3 gameTextMenu_

```
std::vector<text> Menu::gameTextMenu_ [private]
```

4.9.3.4 graphPaths_

```
std::vector<std::string> Menu::graphPaths_ [private]
```

4.9.3.5 guiTextMenu_

```
std::vector<text> Menu::guiTextMenu_ [private]
```

4.9.3.6 incrGame_

```
int Menu::incrGame_ {0} [private]
```

4.9.3.7 incrLib_

```
int Menu::incrLib_ {0} [private]
```

4.9.3.8 isGameSelected_

```
bool Menu::isGameSelected_ {false} [private]
```

4.9.3.9 isUserTyping_

```
bool Menu::isUserTyping_ {true} [private]
```

4.9.3.10 keyMap_

```
std::unordered_map<eventKey, char> Menu::keyMap_ [private]
```

4.9.3.11 lastUpdateTime_

```
std::chrono::steady_clock::time_point Menu::lastUpdateTime_ [private]
```

4.9.3.12 libTextMenu_

```
std::vector<text> Menu::libTextMenu_ [private]
```

4.9.3.13 scoreText_

```
std::vector<text> Menu::scoreText_ [private]
```

4.9.3.14 titleMenu_

```
std::vector<text> Menu::titleMenu_ [private]
```

4.9.3.15 userName_

```
text Menu::userName_ [private]
```

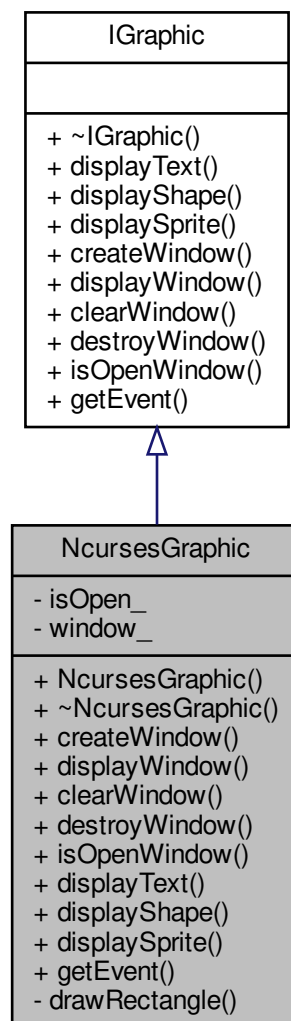
The documentation for this class was generated from the following files:

- include/core/Menu.hpp
- src/core/ActionsMenu.cpp
- src/core/Menu.cpp
- src/core/SetMenuText.cpp
- src/core/UserNameMenu.cpp

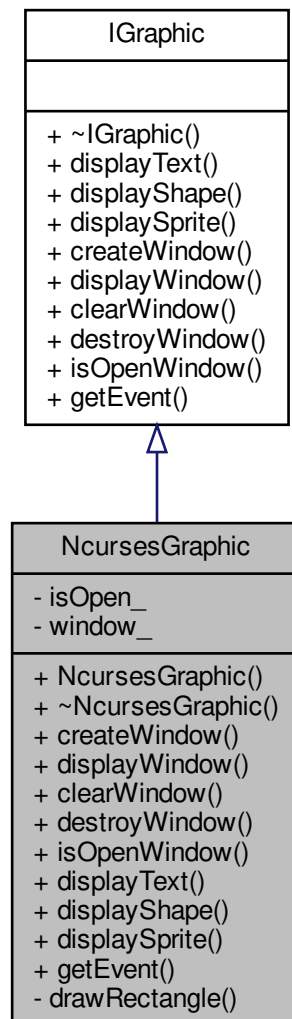
4.10 NcursesGraphic Class Reference

```
#include <NcursesGraphic.hpp>
```

Inheritance diagram for NcursesGraphic:



Collaboration diagram for NcursesGraphic:



Public Member Functions

- `NcursesGraphic ()`=default
- `~NcursesGraphic ()` noexcept=default
- void `createWindow` (std::string title, int width, int height) override
- void `displayWindow` () override
- void `clearWindow` () override
- void `destroyWindow` () override
- bool `isOpenWindow` () override
- void `displayText` (const text &text) override
- void `displayShape` (const shape &shape) override
- void `displaySprite` (const sprite &sprite) override
- eventKey `getEvent` () override

Private Member Functions

- void [drawRectangle](#) (const [shape](#) &[shape](#))

Private Attributes

- bool [isOpen_](#) { false }
- WINDOW * [window_](#)

4.10.1 Constructor & Destructor Documentation

4.10.1.1 NcursesGraphic()

```
NcursesGraphic::NcursesGraphic ( ) [default]
```

4.10.1.2 ~NcursesGraphic()

```
NcursesGraphic::~~NcursesGraphic ( ) [default], [noexcept]
```

4.10.2 Member Function Documentation

4.10.2.1 clearWindow()

```
void NcursesGraphic::clearWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.2.2 createWindow()

```
void NcursesGraphic::createWindow (
    std::string title,
    int width,
    int height ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.2.3 destroyWindow()

```
void NcursesGraphic::destroyWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.2.4 displayShape()

```
void NcursesGraphic::displayShape (
    const shape & shape ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.2.5 displaySprite()

```
void NcursesGraphic::displaySprite (
    const sprite & sprite ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.2.6 displayText()

```
void NcursesGraphic::displayText (
    const text & text ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.2.7 displayWindow()

```
void NcursesGraphic::displayWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.2.8 drawRectangle()

```
void NcursesGraphic::drawRectangle (
    const shape & shape ) [private]
```

4.10.2.9 `getEvent()`

```
eventKey NcursesGraphic::getEvent ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.2.10 `isOpenWindow()`

```
bool NcursesGraphic::isOpenWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.10.3 Field Documentation

4.10.3.1 `isOpen_`

```
bool NcursesGraphic::isOpen_ { false } [private]
```

4.10.3.2 `window_`

```
WINDOW* NcursesGraphic::window_ [private]
```

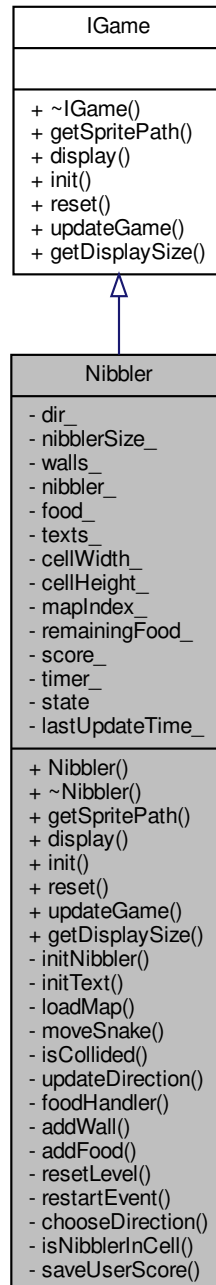
The documentation for this class was generated from the following files:

- [include/graphics/ncurses/NcursesGraphic.hpp](#)
- [src/graphics/ncurses/NcursesEvent.cpp](#)
- [src/graphics/ncurses/NcursesGraphic.cpp](#)

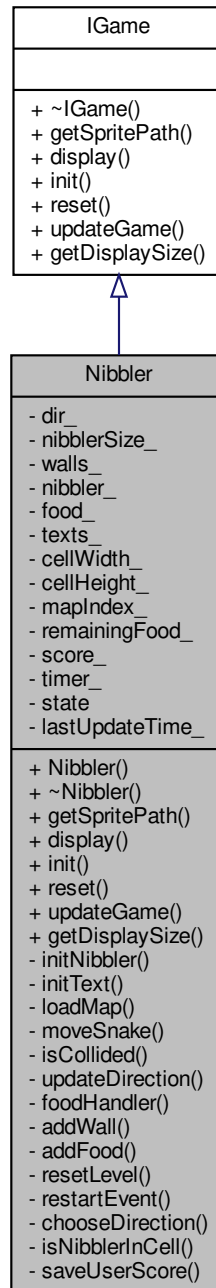
4.11 Nibbler Class Reference

```
#include <Nibbler.hpp>
```

Inheritance diagram for Nibbler:



Collaboration diagram for Nibbler:



Public Member Functions

- `Nibbler()`=default
- `~Nibbler()`=default
- `std::string getSpritePath()` override
- `void display (IGraphic &)` override
- `int init()` override

- void `reset` () override
- int `updateGame` (`eventKey` evtKey) override
- `elemSize` `getDisplaySize` () override

Private Member Functions

- void `initNibbler` (int x, int y)
- void `initText` ()
- void `loadMap` (int index)
- void `moveSnake` (std::vector< `shape` > &tmp)
- bool `isCollided` (const `shape` &s1, const `shape` &s2)
- void `updateDirection` (`eventKey` evtKey)
- void `foodHandler` ()
- void `addWall` (int x, int y)
- void `addFood` (int x, int y)
- void `resetLevel` ()
- void `restartEvent` (`eventKey` evtKey)
- void `chooseDirection` ()
- bool `isNibblerInCell` (int x, int y)
- void `saveUserScore` ()

Private Attributes

- `direction` `dir_` { `direction::RIGHT` }
- int `nibblerSize_` { 4 }
- std::vector< `shape` > `walls_`
- std::vector< `shape` > `nibbler_`
- std::vector< `shape` > `food_`
- std::vector< `text` > `texts_`
- int `cellWidth_` { 20 }
- int `cellHeight_` { 20 }
- int `mapIndex_` { 0 }
- int `remainingFood_` { 0 }
- int `score_` { 0 }
- int `timer_` { 0 }
- `playerState` `state` { `playerState::ALIVE` }
- std::chrono::steady_clock::time_point `lastUpdateTime_`

4.11.1 Constructor & Destructor Documentation

4.11.1.1 Nibbler()

`Nibbler::Nibbler ()` [default]

4.11.1.2 ~Nibbler()

```
Nibbler::~Nibbler ( ) [default]
```

4.11.2 Member Function Documentation

4.11.2.1 addFood()

```
void Nibbler::addFood (
    int x,
    int y ) [private]
```

4.11.2.2 addWall()

```
void Nibbler::addWall (
    int x,
    int y ) [private]
```

4.11.2.3 chooseDirection()

```
void Nibbler::chooseDirection ( ) [private]
```

4.11.2.4 display()

```
void Nibbler::display (
    IGraphic & graphLib ) [override], [virtual]
```

Implements [IGame](#).

4.11.2.5 foodHandler()

```
void Nibbler::foodHandler ( ) [private]
```

4.11.2.6 `getDisplaySize()`

```
elemSize Nibbler::getDisplaySize ( ) [override], [virtual]
```

Implements [IGame](#).

4.11.2.7 `getSpritePath()`

```
std::string Nibbler::getSpritePath ( ) [override], [virtual]
```

Implements [IGame](#).

4.11.2.8 `init()`

```
int Nibbler::init ( ) [override], [virtual]
```

Implements [IGame](#).

4.11.2.9 `initNibbler()`

```
void Nibbler::initNibbler (
    int x,
    int y ) [private]
```

4.11.2.10 `initText()`

```
void Nibbler::initText ( ) [private]
```

4.11.2.11 `isCollided()`

```
bool Nibbler::isCollided (
    const shape & s1,
    const shape & s2 ) [private]
```

4.11.2.12 isNibblerInCell()

```
bool Nibbler::isNibblerInCell (
    int x,
    int y ) [private]
```

4.11.2.13 loadMap()

```
void Nibbler::loadMap (
    int index ) [private]
```

4.11.2.14 moveSnake()

```
void Nibbler::moveSnake (
    std::vector< shape > & tmp ) [private]
```

4.11.2.15 reset()

```
void Nibbler::reset ( ) [override], [virtual]
```

Implements [IGame](#).

4.11.2.16 resetLevel()

```
void Nibbler::resetLevel ( ) [private]
```

4.11.2.17 restartEvent()

```
void Nibbler::restartEvent (
    eventKey evtKey ) [private]
```

4.11.2.18 saveUserScore()

```
void Nibbler::saveUserScore ( ) [private]
```

4.11.2.19 updateDirection()

```
void Nibbler::updateDirection (
    eventKey evtKey ) [private]
```

4.11.2.20 updateGame()

```
int Nibbler::updateGame (
    eventKey evtKey ) [override], [virtual]
```

Implements [IGame](#).

4.11.3 Field Documentation

4.11.3.1 cellHeight_

```
int Nibbler::cellHeight_ { 20 } [private]
```

4.11.3.2 cellWidth_

```
int Nibbler::cellWidth_ { 20 } [private]
```

4.11.3.3 dir_

```
direction Nibbler::dir_ { direction::RIGHT } [private]
```

4.11.3.4 food_

```
std::vector<shape> Nibbler::food_ [private]
```

4.11.3.5 lastUpdateTime_

```
std::chrono::steady_clock::time_point Nibbler::lastUpdateTime_ [private]
```

4.11.3.6 mapIndex_

```
int Nibbler::mapIndex_ { 0 } [private]
```

4.11.3.7 nibbler_

```
std::vector<shape> Nibbler::nibbler_ [private]
```

4.11.3.8 nibblerSize_

```
int Nibbler::nibblerSize_ { 4 } [private]
```

4.11.3.9 remainingFood_

```
int Nibbler::remainingFood_ { 0 } [private]
```

4.11.3.10 score_

```
int Nibbler::score_ { 0 } [private]
```

4.11.3.11 state

```
playerState Nibbler::state { playerState::ALIVE } [private]
```

4.11.3.12 texts_

```
std::vector<text> Nibbler::texts_ [private]
```

4.11.3.13 timer_

```
int Nibbler::timer_ { 0 } [private]
```

4.11.3.14 walls_

```
std::vector<shape> Nibbler::walls_ [private]
```

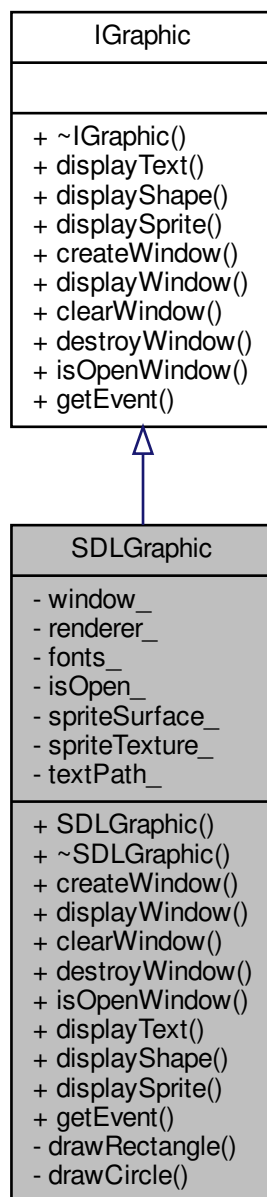
The documentation for this class was generated from the following files:

- [include/games/nibbler/Nibbler.hpp](#)
- [src/games/nibbler/Compute.cpp](#)
- [src/games/nibbler/Display.cpp](#)
- [src/games/nibbler/Nibbler.cpp](#)

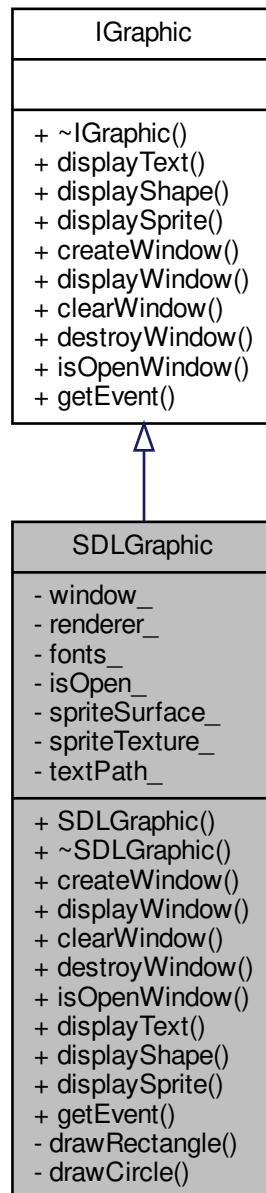
4.12 SDLGraphic Class Reference

```
#include <SDLGraphic.hpp>
```


Inheritance diagram for SDLGraphic:



Collaboration diagram for SDLGraphic:



Public Member Functions

- `SDLGraphic()`=default
- `~SDLGraphic()` noexcept=default
- void `createWindow` (std::string title, int width, int height) override
- void `displayWindow` () override
- void `clearWindow` () override
- void `destroyWindow` () override
- bool `isOpenWindow` () override

- void [displayText](#) (const [text](#) &[text](#)) override
- void [displayShape](#) (const [shape](#) &[shape](#)) override
- void [displaySprite](#) (const [sprite](#) &[sprite](#)) override
- [eventKey](#) [getEvent](#) () override

Private Member Functions

- void [drawRectangle](#) (const [shape](#) &[shape](#))
- void [drawCircle](#) (const [shape](#) &[shape](#))

Private Attributes

- SDL_Window * [window_](#)
- SDL_Renderer * [renderer_](#)
- std::unordered_map< int, TTF_Font * > [fonts_](#)
- bool [isOpen_](#) { false }
- SDL_Surface * [spriteSurface_](#) { NULL }
- SDL_Texture * [spriteTexture_](#) { NULL }
- std::string [textPath_](#)

4.12.1 Constructor & Destructor Documentation

4.12.1.1 SDLGraphic()

```
SDLGraphic::SDLGraphic ( ) [default]
```

4.12.1.2 ~SDLGraphic()

```
SDLGraphic::~SDLGraphic ( ) [default], [noexcept]
```

4.12.2 Member Function Documentation

4.12.2.1 clearWindow()

```
void SDLGraphic::clearWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.2.2 createWindow()

```
void SDLGraphic::createWindow (
    std::string title,
    int width,
    int height ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.2.3 destroyWindow()

```
void SDLGraphic::destroyWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.2.4 displayShape()

```
void SDLGraphic::displayShape (
    const shape & shape ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.2.5 displaySprite()

```
void SDLGraphic::displaySprite (
    const sprite & sprite ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.2.6 displayText()

```
void SDLGraphic::displayText (
    const text & text ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.2.7 displayWindow()

```
void SDLGraphic::displayWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.2.8 drawCircle()

```
void SDLGraphic::drawCircle (
    const shape & shape ) [private]
```

4.12.2.9 drawRectangle()

```
void SDLGraphic::drawRectangle (
    const shape & shape ) [private]
```

4.12.2.10 getEvent()

```
eventKey SDLGraphic::getEvent ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.2.11 isOpenWindow()

```
bool SDLGraphic::isOpenWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.12.3 Field Documentation

4.12.3.1 fonts_

```
std::unordered_map<int, TTF_Font*> SDLGraphic::fonts_ [private]
```

4.12.3.2 isOpen_

```
bool SDLGraphic::isOpen_ { false } [private]
```

4.12.3.3 renderer_

```
SDL_Renderer* SDLGraphic::renderer_ [private]
```

4.12.3.4 spriteSurface_

```
SDL_Surface* SDLGraphic::spriteSurface_ { NULL } [private]
```

4.12.3.5 spriteTexture_

```
SDL_Texture* SDLGraphic::spriteTexture_ { NULL } [private]
```

4.12.3.6 textPath_

```
std::string SDLGraphic::textPath_ [private]
```

4.12.3.7 window_

```
SDL_Window* SDLGraphic::window_ [private]
```

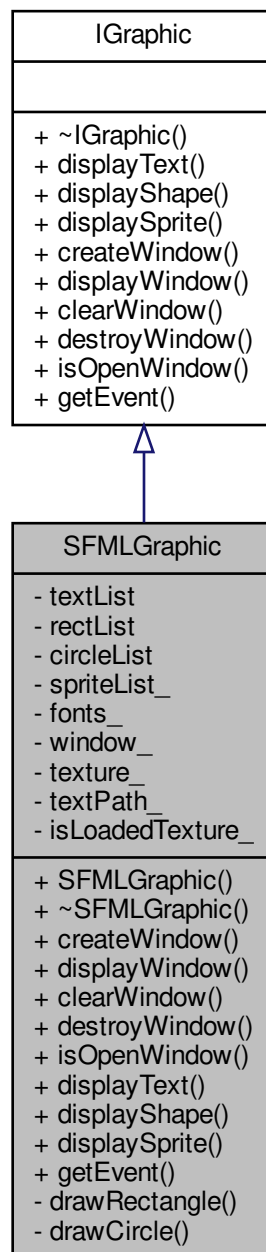
The documentation for this class was generated from the following files:

- include/graphics/sdl/[SDLGraphic.hpp](#)
- src/graphics/sdl/[SDLEvent.cpp](#)
- src/graphics/sdl/[SDLGraphic.cpp](#)

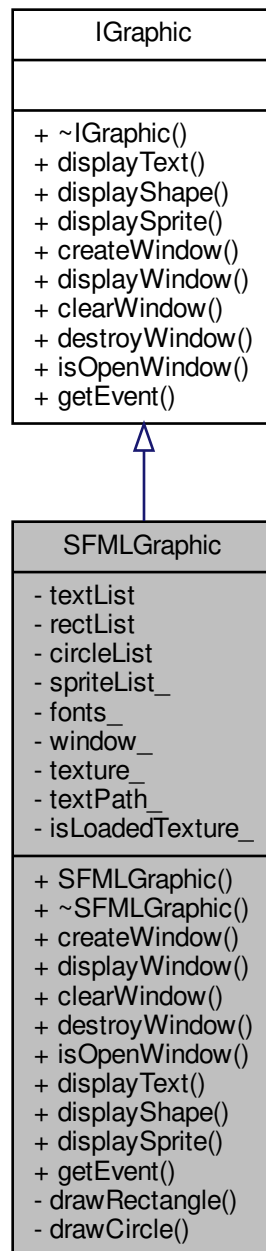
4.13 SFMLGraphic Class Reference

```
#include <SFMLGraphic.hpp>
```

Inheritance diagram for SFMLGraphic:



Collaboration diagram for SFMLGraphic:



Public Member Functions

- `SFMLGraphic()`=default
- `~SFMLGraphic()` noexcept=default
- void `createWindow` (std::string title, int width, int height) override
- void `displayWindow` () override
- void `clearWindow` () override

- void [destroyWindow](#) () override
- bool [isOpenWindow](#) () override
- void [displayText](#) (const [text](#) &[text](#)) override
- void [displayShape](#) (const [shape](#) &[shape](#)) override
- void [displaySprite](#) (const [sprite](#) &[sprite](#)) override
- [eventKey](#) [getEvent](#) () override

Private Member Functions

- void [drawRectangle](#) (const [shape](#) &[shape](#))
- void [drawCircle](#) (const [shape](#) &[shape](#))

Private Attributes

- std::unordered_map< const [text](#) *, sf::Text > [textList](#)
- std::unordered_map< const [shape](#) *, sf::RectangleShape > [rectList](#)
- std::unordered_map< const [shape](#) *, sf::CircleShape > [circleList](#)
- std::unordered_map< const [sprite](#) *, sf::Sprite > [spriteList](#)
- std::unordered_map< int, sf::Font > [fonts](#)
- sf::RenderWindow [window](#)
- sf::Texture [texture](#)
- std::string [textPath](#)
- bool [isLoadingTexture](#) = true

4.13.1 Constructor & Destructor Documentation

4.13.1.1 SFMLGraphic()

```
SFMLGraphic::SFMLGraphic ( ) [default]
```

4.13.1.2 ~SFMLGraphic()

```
SFMLGraphic::~SFMLGraphic ( ) [default], [noexcept]
```

4.13.2 Member Function Documentation

4.13.2.1 clearWindow()

```
void SFMLGraphic::clearWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.2.2 createWindow()

```
void SFMLGraphic::createWindow (
    std::string title,
    int width,
    int height ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.2.3 destroyWindow()

```
void SFMLGraphic::destroyWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.2.4 displayShape()

```
void SFMLGraphic::displayShape (
    const shape & shape ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.2.5 displaySprite()

```
void SFMLGraphic::displaySprite (
    const sprite & sprite ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.2.6 displayText()

```
void SFMLGraphic::displayText (
    const text & text ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.2.7 displayWindow()

```
void SFMLGraphic::displayWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.2.8 drawCircle()

```
void SFMLGraphic::drawCircle (
    const shape & shape ) [private]
```

4.13.2.9 drawRectangle()

```
void SFMLGraphic::drawRectangle (
    const shape & shape ) [private]
```

4.13.2.10 getEvent()

```
eventKey SFMLGraphic::getEvent ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.2.11 isOpenWindow()

```
bool SFMLGraphic::isOpenWindow ( ) [override], [virtual]
```

Implements [IGraphic](#).

4.13.3 Field Documentation

4.13.3.1 circleList

```
std::unordered_map<const shape*, sf::CircleShape> SFMLGraphic::circleList [private]
```

4.13.3.2 fonts_

```
std::unordered_map<int, sf::Font> SFMLGraphic::fonts_ [private]
```

4.13.3.3 isLoadedTexture_

```
bool SFMLGraphic::isLoadedTexture_ = true [private]
```

4.13.3.4 rectList

```
std::unordered_map<const shape*, sf::RectangleShape> SFMLGraphic::rectList [private]
```

4.13.3.5 spriteList_

```
std::unordered_map<const sprite*, sf::Sprite> SFMLGraphic::spriteList_ [private]
```

4.13.3.6 textList

```
std::unordered_map<const text*, sf::Text> SFMLGraphic::textList [private]
```

4.13.3.7 textPath_

```
std::string SFMLGraphic::textPath_ [private]
```

4.13.3.8 texture_

```
sf::Texture SFMLGraphic::texture_ [private]
```

4.13.3.9 window_

```
sf::RenderWindow SFMLGraphic::window_ [private]
```

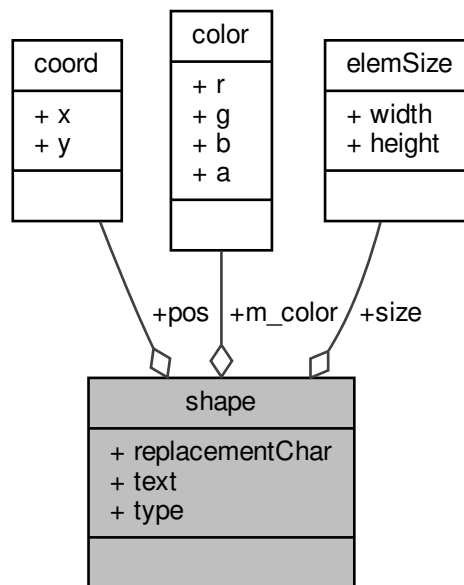
The documentation for this class was generated from the following files:

- [include/graphics/sfml/SFMLGraphic.hpp](#)
- [src/graphics/sfml/SFMLEvent.cpp](#)
- [src/graphics/sfml/SFMLGraphic.cpp](#)

4.14 shape Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for shape:



Data Fields

- [coord](#) pos
- [elemSize](#) size
- [color](#) m_color
- [char](#) replacementChar
- [std::string](#) text
- [shapeType](#) type

4.14.1 Field Documentation

4.14.1.1 m_color

`color` shape::m_color

4.14.1.2 pos

`coord` shape::pos

4.14.1.3 replacementChar

`char` shape::replacementChar

4.14.1.4 size

`elemSize` shape::size

4.14.1.5 text

`std::string` shape::text

4.14.1.6 type

`shapeType` shape::type

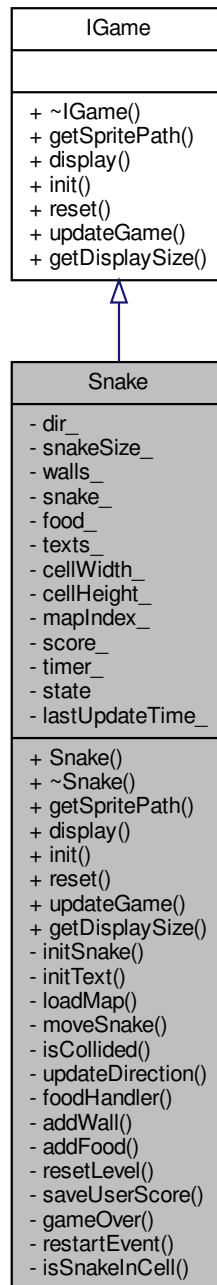
The documentation for this struct was generated from the following file:

- [include/graphics/IGraphic.hpp](#)

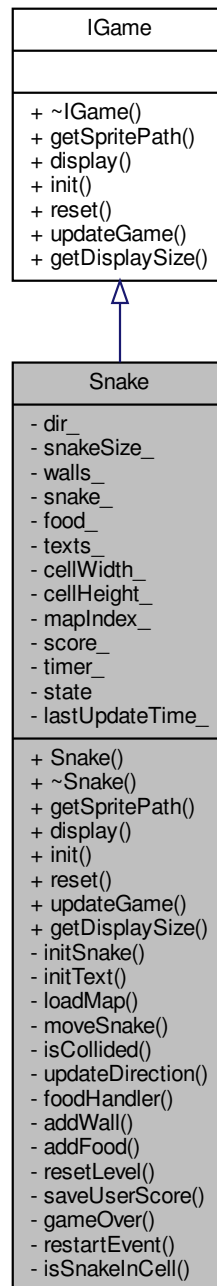
4.15 Snake Class Reference

```
#include <Snake.hpp>
```

Inheritance diagram for Snake:



Collaboration diagram for Snake:



Public Member Functions

- `Snake()`=default
- `~Snake()`=default
- `std::string getSpritePath()` override
- `void display (IGraphic &)` override
- `int init()` override

- void `reset` () override
- int `updateGame` (`eventKey` evtKey) override
- `elemSize` `getDisplaySize` () override

Private Member Functions

- void `initSnake` (int x, int y)
- void `initText` ()
- void `loadMap` (int index)
- void `moveSnake` (std::vector< `shape` > &tmp)
- bool `isCollided` (const `shape` &s1, const `shape` &s2)
- void `updateDirection` (`eventKey` evtKey)
- void `foodHandler` ()
- void `addWall` (int x, int y)
- void `addFood` (int x, int y)
- void `resetLevel` ()
- void `saveUserScore` ()
- void `gameOver` ()
- void `restartEvent` (`eventKey` evtKey)
- bool `isSnakeInCell` (int x, int y)

Private Attributes

- `direction` `dir_` { `direction::RIGHT` }
- int `snakeSize_` { 4 }
- std::vector< `shape` > `walls_`
- std::vector< `shape` > `snake_`
- std::vector< `shape` > `food_`
- std::vector< `text` > `texts_`
- int `cellWidth_` { 20 }
- int `cellHeight_` { 20 }
- int `mapIndex_` { 0 }
- int `score_` { 0 }
- int `timer_` { 0 }
- `playerState` `state` { `playerState::ALIVE` }
- std::chrono::steady_clock::time_point `lastUpdateTime_`

4.15.1 Constructor & Destructor Documentation

4.15.1.1 Snake()

```
Snake::Snake ( ) [default]
```

4.15.1.2 ~Snake()

```
Snake::~~Snake ( ) [default]
```

4.15.2 Member Function Documentation

4.15.2.1 addFood()

```
void Snake::addFood (
    int x,
    int y ) [private]
```

4.15.2.2 addWall()

```
void Snake::addWall (
    int x,
    int y ) [private]
```

4.15.2.3 display()

```
void Snake::display (
    IGraphic & graphLib ) [override], [virtual]
```

Implements [IGame](#).

4.15.2.4 foodHandler()

```
void Snake::foodHandler ( ) [private]
```

4.15.2.5 gameOver()

```
void Snake::gameOver ( ) [private]
```

4.15.2.6 `getDisplaySize()`

```
elemSize Snake::getDisplaySize ( ) [override], [virtual]
```

Implements [IGame](#).

4.15.2.7 `getSpritePath()`

```
std::string Snake::getSpritePath ( ) [override], [virtual]
```

Implements [IGame](#).

4.15.2.8 `init()`

```
int Snake::init ( ) [override], [virtual]
```

Implements [IGame](#).

4.15.2.9 `initSnake()`

```
void Snake::initSnake (
    int x,
    int y ) [private]
```

4.15.2.10 `initText()`

```
void Snake::initText ( ) [private]
```

4.15.2.11 `isCollided()`

```
bool Snake::isCollided (
    const shape & s1,
    const shape & s2 ) [private]
```

4.15.2.12 isSnakeInCell()

```
bool Snake::isSnakeInCell (
    int x,
    int y ) [private]
```

4.15.2.13 loadMap()

```
void Snake::loadMap (
    int index ) [private]
```

4.15.2.14 moveSnake()

```
void Snake::moveSnake (
    std::vector< shape > & tmp ) [private]
```

4.15.2.15 reset()

```
void Snake::reset ( ) [override], [virtual]
```

Implements [IGame](#).

4.15.2.16 resetLevel()

```
void Snake::resetLevel ( ) [private]
```

4.15.2.17 restartEvent()

```
void Snake::restartEvent (
    eventKey evtKey ) [private]
```

4.15.2.18 saveUserScore()

```
void Snake::saveUserScore ( ) [private]
```

4.15.2.19 updateDirection()

```
void Snake::updateDirection (
    eventKey evtKey ) [private]
```

4.15.2.20 updateGame()

```
int Snake::updateGame (
    eventKey evtKey ) [override], [virtual]
```

Implements [IGame](#).

4.15.3 Field Documentation

4.15.3.1 cellHeight_

```
int Snake::cellHeight_ { 20 } [private]
```

4.15.3.2 cellWidth_

```
int Snake::cellWidth_ { 20 } [private]
```

4.15.3.3 dir_

```
direction Snake::dir_ { direction::RIGHT } [private]
```

4.15.3.4 food_

```
std::vector<shape> Snake::food_ [private]
```

4.15.3.5 lastUpdateTime_

```
std::chrono::steady_clock::time_point Snake::lastUpdateTime_ [private]
```

4.15.3.6 mapIndex_

```
int Snake::mapIndex_ { 0 } [private]
```

4.15.3.7 score_

```
int Snake::score_ { 0 } [private]
```

4.15.3.8 snake_

```
std::vector<shape> Snake::snake_ [private]
```

4.15.3.9 snakeSize_

```
int Snake::snakeSize_ { 4 } [private]
```

4.15.3.10 state

```
playerState Snake::state { playerState::ALIVE } [private]
```

4.15.3.11 texts_

```
std::vector<text> Snake::texts_ [private]
```

4.15.3.12 timer_

```
int Snake::timer_ { 0 } [private]
```

4.15.3.13 walls_

```
std::vector<shape> Snake::walls_ [private]
```

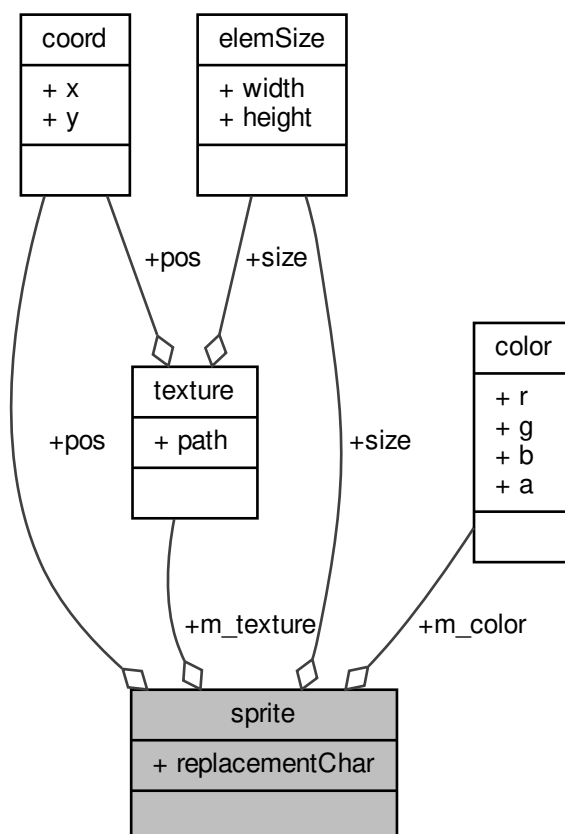
The documentation for this class was generated from the following files:

- include/games/snake/[Snake.hpp](#)
- src/games/snake/[Compute.cpp](#)
- src/games/snake/[Display.cpp](#)
- src/games/snake/[Snake.cpp](#)

4.16 sprite Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for sprite:



Data Fields

- [coord pos](#)
- [elemSize size](#)
- [texture m_texture](#)
- char [replacementChar](#)
- [color m_color](#)

4.16.1 Field Documentation

4.16.1.1 m_color

[color](#) sprite::m_color

4.16.1.2 m_texture

[texture](#) sprite::m_texture

4.16.1.3 pos

[coord](#) sprite::pos

4.16.1.4 replacementChar

char sprite::replacementChar

4.16.1.5 size

[elemSize](#) sprite::size

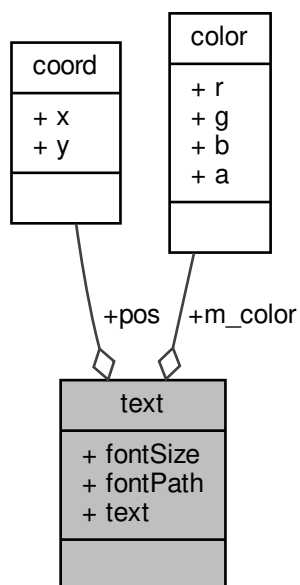
The documentation for this struct was generated from the following file:

- [include/graphics/IGraphic.hpp](#)

4.17 text Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for text:



Data Fields

- `coord pos`
- `int fontSize`
- `std::string fontPath`
- `color m_color`
- `std::string text`

4.17.1 Field Documentation

4.17.1.1 fontPath

```
std::string text::fontPath
```

4.17.1.2 fontSize

```
int text::fontSize
```

4.17.1.3 m_color

```
color text::m_color
```

4.17.1.4 pos

```
coord text::pos
```

4.17.1.5 text

```
std::string text::text
```

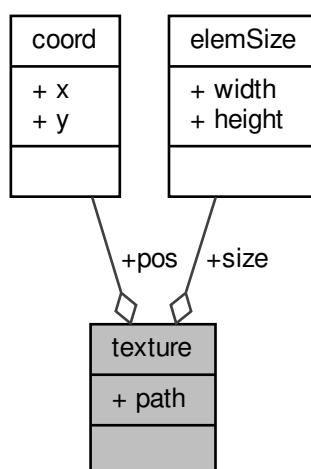
The documentation for this struct was generated from the following file:

- [include/graphics/IGraphic.hpp](#)

4.18 texture Struct Reference

```
#include <IGraphic.hpp>
```

Collaboration diagram for texture:



Data Fields

- [coord](#) [pos](#)
- [elemSize](#) [size](#)
- `std::string` [path](#)

4.18.1 Field Documentation

4.18.1.1 `path`

`std::string texture::path`

4.18.1.2 `pos`

[coord](#) `texture::pos`

4.18.1.3 `size`

[elemSize](#) `texture::size`

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

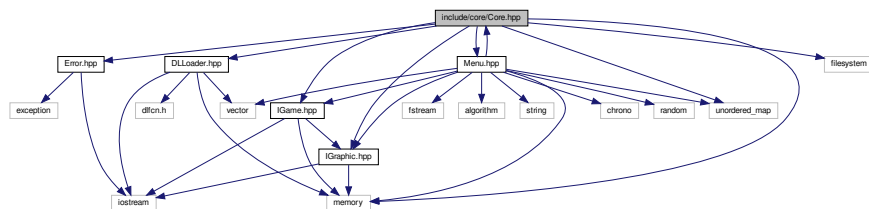
- `include/graphics/IGraphic.hpp`

Chapter 5

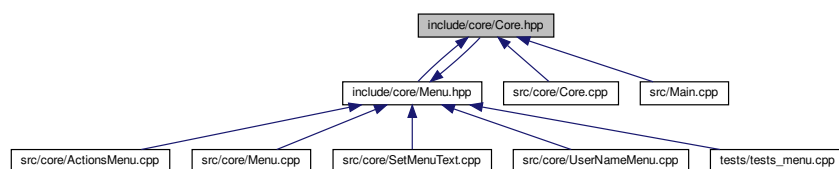
File Documentation

5.1 include/core/Core.hpp File Reference

```
#include "DLoader.hpp"
#include "Error.hpp"
#include "IGame.hpp"
#include "IGraphic.hpp"
#include "Menu.hpp"
#include <filesystem>
#include <memory>
#include <unordered_map>
Include dependency graph for Core.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Core](#)

Enumerations

- enum class `GState` {
 `PLAY` , `PAUSE` , `MENU` , `QUIT` ,
 `CONTINUE` }
- enum class `libType` { `GRAPHICAL` , `GAME` }

Variables

- static const `std::unordered_map< std::string, libType > validLibs`

5.1.1 Enumeration Type Documentation

5.1.1.1 GState

```
enum GState [strong]
```

Enumerator

PLAY	
PAUSE	
MENU	
QUIT	
CONTINUE	

5.1.1.2 libType

```
enum libType [strong]
```

Enumerator

GRAPHICAL	
GAME	

5.1.2 Variable Documentation

5.1.2.1 validLibs

```
const std::unordered_map<std::string, libType> validLibs [static]
```

Initial value:

```

{
    {"arcade_ncurses.so", libType::GRAPHICAL},
    {"arcade_sdl2.so", libType::GRAPHICAL},
    {"arcade_ndk++.so", libType::GRAPHICAL},
    {"arcade_aalib.so", libType::GRAPHICAL},
    {"arcade_libcaca.so", libType::GRAPHICAL},
    {"arcade_allegro5.so", libType::GRAPHICAL},
    {"arcade_xlib.so", libType::GRAPHICAL},
    {"arcade_gtk+.so", libType::GRAPHICAL},
    {"arcade_sfml.so", libType::GRAPHICAL},
    {"arcade_irrlicht.so", libType::GRAPHICAL},
    {"arcade_opengl.so", libType::GRAPHICAL},
    {"arcade_vulkan.so", libType::GRAPHICAL},
    {"arcade_qt5.so", libType::GRAPHICAL},
    {"arcade_snake.so", libType::GAME},
    {"arcade_nibbler.so", libType::GAME},
    {"test_game.so", libType::GAME},
    {"arcade_pacman.so", libType::GAME},
    {"arcade_gix.so", libType::GAME},
    {"arcade_centipede.so", libType::GAME},
    {"arcade_solarfox.so", libType::GAME}
}

```

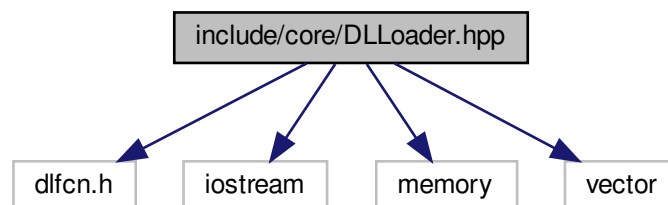
5.2 include/core/DLLoader.hpp File Reference

```

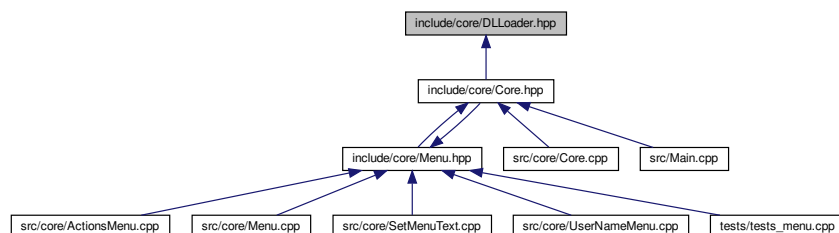
#include <dlfcn.h>
#include <iostream>
#include <memory>
#include <vector>

```

Include dependency graph for DLLoader.hpp:



This graph shows which files directly or indirectly include this file:



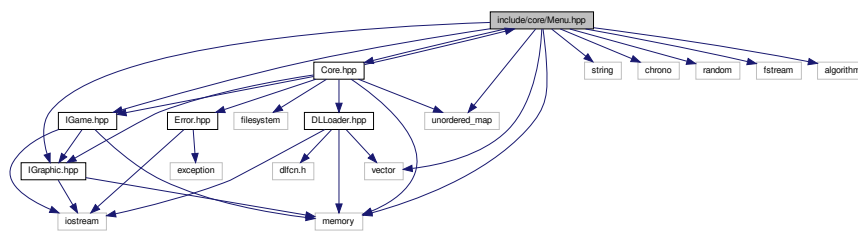
Data Structures

- class [DLLoader< T >](#)

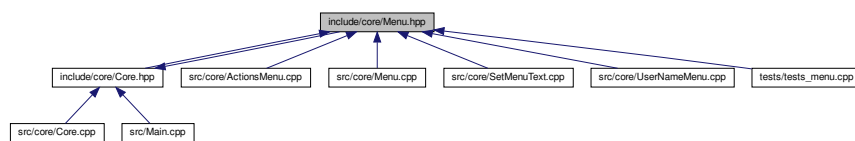
5.3 include/core/Menu.hpp File Reference

```
#include "IGraphic.hpp"
#include "IGame.hpp"
#include "Core.hpp"
#include <string>
#include <vector>
#include <memory>
#include <chrono>
#include <random>
#include <fstream>
#include <algorithm>
#include <unordered_map>
```

Include dependency graph for Menu.hpp:



This graph shows which files directly or indirectly include this file:

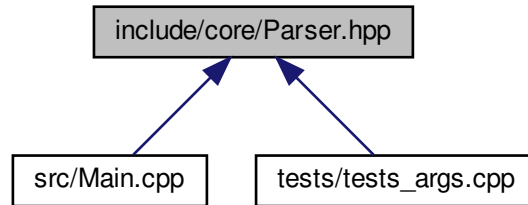


Data Structures

- class [Menu](#)

5.4 include/core/Parser.hpp File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int [checkArgs](#) (int ac, char **av)
- int [checkEnv](#) (char **env)
- void [displayUsage](#) ()
- int [checkLibrary](#) (const char *library)

5.4.1 Function Documentation

5.4.1.1 checkArgs()

```
int checkArgs (  
    int ac,  
    char ** av )
```

5.4.1.2 checkEnv()

```
int checkEnv (  
    char ** env )
```

5.4.1.3 checkLibrary()

```
int checkLibrary (  
    const char * library )
```

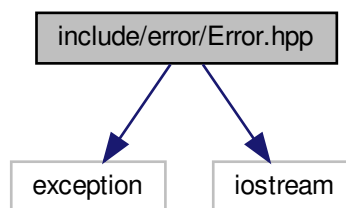
5.4.1.4 displayUsage()

```
void displayUsage ( )
```

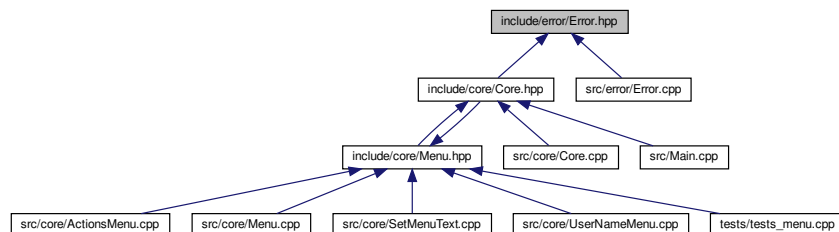
5.5 include/error/Error.hpp File Reference

```
#include <exception>
#include <iostream>
```

Include dependency graph for Error.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

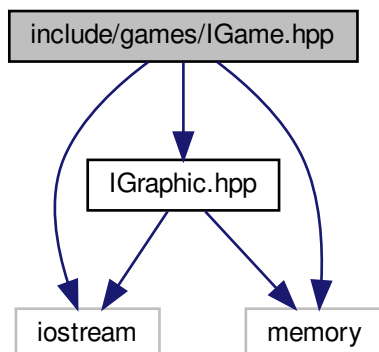
- class [ArcadeError](#)

5.6 include/games/IGame.hpp File Reference

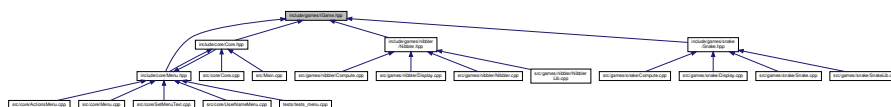
```
#include <iostream>
#include <memory>
```

```
#include "IGraphic.hpp"
```

Include dependency graph for IGame.hpp:



This graph shows which files directly or indirectly include this file:



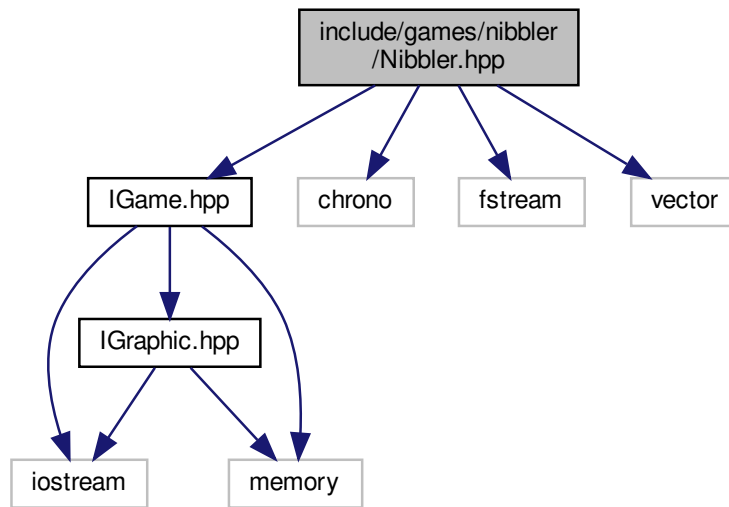
Data Structures

- class `IGame`

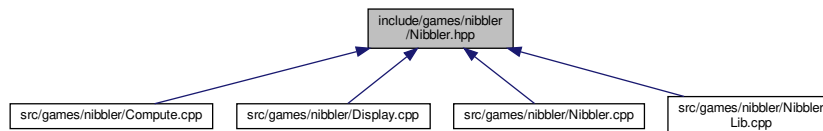
5.7 include/games/nibbler/Nibbler.hpp File Reference

```
#include "IGame.hpp"
#include <chrono>
#include <fstream>
#include <vector>
```

Include dependency graph for Nibbler.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class `Nibbler`

Enumerations

- enum class `direction` {
`UP`, `DOWN`, `LEFT`, `RIGHT`,
`UP`, `DOWN`, `LEFT`, `RIGHT` }
- enum class `playerState` {
`ALIVE`, `DEAD`, `STOP`, `WON`,
`ALIVE`, `DEAD`, `STOP` }

Variables

- `const std::vector< std::vector< std::string > > allMaps`

5.7.1 Enumeration Type Documentation

5.7.1.1 direction

```
enum direction [strong]
```

Enumerator

UP	
DOWN	
LEFT	
RIGHT	
UP	
DOWN	
LEFT	
RIGHT	

5.7.1.2 playerState

```
enum playerState [strong]
```

Enumerator

ALIVE	
DEAD	
STOP	
WON	
ALIVE	
DEAD	
STOP	

5.7.2 Variable Documentation

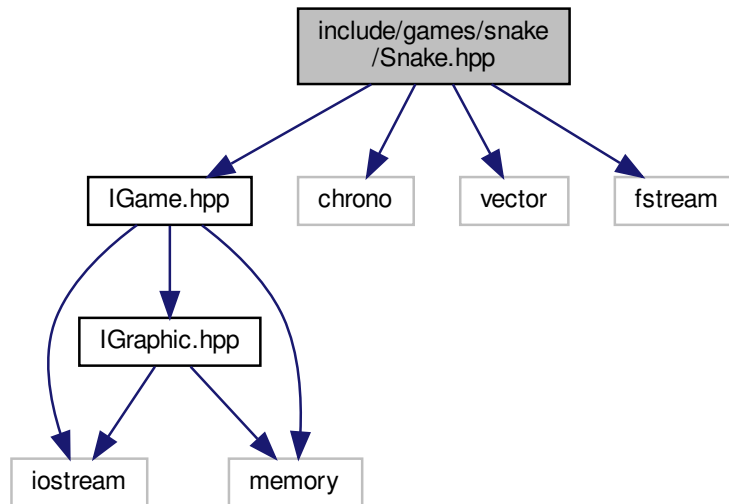
5.7.2.1 allMaps

```
const std::vector<std::vector<std::string> > allMaps
```

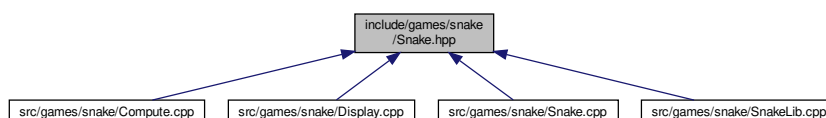
5.8 include/games/snake/Snake.hpp File Reference

```
#include "IGame.hpp"
#include <chrono>
#include <vector>
#include <fstream>
```

Include dependency graph for Snake.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Snake](#)

Enumerations

- enum class [direction](#) {
UP, DOWN, LEFT, RIGHT,
UP, DOWN, LEFT, RIGHT }
- enum class [playerState](#) {
ALIVE, DEAD, STOP, WON,
ALIVE, DEAD, STOP }

Variables

- `const std::vector< std::vector< std::string > > allMaps`

5.8.1 Enumeration Type Documentation

5.8.1.1 direction

```
enum direction [strong]
```

Enumerator

UP	
DOWN	
LEFT	
RIGHT	
UP	
DOWN	
LEFT	
RIGHT	

5.8.1.2 playerState

```
enum playerState [strong]
```

Enumerator

ALIVE	
DEAD	
STOP	
WON	
ALIVE	
DEAD	
STOP	

5.8.2 Variable Documentation

5.8.2.1 allMaps

```
const std::vector<std::vector<std::string> > allMaps
```

Initial value:

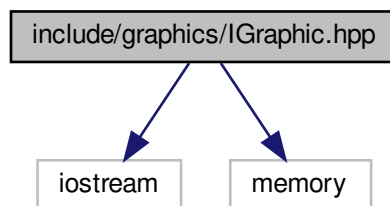
```
= {
    {
        "11111111111111111111",
        "10000000000000000001",
        "10000000000000000001",
        "10020000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000003000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "10000000000000000001",
        "11111111111111111111",
    },
}
```

5.9 include/graphics/IGraphic.hpp File Reference

```
#include <iostream>
```

```
#include <memory>
```

Include dependency graph for IGraphic.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [coord](#)
- struct [elemSize](#)
- struct [texture](#)
- struct [color](#)
- struct [sprite](#)
- struct [text](#)
- struct [shape](#)
- class [IGraphic](#)

Enumerations

- enum class `eventKey` {
`NULL_EVENT`, `A`, `Z`, `E`,
`R`, `T`, `Y`, `U`,
`I`, `O`, `P`, `Q`,
`S`, `D`, `F`, `G`,
`H`, `J`, `K`, `L`,
`M`, `W`, `X`, `C`,
`V`, `B`, `N`, `SPACE`,
`ESCAPE`, `ENTER`, `LARROW`, `RRROW`,
`BARROW`, `UARROW`, `DELETE`, `SUPR`,
`TAB`, `ONE`, `TWO`, `THREE`,
`FOUR`, `FIVE`, `SIX`, `SEVEN`,
`EIGHT`, `NINE`, `ZERO`, `CTRL`,
`ALT`, `SHIFT`, `QUIT` }
- enum class `shapeType` { `RECTANGLE`, `CIRCLE`, `TRIANGLE` }

5.9.1 Enumeration Type Documentation

5.9.1.1 `eventKey`

enum `eventKey` [strong]

Enumerator

NULL_EVENT	
A	
Z	
E	
R	
T	
Y	
U	
I	
O	
P	
Q	
S	
D	
F	
G	
H	
J	
K	
L	
M	
W	
X	
C	

Enumerator

V	
B	
N	
SPACE	
ESCAPE	
ENTER	
LARROW	
RARROW	
BARROW	
UARROW	
DELETE	
SUPR	
TAB	
ONE	
TWO	
THREE	
FOUR	
FIVE	
SIX	
SEVEN	
EIGHT	
NINE	
ZERO	
CTRL	
ALT	
SHIFT	
QUIT	

5.9.1.2 shapeType

```
enum shapeType [strong]
```

Enumerator

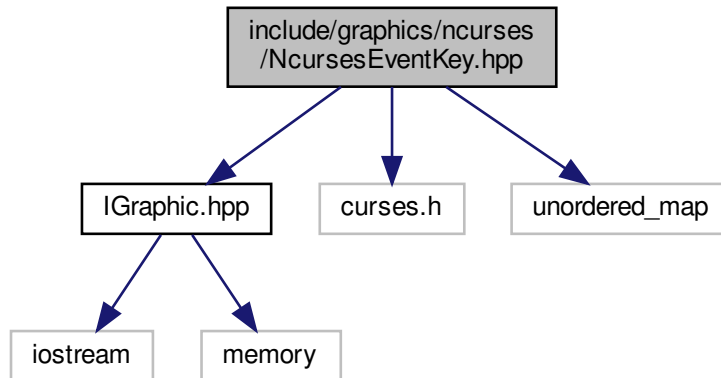
RECTANGLE	
CIRCLE	
TRIANGLE	

5.10 include/graphics/ncurses/NcursesEventKey.hpp File Reference

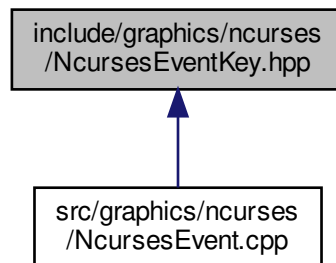
```
#include "IGraphic.hpp"  
#include <curses.h>
```

```
#include <unordered_map>
```

Include dependency graph for NcursesEventKey.hpp:



This graph shows which files directly or indirectly include this file:



Variables

- `std::unordered_map< char, eventKey > keyEvent`

5.10.1 Variable Documentation

5.10.1.1 keyEvent

```
std::unordered_map<char, eventKey> keyEvent
```

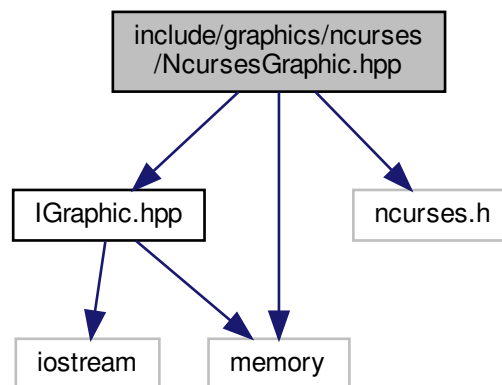
Initial value:

```
= { { 'a', eventKey::A },
    { 'b', eventKey::B }, { 'c', eventKey::C }, { 'd', eventKey::D },
    { 'e', eventKey::E }, { 'f', eventKey::F }, { 'g', eventKey::G },
    { 'h', eventKey::H }, { 'i', eventKey::I }, { 'j', eventKey::J },
    { 'k', eventKey::K }, { 'l', eventKey::L }, { 'm', eventKey::M },
    { 'n', eventKey::N }, { 'o', eventKey::O }, { 'p', eventKey::P },
    { 'q', eventKey::Q }, { 'r', eventKey::R }, { 's', eventKey::S },
    { 't', eventKey::T }, { 'u', eventKey::U }, { 'v', eventKey::V },
    { 'w', eventKey::W }, { 'x', eventKey::X }, { 'y', eventKey::Y },
    { 'z', eventKey::Z }, { KEY_F0, eventKey::SPACE }, { '\n', eventKey::ENTER },
    { KEY_LEFT, eventKey::LARROW }, { KEY_RIGHT, eventKey::RARROW }, { KEY_DOWN, eventKey::BARROW },
    { KEY_UP, eventKey::UARROW }, { KEY_BACKSPACE, eventKey::DELETE }, { KEY_STAB, eventKey::TAB } }
```

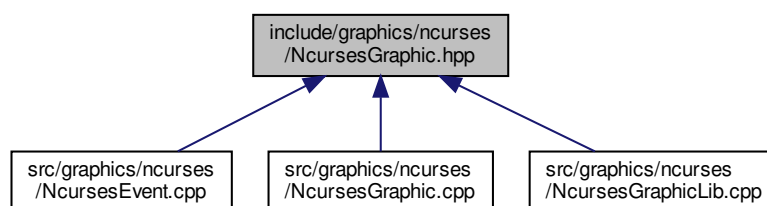
5.11 include/graphics/ncurses/NcursesGraphic.hpp File Reference

```
#include "IGraphic.hpp"
#include <memory>
#include <ncurses.h>
```

Include dependency graph for NcursesGraphic.hpp:



This graph shows which files directly or indirectly include this file:



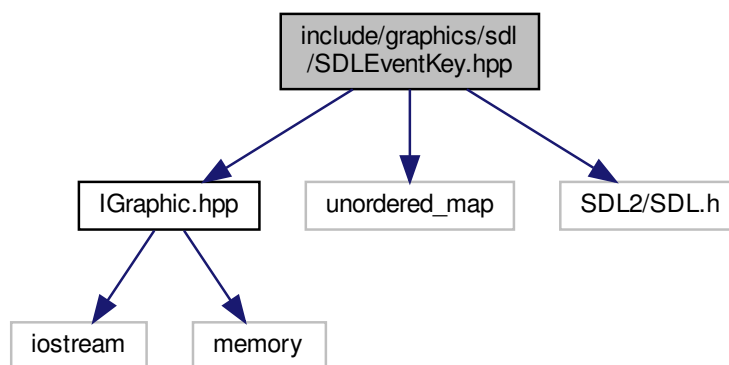
Data Structures

- class [NcursesGraphic](#)

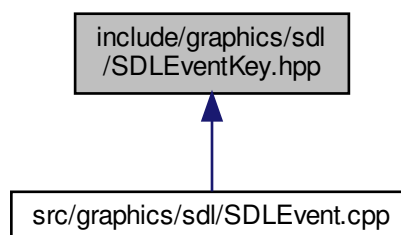
5.12 include/graphics/sdl/SDLEventKey.hpp File Reference

```
#include "IGraphic.hpp"  
#include <unordered_map>  
#include <SDL2/SDL.h>
```

Include dependency graph for SDLEventKey.hpp:



This graph shows which files directly or indirectly include this file:



Variables

- `std::unordered_map< char, eventKey > keyEvent`

5.12.1 Variable Documentation

5.12.1.1 keyEvent

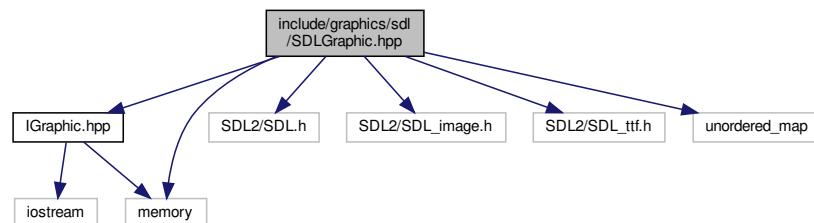
```
std::unordered_map<char, eventKey> keyEvent
```

Initial value:

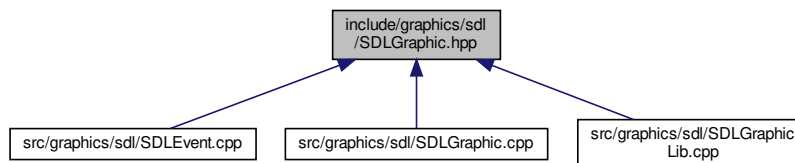
```
= { { 'a', eventKey::A },
    { 'b', eventKey::B }, { 'c', eventKey::C }, { 'd', eventKey::D },
    { 'e', eventKey::E }, { 'f', eventKey::F }, { 'g', eventKey::G },
    { 'h', eventKey::H }, { 'i', eventKey::I }, { 'j', eventKey::J },
    { 'k', eventKey::K }, { 'l', eventKey::L }, { 'm', eventKey::M },
    { 'n', eventKey::N }, { 'o', eventKey::O }, { 'p', eventKey::P },
    { 'q', eventKey::Q }, { 'r', eventKey::R }, { 's', eventKey::S },
    { 't', eventKey::T }, { 'u', eventKey::U }, { 'v', eventKey::V },
    { 'w', eventKey::W }, { 'x', eventKey::X }, { 'y', eventKey::Y },
    { 'z', eventKey::Z }, { SDLK_SPACE, eventKey::SPACE },
    { SDLK_ESCAPE, eventKey::ESCAPE }, { SDLK_RETURN, eventKey::ENTER },
    { SDLK_LEFT, eventKey::LARROW }, { SDLK_RIGHT, eventKey::RARROW },
    { SDLK_DOWN, eventKey::BARROW }, { SDLK_UP, eventKey::UARROW },
    { SDLK_BACKSPACE, eventKey::DELETE }, { SDLK_DELETE, eventKey::SUPR },
    { SDLK_KP_TAB, eventKey::TAB }, { SDLK_1, eventKey::ONE },
    { SDLK_2, eventKey::TWO }, { SDLK_3, eventKey::THREE },
    { SDLK_4, eventKey::FOUR }, { SDLK_5, eventKey::FIVE },
    { SDLK_6, eventKey::SIX }, { SDLK_7, eventKey::SEVEN },
    { SDLK_8, eventKey::EIGHT }, { SDLK_9, eventKey::NINE },
    { SDLK_0, eventKey::ZERO }, { SDLK_LCTRL, eventKey::CTRL },
    { SDLK_LALT, eventKey::ALT }, { SDLK_LSHIFT, eventKey::SHIFT } }
```

5.13 include/graphics/sdl/SDLGraphic.hpp File Reference

```
#include "IGraphic.hpp"
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_ttf.h>
#include <memory>
#include <unordered_map>
Include dependency graph for SDLGraphic.hpp:
```



This graph shows which files directly or indirectly include this file:



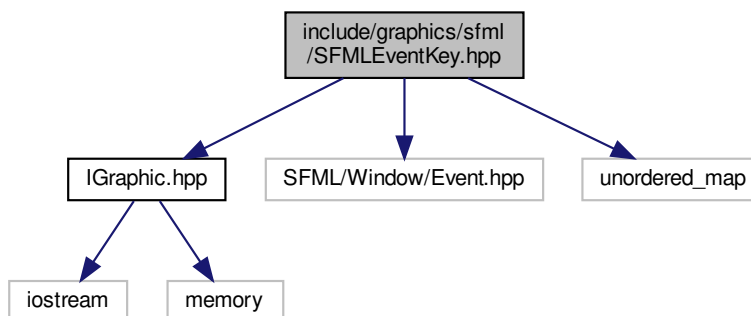
Data Structures

- class [SDLGraphic](#)

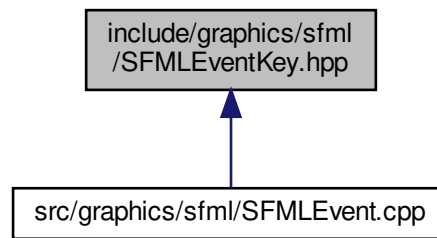
5.14 include/graphics/sfml/SFMLEventKey.hpp File Reference

```
#include "IGraphic.hpp"
#include <SFML/Window/Event.hpp>
#include <unordered_map>
```

Include dependency graph for SFMLEventKey.hpp:



This graph shows which files directly or indirectly include this file:



Variables

- `std::unordered_map< sf::Keyboard::Key, eventKey > keyEvent`

5.14.1 Variable Documentation

5.14.1.1 keyEvent

`std::unordered_map<sf::Keyboard::Key, eventKey> keyEvent`

Initial value:

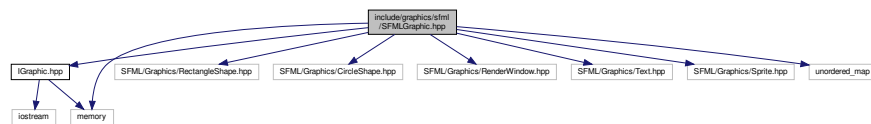
```

= { { sf::Keyboard::Key::A, eventKey::A },
    { sf::Keyboard::Key::B, eventKey::B }, { sf::Keyboard::Key::C, eventKey::C }, { sf::Keyboard::Key::D,
      eventKey::D },
    { sf::Keyboard::Key::E, eventKey::E }, { sf::Keyboard::Key::F, eventKey::F }, { sf::Keyboard::Key::G,
      eventKey::G },
    { sf::Keyboard::Key::H, eventKey::H }, { sf::Keyboard::Key::I, eventKey::I }, { sf::Keyboard::Key::J,
      eventKey::J },
    { sf::Keyboard::Key::K, eventKey::K }, { sf::Keyboard::Key::L, eventKey::L }, { sf::Keyboard::Key::M,
      eventKey::M },
    { sf::Keyboard::Key::N, eventKey::N }, { sf::Keyboard::Key::O, eventKey::O }, { sf::Keyboard::Key::P,
      eventKey::P },
    { sf::Keyboard::Key::Q, eventKey::Q }, { sf::Keyboard::Key::R, eventKey::R }, { sf::Keyboard::Key::S,
      eventKey::S },
    { sf::Keyboard::Key::T, eventKey::T }, { sf::Keyboard::Key::U, eventKey::U }, { sf::Keyboard::Key::V,
      eventKey::V },
    { sf::Keyboard::Key::W, eventKey::W }, { sf::Keyboard::Key::X, eventKey::X }, { sf::Keyboard::Key::Y,
      eventKey::Y },
    { sf::Keyboard::Key::Z, eventKey::Z }, { sf::Keyboard::Key::Space, eventKey::SPACE },
    { sf::Keyboard::Key::Escape, eventKey::ESCAPE }, { sf::Keyboard::Key::Enter, eventKey::ENTER },
    { sf::Keyboard::Key::Left, eventKey::LARROW }, { sf::Keyboard::Key::Right, eventKey::RARROW },
    { sf::Keyboard::Key::Down, eventKey::BARROW }, { sf::Keyboard::Key::Up, eventKey::UARROW },
    { sf::Keyboard::Key::BackSpace, eventKey::DELETE }, { sf::Keyboard::Key::Delete, eventKey::SUPR },
    { sf::Keyboard::Key::Tab, eventKey::TAB }, { sf::Keyboard::Key::Num1, eventKey::ONE },
    { sf::Keyboard::Key::Num2, eventKey::TWO }, { sf::Keyboard::Key::Num3, eventKey::THREE },
    { sf::Keyboard::Key::Num4, eventKey::FOUR }, { sf::Keyboard::Key::Num5, eventKey::FIVE },
    { sf::Keyboard::Key::Num6, eventKey::SIX }, { sf::Keyboard::Key::Num7, eventKey::SEVEN },
    { sf::Keyboard::Key::Num8, eventKey::EIGHT }, { sf::Keyboard::Key::Num9, eventKey::NINE },
    { sf::Keyboard::Key::Num0, eventKey::ZERO }, { sf::Keyboard::Key::LControl, eventKey::CTRL },
    { sf::Keyboard::Key::LAlt, eventKey::ALT }, { sf::Keyboard::Key::LShift, eventKey::SHIFT } }
  
```

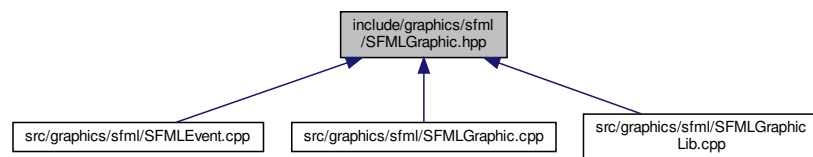

5.15 include/graphics/sfml/SFMLGraphic.hpp File Reference

```
#include "IGraphic.hpp"
#include <SFML/Graphics/RectangleShape.hpp>
#include <SFML/Graphics/CircleShape.hpp>
#include <SFML/Graphics/RenderWindow.hpp>
#include <SFML/Graphics/Text.hpp>
#include <SFML/Graphics/Sprite.hpp>
#include <memory>
#include <unordered_map>
```

Include dependency graph for SFMLGraphic.hpp:



This graph shows which files directly or indirectly include this file:



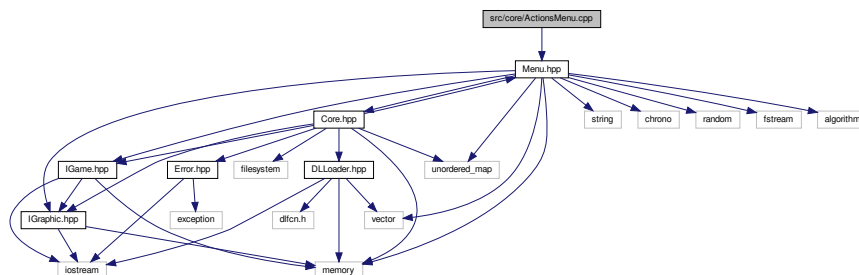
Data Structures

- class [SFMLGraphic](#)

5.16 src/core/ActionsMenu.cpp File Reference

```
#include "Menu.hpp"
```

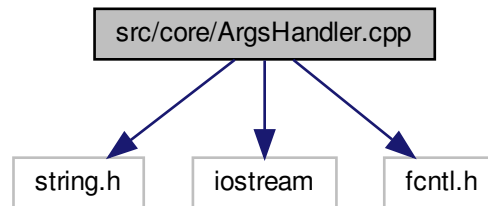
Include dependency graph for ActionsMenu.cpp:



5.17 src/core/ArgsHandler.cpp File Reference

```
#include <string.h>
#include <iostream>
#include <fcntl.h>
```

Include dependency graph for ArgsHandler.cpp:



Functions

- int [checkLibrary](#) (const char *library)
- void [displayUsage](#) ()
- int [checkArgs](#) (int ac, char **av)
- int [checkEnv](#) (char **env)

5.17.1 Function Documentation

5.17.1.1 checkArgs()

```
int checkArgs (
    int ac,
    char ** av )
```

5.17.1.2 checkEnv()

```
int checkEnv (
    char ** env )
```

5.17.1.3 checkLibrary()

```
int checkLibrary (
    const char * library )
```

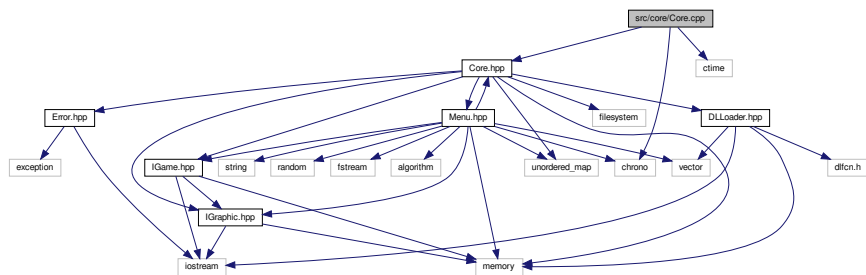
5.17.1.4 displayUsage()

```
void displayUsage ( )
```

5.18 src/core/Core.cpp File Reference

```
#include "Core.hpp"
#include <chrono>
#include <ctime>
```

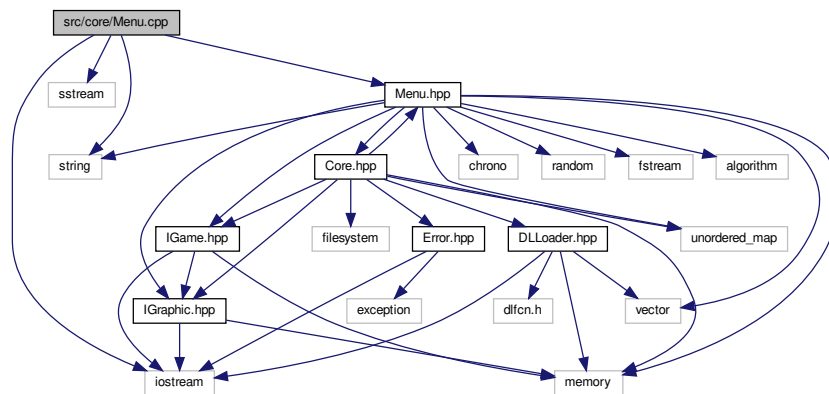
Include dependency graph for Core.cpp:



5.19 src/core/Menu.cpp File Reference

```
#include <iostream>
#include <sstream>
#include <string>
#include "Menu.hpp"
```

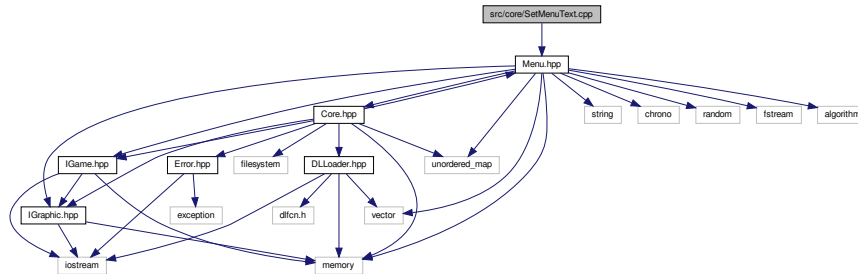
Include dependency graph for Menu.cpp:



5.20 src/core/SetMenuText.cpp File Reference

```
#include "Menu.hpp"
```

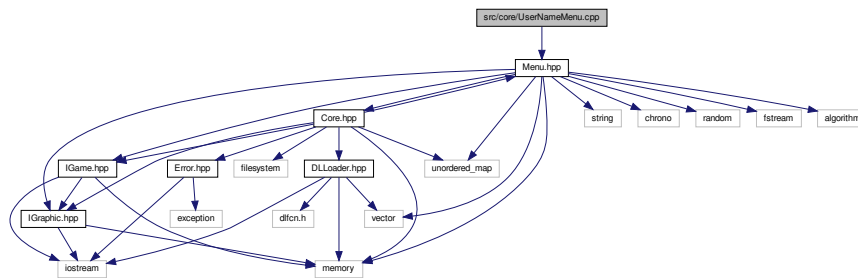
Include dependency graph for SetMenuText.cpp:



5.21 src/core/UserNameMenu.cpp File Reference

```
#include "Menu.hpp"
```

Include dependency graph for UserNameMenu.cpp:



Functions

- int [random](#) (int low, int high)

5.21.1 Function Documentation

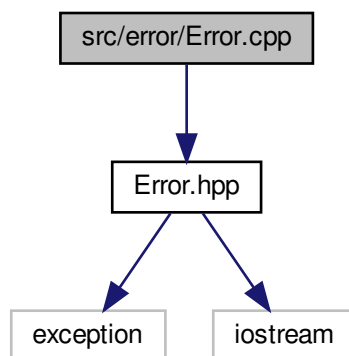
5.21.1.1 random()

```
int random (
    int low,
    int high )
```

5.22 src/error/Error.cpp File Reference

```
#include "Error.hpp"
```

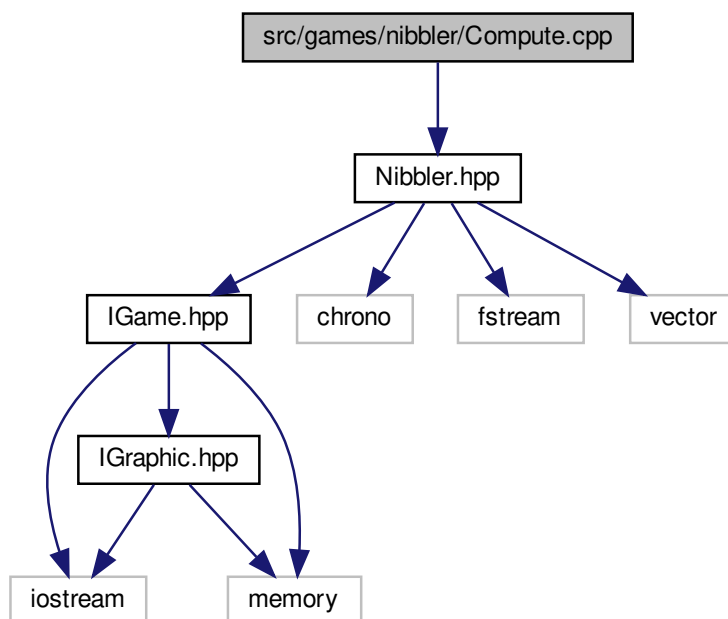
Include dependency graph for Error.cpp:



5.23 src/games/nibbler/Compute.cpp File Reference

```
#include "Nibbler.hpp"
```

Include dependency graph for Compute.cpp:

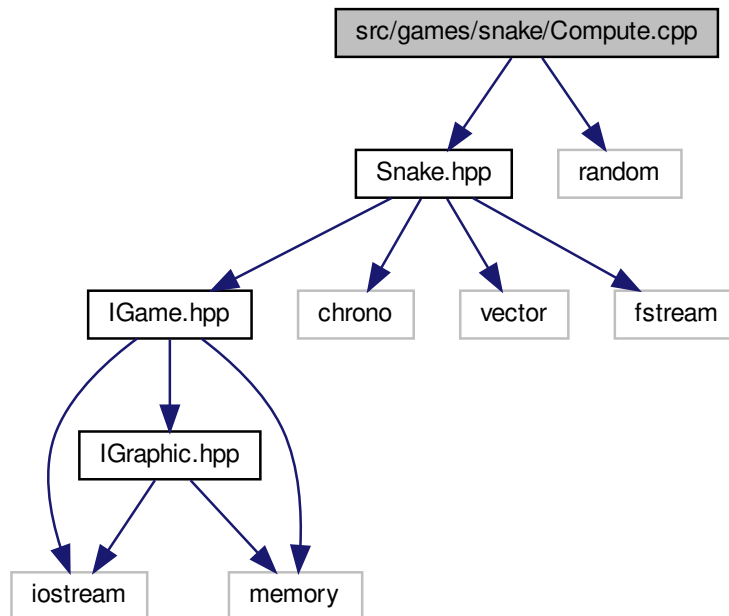


5.24 src/games/snake/Compute.cpp File Reference

```
#include "Snake.hpp"
```

```
#include <random>
```

Include dependency graph for Compute.cpp:



Functions

- int `random` (int low, int high)

5.24.1 Function Documentation

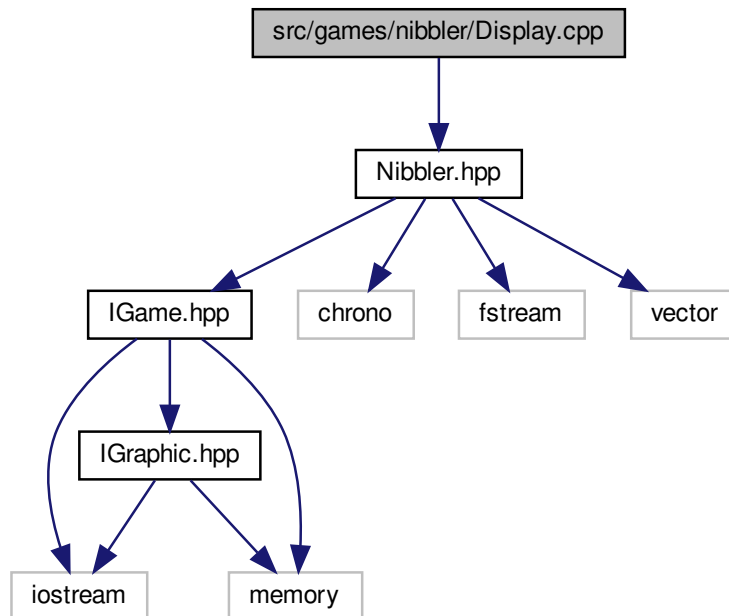
5.24.1.1 `random()`

```
int random (  
    int low,  
    int high )
```

5.25 src/games/nibbler/Display.cpp File Reference

```
#include "Nibbler.hpp"
```

Include dependency graph for Display.cpp:

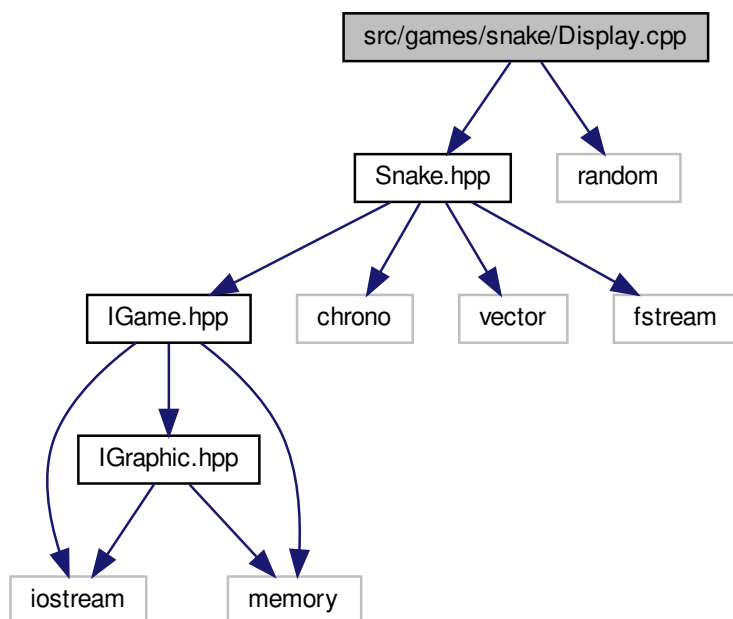


5.26 src/games/snake/Display.cpp File Reference

```
#include "Snake.hpp"
```

```
#include <random>
```

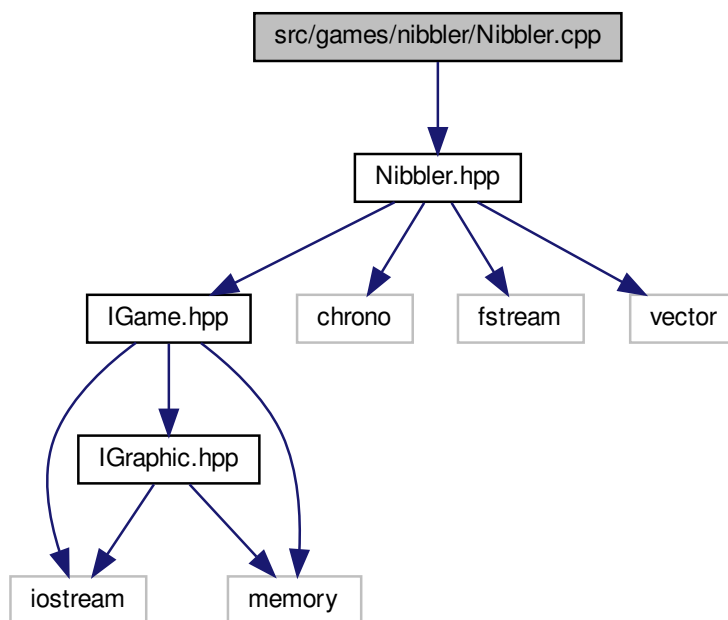
Include dependency graph for Display.cpp:



5.27 src/games/nibbler/Nibbler.cpp File Reference

```
#include "Nibbler.hpp"
```

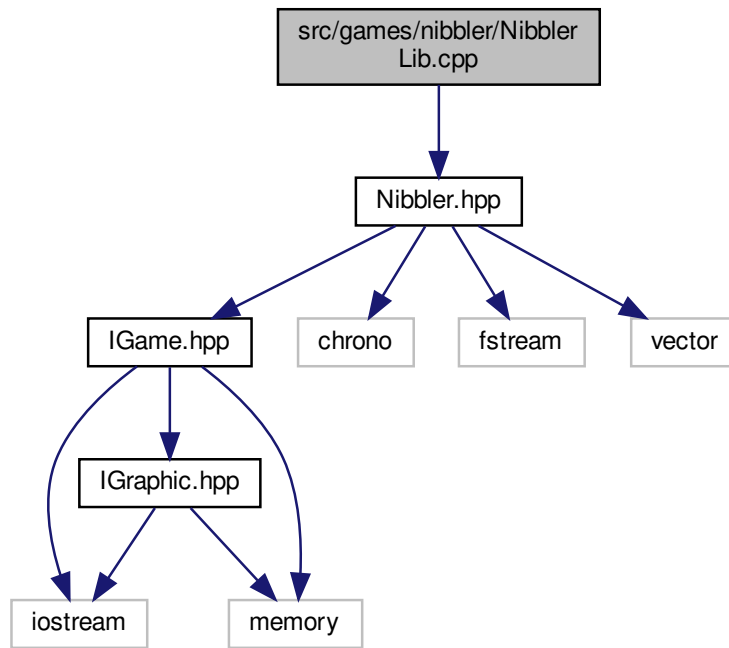
Include dependency graph for Nibbler.cpp:



5.28 src/games/nibbler/NibblerLib.cpp File Reference

```
#include "Nibbler.hpp"
```

Include dependency graph for NibblerLib.cpp:



Functions

- `std::unique_ptr< IGame > entryPoint (void)`

5.28.1 Function Documentation

5.28.1.1 `entryPoint()`

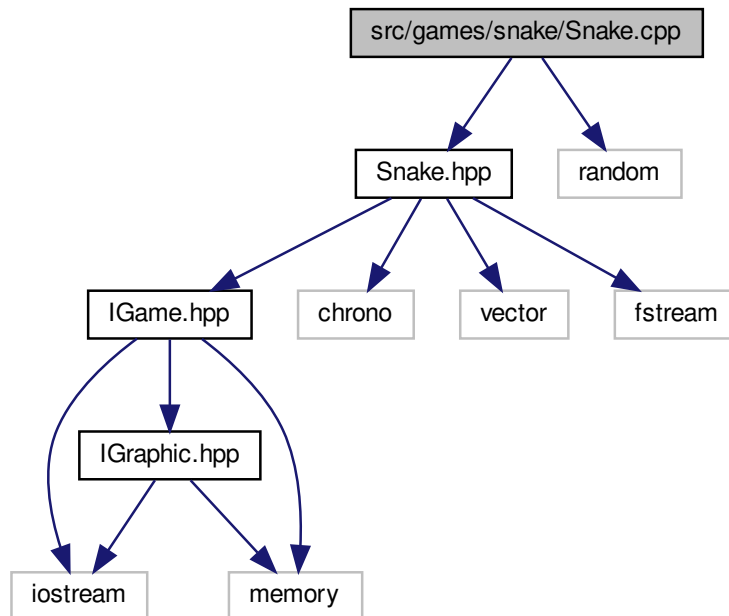
```
std::unique_ptr<IGame> entryPoint (  
    void )
```

5.29 src/games/snake/Snake.cpp File Reference

```
#include "Snake.hpp"
```

```
#include <random>
```

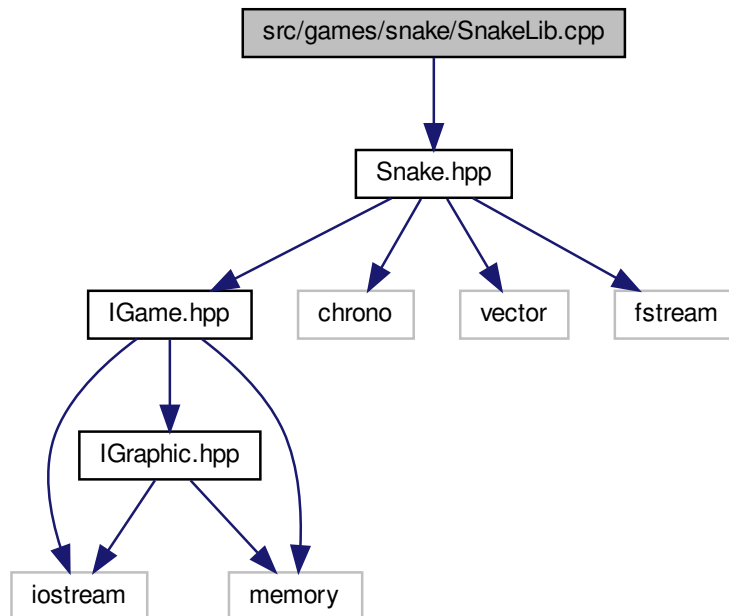
Include dependency graph for Snake.cpp:



5.30 src/games/snake/SnakeLib.cpp File Reference

```
#include "Snake.hpp"
```

Include dependency graph for SnakeLib.cpp:



Functions

- `std::unique_ptr< IGame > entryPoint (void)`

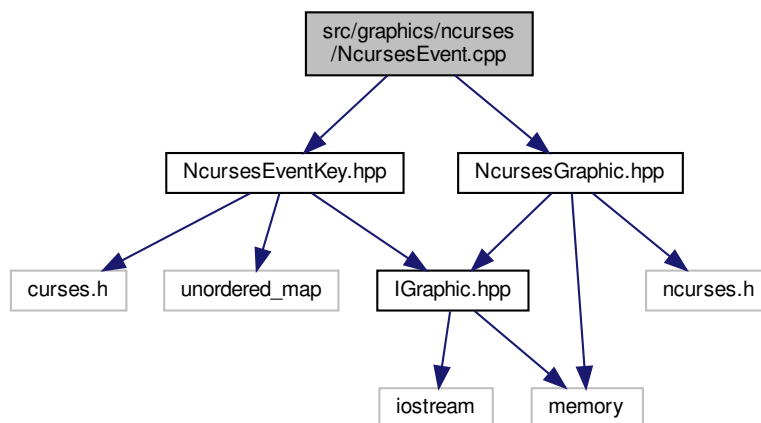
5.30.1 Function Documentation

5.30.1.1 entryPoint()

```
std::unique_ptr<IGame> entryPoint (  
    void )
```

5.31 src/graphics/ncurses/NcursesEvent.cpp File Reference

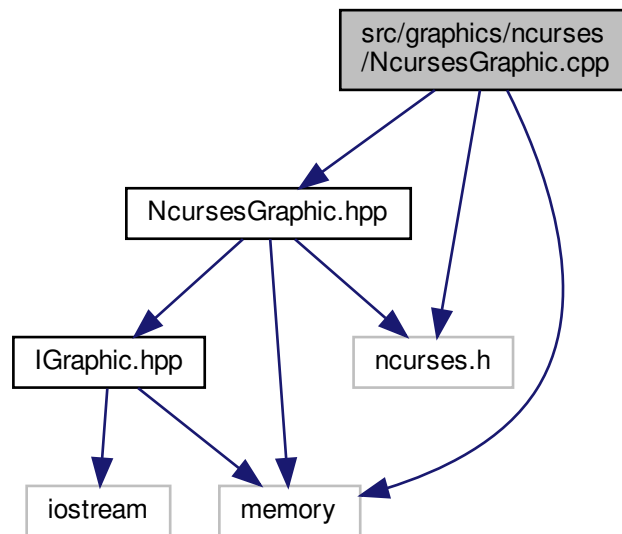
```
#include "NcursesEventKey.hpp"  
#include "NcursesGraphic.hpp"  
Include dependency graph for NcursesEvent.cpp:
```



5.32 src/graphics/ncurses/NcursesGraphic.cpp File Reference

```
#include "NcursesGraphic.hpp"  
#include <memory>  
#include <ncurses.h>
```

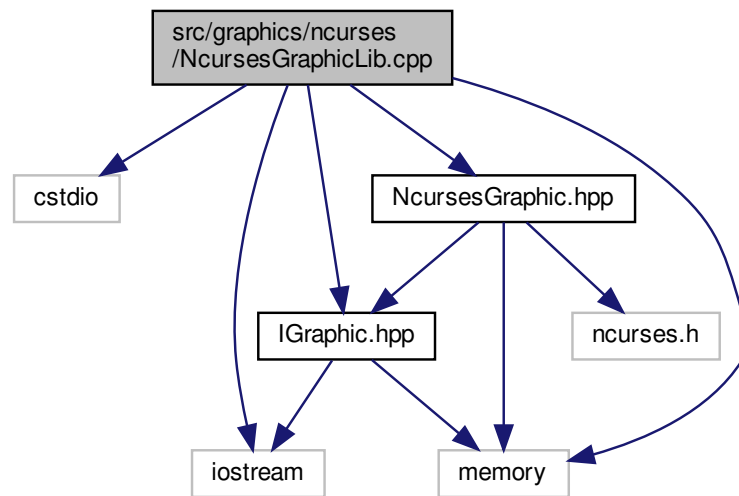
Include dependency graph for NcursesGraphic.cpp:



5.33 src/graphics/ncurses/NcursesGraphicLib.cpp File Reference

```
#include <stdio>
#include <iostream>
#include <memory>
#include "IGraphic.hpp"
#include "NcursesGraphic.hpp"
```

Include dependency graph for NcursesGraphicLib.cpp:



Functions

- `std::unique_ptr<IGraphic> entryPoint (void)`

5.33.1 Function Documentation

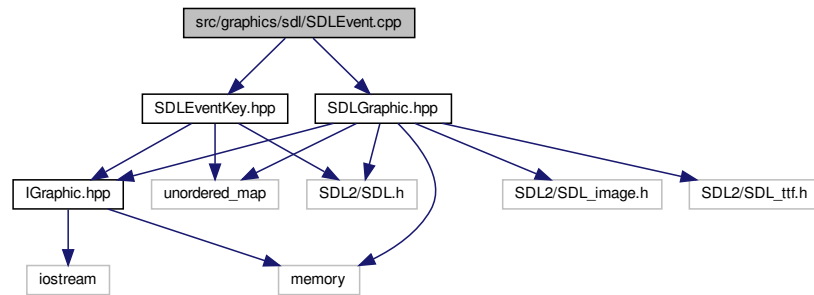
5.33.1.1 entryPoint()

```
std::unique_ptr<IGraphic> entryPoint (
    void )
```

5.34 src/graphics/sdl/SDLEvent.cpp File Reference

```
#include "SDLEventKey.hpp"
#include "SDLGraphic.hpp"
```

Include dependency graph for `SDLEvent.cpp`:



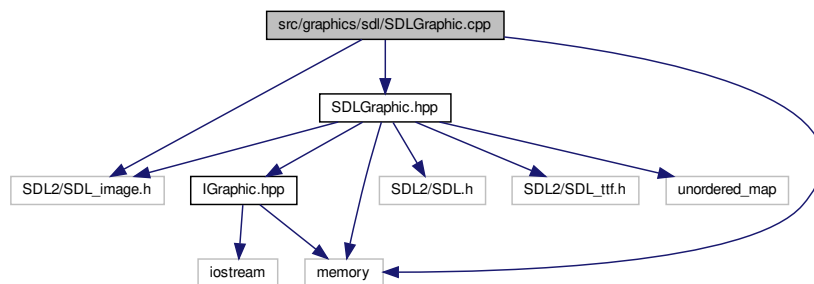
5.35 `src/graphics/sdl/SDLGraphic.cpp` File Reference

```

#include <SDL2/SDL_image.h>
#include <SDLGraphic.hpp>
#include <memory>

```

Include dependency graph for `SDLGraphic.cpp`:



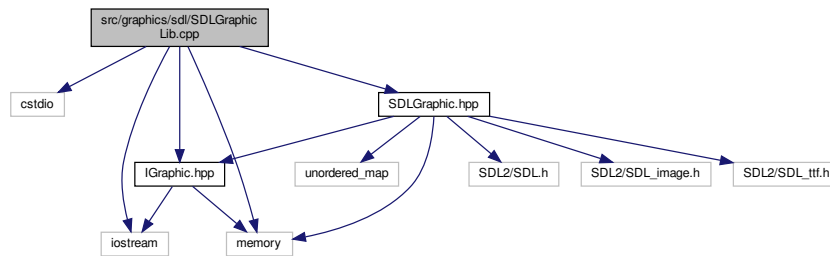
5.36 `src/graphics/sdl/SDLGraphicLib.cpp` File Reference

```

#include <cstdio>
#include <iostream>
#include <memory>
#include "IGraphic.hpp"
#include "SDLGraphic.hpp"

```


Include dependency graph for SDLGraphicLib.cpp:



Functions

- `std::unique_ptr< IGraphic > entryPoint (void)`

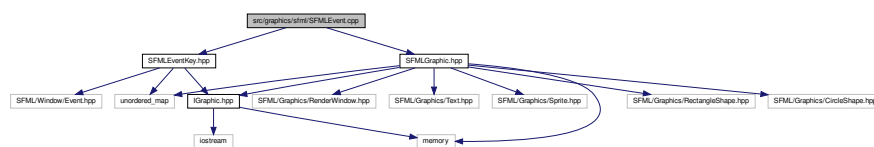
5.36.1 Function Documentation

5.36.1.1 entryPoint()

```
std::unique_ptr< IGraphic > entryPoint (
    void )
```

5.37 src/graphics/sfml/SFMLEvent.cpp File Reference

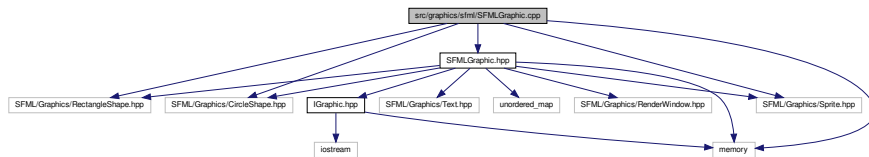
```
#include "SFMLEventKey.hpp"
#include "SFMLGraphic.hpp"
Include dependency graph for SFMLEvent.cpp:
```



5.38 src/graphics/sfml/SFMLGraphic.cpp File Reference

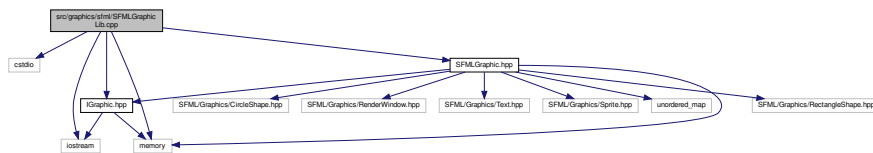
```
#include "SFMLGraphic.hpp"
#include <SFML/Graphics/CircleShape.hpp>
#include <SFML/Graphics/RectangleShape.hpp>
#include <SFML/Graphics/Sprite.hpp>
#include <memory>
```

Include dependency graph for SFMLGraphic.cpp:



5.39 src/graphics/sfml/SFMLGraphicLib.cpp File Reference

```
#include <cstdio>
#include <iostream>
#include <memory>
#include "IGraphic.hpp"
#include "SFMLGraphic.hpp"
Include dependency graph for SFMLGraphicLib.cpp:
```



Functions

- `std::unique_ptr< IGraphic > entryPoint (void)`

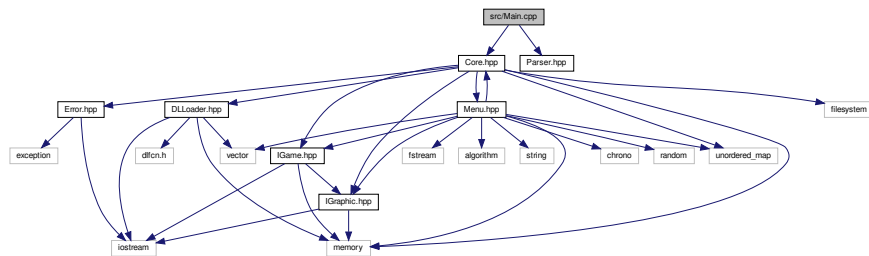
5.39.1 Function Documentation

5.39.1.1 entryPoint()

```
std::unique_ptr< IGraphic > entryPoint (
    void )
```

5.40 src/Main.cpp File Reference

```
#include "Core.hpp"
#include "Parser.hpp"
Include dependency graph for Main.cpp:
```



Functions

- int [main](#) (int ac, char **av, char **env)

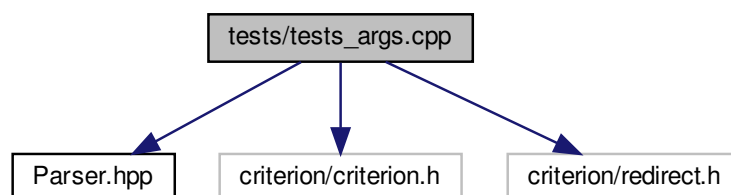
5.40.1 Function Documentation

5.40.1.1 main()

```
int main (
    int ac,
    char ** av,
    char ** env )
```

5.41 tests/tests_args.cpp File Reference

```
#include "Parser.hpp"
#include <riterion/criterion.h>
#include <riterion/redirect.h>
Include dependency graph for tests_args.cpp:
```



Functions

- [Test](#) ([checkArgs](#), no_args)
- [Test](#) ([checkArgs](#), no_args2)
- [Test](#) ([checkArgs](#), bad_args)
- [Test](#) ([checkArgs](#), wrong_lib)
- [Test](#) ([checkArgs](#), wrong_lib2)
- [Test](#) ([checkArgs](#), one_arg)
- [Test](#) ([checkArgs](#), two_args)
- [Test](#) ([checkArgs](#), help)
- [Test](#) ([checkEnv](#), no_env)

5.41.1 Function Documentation

5.41.1.1 Test() [1/9]

```
Test (
    checkArgs ,
    bad_args )
```

5.41.1.2 Test() [2/9]

```
Test (
    checkArgs ,
    help )
```

5.41.1.3 Test() [3/9]

```
Test (
    checkArgs ,
    no_args )
```

5.41.1.4 Test() [4/9]

```
Test (
    checkArgs ,
    no_args2 )
```

5.41.1.5 Test() [5/9]

```
Test (
    checkArgs ,
    one_arg )
```

5.41.1.6 Test() [6/9]

```
Test (
    checkArgs ,
    two_args )
```

5.41.1.7 Test() [7/9]

```
Test (
    checkArgs ,
    wrong_lib )
```

5.41.1.8 Test() [8/9]

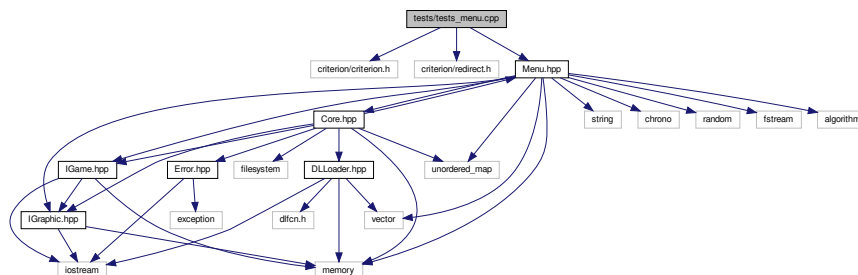
```
Test (
    checkArgs ,
    wrong_lib2 )
```

5.41.1.9 Test() [9/9]

```
Test (
    checkEnv ,
    no_env )
```

5.42 tests/tests_menu.cpp File Reference

```
#include <criterion/criterion.h>
#include <criterion/redirect.h>
#include "Menu.hpp"
Include dependency graph for tests_menu.cpp:
```



Functions

- [Test](#) (MenuTest, coreStateNotModifiedWhenNotTyping)
- [Test](#) (MenuTest, deleteCharIsCalledOnDeleteKey)
- [Test](#) (MenuTest, coreStateIsSetToQuitOnQuitKey)
- [Test](#) (MenuTest, characterIsAddedToUserNameOnValidKey)

5.42.1 Function Documentation

5.42.1.1 Test() [1/4]

```
Test (
    MenuTest ,
    characterIsAddedToUserNameOnValidKey )
```

5.42.1.2 Test() [2/4]

```
Test (
    MenuTest ,
    coreStateIsSetToQuitOnQuitKey )
```

5.42.1.3 Test() [3/4]

```
Test (
    MenuTest ,
    coreStateNotModifiedWhenNotTyping )
```

5.42.1.4 Test() [4/4]

```
Test (
    MenuTest ,
    deleteCharIsCalledOnDeleteKey )
```

Index

- ~ArcadeError
 - ArcadeError, [8](#)
- ~Core
 - Core, [13](#)
- ~DLLoader
 - DLLoader< T >, [19](#)
- ~IGame
 - IGame, [23](#)
- ~IGraphic
 - IGraphic, [26](#)
- ~Menu
 - Menu, [31](#)
- ~NcursesGraphic
 - NcursesGraphic, [39](#)
- ~Nibbler
 - Nibbler, [44](#)
- ~SDLGraphic
 - SDLGraphic, [53](#)
- ~SFMLGraphic
 - SFMLGraphic, [59](#)
- ~Snake
 - Snake, [67](#)

- A
 - IGraphic.hpp, [91](#)
- a
 - color, [10](#)
- actualLib_
 - DLLoader< T >, [20](#)
- addFood
 - Nibbler, [45](#)
 - Snake, [68](#)
- addWall
 - Nibbler, [45](#)
 - Snake, [68](#)
- ALIVE
 - Nibbler.hpp, [87](#)
 - Snake.hpp, [89](#)
- allMaps
 - Nibbler.hpp, [87](#)
 - Snake.hpp, [89](#)
- ALT
 - IGraphic.hpp, [92](#)
- applyChanges
 - Menu, [31](#)
- ArcadeError, [7](#)
 - ~ArcadeError, [8](#)
 - ArcadeError, [8](#)
 - msg_, [9](#)
 - what, [9](#)

- ArgsHandler.cpp
 - checkArgs, [100](#)
 - checkEnv, [100](#)
 - checkLibrary, [100](#)
 - displayUsage, [101](#)
- B
 - IGraphic.hpp, [92](#)
- b
 - color, [10](#)
- BARROW
 - IGraphic.hpp, [92](#)
- C
 - IGraphic.hpp, [91](#)
- cellHeight_
 - Nibbler, [48](#)
 - Snake, [71](#)
- cellWidth_
 - Nibbler, [48](#)
 - Snake, [71](#)
- checkArgs
 - ArgsHandler.cpp, [100](#)
 - Parser.hpp, [83](#)
- checkEnv
 - ArgsHandler.cpp, [100](#)
 - Parser.hpp, [83](#)
- checkLibrary
 - ArgsHandler.cpp, [100](#)
 - Parser.hpp, [83](#)
- chooseDirection
 - Nibbler, [45](#)
- chooseGame
 - Menu, [31](#)
- chooseLib
 - Menu, [31](#)
- CIRCLE
 - IGraphic.hpp, [92](#)
- circleList
 - SFMLGraphic, [61](#)
- clearWindow
 - IGraphic, [26](#)
 - NcursesGraphic, [39](#)
 - SDLGraphic, [53](#)
 - SFMLGraphic, [59](#)
- closeLib
 - DLLoader< T >, [19](#)
- color, [9](#)
 - a, [10](#)
 - b, [10](#)

- g, 10
- r, 10
- Compute.cpp
 - random, 104
- CONTINUE
 - Core.hpp, 80
- coord, 10
 - x, 11
 - y, 11
- Core, 11
 - ~Core, 13
 - Core, 13
 - coreStateHandler, 14
 - currentGame_, 17
 - currentGraph_, 17
 - findPathIndex, 14
 - gameLib_, 17
 - gameLoader_, 17
 - gameLoopHandler, 14
 - gamePaths_, 17
 - gameState_, 17
 - getAilLib, 14
 - getCoreState, 14
 - getCurrentGame, 14
 - getCurrentGraph, 14
 - getGamePaths, 14
 - getGraphPaths, 15
 - graphLib_, 17
 - graphLoader_, 17
 - graphPaths_, 18
 - handleEvent, 15
 - isLibGraphical, 15
 - loadNextGame, 15
 - loadNextGraph, 15
 - loadSpecificGame, 15
 - loadSpecificGraph, 15
 - menu_, 18
 - pushLib, 16
 - quitArcade, 16
 - restartGame, 16
 - setCoreState, 16
 - setCurrentGame, 16
 - setCurrentGraph, 16
- Core.hpp
 - CONTINUE, 80
 - GAME, 80
 - GRAPHICAL, 80
 - GState, 80
 - libType, 80
 - MENU, 80
 - PAUSE, 80
 - PLAY, 80
 - QUIT, 80
 - validLibs, 80
- coreStateHandler
 - Core, 14
- counter_
 - Menu, 35
- createGuiTextMenu
 - Menu, 31
- createNewUser
 - Menu, 31
- createTitleMenu
 - Menu, 31
- createWindow
 - IGraphic, 27
 - NcursesGraphic, 39
 - SDLGraphic, 53
 - SFMLGraphic, 60
- CTRL
 - IGraphic.hpp, 92
- currentGame_
 - Core, 17
- currentGraph_
 - Core, 17
- D
 - IGraphic.hpp, 91
- DEAD
 - Nibbler.hpp, 87
 - Snake.hpp, 89
- DELETE
 - IGraphic.hpp, 92
- deleteChar
 - Menu, 32
- destroyWindow
 - IGraphic, 27
 - NcursesGraphic, 39
 - SDLGraphic, 54
 - SFMLGraphic, 60
- dir_
 - Nibbler, 48
 - Snake, 71
- direction
 - Nibbler.hpp, 87
 - Snake.hpp, 89
- display
 - IGame, 23
 - Nibbler, 45
 - Snake, 68
- displayShape
 - IGraphic, 27
 - NcursesGraphic, 40
 - SDLGraphic, 54
 - SFMLGraphic, 60
- displaySprite
 - IGraphic, 27
 - NcursesGraphic, 40
 - SDLGraphic, 54
 - SFMLGraphic, 60
- displayText
 - IGraphic, 27
 - NcursesGraphic, 40
 - SDLGraphic, 54
 - SFMLGraphic, 60
- displayUsage
 - ArgsHandler.cpp, 101

- Parser.hpp, [83](#)
- displayWindow
 - IGraphic, [27](#)
 - NcursesGraphic, [40](#)
 - SDLGraphic, [54](#)
 - SFMLGraphic, [60](#)
- DLLoader
 - DLLoader< T >, [19](#)
- DLLoader< T >, [18](#)
 - ~DLLoader, [19](#)
 - actualLib_, [20](#)
 - closeLib, [19](#)
 - DLLoader, [19](#)
 - getInstance, [20](#)
- DOWN
 - Nibbler.hpp, [87](#)
 - Snake.hpp, [89](#)
- drawCircle
 - SDLGraphic, [55](#)
 - SFMLGraphic, [61](#)
- drawRectangle
 - NcursesGraphic, [40](#)
 - SDLGraphic, [55](#)
 - SFMLGraphic, [61](#)
- E
 - IGraphic.hpp, [91](#)
- EIGHT
 - IGraphic.hpp, [92](#)
- elemSize, [20](#)
 - height, [21](#)
 - width, [21](#)
- ENTER
 - IGraphic.hpp, [92](#)
- entryPoint
 - NcursesGraphicLib.cpp, [113](#)
 - NibblerLib.cpp, [108](#)
 - SDLGraphicLib.cpp, [115](#)
 - SFMLGraphicLib.cpp, [116](#)
 - SnakeLib.cpp, [110](#)
- ESCAPE
 - IGraphic.hpp, [92](#)
- eventKey
 - IGraphic.hpp, [91](#)
- F
 - IGraphic.hpp, [91](#)
- findPathIndex
 - Core, [14](#)
- FIVE
 - IGraphic.hpp, [92](#)
- fontPath
 - text, [75](#)
- fonts_
 - SDLGraphic, [55](#)
 - SFMLGraphic, [61](#)
- fontSize
 - text, [75](#)
- food_
 - Nibbler, [48](#)
 - Snake, [71](#)
- foodHandler
 - Nibbler, [45](#)
 - Snake, [68](#)
- FOUR
 - IGraphic.hpp, [92](#)
- G
 - IGraphic.hpp, [91](#)
- g
 - color, [10](#)
- GAME
 - Core.hpp, [80](#)
- gameLib_
 - Core, [17](#)
- gameLoader_
 - Core, [17](#)
- gameLoopHandler
 - Core, [14](#)
- gameOver
 - Snake, [68](#)
- gamePaths_
 - Core, [17](#)
 - Menu, [35](#)
- gameState_
 - Core, [17](#)
- gameTextMenu_
 - Menu, [35](#)
- getAllLib
 - Core, [14](#)
- getCoreState
 - Core, [14](#)
- getCurrentGame
 - Core, [14](#)
- getCurrentGraph
 - Core, [14](#)
- getDisplaySize
 - IGame, [23](#)
 - Nibbler, [45](#)
 - Snake, [68](#)
- getEvent
 - IGraphic, [28](#)
 - NcursesGraphic, [40](#)
 - SDLGraphic, [55](#)
 - SFMLGraphic, [61](#)
- getGamePaths
 - Core, [14](#)
- getGraphPaths
 - Core, [15](#)
- getInstance
 - DLLoader< T >, [20](#)
- getSpritePath
 - IGame, [24](#)
 - Nibbler, [46](#)
 - Snake, [69](#)
- getTop3Scores
 - Menu, [32](#)
- getUserName

- Menu, [32](#)
- GRAPHICAL
 - Core.hpp, [80](#)
- graphLib_
 - Core, [17](#)
- graphLoader_
 - Core, [17](#)
- graphPaths_
 - Core, [18](#)
 - Menu, [35](#)
- GState
 - Core.hpp, [80](#)
- guiTextMenu_
 - Menu, [35](#)
- H
 - IGraphic.hpp, [91](#)
- handleEvent
 - Core, [15](#)
 - Menu, [32](#)
- handleUserInput
 - Menu, [32](#)
- height
 - elemSize, [21](#)
- highlightSelected
 - Menu, [32](#)
- highlightTitle
 - Menu, [32](#)
- I
 - IGraphic.hpp, [91](#)
- IGame, [21](#)
 - ~IGame, [23](#)
 - display, [23](#)
 - getDisplaySize, [23](#)
 - getSpritePath, [24](#)
 - init, [24](#)
 - reset, [24](#)
 - updateGame, [24](#)
- IGraphic, [25](#)
 - ~IGraphic, [26](#)
 - clearWindow, [26](#)
 - createWindow, [27](#)
 - destroyWindow, [27](#)
 - displayShape, [27](#)
 - displaySprite, [27](#)
 - displayText, [27](#)
 - displayWindow, [27](#)
 - getEvent, [28](#)
 - isOpenWindow, [28](#)
- IGraphic.hpp
 - A, [91](#)
 - ALT, [92](#)
 - B, [92](#)
 - BARROW, [92](#)
 - C, [91](#)
 - CIRCLE, [92](#)
 - CTRL, [92](#)
 - D, [91](#)
 - DELETE, [92](#)
 - E, [91](#)
 - EIGHT, [92](#)
 - ENTER, [92](#)
 - ESCAPE, [92](#)
 - eventKey, [91](#)
 - F, [91](#)
 - FIVE, [92](#)
 - FOUR, [92](#)
 - G, [91](#)
 - H, [91](#)
 - I, [91](#)
 - J, [91](#)
 - K, [91](#)
 - L, [91](#)
 - LARROW, [92](#)
 - M, [91](#)
 - N, [92](#)
 - NINE, [92](#)
 - NULL_EVENT, [91](#)
 - O, [91](#)
 - ONE, [92](#)
 - P, [91](#)
 - Q, [91](#)
 - QUIT, [92](#)
 - R, [91](#)
 - RARROW, [92](#)
 - RECTANGLE, [92](#)
 - S, [91](#)
 - SEVEN, [92](#)
 - shapeType, [92](#)
 - SHIFT, [92](#)
 - SIX, [92](#)
 - SPACE, [92](#)
 - SUPR, [92](#)
 - T, [91](#)
 - TAB, [92](#)
 - THREE, [92](#)
 - TRIANGLE, [92](#)
 - TWO, [92](#)
 - U, [91](#)
 - UARROW, [92](#)
 - V, [92](#)
 - W, [91](#)
 - X, [91](#)
 - Y, [91](#)
 - Z, [91](#)
 - ZERO, [92](#)
- include/core/Core.hpp, [79](#)
- include/core/DLLoader.hpp, [81](#)
- include/core/Menu.hpp, [82](#)
- include/core/Parser.hpp, [83](#)
- include/error/Error.hpp, [84](#)
- include/games/IGame.hpp, [84](#)
- include/games/nibbler/Nibbler.hpp, [85](#)
- include/games/snake/Snake.hpp, [88](#)
- include/graphics/IGraphic.hpp, [90](#)
- include/graphics/ncurses/NcursesEventKey.hpp, [92](#)

- include/graphics/ncurses/NcursesGraphic.hpp, 94
- include/graphics/sdl/SDL_EventKey.hpp, 95
- include/graphics/sdl/SDL_Graphic.hpp, 96
- include/graphics/sfml/SFML_EventKey.hpp, 97
- include/graphics/sfml/SFML_Graphic.hpp, 99
- incrGame_
 - Menu, 35
- incrLib_
 - Menu, 35
- init
 - IGame, 24
 - Nibbler, 46
 - Snake, 69
- initNibbler
 - Nibbler, 46
- initSnake
 - Snake, 69
- initText
 - Nibbler, 46
 - Snake, 69
- isCollided
 - Nibbler, 46
 - Snake, 69
- isGameSelected_
 - Menu, 35
- isLibGraphical
 - Core, 15
- isLoadedTexture_
 - SFML_Graphic, 62
- isNibblerInCell
 - Nibbler, 46
- isOpen_
 - NcursesGraphic, 41
 - SDL_Graphic, 55
- isOpenWindow
 - IGraphic, 28
 - NcursesGraphic, 41
 - SDL_Graphic, 55
 - SFML_Graphic, 61
- isSnakeInCell
 - Snake, 69
- isUserTyping
 - Menu, 33
- isUserTyping_
 - Menu, 36
- J
 - IGraphic.hpp, 91
- K
 - IGraphic.hpp, 91
- keyEvent
 - NcursesEventKey.hpp, 93
 - SDL_EventKey.hpp, 96
 - SFML_EventKey.hpp, 98
- keyMap_
 - Menu, 36
- L
 - IGraphic.hpp, 91
- LARROW
 - IGraphic.hpp, 92
- lastUpdateTime_
 - Menu, 36
 - Nibbler, 48
 - Snake, 71
- LEFT
 - Nibbler.hpp, 87
 - Snake.hpp, 89
- libTextMenu_
 - Menu, 36
- libType
 - Core.hpp, 80
- loadMap
 - Nibbler, 47
 - Snake, 70
- loadNextGame
 - Core, 15
- loadNextGraph
 - Core, 15
- loadSpecificGame
 - Core, 15
- loadSpecificGraph
 - Core, 15
- loopTitle
 - Menu, 33
- M
 - IGraphic.hpp, 91
- m_color
 - shape, 64
 - sprite, 74
 - text, 76
- m_texture
 - sprite, 74
- main
 - Main.cpp, 117
- Main.cpp
 - main, 117
- mapIndex_
 - Nibbler, 48
 - Snake, 71
- MENU
 - Core.hpp, 80
- Menu, 28
 - ~Menu, 31
 - applyChanges, 31
 - chooseGame, 31
 - chooseLib, 31
 - counter_, 35
 - createGuiTextMenu, 31
 - createNewUser, 31
 - createTitleMenu, 31
 - deleteChar, 32
 - gamePaths_, 35
 - gameTextMenu_, 35
 - getTop3Scores, 32
 - getUserName, 32

- graphPaths_, 35
- guiTextMenu_, 35
- handleEvent, 32
- handleUserInput, 32
- highlightSelected, 32
- highlightTitle, 32
- incrGame_, 35
- incrLib_, 35
- isGameSelected_, 35
- isUserTyping, 33
- isUserTyping_, 36
- keyMap_, 36
- lastUpdateTime_, 36
- libTextMenu_, 36
- loopTitle, 33
- Menu, 30
- menuLoopHandler, 33
- moveDown, 33
- moveUp, 33
- saveUserName, 33
- scoreText_, 36
- setAvailableLibText, 33
- setCursorsMenu, 34
- setGameLibText, 34
- setGraphLibText, 34
- setHighScoreText, 34
- setLibNameMenu, 34
- setScoreboardTitle, 34
- setUserNameText, 34
- titleMenu_, 36
- userName_, 36
- menu_
 - Core, 18
- menuLoopHandler
 - Menu, 33
- moveDown
 - Menu, 33
- moveSnake
 - Nibbler, 47
 - Snake, 70
- moveUp
 - Menu, 33
- msg_
 - ArcadeError, 9
- N
 - IGraphic.hpp, 92
- NcursesEventKey.hpp
 - keyEvent, 93
- NcursesGraphic, 37
 - ~NcursesGraphic, 39
 - clearWindow, 39
 - createWindow, 39
 - destroyWindow, 39
 - displayShape, 40
 - displaySprite, 40
 - displayText, 40
 - displayWindow, 40
 - drawRectangle, 40
 - getEvent, 40
 - isOpen_, 41
 - isOpenWindow, 41
 - NcursesGraphic, 39
 - window_, 41
- NcursesGraphicLib.cpp
 - entryPoint, 113
- Nibbler, 42
 - ~Nibbler, 44
 - addFood, 45
 - addWall, 45
 - cellHeight_, 48
 - cellWidth_, 48
 - chooseDirection, 45
 - dir_, 48
 - display, 45
 - food_, 48
 - foodHandler, 45
 - getDisplaySize, 45
 - getSpritePath, 46
 - init, 46
 - initNibbler, 46
 - initText, 46
 - isCollided, 46
 - isNibblerInCell, 46
 - lastUpdateTime_, 48
 - loadMap, 47
 - mapIndex_, 48
 - moveSnake, 47
 - Nibbler, 44
 - nibbler_, 49
 - nibblerSize_, 49
 - remainingFood_, 49
 - reset, 47
 - resetLevel, 47
 - restartEvent, 47
 - saveUserScore, 47
 - score_, 49
 - state, 49
 - texts_, 49
 - timer_, 49
 - updateDirection, 47
 - updateGame, 48
 - walls_, 49
- Nibbler.hpp
 - ALIVE, 87
 - allMaps, 87
 - DEAD, 87
 - direction, 87
 - DOWN, 87
 - LEFT, 87
 - playerState, 87
 - RIGHT, 87
 - STOP, 87
 - UP, 87
 - WON, 87
- nibbler_
 - Nibbler, 49

NibblerLib.cpp
 entryPoint, 108
nibblerSize_
 Nibbler, 49
NINE
 IGraphic.hpp, 92
NULL_EVENT
 IGraphic.hpp, 91

O
 IGraphic.hpp, 91
ONE
 IGraphic.hpp, 92

P
 IGraphic.hpp, 91
Parser.hpp
 checkArgs, 83
 checkEnv, 83
 checkLibrary, 83
 displayUsage, 83
path
 texture, 77
PAUSE
 Core.hpp, 80
PLAY
 Core.hpp, 80
playerState
 Nibbler.hpp, 87
 Snake.hpp, 89
pos
 shape, 64
 sprite, 74
 text, 76
 texture, 77
pushLib
 Core, 16

Q
 IGraphic.hpp, 91
QUIT
 Core.hpp, 80
 IGraphic.hpp, 92
quitArcade
 Core, 16

R
 IGraphic.hpp, 91
r
 color, 10
random
 Compute.cpp, 104
 UserNameMenu.cpp, 102
RARROW
 IGraphic.hpp, 92
RECTANGLE
 IGraphic.hpp, 92
rectList
 SFMLGraphic, 62
 remainingFood_
 Nibbler, 49
renderer_
 SDLGraphic, 56
replacementChar
 shape, 64
 sprite, 74
reset
 IGame, 24
 Nibbler, 47
 Snake, 70
resetLevel
 Nibbler, 47
 Snake, 70
restartEvent
 Nibbler, 47
 Snake, 70
restartGame
 Core, 16
RIGHT
 Nibbler.hpp, 87
 Snake.hpp, 89

S
 IGraphic.hpp, 91
saveUserName
 Menu, 33
saveUserScore
 Nibbler, 47
 Snake, 70
score_
 Nibbler, 49
 Snake, 72
scoreText_
 Menu, 36
SDLEventKey.hpp
 keyEvent, 96
SDLGraphic, 50
 ~SDLGraphic, 53
 clearWindow, 53
 createWindow, 53
 destroyWindow, 54
 displayShape, 54
 displaySprite, 54
 displayText, 54
 displayWindow, 54
 drawCircle, 55
 drawRectangle, 55
 fonts_, 55
 getEvent, 55
 isOpen_, 55
 isOpenWindow, 55
 renderer_, 56
 SDLGraphic, 53
 spriteSurface_, 56
 spriteTexture_, 56
 textPath_, 56
 window_, 56
SDLGraphicLib.cpp

- entryPoint, 115
- setAvailableLibText
 - Menu, 33
- setCoreState
 - Core, 16
- setCurrentGame
 - Core, 16
- setCurrentGraph
 - Core, 16
- setCursorsMenu
 - Menu, 34
- setGameLibText
 - Menu, 34
- setGraphLibText
 - Menu, 34
- setHighScoreText
 - Menu, 34
- setLibNameMenu
 - Menu, 34
- setScoreboardTitle
 - Menu, 34
- setUserNameText
 - Menu, 34
- SEVEN
 - IGraphic.hpp, 92
- SFMLEventKey.hpp
 - keyEvent, 98
- SFMLGraphic, 57
 - ~SFMLGraphic, 59
 - circleList, 61
 - clearWindow, 59
 - createWindow, 60
 - destroyWindow, 60
 - displayShape, 60
 - displaySprite, 60
 - displayText, 60
 - displayWindow, 60
 - drawCircle, 61
 - drawRectangle, 61
 - fonts_, 61
 - getEvent, 61
 - isLoadingTexture_, 62
 - isOpenWindow, 61
 - rectList, 62
 - SFMLGraphic, 59
 - spriteList_, 62
 - textList, 62
 - textPath_, 62
 - texture_, 62
 - window_, 62
- SFMLGraphicLib.cpp
 - entryPoint, 116
- shape, 63
 - m_color, 64
 - pos, 64
 - replacementChar, 64
 - size, 64
 - text, 64
 - type, 64
- shapeType
 - IGraphic.hpp, 92
- SHIFT
 - IGraphic.hpp, 92
- SIX
 - IGraphic.hpp, 92
- size
 - shape, 64
 - sprite, 74
 - texture, 77
- Snake, 65
 - ~Snake, 67
 - addFood, 68
 - addWall, 68
 - cellHeight_, 71
 - cellWidth_, 71
 - dir_, 71
 - display, 68
 - food_, 71
 - foodHandler, 68
 - gameOver, 68
 - getDisplaySize, 68
 - getSpritePath, 69
 - init, 69
 - initSnake, 69
 - initText, 69
 - isCollided, 69
 - isSnakeInCell, 69
 - lastUpdateTime_, 71
 - loadMap, 70
 - mapIndex_, 71
 - moveSnake, 70
 - reset, 70
 - resetLevel, 70
 - restartEvent, 70
 - saveUserScore, 70
 - score_, 72
 - Snake, 67
 - snake_, 72
 - snakeSize_, 72
 - state, 72
 - texts_, 72
 - timer_, 72
 - updateDirection, 70
 - updateGame, 71
 - walls_, 72
- Snake.hpp
 - ALIVE, 89
 - allMaps, 89
 - DEAD, 89
 - direction, 89
 - DOWN, 89
 - LEFT, 89
 - playerState, 89
 - RIGHT, 89
 - STOP, 89
 - UP, 89

- WON, [89](#)
- snake_
 - Snake, [72](#)
- SnakeLib.cpp
 - entryPoint, [110](#)
- snakeSize_
 - Snake, [72](#)
- SPACE
 - IGraphic.hpp, [92](#)
- sprite, [73](#)
 - m_color, [74](#)
 - m_texture, [74](#)
 - pos, [74](#)
 - replacementChar, [74](#)
 - size, [74](#)
- spriteList_
 - SFMLGraphic, [62](#)
- spriteSurface_
 - SDLGraphic, [56](#)
- spriteTexture_
 - SDLGraphic, [56](#)
- src/core/ActionsMenu.cpp, [99](#)
- src/core/ArgsHandler.cpp, [100](#)
- src/core/Core.cpp, [101](#)
- src/core/Menu.cpp, [101](#)
- src/core/SetMenuText.cpp, [102](#)
- src/core/UserNameMenu.cpp, [102](#)
- src/error/Error.cpp, [103](#)
- src/games/nibbler/Compute.cpp, [103](#)
- src/games/nibbler/Display.cpp, [105](#)
- src/games/nibbler/Nibbler.cpp, [107](#)
- src/games/nibbler/NibblerLib.cpp, [108](#)
- src/games/snake/Compute.cpp, [104](#)
- src/games/snake/Display.cpp, [105](#)
- src/games/snake/Snake.cpp, [109](#)
- src/games/snake/SnakeLib.cpp, [110](#)
- src/graphics/ncurses/NcursesEvent.cpp, [111](#)
- src/graphics/ncurses/NcursesGraphic.cpp, [111](#)
- src/graphics/ncurses/NcursesGraphicLib.cpp, [112](#)
- src/graphics/sdl/SDLEvent.cpp, [113](#)
- src/graphics/sdl/SDLGraphic.cpp, [114](#)
- src/graphics/sdl/SDLGraphicLib.cpp, [114](#)
- src/graphics/sfm/SFMLEvent.cpp, [115](#)
- src/graphics/sfm/SFMLGraphic.cpp, [116](#)
- src/graphics/sfm/SFMLGraphicLib.cpp, [116](#)
- src/Main.cpp, [117](#)
- state
 - Nibbler, [49](#)
 - Snake, [72](#)
- STOP
 - Nibbler.hpp, [87](#)
 - Snake.hpp, [89](#)
- SUPR
 - IGraphic.hpp, [92](#)
- T
 - IGraphic.hpp, [91](#)
- TAB
 - IGraphic.hpp, [92](#)
- Test
 - tests_args.cpp, [118](#), [119](#)
 - tests_menu.cpp, [120](#)
 - tests/tests_args.cpp, [117](#)
 - tests/tests_menu.cpp, [119](#)
 - tests_args.cpp
 - Test, [118](#), [119](#)
 - tests_menu.cpp
 - Test, [120](#)
- text, [75](#)
 - fontPath, [75](#)
 - fontSize, [75](#)
 - m_color, [76](#)
 - pos, [76](#)
 - shape, [64](#)
 - text, [76](#)
- textList
 - SFMLGraphic, [62](#)
- textPath_
 - SDLGraphic, [56](#)
 - SFMLGraphic, [62](#)
- texts_
 - Nibbler, [49](#)
 - Snake, [72](#)
- texture, [76](#)
 - path, [77](#)
 - pos, [77](#)
 - size, [77](#)
- texture_
 - SFMLGraphic, [62](#)
- THREE
 - IGraphic.hpp, [92](#)
- timer_
 - Nibbler, [49](#)
 - Snake, [72](#)
- titleMenu_
 - Menu, [36](#)
- TRIANGLE
 - IGraphic.hpp, [92](#)
- TWO
 - IGraphic.hpp, [92](#)
- type
 - shape, [64](#)
- U
 - IGraphic.hpp, [91](#)
- UARROW
 - IGraphic.hpp, [92](#)
- UP
 - Nibbler.hpp, [87](#)
 - Snake.hpp, [89](#)
- updateDirection
 - Nibbler, [47](#)
 - Snake, [70](#)
- updateGame
 - IGame, [24](#)
 - Nibbler, [48](#)
 - Snake, [71](#)
- userName_

- Menu, [36](#)
- UserNameMenu.cpp
 - random, [102](#)
- V
 - IGraphic.hpp, [92](#)
- validLibs
 - Core.hpp, [80](#)
- W
 - IGraphic.hpp, [91](#)
- walls_
 - Nibbler, [49](#)
 - Snake, [72](#)
- what
 - ArcadeError, [9](#)
- width
 - elemSize, [21](#)
- window_
 - NcursesGraphic, [41](#)
 - SDLGraphic, [56](#)
 - SFMLGraphic, [62](#)
- WON
 - Nibbler.hpp, [87](#)
 - Snake.hpp, [89](#)
- X
 - IGraphic.hpp, [91](#)
- x
 - coord, [11](#)
- Y
 - IGraphic.hpp, [91](#)
- y
 - coord, [11](#)
- Z
 - IGraphic.hpp, [91](#)
- ZERO
 - IGraphic.hpp, [92](#)