Build-a-cap

1

Generated by Doxygen 1.9.5

1 Namespace Index 1
1.1 Package List . . . 1

2 Hierarchical Index 3
2.1 Class Hierarchy . . . 3

3 Class Index 7
3.1 Class List . . . 7

4 Namespace Documentation 13
4.1 BehaviourTree Namespace Reference . . . 13

5 Class Documentation 15
5.1 Ability Class Reference . . . 15
5.1.1 Detailed Description . . . 16
5.1.2 Member Function Documentation . . . 16
5.1.2.1 OnBegin() . . . 16
5.1.2.2 OnContinue() . . . 16
5.1.2.3 OnEnd() . . . 17
5.2 AbilityOnCD Class Reference . . . 17
5.2.1 Detailed Description . . . 17
5.2.2 Constructor & Destructor Documentation . . . 17
5.2.2.1 AbilityOnCD() . . . 17
5.2.3 Member Function Documentation . . . 18
5.2.3.1 Check() . . . 18
5.3 AbilitySO Class Reference . . . 18
5.3.1 Detailed Description . . . 19
5.4 ActivateTrapTask Class Reference . . . 19
5.4.1 Detailed Description . . . 19
5.4.2 Constructor & Destructor Documentation . . . 19
5.4.2.1 ActivateTrapTask() . . . 19
5.4.3 Member Function Documentation . . . 20
5.4.3.1 OnBegin() . . . 20
5.4.3.2 OnContinue() . . . 20
5.5 AIGenerator Class Reference . . . 20
5.5.1 Detailed Description . . . 21
5.5.2 Member Function Documentation . . . 21
5.5.2.1 GenerateEnemySpawnPoints() . . . 21
5.5.2.2 Init() . . . 21
5.6 AimRangedAttack Class Reference . . . 22
5.6.1 Detailed Description . . . 22
5.6.2 Constructor & Destructor Documentation . . . 22
5.6.2.1 AimRangedAttack() . . . 22
5.6.3 Member Function Documentation . . . 23

5.6.3.1 OnBegin() . . . 23
5.6.3.2 OnEnd() . . . 23
5.7 AnimationClipGenerator Class Reference . . . 23
5.7.1 Detailed Description . . . 24
5.7.2 Member Function Documentation . . . 24
5.7.2.1 GenerateAllAnimations() . . . 24
5.7.2.2 GenerateAnimations() . . . 24
5.8 AnimationClipProperties Struct Reference . . . 24
5.8.1 Detailed Description . . . 25
5.9 AnimatorGenerator Class Reference . . . 25
5.9.1 Detailed Description . . . 25
5.9.2 Member Function Documentation . . . 25
5.9.2.1 GenerateAnimator() . . . 26
5.10 AnimatorTransitionProperties Struct Reference . . . 26
5.10.1 Detailed Description . . . 26
5.11 Attack Class Reference . . . 27
5.11.1 Detailed Description . . . 27
5.11.2 Member Function Documentation . . . 27
5.11.2.1 OnBegin() . . . 28
5.12 AttackFrames Class Reference . . . 28
5.12.1 Detailed Description . . . 28
5.13 AttackSO Class Reference . . . 28
5.13.1 Detailed Description . . . 29
5.14 AttackStateMachine Class Reference . . . 29
5.14.1 Detailed Description . . . 29
5.15 AttackTarget Class Reference . . . 30
5.15.1 Detailed Description . . . 30
5.15.2 Constructor & Destructor Documentation . . . 30
5.15.2.1 AttackTarget() . . . 30
5.15.3 Member Function Documentation . . . 31
5.15.3.1 OnBegin() . . . 31
5.15.3.2 OnContinue() . . . 31
5.16 AudioManager Class Reference . . . 31
5.16.1 Detailed Description . . . 32
5.17 BridgetBT Class Reference . . . 32
5.17.1 Detailed Description . . . 32
5.17.2 Member Function Documentation . . . 32
5.17.2.1 Init() . . . 32
5.18 Character Class Reference . . . 33
5.18.1 Detailed Description . . . 33
5.18.2 Member Function Documentation . . . 33
5.18.2.1 Init() . . . 33

5.19 CharacterSO Class Reference . . . 34
5.19.1 Detailed Description . . . 34
5.20 CharacterTreeBase Class Reference . . . 34
5.20.1 Detailed Description . . . 36
5.20.2 Member Function Documentation . . . 36
5.20.2.1 GetDashBT() . . . 36
5.21 CheckBase Class Reference . . . 36
5.21.1 Detailed Description . . . 37
5.21.2 Constructor & Destructor Documentation . . . 37
5.21.2.1 CheckBase() . . . 37
5.21.3 Member Function Documentation . . . 37
5.21.3.1 Update() . . . 37
5.22 ClickRangedAttack Class Reference . . . 38
5.22.1 Detailed Description . . . 38
5.22.2 Constructor & Destructor Documentation . . . 38
5.22.2.1 ClickRangedAttack() . . . 38
5.22.3 Member Function Documentation . . . 39
5.22.3.1 OnBegin() . . . 39
5.23 ColorPaletteSO Class Reference . . . 39
5.23.1 Detailed Description . . . 39
5.24 CombatCharacter Class Reference . . . 40
5.24.1 Detailed Description . . . 41
5.24.2 Member Function Documentation . . . 41
5.24.2.1 ActivateTrap() . . . 41
5.24.2.2 Attack() . . . 42
5.24.2.3 AttackTarget() . . . 42
5.24.2.4 Die() . . . 42
5.24.2.5 Init() . . . 43
5.24.2.6 Move() . . . 43
5.24.2.7 Rotate() . . . 43
5.24.2.8 SetMovementSpeed() . . . 43
5.24.2.9 TakeDamage() . . . 44
5.25 BehaviourTree.Composite Class Reference . . . 44
5.25.1 Detailed Description . . . 45
5.26 CupsBT Class Reference . . . 45
5.26.1 Detailed Description . . . 45
5.26.2 Member Function Documentation . . . 45
5.26.2.1 Init() . . . 46
5.27 Dash Class Reference . . . 46
5.27.1 Detailed Description . . . 46
5.27.2 Constructor & Destructor Documentation . . . 46
5.27.2.1 Dash() . . . 46

5.27.3 Member Function Documentation . . . 47
5.27.3.1 OnBegin() . . . 47
5.27.3.2 OnContinue() . . . 47
5.27.3.3 OnEnd() . . . 47
5.28 DashSO Class Reference . . . 48
5.28.1 Detailed Description . . . 48
5.29 DashToTarget Class Reference . . . 48
5.29.1 Detailed Description . . . 49
5.29.2 Constructor & Destructor Documentation . . . 49
5.29.2.1 DashToTarget() . . . 49
5.29.3 Member Function Documentation . . . 49
5.29.3.1 OnBegin() . . . 49
5.29.3.2 OnContinue() . . . 50
5.30 DeathStateMachine Class Reference . . . 50
5.30.1 Detailed Description . . . 50
5.31 BehaviourTree.Decorator Class Reference . . . 50
5.31.1 Detailed Description . . . 51
5.32 DroppedItem Class Reference . . . 51
5.32.1 Detailed Description . . . 51
5.32.2 Member Function Documentation . . . 51
5.32.2.1 Init() [1/2] . . . 51
5.32.2.2 Init() [2/2] . . . 52
5.33 EnemyCharacter Class Reference . . . 52
5.33.1 Detailed Description . . . 53
5.33.2 Member Function Documentation . . . 53
5.33.2.1 Die() . . . 53
5.33.2.2 Freeze() . . . 53
5.34 EnemyCharacterSO Class Reference . . . 53
5.34.1 Detailed Description . . . 54
5.35 EnemySpawner Class Reference . . . 54
5.35.1 Detailed Description . . . 54
5.35.2 Member Function Documentation . . . 54
5.35.2.1 FreezeEnemies() . . . 54
5.35.2.2 Init() . . . 55
5.35.2.3 SpawnEnemy() . . . 55
5.36 FindTargetInRange Class Reference . . . 55
5.36.1 Detailed Description . . . 56
5.36.2 Constructor & Destructor Documentation . . . 56
5.36.2.1 FindTargetInRange() . . . 56
5.36.3 Member Function Documentation . . . 57
5.36.3.1 Check() . . . 57
5.37 FollowTarget Class Reference . . . 57

5.37.1 Detailed Description . . . 57
5.37.2 Member Function Documentation . . . 57
5.37.2.1 Init() . . . 57
5.38 GameManager Class Reference . . . 58
5.38.1 Detailed Description . . . 58
5.38.2 Member Function Documentation . . . 59
5.38.2.1 GameEnd() . . . 59
5.38.2.2 PauseGame() . . . 59
5.39 GFXSetUpWindow Class Reference . . . 59
5.39.1 Detailed Description . . . 60
5.40 MapGenerator.GridTile Struct Reference . . . 60
5.41 Health Class Reference . . . 60
5.41.1 Detailed Description . . . 60
5.42 HealthUI Class Reference . . . 61
5.42.1 Detailed Description . . . 61
5.43 HitBoxController Class Reference . . . 61
5.43.1 Detailed Description . . . 62
5.43.2 Member Function Documentation . . . 62
5.43.2.1 HandleCollision() . . . 62
5.43.2.2 Init() . . . 62
5.44 HUDManager Class Reference . . . 62
5.44.1 Detailed Description . . . 63
5.44.2 Member Function Documentation . . . 63
5.44.2.1 Init() . . . 63
5.45 IDamagable Interface Reference . . . 63
5.45.1 Detailed Description . . . 64
5.45.2 Member Function Documentation . . . 64
5.45.2.1 Die() . . . 64
5.45.2.2 TakeDamage() . . . 64
5.46 Interactable Class Reference . . . 64
5.46.1 Detailed Description . . . 65
5.46.2 Member Function Documentation . . . 65
5.46.2.1 SetTooltip() . . . 65
5.47 BehaviourTree.Inverter Class Reference . . . 65
5.47.1 Detailed Description . . . 66
5.47.2 Member Function Documentation . . . 66
5.47.2.1 Update() . . . 66
5.48 IPushable Interface Reference . . . 67
5.48.1 Detailed Description . . . 67
5.48.2 Member Function Documentation . . . 67
5.48.2.1 Push() . . . 67
5.49 Item Class Reference . . . 68

5.49.1 Detailed Description . . . 68
5.49.2 Constructor & Destructor Documentation . . . 68
5.49.2.1 Item() . . . 68
5.49.3 Member Function Documentation . . . 68
5.49.3.1 ReceiveDamage() . . . 68
5.50 ItemEffect Class Reference . . . 69
5.51 ItemSetUpWindow Class Reference . . . 69
5.51.1 Detailed Description . . . 69
5.52 ItemSO Class Reference . . . 70
5.52.1 Detailed Description . . . 70
5.53 ItemUI Class Reference . . . 70
5.53.1 Detailed Description . . . 71
5.54 JimmyBT Class Reference . . . 71
5.54.1 Detailed Description . . . 71
5.54.2 Member Function Documentation . . . 71
5.54.2.1 Init() . . . 71
5.55 JoeBT Class Reference . . . 72
5.55.1 Detailed Description . . . 72
5.55.2 Member Function Documentation . . . 72
5.55.2.1 Init() . . . 72
5.56 KeepDistanceFromTarget Class Reference . . . 73
5.56.1 Member Function Documentation . . . 73
5.56.1.1 OnBegin() . . . 73
5.56.1.2 OnContinue() . . . 73
5.57 BehaviourTree.Leaf Class Reference . . . 74
5.57.1 Detailed Description . . . 74
5.58 LevelManager Class Reference . . . 75
5.58.1 Detailed Description . . . 75
5.58.2 Member Function Documentation . . . 75
5.58.2.1 FreezeLevel() . . . 75
5.58.2.2 LoadFinalRoom() . . . 76
5.58.2.3 LoadRoom() [1/2] . . . 76
5.58.2.4 LoadRoom() [2/2] . . . 76
5.59 LevelSO Class Reference . . . 77
5.59.1 Detailed Description . . . 77
5.59.2 Member Function Documentation . . . 77
5.59.2.1 GetEnemiesToSpawn() [1/2] . . . 78
5.59.2.2 GetEnemiesToSpawn() [2/2] . . . 78
5.60 MallGenerator Class Reference . . . 78
5.60.1 Member Function Documentation . . . 79
5.60.1.1 Generate() . . . 79
5.60.1.2 SetUpParameters() . . . 79

5.61 MapGenerator Class Reference . . . 79
5.61.1 Detailed Description . . . 81
5.61.2 Member Function Documentation . . . 81
5.61.2.1 Generate() . . . 81
5.61.2.2 GetGrid() . . . 81
5.61.2.3 GetGridTileWorldCoordinates() . . . 81
5.61.2.4 GetGridTileWorldCoordinatesMiddle() . . . 82
5.61.2.5 GetSmallGridTileWorldCoordinates() . . . 82
5.62 MeleeAttack Class Reference . . . 83
5.62.1 Detailed Description . . . 83
5.62.2 Constructor & Destructor Documentation . . . 83
5.62.2.1 MeleeAttack() . . . 83
5.62.3 Member Function Documentation . . . 84
5.62.3.1 OnBegin() . . . 84
5.63 MerchantCharacter Class Reference . . . 84
5.63.1 Detailed Description . . . 84
5.63.2 Member Function Documentation . . . 85
5.63.2.1 Init() . . . 85
5.63.2.2 Sell() . . . 85
5.64 MerchantCharacterSO Class Reference . . . 85
5.64.1 Detailed Description . . . 86
5.65 Money Class Reference . . . 86
5.65.1 Detailed Description . . . 86
5.66 NickBT Class Reference . . . 86
5.66.1 Detailed Description . . . 87
5.66.2 Member Function Documentation . . . 87
5.66.2.1 Init() . . . 87
5.67 NickITBT Class Reference . . . 87
5.67.1 Detailed Description . . . 88
5.67.2 Member Function Documentation . . . 88
5.67.2.1 Init() . . . 88
5.68 BehaviourTree.Node Class Reference . . . 88
5.68.1 Detailed Description . . . 89
5.68.2 Member Function Documentation . . . 89
5.68.2.1 Update() . . . 89
5.69 OfficeRoomGenerator Class Reference . . . 89
5.69.1 Member Function Documentation . . . 90
5.69.1.1 Generate() . . . 90
5.69.1.2 SetUpParameters() . . . 90
5.69.1.3 UnityToScriptCoord() . . . 90
5.70 Pivot Struct Reference . . . 91
5.71 PlayerCharacter Class Reference . . . 91

5.71.1 Detailed Description . . . 92
5.71.2 Member Function Documentation . . . 92
5.71.2.1 Attack() . . . 92
5.71.2.2 CollectItem() . . . 92
5.71.2.3 CollectMoney() . . . 93
5.71.2.4 Init() . . . 93
5.71.2.5 Push() . . . 93
5.71.2.6 TakeDamage() . . . 93
5.72 PlayerCharacterSO Class Reference . . . 94
5.72.1 Detailed Description . . . 94
5.73 PlayerController Class Reference . . . 94
5.73.1 Detailed Description . . . 95
5.74 PlayerInput Class Reference . . . 95
5.74.1 Detailed Description . . . 96
5.75 PlayerInventory Class Reference . . . 96
5.75.1 Detailed Description . . . 96
5.75.2 Member Function Documentation . . . 96
5.75.2.1 AddItem() . . . 96
5.75.2.2 HasSpace() . . . 97
5.75.2.3 ItemEquipped() . . . 97
5.75.2.4 ReceiveDamage() . . . 97
5.76 PlayerSpawner Class Reference . . . 98
5.76.1 Member Function Documentation . . . 98
5.76.1.1 PlacePlayer() . . . 98
5.76.1.2 SpawnPlayerAndInit() . . . 99
5.77 PottsBT Class Reference . . . 99
5.77.1 Detailed Description . . . 100
5.77.2 Member Function Documentation . . . 100
5.77.2.1 Init() . . . 100
5.78 ProjectileController Class Reference . . . 100
5.78.1 Detailed Description . . . 101
5.78.2 Member Function Documentation . . . 101
5.78.2.1 Init() . . . 101
5.78.2.2 Shoot() . . . 101
5.78.2.3 ShootAt() . . . 101
5.79 RangedAttack Class Reference . . . 102
5.79.1 Detailed Description . . . 102
5.80 BehaviourTree.Repeater Class Reference . . . 102
5.80.1 Detailed Description . . . 103
5.80.2 Member Function Documentation . . . 103
5.80.2.1 Update() . . . 103
5.81 Resource Class Reference . . . 103

5.81.1 Detailed Description . . . 104
5.81.2 Member Function Documentation . . . 104
5.81.2.1 ChangeCurrent() . . . 104
5.82 ResourceUI Class Reference . . . 105
5.82.1 Detailed Description . . . 105
5.82.2 Member Function Documentation . . . 105
5.82.2.1 Init() . . . 105
5.83 Respect Class Reference . . . 105
5.83.1 Detailed Description . . . 106
5.84 BehaviourTree.Selector Class Reference . . . 106
5.84.1 Detailed Description . . . 107
5.84.2 Member Function Documentation . . . 107
5.84.2.1 Update() . . . 107
5.85 BehaviourTree.Sequence Class Reference . . . 107
5.85.1 Detailed Description . . . 108
5.85.2 Member Function Documentation . . . 108
5.85.2.1 Update() . . . 108
5.86 BehaviourTree.SequenceWithCachedLastChild Class Reference . . . 108
5.86.1 Detailed Description . . . 109
5.86.2 Member Function Documentation . . . 109
5.86.2.1 Update() . . . 109
5.87 ShopItemUI Class Reference . . . 109
5.87.1 Detailed Description . . . 110
5.87.2 Member Function Documentation . . . 110
5.87.2.1 UpdateItem() . . . 110
5.88 ShopManager Class Reference . . . 110
5.88.1 Detailed Description . . . 110
5.88.2 Member Function Documentation . . . 111
5.88.2.1 ShowShop() . . . 111
5.89 SmithBT Class Reference . . . 111
5.89.1 Detailed Description . . . 111
5.89.2 Member Function Documentation . . . 112
5.89.2.1 Init() . . . 112
5.90 SpecialRoom Class Reference . . . 112
5.91 SpriteImportSettings Struct Reference . . . 112
5.92 SpriteOrganizer Class Reference . . . 113
5.92.1 Detailed Description . . . 113
5.92.2 Member Function Documentation . . . 113
5.92.2.1 Delete() . . . 113
5.92.2.2 Rename() . . . 113
5.93 SpriteSetUp Class Reference . . . 114
5.93.1 Detailed Description . . . 114

5.93.2 Member Function Documentation . . . 114
5.93.2.1 SetSpriteImportSettings() . . . 114
5.94 StreetGenerator Class Reference . . . 114
5.94.1 Member Function Documentation . . . 115
5.94.1.1 Generate() . . . 115
5.94.1.2 SetUpParameters() . . . 116
5.95 BehaviourTree.Succeeder Class Reference . . . 116
5.95.1 Detailed Description . . . 116
5.95.2 Member Function Documentation . . . 116
5.95.2.1 Update() . . . 117
5.96 TableHandler Class Reference . . . 117
5.97 TargetInRange Class Reference . . . 118
5.97.1 Detailed Description . . . 118
5.97.2 Constructor & Destructor Documentation . . . 118
5.97.2.1 TargetInRange() . . . 118
5.97.3 Member Function Documentation . . . 119
5.97.3.1 Check() . . . 119
5.98 TaskBase Class Reference . . . 119
5.98.1 Detailed Description . . . 120
5.98.2 Constructor & Destructor Documentation . . . 120
5.98.2.1 TaskBase() . . . 120
5.98.3 Member Function Documentation . . . 120
5.98.3.1 OnEnd() . . . 120
5.98.3.2 Update() . . . 121
5.99 TooltipUI Class Reference . . . 121
5.99.1 Detailed Description . . . 121
5.99.2 Member Function Documentation . . . 121
5.99.2.1 SetText() . . . 121
5.99.2.2 ShowToolTip() . . . 122
5.100 Trap Class Reference . . . 122
5.100.1 Detailed Description . . . 123
5.100.2 Constructor & Destructor Documentation . . . 123
5.100.2.1 Trap() . . . 123
5.100.3 Member Function Documentation . . . 123
5.100.3.1 OnBegin() . . . 123
5.100.3.2 OnEnd() . . . 123
5.101 TrapController Class Reference . . . 124
5.101.1 Detailed Description . . . 124
5.101.2 Member Function Documentation . . . 124
5.101.2.1 ActivateTrap() . . . 124
5.101.2.2 Init() . . . 124
5.102 TrapDamager Class Reference . . . 125

5.102.1 Detailed Description . . . 125
5.102.2 Member Function Documentation . . . 125
5.102.2.1 Init() . . . 125
5.103 TrapSO Class Reference . . . 126
5.103.1 Detailed Description . . . 126
5.104 TrapTrigger Class Reference . . . 126
5.104.1 Detailed Description . . . 127
5.105 BehaviourTree.TreeBase Class Reference . . . 127
5.105.1 Detailed Description . . . 128
5.105.2 Member Function Documentation . . . 128
5.105.2.1 Init() . . . 128
5.106 UIInput Class Reference . . . 128
5.106.1 Detailed Description . . . 129
5.107 UIOverlayManager Class Reference . . . 129
5.107.1 Detailed Description . . . 129
5.107.2 Member Function Documentation . . . 129
5.107.2.1 ChangeTitle() . . . 129
5.108 WaitFor Class Reference . . . 130
5.108.1 Detailed Description . . . 130
5.108.2 Constructor & Destructor Documentation . . . 130
5.108.2.1 WaitFor() . . . 130
5.108.3 Member Function Documentation . . . 131
5.108.3.1 OnBegin() . . . 131
5.108.3.2 OnContinue() . . . 131
5.109 WalkToTarget Class Reference . . . 131
5.109.1 Detailed Description . . . 132
5.109.2 Member Function Documentation . . . 132
5.109.2.1 OnBegin() . . . 132
5.109.2.2 OnContinue() . . . 132

Index **133**

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

BehaviourTree 13

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

- Ability . . . 15
  - Attack . . . 27
    - MeleeAttack . . . 83
    - RangedAttack . . . 102
      - AimRangedAttack . . . 22
      - ClickRangedAttack . . . 38
  - Dash . . . 46
  - Trap . . . 122
- AnimationClipGenerator . . . 23
- AnimationClipProperties . . . 24
- AnimatorGenerator . . . 25
- AnimatorTransitionProperties . . . 26
- AttackFrames . . . 28
- EditorWindow
  - GFXSetUpWindow . . . 59
  - ItemSetUpWindow . . . 69
- MapGenerator.GridTile . . . 60
- IDamagable . . . 63
  - CombatCharacter . . . 40
    - EnemyCharacter . . . 52
    - PlayerCharacter . . . 91
- IPushable . . . 67
  - PlayerCharacter . . . 91
- Item . . . 68
- ItemEffect . . . 69
- MonoBehaviour
  - AIGenerator . . . 20
  - AudioManager . . . 31
  - BehaviourTree.TreeBase . . . 127
    - CharacterTreeBase . . . 34
      - BridgetBT . . . 32
      - CupsBT . . . 45
      - JimmyBT . . . 71
      - JoeBT . . . 72

NickBT . . . 86
NickITBT . . . 87
PottsBT . . . 99
SmithBT . . . 111
Character . . . 33
CombatCharacter . . . 40
MerchantCharacter . . . 84
DroppedItem . . . 51
EnemySpawner . . . 54
FollowTarget . . . 57
GameManager . . . 58
HUDManager . . . 62
HealthUI . . . 61
HitBoxController . . . 61
Interactable . . . 64
ItemUI . . . 70
LevelManager . . . 75
MapGenerator . . . 79
MallGenerator . . . 78
OfficeRoomGenerator . . . 89
StreetGenerator . . . 114
PlayerController . . . 94
PlayerInput . . . 95
PlayerSpawner . . . 98
ProjectileController . . . 100
ResourceUI . . . 105
ShopItemUI . . . 109
ShopManager . . . 110
SpecialRoom . . . 112
TableHandler . . . 117
TooltipUI . . . 121
TrapController . . . 124
TrapDamager . . . 125
TrapTrigger . . . 126
UIInput . . . 128
UIOverlayManager . . . 129
BehaviourTree.Node . . . 88
BehaviourTree.Composite . . . 44
BehaviourTree.Selector . . . 106
BehaviourTree.Sequence . . . 107
BehaviourTree.SequenceWithCachedLastChild . . . 108
BehaviourTree.Decorator . . . 50
BehaviourTree.Inverter . . . 65
BehaviourTree.Repeater . . . 102
BehaviourTree.Succeeder . . . 116
BehaviourTree.Leaf . . . 74
CheckBase . . . 36
AbilityOnCD . . . 17
FindTargetInRange . . . 55
TargetInRange . . . 118
TaskBase . . . 119
ActivateTrapTask . . . 19
AttackTarget . . . 30
DashToTarget . . . 48
KeepDistanceFromTarget . . . 73
WaitFor . . . 130
WalkToTarget . . . 131

Pivot . . . 91
PlayerInventory . . . 96
Resource . . . 103
Health . . . 60
Money . . . 86
Respect . . . 105
ScriptableObject
AbilitySO . . . 18
AttackSO . . . 28
DashSO . . . 48
TrapSO . . . 126
CharacterSO . . . 34
EnemyCharacterSO . . . 53
MerchantCharacterSO . . . 85
PlayerCharacterSO . . . 94
ColorPaletteSO . . . 39
ItemSO . . . 70
LevelSO . . . 77
SpriteImportSettings . . . 112
SpriteOrganizer . . . 113
SpriteSetUp . . . 114
StateMachineBehaviour
AttackStateMachine . . . 29
DeathStateMachine . . . 50

Chapter 3

Class Index

3.1 Class List

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

Ability
: Wrapper class for all characters' abilities. . . . 15

AbilityOnCD
: Checks whether an ability is on cool down. . . . 17

AbilitySO
: Wrapper class for data of all abilities. . . . 18

ActivateTrapTask
: A task which activates a trap . . . 19

AIGenerator
: Handles all AI related procedural content generation. . . . 20

AimRangedAttack
: Class for ranged attacks aimed by the player. . . . 22

AnimationClipGenerator
: Generator for animation clips. The clips are placed in a hierarchy according to file names. Expects .png files on input, with the following naming convention: {character-name}_{animation-name}_{direction}_{frame-number}.png . . . 23

AnimationClipProperties
: A struct for custom animation clip properties. . . . 24

AnimatorGenerator
: Generates an animator controller for one character. Expects animation clips to already be present in the folder specified by the character name. . . . 25

AnimatorTransitionProperties
: Struct for storing animator transition properties. . . . 26

Attack
: Base wrapper class for all possible attacks. . . . 27

AttackFrames
: Specifies the frame count per each attack phase - startup, active, recovery. (So far) only utilized for animation clip generation - depending on the type of attack, some of the counts will always be zero. . . . 28

AttackSO
: Attack data class. . . . 28

AttackStateMachine
: State machine for attack states of all characters. . . . 29

AttackTarget
: Performs the given attack in the direction of the target set in shared data. . . . 30

AudioManager
Manages the ambience and the UI sounds in the game. . . . 31
BridgetBT
Behavioral tree for the boss of the last level. . . . 32
Character
Base class for all in-game characters. . . . 33
CharacterSO
Character data class. . . . 34
CharacterTreeBase
Base class for behavior trees of all characters in the game. . . . 34
CheckBase
Base class for all checks. Checks only check whether this branch is valid. . . . 36
ClickRangedAttack
Class for a basic ranged attack. . . . 38
ColorPaletteSO
Game color palette data. . . . 39
CombatCharacter
Base class for all aggressive characters - i.e. characters that can engage in combat. Handles character movement and combat - both animation and physics. . . . 40
BehaviourTree.Composite
Base class for all behavior tree composite nodes. Composite nodes have one or more children which are processed either in a first to last sequence or in random order. . . . 44
CupsBT
Behavioral tree for a mob from the first level. . . . 45
Dash
Class for the dash ability. . . . 46
DashSO
Dash data class. . . . 48
DashToTarget
Performs a dash in the direction of the target. . . . 48
DeathStateMachine
State machine for death state of all characters. . . . 50
BehaviourTree.Decorator
Base class for all behavior tree decorator nodes. Decorator nodes have exactly one child. . . . 50
DroppedItem
Handles items dropped by the enemy characters and their collection by the player. . . . 51
EnemyCharacter
Class for all enemy characters. . . . 52
EnemyCharacterSO
Enemy character data class. . . . 53
EnemySpawner
Responsible for spawning and managing enemy characters in the scene. . . . 54
FindTargetInRange
A task which tries to find a target (typically the player) and save it. . . . 55
FollowTarget
Makes this game object follow a specified target. . . . 57
GameManager
The game manager - main entry point of the game. . . . 58
GFXSetUpWindow
Editor window for all graphics' set-up. . . . 59
MapGenerator.GridTile . . . 60
Health
Wrapper class for the health resource. Represents the current and max health for all characters that can engage in combat. . . . 60
HealthUI
Health bar UI manager class. . . . 61
HitBoxController
Controls the hitbox for all melee attacks. . . . 61

HUDManager
62
IDamagable
Any damageable entity should implement this. . . . 63
Interactable
Class for game objects with which the player can interact by pressing the interact button. . . . 64
BehaviourTree.Inverter
A decorator node which inverts the result of the child. . . . 65
IPushable
Any pushable entity should implement this. . . . 67
Item
Only for equippable items. . . . 68
ItemEffect . . . 69
ItemSetUpWindow
An editor window for setting up the in-game items. . . . 69
ItemSO
Item data class. . . . 70
ItemUI
Manager class for UI of the items currently equipped by the player. . . . 70
JimmyBT
Behavioral tree for the second level boss. . . . 71
JoeBT
Behavioral tree for a mob from the second level. . . . 72
KeepDistanceFromTarget . . . 73
BehaviourTree.Leaf
Leaf node base class. Leaves have no children and their function is to perform character specific actions or checks. . . . 74
LevelManager
Manager for one game level (each level should have its own LevelManager). Last level (the mall roof) probably won't be able to make use of this -> special prefab instead. . . . 75
LevelSO
Level data class. . . . 77
MallGenerator . . . 78
MapGenerator
Base class for map generation . . . 79
MeleeAttack
Base wrapper class for all melee attacks. . . . 83
MerchantCharacter
Base class for the merchant. . . . 84
MerchantCharacterSO
Merchant character data class. . . . 85
Money
Wrapper class for the money resource. For the player and merchant NPC it should represent their money amount, for aggressive NPCs the amount of money they drop. . . . 86
NickBT
Behavioral tree for a mob from the first level. . . . 86
NickITBT
Behavioral tree for a mob from the first level. . . . 87
BehaviourTree.Node
Base class for all behavior tree nodes. . . . 88
OfficeRoomGenerator . . . 89
Pivot . . . 91
PlayerCharacter
Handles player movement and combat - both animation and physics. . . . 91
PlayerCharacterSO
Player character data class. . . . 94

PlayerController
Connects the player input and player character, issuing commands to the character as a reaction to player input. . . . 94
PlayerInput
Handles player input which pertains to the controlling of the player character. . . . 95
PlayerInventory
Player inventory wrapper class. The player inventory equals the items the player has currently equipped. . . . 96
PlayerSpawner . . . 98
PottsBT
Behavioral tree for a mob from the first level. . . . 99
ProjectileController
Manages the shooting of projectiles for all ranged attacks. . . . 100
RangedAttack
Base wrapper class for all ranged attacks. . . . 102
BehaviourTree.Repeater
A decorator node which repeatedly processes its child after it returns a result (meaning either a success or a failure). Example usage: at the top of the tree to make the tree run continuously. . 102
Resource
Base wrapper class for all resources. . . . 103
ResourceUI
Handles the UI of any resource. Expects either a Slider or a Text component (or both) on one of this object's children. . . . 105
Respect
Wrapper class for the respect resource. Exclusive to the player, represents the current respect value. . . . 105
BehaviourTree.Selector
A composite node which will process all its children in order - when a child reports success it reports success and does not process any further children. Equivalent to an OR operation - returns success if any of its children returned success. . . . 106
BehaviourTree.Sequence
A composite node which will process all its children in order - it proceeds to the next child when the previous one reports success. Equivalent to an AND operation - returns success only if all of its children returned success. . . . 107
BehaviourTree.SequenceWithCachedLastChild
A composite node which will process all its children in order - it proceeds to the next child when the previous one reports success. Equivalent to an AND operation - returns success only if all of its children returned success. When entering the sequence proceeds from the child processed in the previous tick - useful for a sequence of tasks that need to be completed one after another. 108
ShopItemUI
Manager for the UI of items in the shop. . . . 109
ShopManager
Class that manages the shop UI and connected events. . . . 110
SmithBT
Behavioral tree for the boss of the first level. . . . 111
SpecialRoom . . . 112
SpriteImportSettings . . . 112
SpriteOrganizer
Used for general sprite organization. . . . 113
SpriteSetUp
Class which handles setting up the sprites for use in the game. . . . 114
StreetGenerator . . . 114
BehaviourTree.Succeeder
A decorator node which always returns success. . . . 116
TableHandler . . . 117
TargetInRange
Checks whether target is within specified range. . . . 118

TaskBase
Base class for all tasks. Tasks update the character's animation, physics, etc. 119
TooltipUI
Manager for UI tooltips. 121
Trap
Class for the trap ability. 122
TrapController
Main controller for every trap in the scene. 124
TrapDamager
Handles the behaviour when the trap is triggered, ex. whether it should deal damage. 125
TrapSO
Trap data class. 126
TrapTrigger
Triggers the trap this is a child of when the player enters/stands in this collider. 126
BehaviourTree.TreeBase
Base class for all behaviour trees. 127
UIInput
Class for input which should be independent of the player character. 128
UIOverlayManager
Class for any UI overlay with buttons relating to high-level game logic - ex. menus, game over screen. 129
WaitFor
A task which makes the character wait for a specified time. 130
WalkToTarget
A task which makes the character walk towards a given target. 131

Chapter 4

Namespace Documentation

4.1 BehaviourTree Namespace Reference

Classes

- class Composite

  Base class for all behavior tree composite nodes. Composite nodes have one or more children which are processed either in a first to last sequence or in random order.
- class Decorator

  Base class for all behavior tree decorator nodes. Decorator nodes have exactly one child.
- class Inverter

  A decorator node which inverts the result of the child.
- class Leaf

  Leaf node base class. Leaves have no children and their function is to perform character specific actions or checks.
- class Node

  Base class for all behavior tree nodes.
- class Repeater

  A decorator node which repeatedly processes its child after it returns a result (meaning either a success or a failure). Example usage: at the top of the tree to make the tree run continuously.
- class Selector

  A composite node which will process all its children in order - when a child reports success it reports success and does not process any further children. Equivalent to an OR operation - returns success if any of its children returned success.
- class Sequence

  A composite node which will process all its children in order - it proceeds to the next child when the previous one reports success. Equivalent to an AND operation - returns success only if all of its children returned success.
- class SequenceWithCachedLastChild

  A composite node which will process all its children in order - it proceeds to the next child when the previous one reports success. Equivalent to an AND operation - returns success only if all of its children returned success. When entering the sequence proceeds from the child processed in the previous tick - useful for a sequence of tasks that need to be completed one after another.
- class Succeeder

  A decorator node which always returns success.
- class TreeBase

  Base class for all behaviour trees.

Enumerations

- enum NodeStatus

  Enum which describes the node status. Success and Failure are results, while a Running status means the node has not yet determined whether it will return a Success or a Failure.

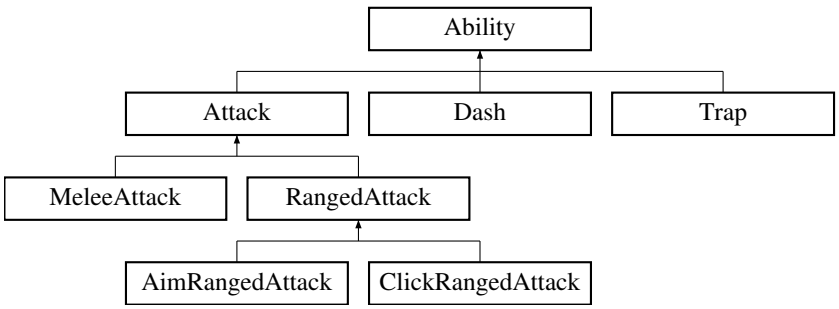
Chapter 5

Class Documentation

5.1 Ability Class Reference

Wrapper class for all characters' abilities.

Inheritance diagram for Ability:



Public Member Functions

- virtual void OnBegin ()

  Begins performing the ability.
- virtual IEnumerator OnContinue ()

  Updates the performing of the ability if the ability is in use.
- virtual void OnEnd ()

  Ends the performing of the ability.

Public Attributes

- EAbilityType **type**

Protected Member Functions

- **Ability** (CombatCharacter character, AbilitySO data, EAbilityType type)

Protected Attributes

- CombatCharacter **character**
- AbilitySO **data**
- float **_lastUsed**

Properties

- bool **InUse** `[get, set]`
- bool **OnCoolDown** `[get]`

5.1.1 Detailed Description

Wrapper class for all characters' abilities.

5.1.2 Member Function Documentation

5.1.2.1 OnBegin()

```
virtual void Ability.OnBegin ( )  [virtual]
```

Begins performing the ability.

Reimplemented in AimRangedAttack, Attack, ClickRangedAttack, Dash, MeleeAttack, and Trap.

5.1.2.2 OnContinue()

```
virtual IEnumerator Ability.OnContinue ( )  [virtual]
```

Updates the performing of the ability if the ability is in use.

Returns

Reimplemented in Dash.

5.1.2.3 OnEnd()

```
virtual void Ability.OnEnd ( )  [virtual]
```

Ends the performing of the ability.

Reimplemented in AimRangedAttack, Dash, and Trap.

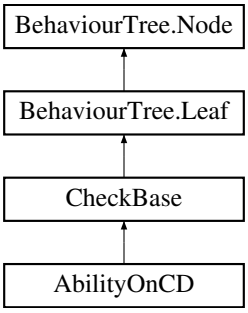
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/Ability.cs

5.2 AbilityOnCD Class Reference

Checks whether an ability is on cool down.

Inheritance diagram for AbilityOnCD:



Public Member Functions

- AbilityOnCD (CharacterTreeBase characterBT, Ability ability, string debugName="")
  Creates a task instance.

Protected Member Functions

- override bool Check ()

Additional Inherited Members

5.2.1 Detailed Description

Checks whether an ability is on cool down.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 AbilityOnCD()

```
AbilityOnCD.AbilityOnCD (
            CharacterTreeBase characterBT,
            Ability ability,
            string debugName = "" )
```

Creates a task instance.

Parameters

characterBT	The behavioral tree of this character
ability	The ability to check
debugName	

5.2.3 Member Function Documentation

5.2.3.1 Check()

```
override bool AbilityOnCD.Check ( )  [protected], [virtual]
```

Implements CheckBase.

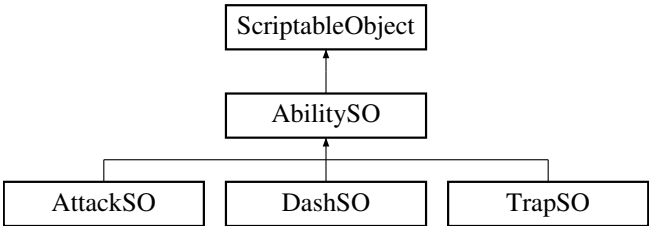
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/AbilityOnCD.cs

5.3 AbilitySO Class Reference

Wrapper class for data of all abilities.

Inheritance diagram for AbilitySO:



Public Attributes

- float **coolDown**
- int **damage** = 0
- float **enemyPushbackDistance** = 0
- float **enemyPushbackSpeed** = 40
- FMODUnity.EventReference **onBeginSound**
- FMODUnity.EventReference **onEndSound**

5.3.1 Detailed Description

Wrapper class for data of all abilities.

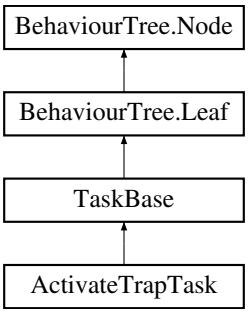
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/AbilitySO.cs

5.4 ActivateTrapTask Class Reference

A task which activates a trap

Inheritance diagram for ActivateTrapTask:



Public Member Functions

- ActivateTrapTask (CharacterTreeBase characterBT, Trap trap, string debugName="")

Protected Member Functions

- override void OnBegin ()
- override void OnContinue ()

Additional Inherited Members

5.4.1 Detailed Description

A task which activates a trap

5.4.2 Constructor & Destructor Documentation

5.4.2.1 ActivateTrapTask()

```
ActivateTrapTask.ActivateTrapTask (
            CharacterTreeBase characterBT,
            Trap trap,
            string debugName = "" )
```

Parameters

characterBT	The behavioral tree of this character
trap	The trap to activate
debugName	

5.4.3 Member Function Documentation

5.4.3.1 OnBegin()

```
override void ActivateTrapTask.OnBegin ( )  [protected], [virtual]
```

Implements TaskBase.

5.4.3.2 OnContinue()

```
override void ActivateTrapTask.OnContinue ( )  [protected], [virtual]
```

Implements TaskBase.

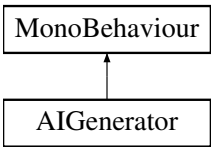
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/ActivateTrapTask.cs

5.5 AIGenerator Class Reference

Handles all AI related procedural content generation.

Inheritance diagram for AIGenerator:



Public Member Functions

- void **ClearGraph** ()

  Removes the current navmesh if present.
- void Init (MapGenerator roomGenerator)

  Initializes the AI generator.
- void **GenerateGraph** ()

  Generates a navigation point graph to be used by the astar path.
- Vector3[ ] GenerateEnemySpawnPoints (int numEnemies)

  Randomly generates enemy spawn points in the current room.
- void **CleanUp** (UnityAction onEnd)

  Removes the current astar path from the scene - should be called before instantiating a prefab which already has an astar path component.

5.5.1 Detailed Description

Handles all AI related procedural content generation.

5.5.2 Member Function Documentation

5.5.2.1 GenerateEnemySpawnPoints()

```
Vector3[] AIGenerator.GenerateEnemySpawnPoints (
            int numEnemies )
```

Randomly generates enemy spawn points in the current room.

Parameters

numEnemies	The number of spawn points to generate

Returns

The generated spawn points

5.5.2.2 Init()

```
void AIGenerator.Init (
            MapGenerator roomGenerator )
```

Initializes the AI generator.

Parameters

roomGenerator	Procedural content generator which generates a room

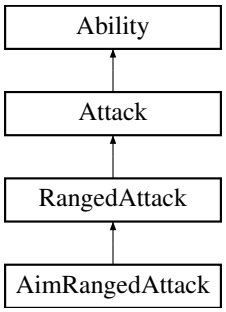
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/AIGenerator.cs

5.6 AimRangedAttack Class Reference

Class for ranged attacks aimed by the player.

Inheritance diagram for AimRangedAttack:



Public Member Functions

- AimRangedAttack (CombatCharacter character, AttackSO data)

  Creates a new aimed ranged attack instance - a ranged attack the player can aim with and which shoots the projectile on button release.
- override void OnBegin ()

  Begins performing the ability.
- override void OnEnd ()

  Ends the performing of the ability.

Additional Inherited Members

5.6.1 Detailed Description

Class for ranged attacks aimed by the player.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 AimRangedAttack()

```
AimRangedAttack.AimRangedAttack (
            CombatCharacter character,
            AttackSO data )
```

Creates a new aimed ranged attack instance - a ranged attack the player can aim with and which shoots the projectile on button release.

Parameters

character	The character to which the attack belongs
data	The attack data

5.6.3 Member Function Documentation

5.6.3.1 OnBegin()

```
override void AimRangedAttack.OnBegin ( )  [virtual]
```

Begins performing the ability.

Reimplemented from Ability.

5.6.3.2 OnEnd()

```
override void AimRangedAttack.OnEnd ( )  [virtual]
```

Ends the performing of the ability.

Reimplemented from Ability.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/AimRangedAttack.cs

5.7 AnimationClipGenerator Class Reference

Generator for animation clips. The clips are placed in a hierarchy according to file names. Expects .png files on input, with the following naming convention: {character-name}_{animation-name}_{direction}_{frame-number}.png

Public Member Functions

- **AnimationClipGenerator** (string characterName)
- **AnimationClipGenerator** (string characterName, AnimationClipProperties animationProperties)
- int GenerateAnimations (EAbilityType ability, AttackFrames attackFrames=null)

  Generates animation clips from sprites for the given ability for the given character.
- int GenerateAllAnimations ()

  Generates animation clips from sprites for the given character.

5.7.1 Detailed Description

Generator for animation clips. The clips are placed in a hierarchy according to file names. Expects .png files on input, with the following naming convention: {character-name}_{animation-name}_{direction}_{frame-number}.png

5.7.2 Member Function Documentation

5.7.2.1 GenerateAllAnimations()

```
int AnimationClipGenerator.GenerateAllAnimations ( )
```

Generates animation clips from sprites for the given character.

Returns

Number of clips generated.

5.7.2.2 GenerateAnimations()

```
int AnimationClipGenerator.GenerateAnimations (
            EAbilityType ability,
            AttackFrames attackFrames = null )
```

Generates animation clips from sprites for the given ability for the given character.

Parameters

characterName	

Returns

Number of clips generated.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/AnimationClipGenerator.cs

5.8 AnimationClipProperties Struct Reference

A struct for custom animation clip properties.

Public Member Functions

- **AnimationClipProperties** (float frameRate, bool loop, Color spriteColor, bool duplicateSingleFrame, string characterName, int meleeHitBoxOnFrame=-1)

Public Attributes

- float **frameRate**
- bool **loop**
- Color **spriteColor**
- bool **duplicateSingleFrame**
- string **characterName**
- int **meleeHitBoxOnFrame**

5.8.1 Detailed Description

A struct for custom animation clip properties.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/AnimationClipGenerator.cs

5.9 AnimatorGenerator Class Reference

Generates an animator controller for one character. Expects animation clips to already be present in the folder specified by the character name.

Public Member Functions

- **AnimatorGenerator** (string characterName)
- void GenerateAnimator ()

  Generates an animator controller for the given character. The state machine logic is hardcoded here.

5.9.1 Detailed Description

Generates an animator controller for one character. Expects animation clips to already be present in the folder specified by the character name.

5.9.2 Member Function Documentation

5.9.2.1 GenerateAnimator()

```
void AnimatorGenerator.GenerateAnimator ( )
```

Generates an animator controller for the given character. The state machine logic is hardcoded here.

Returns

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/AnimatorGenerator.cs

5.10 AnimatorTransitionProperties Struct Reference

Struct for storing animator transition properties.

Public Member Functions

- **AnimatorTransitionProperties** (AnimatorConditionMode condition, float conditionThreshold, string conditionParameter, bool hasExit=false, float exitTime=1, float duration=0)

Public Attributes

- AnimatorConditionMode **condition**
- float **conditionThreshold**
- string **conditionParameter**
- bool **hasExit**
- float **exitTime**
- float **duration**

5.10.1 Detailed Description

Struct for storing animator transition properties.

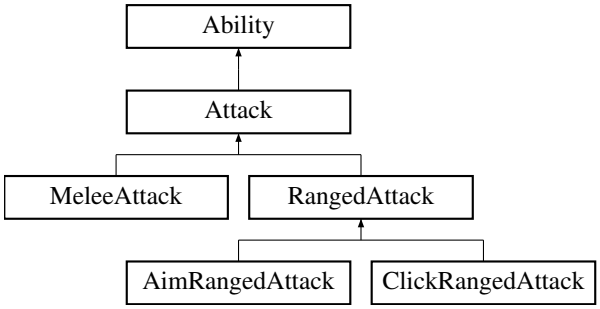
The documentation for this struct was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/AnimatorGenerator.cs

5.11 Attack Class Reference

Base wrapper class for all possible attacks.

Inheritance diagram for Attack:



Public Member Functions

- override void OnBegin ()

  Begins performing the ability.

Protected Member Functions

- **Attack** (CombatCharacter character, AttackSO data, EAbilityType type)

Properties

- Vector3? **Target** `[get, set]`
- AttackSO **Data** `[get]`

Additional Inherited Members

5.11.1 Detailed Description

Base wrapper class for all possible attacks.

5.11.2 Member Function Documentation

5.11.2.1 OnBegin()

```
override void Attack.OnBegin ( )  [virtual]
```

Begins performing the ability.

Reimplemented from Ability.

Reimplemented in ClickRangedAttack, and MeleeAttack.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/Attack.cs

5.12 AttackFrames Class Reference

Specifies the frame count per each attack phase - startup, active, recovery. (So far) only utilized for animation clip generation - depending on the type of attack, some of the counts will always be zero.

Public Member Functions

- **AttackFrames** (EAttackEffect attackEffect=EAttackEffect.Click, int startup=0, int active=0, int recovery=0)
- Tuple< int, int > **GetStartupIndexes** ()
- Tuple< int, int > **GetActiveIndexes** ()
- Tuple< int, int > **GetRecoveryIndexes** ()

Properties

- EAttackEffect **AttackEffect** `[get]`

5.12.1 Detailed Description

Specifies the frame count per each attack phase - startup, active, recovery. (So far) only utilized for animation clip generation - depending on the type of attack, some of the counts will always be zero.

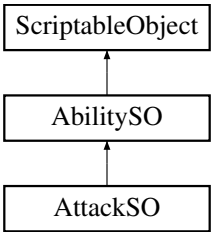
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/General/Utility.cs

5.13 AttackSO Class Reference

Attack data class.

Inheritance diagram for AttackSO:



Public Member Functions

- Attack **GetAttack** (CombatCharacter character)

Public Attributes

- string **attackName**
- int **id**
- EAttackButton **button**
- EAttackEffect **effect**
- int **cost**
- float **recoveryTime** = 1f
- float **attackRange** = 3
- float **movementSpeedFactor** = 0
- ProjectileController **projectilePrefab**
- float **projectileSpeed** = 100
- float **projectileDelay**

5.13.1 Detailed Description

Attack data class.

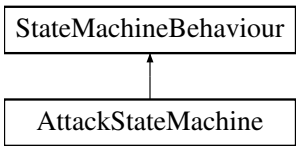
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/AttackSO.cs

5.14 AttackStateMachine Class Reference

State machine for attack states of all characters.

Inheritance diagram for AttackStateMachine:



Public Member Functions

- override void **OnStateEnter** (Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
- override void **OnStateExit** (Animator animator, AnimatorStateInfo stateinfo, int layerindex)

5.14.1 Detailed Description

State machine for attack states of all characters.

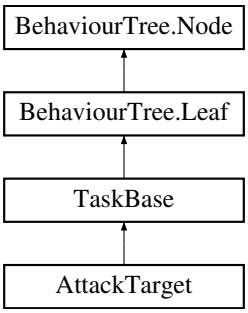
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/StateMachineBehaviors/AttackStateMachine.cs

5.15 AttackTarget Class Reference

Performs the given attack in the direction of the target set in shared data.

Inheritance diagram for AttackTarget:



Public Member Functions

- AttackTarget (CharacterTreeBase characterBT, Attack attack, string targetName, bool precise=false, string debugName="")

Protected Member Functions

- override void OnBegin ()
- override void OnContinue ()

Additional Inherited Members

5.15.1 Detailed Description

Performs the given attack in the direction of the target set in shared data.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 AttackTarget()

```
AttackTarget.AttackTarget (
            CharacterTreeBase characterBT,
            Attack attack,
            string targetName,
            bool precise = false,
            string debugName = "" )
```

Parameters

characterBT	The behavioral tree of this character
attack	The attack to perform
targetName	The name of the target - this task will try to retrieve the target from shared memory.
precise	Should we attack the target specifically or attack in the direction of the target
debugName	

5.15.3 Member Function Documentation

5.15.3.1 OnBegin()

```
override void AttackTarget.OnBegin ( )  [protected], [virtual]
```

Implements TaskBase.

5.15.3.2 OnContinue()

```
override void AttackTarget.OnContinue ( )  [protected], [virtual]
```

Implements TaskBase.

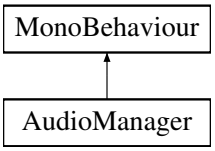
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/AttackTarget.cs

5.16 AudioManager Class Reference

Manages the ambience and the UI sounds in the game.

Inheritance diagram for AudioManager:



Public Member Functions

- void **Refresh** ()
- void **StartAmbience** (int sceneIndex)
- void **PlayButtonClick** ()
- void **PlaySound** (FMOD.GUID id)

5.16.1 Detailed Description

Manages the ambience and the UI sounds in the game.

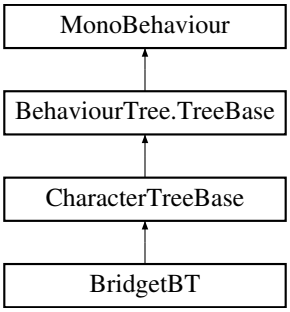
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/General/AudioManager.cs

5.17 BridgetBT Class Reference

Behavioral tree for the boss of the last level.

Inheritance diagram for BridgetBT:



Protected Member Functions

- override void Init ()

  Initiate every tree in this method - called from Start.

Additional Inherited Members

5.17.1 Detailed Description

Behavioral tree for the boss of the last level.

5.17.2 Member Function Documentation

5.17.2.1 Init()

```
override void BridgetBT.Init ( )  [protected], [virtual]
```

Initiate every tree in this method - called from Start.

Implements BehaviourTree.TreeBase.

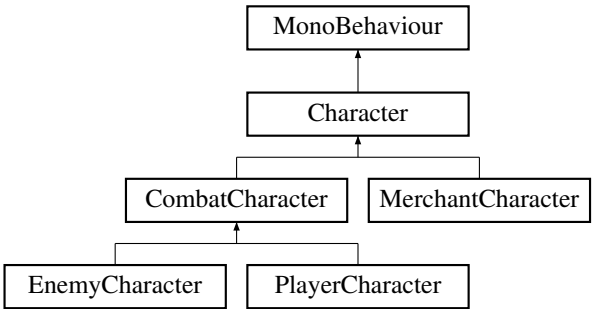
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/BridgetBT.cs

5.18 Character Class Reference

Base class for all in-game characters.

Inheritance diagram for Character:



Public Member Functions

- virtual void Init ()

  Initializes the character.

Protected Attributes

- CharacterSO **data**
- Money **money**

Properties

- Animator **Animator** `[get]`
- Money **Money** `[get]`
- CharacterSO **Data** `[get]`

5.18.1 Detailed Description

Base class for all in-game characters.

5.18.2 Member Function Documentation

5.18.2.1 Init()

```
virtual void Character.Init ( )  [virtual]
```

Initializes the character.

Reimplemented in CombatCharacter, MerchantCharacter, and PlayerCharacter.

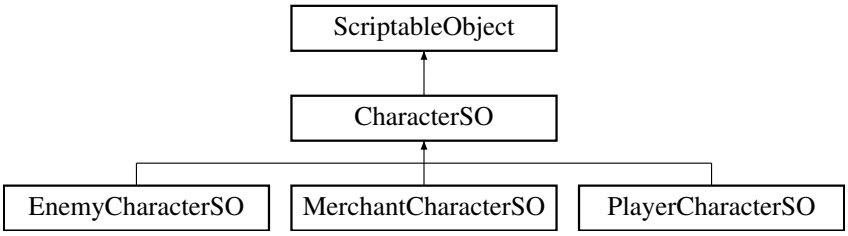
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Characters/Character.cs

5.19 CharacterSO Class Reference

Character data class.

Inheritance diagram for CharacterSO:



Public Attributes

- int **money**
- int **health**
- float **speed**
- AttackSO[ ] **attacks**
- TrapSO[ ] **traps**
- DashSO **dash**
- FMODUnity.EventReference **onHitSound**
- FMODUnity.EventReference **onDeathSound**

5.19.1 Detailed Description

Character data class.

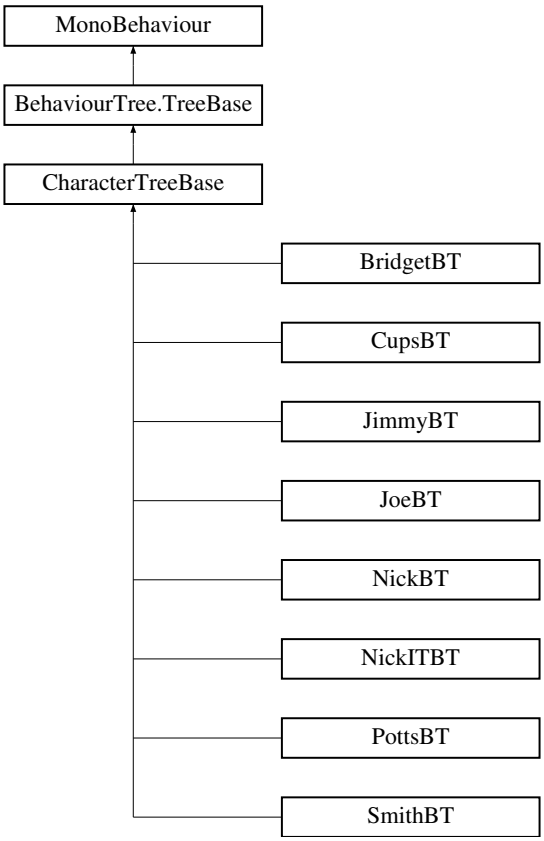
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/CharacterSO.cs

5.20 CharacterTreeBase Class Reference

Base class for behavior trees of all characters in the game.

Inheritance diagram for CharacterTreeBase:



Public Member Functions

- void **AddItem** (string name, object item)
- object **GetItem** (string name)
- void **RemoveItem** (string name)

Public Attributes

- Vector3[ ] **patrolPoints**

Protected Member Functions

- Node **GetPatrolBT** ()
- Node **GetAttackBT** (Attack attack, bool checkCD=false, bool precise=false)
- Node **GetDashAttackBT** (Attack attack, bool checkCD=true)
- Node GetDashBT (bool checkCD=true)

  Specifically for dashes that do damage.
- Node **GetTrapBT** (Trap trap, bool checkCD=true)

Protected Attributes

- CharacterTreeBase **rootTree** = null
- Transform **playerCharacter**

Properties

- CombatCharacter **Character** `[get]`
- Seeker **Seeker** `[get]`
- bool **ShouldUpdate** `[get, set]`

5.20.1 Detailed Description

Base class for behavior trees of all characters in the game.

5.20.2 Member Function Documentation

5.20.2.1 GetDashBT()

```
Node CharacterTreeBase.GetDashBT (
            bool checkCD = true )  [protected]
```

Specifically for dashes that do damage.

Parameters

checkCD	Take cooldown into account

Returns

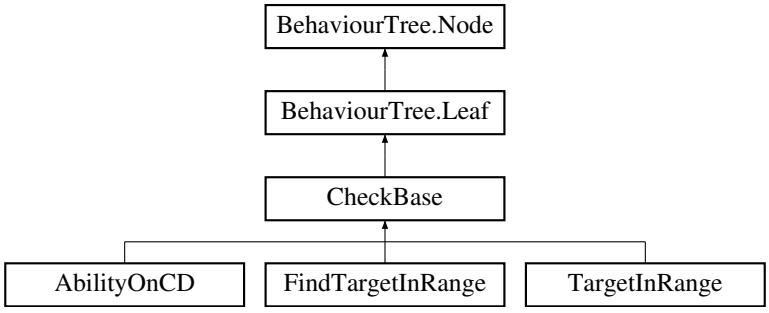
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/CharacterTreeBase.cs

5.21 CheckBase Class Reference

Base class for all checks. Checks only check whether this branch is valid.

Inheritance diagram for CheckBase:



Public Member Functions

- override NodeStatus Update ()

  Updates the node according to the node's specific logic.

Protected Member Functions

- CheckBase (CharacterTreeBase characterBT, string debugName="")
- abstract bool **Check** ()

Protected Attributes

- CharacterTreeBase **bt**

Additional Inherited Members

5.21.1 Detailed Description

Base class for all checks. Checks only check whether this branch is valid.

5.21.2 Constructor & Destructor Documentation

5.21.2.1 CheckBase()

```
CheckBase.CheckBase (
            CharacterTreeBase characterBT,
            string debugName = "" )  [protected]
```

Parameters

characterBT	The behavioral tree of this character
debugName	Used in ToString() for debug purposes

5.21.3 Member Function Documentation

5.21.3.1 Update()

```
override NodeStatus CheckBase.Update ( )  [virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

Implements BehaviourTree.Node.

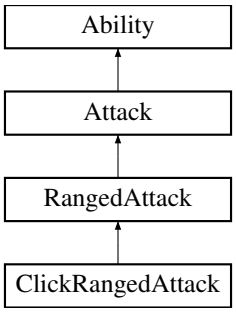
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/CheckBase.cs

5.22 ClickRangedAttack Class Reference

Class for a basic ranged attack.

Inheritance diagram for ClickRangedAttack:



Public Member Functions

- ClickRangedAttack (CombatCharacter character, AttackSO data)

  Creates an instance of a click ranged attack - a ranged attack which shoots the projectile on button down.
- override void OnBegin ()

  Begins performing the ability.

Additional Inherited Members

5.22.1 Detailed Description

Class for a basic ranged attack.

5.22.2 Constructor & Destructor Documentation

5.22.2.1 ClickRangedAttack()

```
ClickRangedAttack.ClickRangedAttack (
            CombatCharacter character,
            AttackSO data )
```

Creates an instance of a click ranged attack - a ranged attack which shoots the projectile on button down.

Parameters

character	The character to which the attack belongs
data	The attack data

5.22.3 Member Function Documentation

5.22.3.1 OnBegin()

```
override void ClickRangedAttack.OnBegin ( )  [virtual]
```

Begins performing the ability.

Reimplemented from Attack.

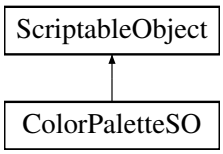
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/ClickRangedAttack.cs

5.23 ColorPaletteSO Class Reference

Game color palette data.

Inheritance diagram for ColorPaletteSO:



Public Attributes

- Color **healthGrey**
- Color **healthRed**
- Color **healthBrightRed**

5.23.1 Detailed Description

Game color palette data.

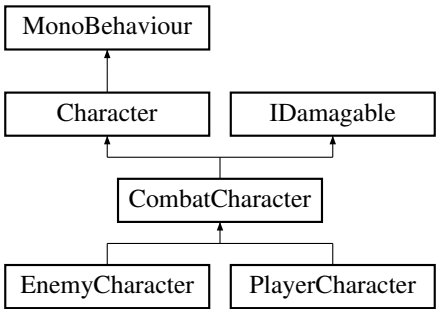
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/ColorPaletteSO.cs

5.24 CombatCharacter Class Reference

Base class for all aggressive characters - i.e. characters that can engage in combat. Handles character movement and combat - both animation and physics.

Inheritance diagram for CombatCharacter:



Public Member Functions

- Dash **GetDash** ()
- Attack **GetAttackByID** (int id)
- Trap **GetMainTrap** ()
- override void Init ()

  Initializes the character.
- void Rotate (Vector2 direction)

  Sets character to face in the given direction.
- void Move (Vector2 move)

  Moves this character in the given direction.
- void **ForceIdle** ()

  Forces the character into an idle animation.
- void **Dash** ()

  Performs a dash.
- bool Attack (Attack attack, EAttackCommand attackCommand=EAttackCommand.Begin)

  Tries to begin or end an attack.
- bool AttackTarget (Attack attack, Vector3 target, EAttackCommand attackCommand=EAttackCommand.↩ Begin)

  Tries to begin or end an attack on the specified target.
- void ActivateTrap (Trap trap=null)

  Activates the given trap. If no trap specified, activates the first trap in the characters trap list, if present.
- virtual void TakeDamage (int amount)

  Deals damage to this character.
- virtual void Die ()

  Kills this character.
- void **CleanUp** ()

  Handles clean up of the character after death.
- void SetMovementSpeed (float movementSpeed)

  Sets the movement speed of this character to the given value.
- void **ResetAttacks** ()

  Ends all attacks of the character.
- void **ResetMovementSpeed** ()

  Resets movement speed of the character.

Public Attributes

- UnityEvent **onDeath** = new UnityEvent()

Protected Attributes

- float **movementSpeed**
- float **currentSpeed**
- Vector2 **facing**
- Health **health**
- bool **canMove**
- List< Attack > **attacks**
- List< Trap > **traps**
- Dash **dash**
- Rigidbody2D **rb**

Properties

- Transform **ProjectileSpawnerTransform** `[get]`
- Vector2 **Facing** `[get]`
- float **ColliderRadius** `[get]`
- Rigidbody2D **RB** `[get]`
- Health **Health** `[get]`

5.24.1 Detailed Description

Base class for all aggressive characters - i.e. characters that can engage in combat. Handles character movement and combat - both animation and physics.

5.24.2 Member Function Documentation

5.24.2.1 ActivateTrap()

```
void CombatCharacter.ActivateTrap (
            Trap trap = null )
```

Activates the given trap. If no trap specified, activates the first trap in the characters trap list, if present.

Parameters

trap	The trap to activate

5.24.2.2 Attack()

```
bool CombatCharacter.Attack (
            Attack attack,
            EAttackCommand attackCommand = EAttackCommand.Begin )
```

Tries to begin or end an attack.

Parameters

attack	The attack to perform
attackCommand	The attack command (i.e. should the attack begin or end)

Returns

5.24.2.3 AttackTarget()

```
bool CombatCharacter.AttackTarget (
            Attack attack,
            Vector3 target,
            EAttackCommand attackCommand = EAttackCommand.Begin )
```

Tries to begin or end an attack on the specified target.

Parameters

attack	The attack to perform
target	The target to attack
attackCommand	

Returns

5.24.2.4 Die()

```
virtual void CombatCharacter.Die ( )  [virtual]
```

Kills this character.

Implements IDamagable.

Reimplemented in EnemyCharacter.

5.24.2.5 Init()

```
override void CombatCharacter.Init ( )  [virtual]
```

Initializes the character.

Reimplemented from Character.

Reimplemented in PlayerCharacter.

5.24.2.6 Move()

```
void CombatCharacter.Move (
            Vector2 move )
```

Moves this character in the given direction.

Parameters

move	Direction to move in

5.24.2.7 Rotate()

```
void CombatCharacter.Rotate (
            Vector2 direction )
```

Sets character to face in the given direction.

Parameters

direction	The direction to look in. Expects a normalized vector.

5.24.2.8 SetMovementSpeed()

```
void CombatCharacter.SetMovementSpeed (
            float movementSpeed )
```

Sets the movement speed of this character to the given value.

Parameters

movementSpeed	

5.24.2.9 TakeDamage()

```
virtual void CombatCharacter.TakeDamage (
            int amount )  [virtual]
```

Deals damage to this character.

Parameters

amount	The amount of damage to deal

Implements IDamagable.

Reimplemented in PlayerCharacter.

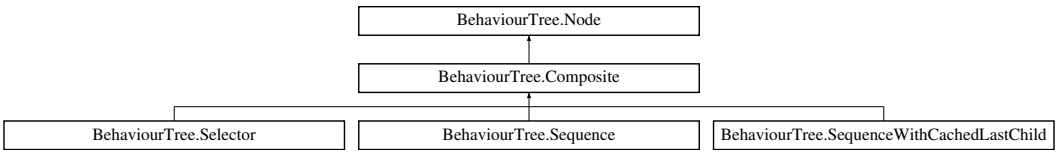
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Characters/CombatCharacter.cs

5.25 BehaviourTree.Composite Class Reference

Base class for all behavior tree composite nodes. Composite nodes have one or more children which are processed either in a first to last sequence or in random order.

Inheritance diagram for BehaviourTree.Composite:



Protected Member Functions

- **Composite** (List< Node > children)

Protected Attributes

- List< Node > **children**
- int **lastProcessedChild** = 0
- bool **sequentialProcessing** = true

Additional Inherited Members

5.25.1 Detailed Description

Base class for all behavior tree composite nodes. Composite nodes have one or more children which are processed either in a first to last sequence or in random order.

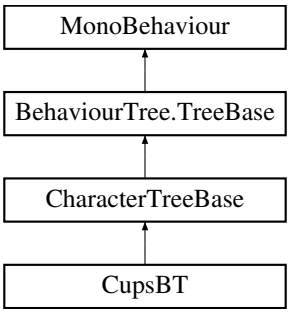
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.26 CupsBT Class Reference

Behavioral tree for a mob from the first level.

Inheritance diagram for CupsBT:



Protected Member Functions

- override void Init ()

  Initiate every tree in this method - called from Start.

Additional Inherited Members

5.26.1 Detailed Description

Behavioral tree for a mob from the first level.

5.26.2 Member Function Documentation

5.26.2.1 Init()

```
override void CupsBT.Init ( )  [protected], [virtual]
```

Initiate every tree in this method - called from Start.

Implements BehaviourTree.TreeBase.

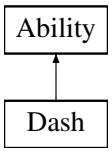
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/CupsBT.cs

5.27 Dash Class Reference

Class for the dash ability.

Inheritance diagram for Dash:



Public Member Functions

- Dash (CombatCharacter character, DashSO data)
  Creates a dash instance.
- override void OnBegin ()
  Begins performing the ability.
- override IEnumerator OnContinue ()
  Updates the performing of the ability if the ability is in use.
- override void OnEnd ()
  Ends the performing of the ability.

Properties

- DashSO **Data** `[get, protected set]`

Additional Inherited Members

5.27.1 Detailed Description

Class for the dash ability.

5.27.2 Constructor & Destructor Documentation

5.27.2.1 Dash()

```
Dash.Dash (
            CombatCharacter character,
            DashSO data )
```

Creates a dash instance.

Parameters

character	The character to which the attack belongs
data	The dash data

5.27.3 Member Function Documentation

5.27.3.1 OnBegin()

```
override void Dash.OnBegin ( )  [virtual]
```

Begins performing the ability.

Reimplemented from Ability.

5.27.3.2 OnContinue()

```
override IEnumerator Dash.OnContinue ( )  [virtual]
```

Updates the performing of the ability if the ability is in use.

Returns

Reimplemented from Ability.

5.27.3.3 OnEnd()

```
override void Dash.OnEnd ( )  [virtual]
```

Ends the performing of the ability.

Reimplemented from Ability.

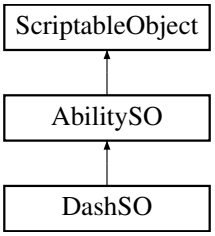
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/Dash.cs

5.28 DashSO Class Reference

Dash data class.

Inheritance diagram for DashSO:



Public Member Functions

- Dash **GetDash** (CombatCharacter character)

Public Attributes

- float **distance**
- float **speed**
- float **deltaBeforeMax**
- float **deltaAfterMax**
- int **maxNumChained**

5.28.1 Detailed Description

Dash data class.

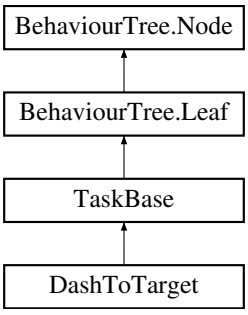
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/DashSO.cs

5.29 DashToTarget Class Reference

Performs a dash in the direction of the target.

Inheritance diagram for DashToTarget:



Public Member Functions

- DashToTarget (CharacterTreeBase characterBT, string targetName, string debugName="")

Protected Member Functions

- override void OnBegin ()
- override void OnContinue ()

Additional Inherited Members

5.29.1 Detailed Description

Performs a dash in the direction of the target.

5.29.2 Constructor & Destructor Documentation

5.29.2.1 DashToTarget()

```
DashToTarget.DashToTarget (
            CharacterTreeBase characterBT,
            string targetName,
            string debugName = "" )
```

Parameters

characterBT	The behavioral tree of this character
targetName	The name of the target - this task will try to retrieve the target from shared memory.
debugName	

5.29.3 Member Function Documentation

5.29.3.1 OnBegin()

```
override void DashToTarget.OnBegin ( )  [protected], [virtual]
```

Implements TaskBase.

5.29.3.2 OnContinue()

```
override void DashToTarget.OnContinue ( )  [protected], [virtual]
```

Implements TaskBase.

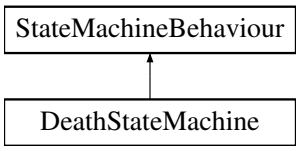
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/DashToTarget.cs

5.30 DeathStateMachine Class Reference

State machine for death state of all characters.

Inheritance diagram for DeathStateMachine:



Public Member Functions

- override void **OnStateEnter** (Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
- override void **OnStateExit** (Animator animator, AnimatorStateInfo stateinfo, int layerindex)

5.30.1 Detailed Description

State machine for death state of all characters.

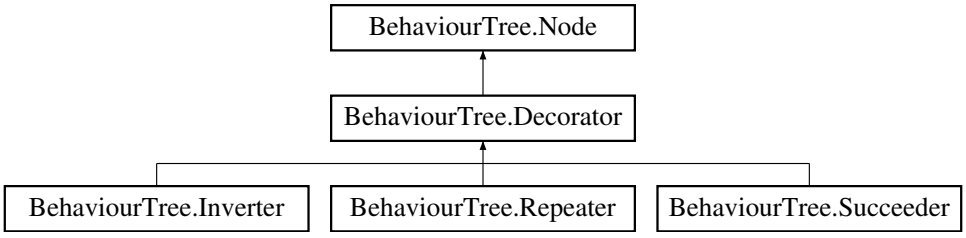
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/StateMachineBehaviors/DeathStateMachine.cs

5.31 BehaviourTree.Decorator Class Reference

Base class for all behavior tree decorator nodes. Decorator nodes have exactly one child.

Inheritance diagram for BehaviourTree.Decorator:



Public Member Functions

- **Decorator** (Node child)

Protected Attributes

- Node **child**

Additional Inherited Members

5.31.1 Detailed Description

Base class for all behavior tree decorator nodes. Decorator nodes have exactly one child.

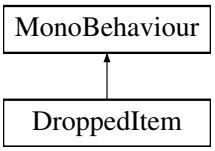
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.32 DroppedItem Class Reference

Handles items dropped by the enemy characters and their collection by the player.

Inheritance diagram for DroppedItem:



Public Member Functions

- void Init (int amount)

  Initializes the dropped item with money.
- void Init (ItemSO item)

  Initializes the dropped item with an item.

5.32.1 Detailed Description

Handles items dropped by the enemy characters and their collection by the player.

5.32.2 Member Function Documentation

5.32.2.1 Init() [1/2]

```
void DroppedItem.Init (
            int amount )
```

Initializes the dropped item with money.

Parameters

amount	The amount of money to drop

5.32.2.2 Init() [2/2]

```
void DroppedItem.Init (
            ItemSO item )
```

Initializes the dropped item with an item.

Parameters

item	The data of the item to drop

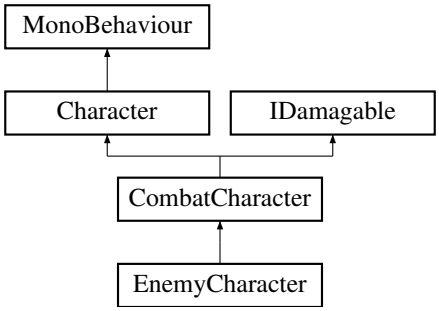
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Items/DroppedItem.cs

5.33 EnemyCharacter Class Reference

Class for all enemy characters.

Inheritance diagram for EnemyCharacter:



Public Member Functions

- void Freeze (bool freeze)

  Stops the AI from updating (useful ex. when pausing the game).
- void **Init** (PlayerCharacter playerCharacter)
- override void Die ()

  Kills this character.

Additional Inherited Members

5.33.1 Detailed Description

Class for all enemy characters.

5.33.2 Member Function Documentation

5.33.2.1 Die()

```
override void EnemyCharacter.Die ( )  [virtual]
```

Kills this character.

Reimplemented from CombatCharacter.

5.33.2.2 Freeze()

```
void EnemyCharacter.Freeze (
            bool freeze )
```

Stops the AI from updating (useful ex. when pausing the game).

Parameters

freeze	

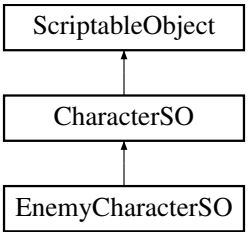
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Characters/EnemyCharacter.cs

5.34 EnemyCharacterSO Class Reference

Enemy character data class.

Inheritance diagram for EnemyCharacterSO:



Public Attributes

- float **lineOfSightRange**
- float **patrolWaitTime** = 1f

5.34.1 Detailed Description

Enemy character data class.

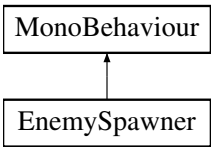
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/EnemyCharacterSO.cs

5.35 EnemySpawner Class Reference

Responsible for spawning and managing enemy characters in the scene.

Inheritance diagram for EnemySpawner:



Public Member Functions

- void Init (GameManager gameManager, PlayerCharacter playerCharacter)

  Initializes the enemy spawner.
- void SpawnEnemy (Vector3 spawnPoint, Vector3[] patrolPoints, EnemyCharacterSO enemy, UnityAction onEnemyDeath=null)

  Spawns a new enemy in the scene and initializes it.
- void FreezeEnemies (bool freeze)

  All enemies currently spawned stop/start being updated.

Public Attributes

- UnityEvent **onAllEnemiesDefeated**

5.35.1 Detailed Description

Responsible for spawning and managing enemy characters in the scene.

5.35.2 Member Function Documentation

5.35.2.1 FreezeEnemies()

```
void EnemySpawner.FreezeEnemies (
            bool freeze )
```

All enemies currently spawned stop/start being updated.

Parameters

freeze	

5.35.2.2 Init()

```
void EnemySpawner.Init (
            GameManager gameManager,
            PlayerCharacter playerCharacter )
```

Initializes the enemy spawner.

Parameters

gameManager	the current game manager
playerCharacter	the current player character instance

5.35.2.3 SpawnEnemy()

```
void EnemySpawner.SpawnEnemy (
            Vector3 spawnPoint,
            Vector3[] patrolPoints,
            EnemyCharacterSO enemy,
            UnityAction onEnemyDeath = null )
```

Spawns a new enemy in the scene and initializes it.

Parameters

spawnPoint	The position to spawn the enemy at.
patrolPoints	The positions between which the enemy will patrol when not engaged in combat
enemy	The enemy initial data
onEnemyDeath	Action to perform after the enemy dies

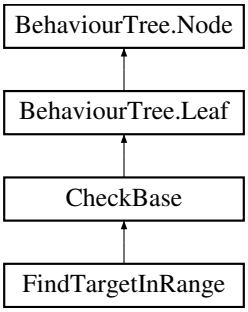
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/EnemySpawner.cs

5.36 FindTargetInRange Class Reference

A task which tries to find a target (typically the player) and save it.

Inheritance diagram for FindTargetInRange:



Public Member Functions

- FindTargetInRange (CharacterTreeBase characterBT, string targetName, float range, EObstacleFilter obstacleFilter=EObstacleFilter.None, string targetTag=Utility.playerTagAndLayer, string debugName="")

Protected Member Functions

- override bool Check ()

Additional Inherited Members

5.36.1 Detailed Description

A task which tries to find a target (typically the player) and save it.

5.36.2 Constructor & Destructor Documentation

5.36.2.1 FindTargetInRange()

```
FindTargetInRange.FindTargetInRange (
            CharacterTreeBase characterBT,
            string targetName,
            float range,
            EObstacleFilter obstacleFilter = EObstacleFilter.None,
            string targetTag = Utility.playerTagAndLayer,
            string debugName = "" )
```

Parameters

characterBT	The behavioral tree of this character
targetName	The name of the target - this task will save any found target to shared memory.
range	
obstacleFilter	
targetTag	The tag of the target we're trying to find
debugName	

5.36.3 Member Function Documentation

5.36.3.1 Check()

```
override bool FindTargetInRange.Check ( )  [protected], [virtual]
```

Implements CheckBase.

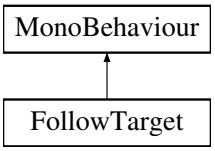
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/FindTargetInRange.cs

5.37 FollowTarget Class Reference

Makes this game object follow a specified target.

Inheritance diagram for FollowTarget:



Public Member Functions

- void Init (Transform target)
  Initializes the follower.

5.37.1 Detailed Description

Makes this game object follow a specified target.

5.37.2 Member Function Documentation

5.37.2.1 Init()

```
void FollowTarget.Init (
            Transform target )
```

Initializes the follower.

Parameters

target	The target to follow

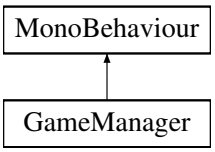
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/FollowTarget.cs

5.38 GameManager Class Reference

The game manager - main entry point of the game.

Inheritance diagram for GameManager:



Public Member Functions

- void **StartGame** ()
  Starts the game.
- void PauseGame (bool pause)
  Pauses the game.
- void GameEnd (bool won)
  Ends the game.
- void **BackToMain** ()
  Returns to main menu.

Public Attributes

- Canvas **worldSpaceCanvas**
- TooltipUI **tooltipUIPrefab**
- ColorPaletteSO **ColorPalette**
- AudioManager **audioManager**
- UIInput **uiInput**
- PlayerSpawner **playerSpawner**

Properties

- PlayerCharacter **PlayerCharacterInstance** `[get]`

5.38.1 Detailed Description

The game manager - main entry point of the game.

5.38.2 Member Function Documentation

5.38.2.1 GameEnd()

```
void GameManager.GameEnd (
            bool won )
```

Ends the game.

Parameters

won	Whether the game was won

5.38.2.2 PauseGame()

```
void GameManager.PauseGame (
            bool pause )
```

Pauses the game.

Parameters

pause	

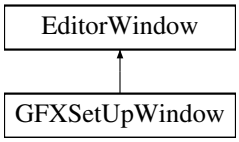
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/General/GameManager.cs

5.39 GFXSetUpWindow Class Reference

Editor window for all graphics' set-up.

Inheritance diagram for GFXSetUpWindow:



5.39.1 Detailed Description

Editor window for all graphics' set-up.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/GFXSetUpWindow.cs

5.40 MapGenerator.GridTile Struct Reference

Public Member Functions

- **GridTile** (int x, int y, bool empty=true)

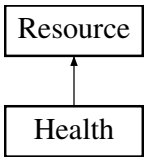
The documentation for this struct was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PCG/MapGenerator.cs

5.41 Health Class Reference

Wrapper class for the health resource. Represents the current and max health for all characters that can engage in combat.

Inheritance diagram for Health:



Public Member Functions

- **Health** (int maxValue)

Additional Inherited Members

5.41.1 Detailed Description

Wrapper class for the health resource. Represents the current and max health for all characters that can engage in combat.

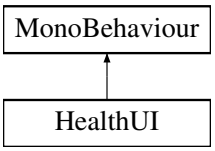
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Resources/Health.cs

5.42 HealthUI Class Reference

Health bar UI manager class.

Inheritance diagram for HealthUI:



Public Member Functions

- void **Init** (GameManager gameManager, PlayerInventory playerInventory)

5.42.1 Detailed Description

Health bar UI manager class.

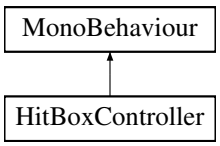
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/HealthUI.cs

5.43 HitBoxController Class Reference

Controls the hitbox for all melee attacks.

Inheritance diagram for HitBoxController:



Public Member Functions

- void Init (AbilitySO data)

  Initializes the hitbox

Static Public Member Functions

- static void HandleCollision (Transform transform, Collider2D collision, AbilitySO abilityData, string friendlyTag)

  Handles all player versus enemy collision.

5.43.1 Detailed Description

Controls the hitbox for all melee attacks.

5.43.2 Member Function Documentation

5.43.2.1 HandleCollision()

```
static void HitBoxController.HandleCollision (
            Transform transform,
            Collider2D collision,
            AbilitySO abilityData,
            string friendlyTag )  [static]
```

Handles all player versus enemy collision.

Parameters

transform	The transform of the object whose collision we're handling
collision	The collider of the object which collided with the transform
abilityData	The data
friendlyTag	

5.43.2.2 Init()

```
void HitBoxController.Init (
            AbilitySO data )
```

Initializes the hitbox

Parameters

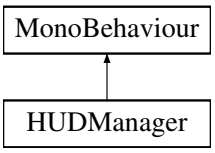
data	

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/HitBoxController.cs

5.44 HUDManager Class Reference

Inheritance diagram for HUDManager:



Public Member Functions

- void Init (GameManager gameManager, PlayerCharacter playerCharacter)

  Should be called from game manager after scene loaded.

5.44.1 Detailed Description

5.44.2 Member Function Documentation

5.44.2.1 Init()

```
void HUDManager.Init (
            GameManager gameManager,
            PlayerCharacter playerCharacter )
```

Should be called from game manager after scene loaded.

Parameters

gameManager	
playerCharacter	The current player character instance

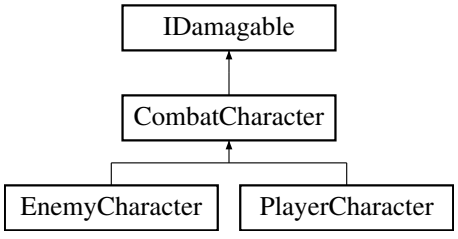
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/HUDManager.cs

5.45 IDamagable Interface Reference

Any damageable entity should implement this.

Inheritance diagram for IDamagable:



Public Member Functions

- void TakeDamage (int amount)

  Deal damage to this damageable.
- void Die ()

5.45.1 Detailed Description

Any damageable entity should implement this.

5.45.2 Member Function Documentation

5.45.2.1 Die()

```
void IDamagable.Die ( )
```

Implemented in CombatCharacter, and EnemyCharacter.

5.45.2.2 TakeDamage()

```
void IDamagable.TakeDamage (
            int amount )
```

Deal damage to this damageable.

Parameters

amount	The amount of damage to deal

Implemented in CombatCharacter, and PlayerCharacter.

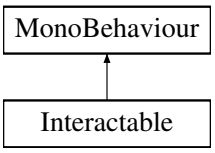
The documentation for this interface was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Interfaces/IDamagable.cs

5.46 Interactable Class Reference

Class for game objects with which the player can interact by pressing the interact button.

Inheritance diagram for Interactable:



Public Member Functions

- void **Init** (string message)
- void SetTooltip (string message)

  Set the text of the tooltip of this interactable, if present.

Public Attributes

- UnityEvent **onInteractPressed**

5.46.1 Detailed Description

Class for game objects with which the player can interact by pressing the interact button.

5.46.2 Member Function Documentation

5.46.2.1 SetTooltip()

```
void Interactable.SetTooltip (
            string message )
```

Set the text of the tooltip of this interactable, if present.

Parameters

message	

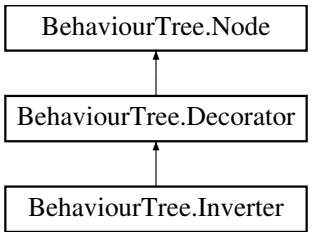
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Interactable.cs

5.47 BehaviourTree.Inverter Class Reference

A decorator node which inverts the result of the child.

Inheritance diagram for BehaviourTree.Inverter:



Public Member Functions

- **Inverter** (Node child)
- override NodeStatus Update ()

  Updates the node according to the node's specific logic.

Additional Inherited Members

5.47.1 Detailed Description

A decorator node which inverts the result of the child.

5.47.2 Member Function Documentation

5.47.2.1 Update()

```
override NodeStatus BehaviourTree.Inverter.Update ( )  [virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

Implements BehaviourTree.Node.

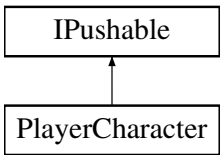
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.48 IPushable Interface Reference

Any pushable entity should implement this.

Inheritance diagram for IPushable:



Public Member Functions

- void Push (Vector2 direction, float distance, float speed)

  Push this pushable in the given direction.

5.48.1 Detailed Description

Any pushable entity should implement this.

5.48.2 Member Function Documentation

5.48.2.1 Push()

```
void IPushable.Push (
            Vector2 direction,
            float distance,
            float speed )
```

Push this pushable in the given direction.

Parameters

direction	
distance	
speed	

Implemented in PlayerCharacter.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Interfaces/IPushable.cs

5.49 Item Class Reference

Only for equippable items.

Public Member Functions

- Item (ItemSO data)

  Creates an item instance.
- void ReceiveDamage (int damage)

  Deal damage to this item.

Properties

- ItemSO **Data** `[get]`
- int **CurrentHealth** `[get]`

5.49.1 Detailed Description

Only for equippable items.

5.49.2 Constructor & Destructor Documentation

5.49.2.1 Item()

```
Item.Item (
            ItemSO data )
```

Creates an item instance.

Parameters

data	

5.49.3 Member Function Documentation

5.49.3.1 ReceiveDamage()

```
void Item.ReceiveDamage (
            int damage )
```

Deal damage to this item.

Parameters

damage	

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Items/Item.cs

5.50 ItemEffect Class Reference

Public Attributes

- EItemEffectType **Type**
- float **ValueModifier**

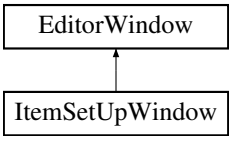
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/ItemSO.cs

5.51 ItemSetUpWindow Class Reference

An editor window for setting up the in-game items.

Inheritance diagram for ItemSetUpWindow:



5.51.1 Detailed Description

An editor window for setting up the in-game items.

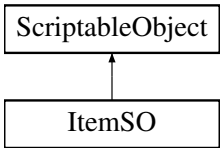
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/ItemSetUpWindow.cs

5.52 ItemSO Class Reference

Item data class.

Inheritance diagram for ItemSO:



Public Attributes

- int **ID**
- string **ItemName**
- List< ItemEffect > **ItemEffects**
- Sprite **UISprite**
- Sprite **WorldSprite**
- int **Health**
- int **Cost**

5.52.1 Detailed Description

Item data class.

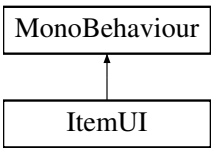
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/ItemSO.cs

5.53 ItemUI Class Reference

Manager class for UI of the items currently equipped by the player.

Inheritance diagram for ItemUI:



Public Member Functions

- void **Init** (GameManager gameManager)
- void **UpdateItem** (Item item)

5.53.1 Detailed Description

Manager class for UI of the items currently equipped by the player.

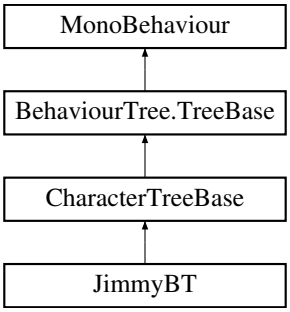
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/ItemUI.cs

5.54 JimmyBT Class Reference

Behavioral tree for the second level boss.

Inheritance diagram for JimmyBT:



Protected Member Functions

- override void Init ()

  Initiate every tree in this method - called from Start.

Additional Inherited Members

5.54.1 Detailed Description

Behavioral tree for the second level boss.

5.54.2 Member Function Documentation

5.54.2.1 Init()

```
override void JimmyBT.Init ( )  [protected], [virtual]
```

Initiate every tree in this method - called from Start.

Implements BehaviourTree.TreeBase.

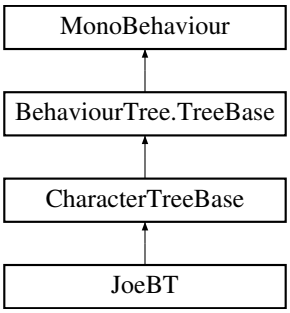
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/JimmyBT.cs

5.55 JoeBT Class Reference

Behavioral tree for a mob from the second level.

Inheritance diagram for JoeBT:



Protected Member Functions

- override void Init ()

  Initiate every tree in this method - called from Start.

Additional Inherited Members

5.55.1 Detailed Description

Behavioral tree for a mob from the second level.

5.55.2 Member Function Documentation

5.55.2.1 Init()

```
override void JoeBT.Init ( )  [protected], [virtual]
```

Initiate every tree in this method - called from Start.

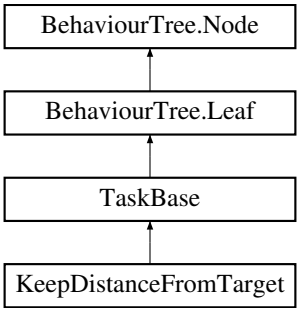
Implements BehaviourTree.TreeBase.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/JoeBT.cs

5.56 KeepDistanceFromTarget Class Reference

Inheritance diagram for KeepDistanceFromTarget:



Public Member Functions

- **KeepDistanceFromTarget** (CharacterTreeBase characterBT, float safeDistance, string debugName="keep target at a distance")

Protected Member Functions

- override void OnBegin ()
- override void OnContinue ()

Additional Inherited Members

5.56.1 Member Function Documentation

5.56.1.1 OnBegin()

```
override void KeepDistanceFromTarget.OnBegin ( )  [protected], [virtual]
```

Implements TaskBase.

5.56.1.2 OnContinue()

```
override void KeepDistanceFromTarget.OnContinue ( )  [protected], [virtual]
```

Implements TaskBase.

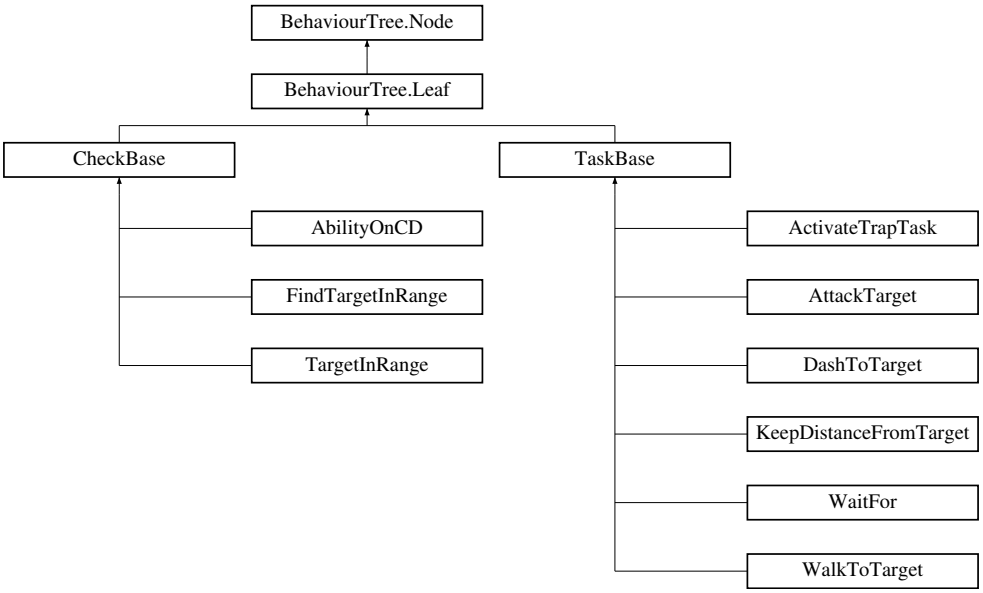
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/KeepDistanceFromTarget.cs

5.57 BehaviourTree.Leaf Class Reference

Leaf node base class. Leaves have no children and their function is to perform character specific actions or checks.

Inheritance diagram for BehaviourTree.Leaf:



Public Member Functions

- override string **ToString** ()

Protected Member Functions

- **Leaf** (string debugName="")

Additional Inherited Members

5.57.1 Detailed Description

Leaf node base class. Leaves have no children and their function is to perform character specific actions or checks.

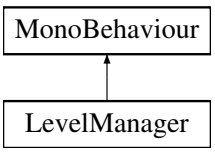
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.58 LevelManager Class Reference

Manager for one game level (each level should have its own LevelManager). Last level (the mall roof) probably won't be able to make use of this -> special prefab instead.

Inheritance diagram for LevelManager:



Public Member Functions

- void **Init** (GameManager gameManager, LevelSO data)
- void FreezeLevel (bool freeze)

  All entities in the level which are normally updated stop being updated. Should be called when pausing the game.
- void LoadRoom (PlayerCharacter player, int roomNumber)

  Loads a room according to the room number.
- void LoadRoom (PlayerCharacter player, RoomType roomType=RoomType.Normal)

  Loads a new room - generates it depending on current level, adds a navmesh, spawns the enemies and places the player. For special room types, no pcg performed - just instantiates a prefab.
- void LoadFinalRoom (PlayerCharacter player)

  Spawns the final room. Should only be used in the final level.

Public Attributes

- UnityEvent **onLevelComplete**

Properties

- static Transform **CurrentRoomTransform** `[get]`

5.58.1 Detailed Description

Manager for one game level (each level should have its own LevelManager). Last level (the mall roof) probably won't be able to make use of this -> special prefab instead.

5.58.2 Member Function Documentation

5.58.2.1 FreezeLevel()

```
void LevelManager.FreezeLevel (
            bool freeze )
```

All entities in the level which are normally updated stop being updated. Should be called when pausing the game.

Parameters

freeze	Should the update stop or start

5.58.2.2 LoadFinalRoom()

```
void LevelManager.LoadFinalRoom (
            PlayerCharacter player )
```

Spawns the final room. Should only be used in the final level.

Parameters

player	the current player character

5.58.2.3 LoadRoom() [1/2]

```
void LevelManager.LoadRoom (
            PlayerCharacter player,
            int roomNumber )
```

Loads a room according to the room number.

Parameters

player	the current player character
roomNumber	the number of the room to load

5.58.2.4 LoadRoom() [2/2]

```
void LevelManager.LoadRoom (
            PlayerCharacter player,
            RoomType roomType = RoomType.Normal )
```

Loads a new room - generates it depending on current level, adds a navmesh, spawns the enemies and places the player. For special room types, no pcg performed - just instantiates a prefab.

Parameters

player	the current player character
roomType	the type of room to spawn

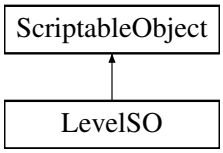
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/General/LevelManager.cs

5.59 LevelSO Class Reference

Level data class.

Inheritance diagram for LevelSO:



Public Member Functions

- EnemyCharacterSO[ ] GetEnemiesToSpawn (int roomNumber, int enemyCount)

  Semi-randomly generates what enemies to spawn depending on the room number.
- EnemyCharacterSO[ ] GetEnemiesToSpawn ()

  Semi-randomly generates what enemies to spawn in the final room - should only be used with the final level.

Public Attributes

- Level **level**
- int **numberOfRooms**
- EnemyCharacterSO[ ] **enemies**
- EnemyCharacterSO **boss**
- MapGenerator **roomGeneratorPrefab**
- SpecialRoom **merchantRoomPrefab**
- SpecialRoom **bossRoomPrefab**
- TextAsset **bossNavGraphData**

5.59.1 Detailed Description

Level data class.

5.59.2 Member Function Documentation

5.59.2.1 GetEnemiesToSpawn() [1/2]

```
EnemyCharacterSO[] LevelSO.GetEnemiesToSpawn ( )
```

Semi-randomly generates what enemies to spawn in the final room - should only be used with the final level.

Returns

The data of the enemies to spawn

5.59.2.2 GetEnemiesToSpawn() [2/2]

```
EnemyCharacterSO[] LevelSO.GetEnemiesToSpawn (
            int roomNumber,
            int enemyCount )
```

Semi-randomly generates what enemies to spawn depending on the room number.

Parameters

roomNumber	The room number for which to spawn enemies
enemyCount	The amount of enemies to spawn

Returns

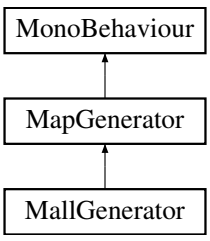
The data of the enemies to spawn

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/LevelSO.cs

5.60 MallGenerator Class Reference

Inheritance diagram for MallGenerator:



Public Member Functions

- override void Generate ()

  Generate a room for the game.

Public Attributes

- List< Tile > **floorTiles**
- List< GameObject > **props**
- List< Tile > **storefrontsSW**
- List< Tile > **storefrontsSE**
- GameObject **pillarObj**

Protected Member Functions

- override void SetUpParameters ()

Additional Inherited Members

5.60.1 Member Function Documentation

5.60.1.1 Generate()

```
override void MallGenerator.Generate ( )  [virtual]
```

Generate a room for the game.

Reimplemented from MapGenerator.

5.60.1.2 SetUpParameters()

```
override void MallGenerator.SetUpParameters ( )  [protected], [virtual]
```

Implements MapGenerator.

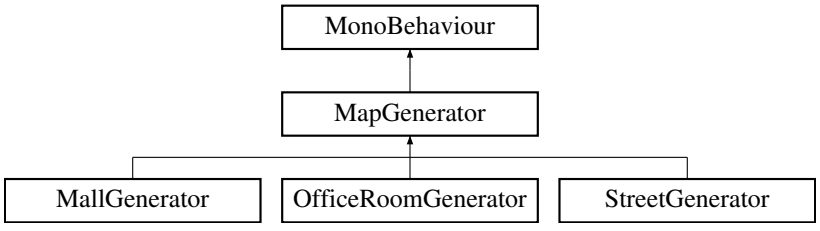
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PCG/MallGenerator.cs

5.61 MapGenerator Class Reference

Base class for map generation

Inheritance diagram for MapGenerator:



Classes

- struct **Entrance**
- struct GridTile
- struct **Room**
- struct **Wall**

Public Member Functions

- void **Start** ()
- virtual void Generate ()

  Generate a room for the game.
- bool **IsTileEmpty** (int x, int y)
- GridTile[,] GetGrid ()

  Get the grid used for obstacles.
- Vector3 GetSmallGridTileWorldCoordinates (int x, int y)

  Converts tile at x y coordinates in the smaller tilemap to world coordinates.
- Vector3 GetGridTileWorldCoordinates (int x, int y)

  Converts tile at x y coordinates in the tilemap to world coordinates.
- Vector3Int **GetEntranceGridCoords** ()
- Vector3Int **GetExitGridCoords** ()
- Transform **GetEntranceCollider** ()
- Vector3 GetGridTileWorldCoordinatesMiddle (int x, int y)

  Converts tile at x y coordinates to the middle point of the tile in world coordinates.
- Vector3 **GetEntranceMiddlePoint** ()
- Transform **GetExitCollider** ()
- Vector3 **GetExitMiddlePoint** ()
- Interactable **GetExitTrigger** ()
- void **DestroyCurrentRoom** ()

Public Attributes

- Tile **collTile**
- GameObject **entranceCollider**
- int **maxWidth**
- int **maxHeight**
- Tile **entranceTile**
- Tile **exitTile**
- Tile **floorEntranceTile**
- GameObject **mapPrefab**

Protected Member Functions

- abstract void **SetUpParameters** ()
- virtual Vector2Int **UnityToScriptCoord** (int x, int y)
- void **SetTilesToMap** (Tile tile, Tilemap tileMap, int startX, int startY, int finX, int finY)
- GameObject **GetExitObject** ()

Protected Attributes

- Transform **_roomHolder**
- Transform **_gridHolder**
- Transform **_obstaclesHolder**

5.61.1 Detailed Description

Base class for map generation

5.61.2 Member Function Documentation

5.61.2.1 Generate()

```
virtual void MapGenerator.Generate ( )  [virtual]
```

Generate a room for the game.

Reimplemented in MallGenerator, OfficeRoomGenerator, and StreetGenerator.

5.61.2.2 GetGrid()

```
GridTile[,] MapGenerator.GetGrid ( )
```

Get the grid used for obstacles.

Returns

GridTile[,] grid

5.61.2.3 GetGridTileWorldCoordinates()

```
Vector3 MapGenerator.GetGridTileWorldCoordinates (
            int x,
            int y )
```

Converts tile at x y coordinates in the tilemap to world coordinates.

Parameters

x	Coordinate x of the tile in the tilemap.
y	Coordinate y of the tile in the tilemap.

Returns

Vector3 world coordinates of the tile.

5.61.2.4 GetGridTileWorldCoordinatesMiddle()

```
Vector3 MapGenerator.GetGridTileWorldCoordinatesMiddle (
            int x,
            int y )
```

Converts tile at x y coordinates to the middle point of the tile in world coordinates.

Parameters

x	Coordinate x of the tile in the tilemap.
y	Coordinate y of the tile in the tilemap.

Returns

Vector3 world coordinates of the middle point of the tile.

5.61.2.5 GetSmallGridTileWorldCoordinates()

```
Vector3 MapGenerator.GetSmallGridTileWorldCoordinates (
            int x,
            int y )
```

Converts tile at x y coordinates in the smaller tilemap to world coordinates.

Parameters

x	Coordinate x of the tile in the tilemap.
y	Coordinate y of the tile in the tilemap.

Returns

Vector3 world coordinates of the tile.

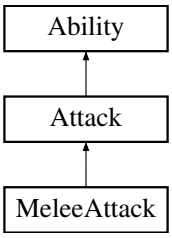
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PCG/MapGenerator.cs

5.62 MeleeAttack Class Reference

Base wrapper class for all melee attacks.

Inheritance diagram for MeleeAttack:



Public Member Functions

- MeleeAttack (CombatCharacter character, AttackSO data)

  Creates a melee attack instance.
- override void OnBegin ()

  Begins performing the ability.

Additional Inherited Members

5.62.1 Detailed Description

Base wrapper class for all melee attacks.

5.62.2 Constructor & Destructor Documentation

5.62.2.1 MeleeAttack()

```
MeleeAttack.MeleeAttack (
            CombatCharacter character,
            AttackSO data )
```

Creates a melee attack instance.

Parameters

character	The character to which the attack belongs
data	The attack data

5.62.3 Member Function Documentation

5.62.3.1 OnBegin()

```
override void MeleeAttack.OnBegin ( )  [virtual]
```

Begins performing the ability.

Reimplemented from Attack.

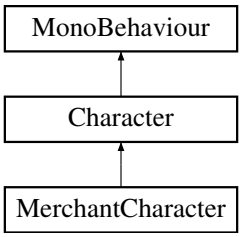
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/MeleeAttack.cs

5.63 MerchantCharacter Class Reference

Base class for the merchant.

Inheritance diagram for MerchantCharacter:



Public Member Functions

- override void Init ()

  Initializes the character.
- void Sell (ItemSO item)

  Sell given item - should be called from shop manager, shop manager is responsible for checking if the player has enough money (and sets the buttons to non-interactable if not). This trades money (according to the item's cost) from player to merchant and puts the item in the player's inventory.

Additional Inherited Members

5.63.1 Detailed Description

Base class for the merchant.

5.63.2 Member Function Documentation

5.63.2.1 Init()

```
override void MerchantCharacter.Init ( )  [virtual]
```

Initializes the character.

Reimplemented from Character.

5.63.2.2 Sell()

```
void MerchantCharacter.Sell (
            ItemSO item )
```

Sell given item - should be called from shop manager, shop manager is responsible for checking if the player has enough money (and sets the buttons to non-interactable if not). This trades money (according to the item's cost) from player to merchant and puts the item in the player's inventory.

Parameters

item	The item data for the item we're selling

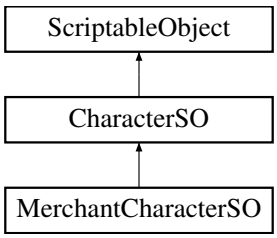
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Characters/MerchantCharacter.cs

5.64 MerchantCharacterSO Class Reference

Merchant character data class.

Inheritance diagram for MerchantCharacterSO:



Public Attributes

- List< ItemSO > **shop**

5.64.1 Detailed Description

Merchant character data class.

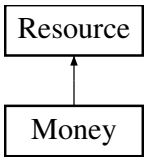
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/MerchantCharacterSO.cs

5.65 Money Class Reference

Wrapper class for the money resource. For the player and merchant NPC it should represent their money amount, for aggressive NPCs the amount of money they drop.

Inheritance diagram for Money:



Public Member Functions

- **Money** (int startingValue)

Additional Inherited Members

5.65.1 Detailed Description

Wrapper class for the money resource. For the player and merchant NPC it should represent their money amount, for aggressive NPCs the amount of money they drop.

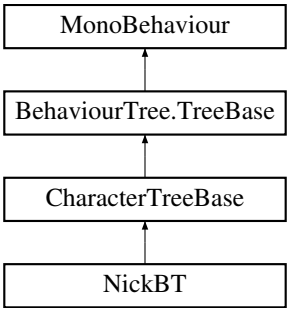
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Resources/Money.cs

5.66 NickBT Class Reference

Behavioral tree for a mob from the first level.

Inheritance diagram for NickBT:



Protected Member Functions

- override void Init ()

  Initiate every tree in this method - called from Start.

Additional Inherited Members

5.66.1 Detailed Description

Behavioral tree for a mob from the first level.

5.66.2 Member Function Documentation

5.66.2.1 Init()

```
override void NickBT.Init ( )  [protected], [virtual]
```

Initiate every tree in this method - called from Start.

Implements BehaviourTree.TreeBase.

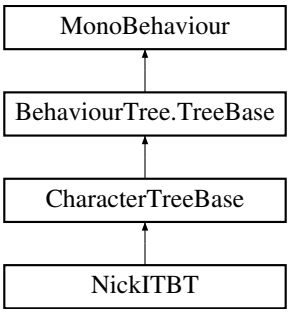
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/NickBT.cs

5.67 NickITBT Class Reference

Behavioral tree for a mob from the first level.

Inheritance diagram for NickITBT:



Protected Member Functions

- override void Init ()

  Initiate every tree in this method - called from Start.

Additional Inherited Members

5.67.1 Detailed Description

Behavioral tree for a mob from the first level.

5.67.2 Member Function Documentation

5.67.2.1 Init()

```
override void NickITBT.Init ( )  [protected], [virtual]
```

Initiate every tree in this method - called from Start.

Implements BehaviourTree.TreeBase.

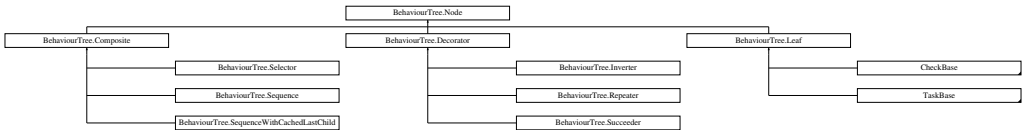
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/NickITBT.cs

5.68 BehaviourTree.Node Class Reference

Base class for all behavior tree nodes.

Inheritance diagram for BehaviourTree.Node:



Public Member Functions

- abstract NodeStatus Update ()

  Updates the node according to the node's specific logic.

Protected Attributes

- NodeStatus **status**

Properties

- Node **Parent** `[get, set]`

5.68.1 Detailed Description

Base class for all behavior tree nodes.

5.68.2 Member Function Documentation

5.68.2.1 Update()

```
abstract NodeStatus BehaviourTree.Node.Update ( )  [pure virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

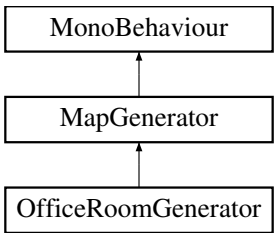
Implemented in BehaviourTree.SequenceWithCachedLastChild, BehaviourTree.Sequence, BehaviourTree.Selector, BehaviourTree.Inverter, BehaviourTree.Succeeder, BehaviourTree.Repeater, CheckBase, and TaskBase.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.69 OfficeRoomGenerator Class Reference

Inheritance diagram for OfficeRoomGenerator:



Public Member Functions

- override void Generate ()

  Generate a room for the game.

Public Attributes

- List< GameObject > **horTables**
- List< GameObject > **verTables**
- List< GameObject > **extraItems**
- int **tableDensity** = 85
- int **extraDensity** = 15
- List< Tile > **tileLst**
- List< Tile > **horWallLst**
- List< Tile > **verWallLst**

Protected Member Functions

- override void SetUpParameters ()
- override Vector2Int UnityToScriptCoord (int x, int y)

Additional Inherited Members

5.69.1 Member Function Documentation

5.69.1.1 Generate()

```
override void OfficeRoomGenerator.Generate ( )  [virtual]
```

Generate a room for the game.

Reimplemented from MapGenerator.

5.69.1.2 SetUpParameters()

```
override void OfficeRoomGenerator.SetUpParameters ( )  [protected], [virtual]
```

Implements MapGenerator.

5.69.1.3 UnityToScriptCoord()

```
override Vector2Int OfficeRoomGenerator.UnityToScriptCoord (
            int x,
            int y )  [protected], [virtual]
```

Reimplemented from MapGenerator.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PCG/OfficeRoomGenerator.cs

5.70 Pivot Struct Reference

Public Attributes

- SpriteAlignment **type**
- Vector2 **vector**

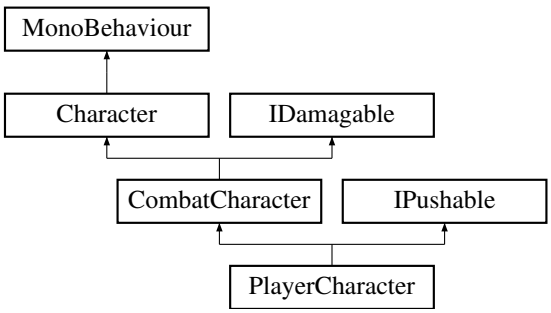
The documentation for this struct was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/SpriteSetUp.cs

5.71 PlayerCharacter Class Reference

Handles player movement and combat - both animation and physics.

Inheritance diagram for PlayerCharacter:



Public Member Functions

- override void Init ()

  Initializes the player character.
- void **StartAiming** ()

  Starts aiming. Should be called when performing an aimed ranged attack.
- void **RotateAimingGFX** ()

  Rotates the aiming graphic of an aimed ranged attack.
- void **StopAiming** ()

  Releases aiming of a ranged attack.
- void Attack (EAttackButton attackButton, EAttackCommand command)

  Tries to begin or end an attack.
- void CollectMoney (int amount)

  Add money to player character.
- void CollectItem (ItemSO item, UnityAction onItemAdded=null)

  Try to add an item to player inventory.
- override void TakeDamage (int amount)

  Deals damage to this character.
- void Push (Vector2 direction, float distance, float speed)

  Pushes the character in the given direction.

Properties

- PlayerInventory **Inventory** `[get]`
- Respect **Respect** `[get]`
- bool **ReadInput** `[set]`

  Should the player character read input?

Additional Inherited Members

5.71.1 Detailed Description

Handles player movement and combat - both animation and physics.

5.71.2 Member Function Documentation

5.71.2.1 Attack()

```
void PlayerCharacter.Attack (
            EAttackButton attackButton,
            EAttackCommand command )
```

Tries to begin or end an attack.

Parameters

attackButton	What button the player pressed
command	What command to perform

5.71.2.2 CollectItem()

```
void PlayerCharacter.CollectItem (
            ItemSO item,
            UnityAction onItemAdded = null )
```

Try to add an item to player inventory.

Parameters

item	Data of the item to add
onItemAdded	Called if the item was added successfully

5.71.2.3 CollectMoney()

```
void PlayerCharacter.CollectMoney (
            int amount )
```

Add money to player character.

Parameters

amount	The amount to collect

5.71.2.4 Init()

```
override void PlayerCharacter.Init ( )  [virtual]
```

Initializes the player character.

Reimplemented from CombatCharacter.

5.71.2.5 Push()

```
void PlayerCharacter.Push (
            Vector2 direction,
            float distance,
            float speed )
```

Pushes the character in the given direction.

Parameters

direction	The direction to push in
distance	The distance of the push
speed	The speed of the push

Implements IPushable.

5.71.2.6 TakeDamage()

```
override void PlayerCharacter.TakeDamage (
            int amount )  [virtual]
```

Deals damage to this character.

Parameters

amount	The amount of damage to deal

Reimplemented from CombatCharacter.

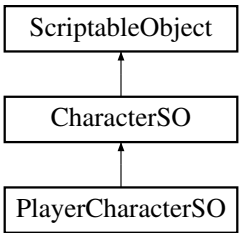
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Characters/PlayerCharacter.cs

5.72 PlayerCharacterSO Class Reference

Player character data class.

Inheritance diagram for PlayerCharacterSO:



Public Attributes

- int **respect**
- List< ItemSO > **startingItems**

5.72.1 Detailed Description

Player character data class.

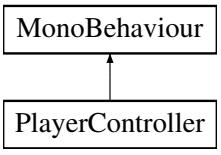
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/PlayerCharacterSO.cs

5.73 PlayerController Class Reference

Connects the player input and player character, issuing commands to the character as a reaction to player input.

Inheritance diagram for PlayerController:



Public Member Functions

- void **Init** ()

Properties

- bool **Aiming** `[get, set]`
- bool **ReadInput** `[get, set]`

  Should we be currently reading input?

5.73.1 Detailed Description

Connects the player input and player character, issuing commands to the character as a reaction to player input.

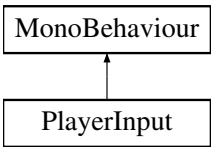
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PlayerController.cs

5.74 PlayerInput Class Reference

Handles player input which pertains to the controlling of the player character.

Inheritance diagram for PlayerInput:



Public Member Functions

- void **Init** ()

  Initializes the input events relevant to player input.

Public Attributes

- Vector2 **movement**
- Vector2 **mousePosition**
- Dictionary< EButtonDown, UnityEvent< EButtonDown > > **buttonDownEvents**
- Dictionary< EButtonUp, UnityEvent< EButtonUp > > **buttonUpEvents**
- bool **readInput**

5.74.1 Detailed Description

Handles player input which pertains to the controlling of the player character.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Input/PlayerInput.cs

5.75 PlayerInventory Class Reference

Player inventory wrapper class. The player inventory equals the items the player has currently equipped.

Public Member Functions

- **PlayerInventory** (PlayerCharacterSO playerData)
- bool AddItem (ItemSO item)

  Adds an item to the inventory - this also adds health to the player.
- int ReceiveDamage (int damage)
- bool HasSpace ()

  Is there space in the inventory for another item?
- bool ItemEquipped (int id)

  Checks whether an item is currently equipped.

Public Attributes

- UnityEvent **InventoryChanged**

Properties

- Item[] **Equipped** `[get]`

5.75.1 Detailed Description

Player inventory wrapper class. The player inventory equals the items the player has currently equipped.

5.75.2 Member Function Documentation

5.75.2.1 AddItem()

```
bool PlayerInventory.AddItem (
            ItemSO item )
```

Adds an item to the inventory - this also adds health to the player.

Parameters

item	Data of the item to add

Returns

Whether an item was added

5.75.2.2 HasSpace()

```
bool PlayerInventory.HasSpace ( )
```

Is there space in the inventory for another item?

Returns

If there is space

5.75.2.3 ItemEquipped()

```
bool PlayerInventory.ItemEquipped (
            int id )
```

Checks whether an item is currently equipped.

Parameters

id	The id of the item to check

Returns

5.75.2.4 ReceiveDamage()

```
int PlayerInventory.ReceiveDamage (
            int damage )
```

Parameters

damage	the amount of damage to receive

Returns

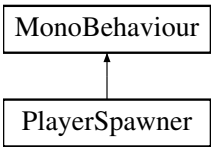
the remaining health of the player (i.e sum of hp of all items in inventory)

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Items/PlayerInventory.cs

5.76 PlayerSpawner Class Reference

Inheritance diagram for PlayerSpawner:



Public Member Functions

- PlayerCharacter SpawnPlayerAndInit (PlayerCharacter prefab, Transform parent=default)

  Should only be called at the beginning of the game - instantiates the player prefab under the specified transform and initializes the player character.

Static Public Member Functions

- static void PlacePlayer (PlayerCharacter player, Vector3 position)

  Places the player instance under the specified parent. Assumes the player has already been instantiated and initialized. Should be called at the beginning of each room after it has been generated.

5.76.1 Member Function Documentation

5.76.1.1 PlacePlayer()

```
static void PlayerSpawner.PlacePlayer (
            PlayerCharacter player,
            Vector3 position )  [static]
```

Places the player instance under the specified parent. Assumes the player has already been instantiated and initialized. Should be called at the beginning of each room after it has been generated.

Parameters

player	the player instance to spawn
position	the position at which it should be spawned

5.76.1.2 SpawnPlayerAndInit()

```
PlayerCharacter PlayerSpawner.SpawnPlayerAndInit (
            PlayerCharacter prefab,
            Transform parent = default )
```

Should only be called at the beginning of the game - instantiates the player prefab under the specified transform and initializes the player character.

Parameters

prefab	player prefab
parent	parent transform to instantiate under, transform of this object if empty

Returns

The instantiated and initiated instance of the player character

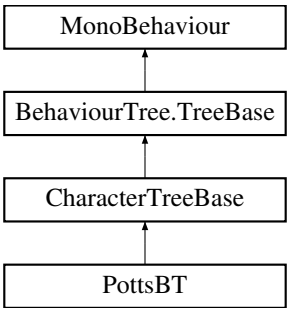
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PlayerSpawner.cs

5.77 PottsBT Class Reference

Behavioral tree for a mob from the first level.

Inheritance diagram for PottsBT:



Protected Member Functions

- override void Init ()

  Initiate every tree in this method - called from Start.

Additional Inherited Members

5.77.1 Detailed Description

Behavioral tree for a mob from the first level.

5.77.2 Member Function Documentation

5.77.2.1 Init()

```
override void PottsBT.Init ( )  [protected], [virtual]
```

Initiate every tree in this method - called from Start.

Implements BehaviourTree.TreeBase.

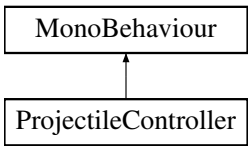
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/PottsBT.cs

5.78 ProjectileController Class Reference

Manages the shooting of projectiles for all ranged attacks.

Inheritance diagram for ProjectileController:



Public Member Functions

- void Init (AbilitySO data, string friendlyTag)

  Initializes the projectile.
- void Shoot (Vector2 force)

  Adds force to the projectile according to the given vector and the attack data.
- void ShootAt (Vector2 position)

  For projectiles shot from the sky - spawns the projectile at the given position. Expects an Animator component to handle enabling the hitbox as well as the destruction of this object.

5.78.1 Detailed Description

Manages the shooting of projectiles for all ranged attacks.

5.78.2 Member Function Documentation

5.78.2.1 Init()

```
void ProjectileController.Init (
            AbilitySO data,
            string friendlyTag )
```

Initializes the projectile.

Parameters

data	The data of the attack which spawned this projectile
friendlyTag	The tag of the game object which spawned this projectile

5.78.2.2 Shoot()

```
void ProjectileController.Shoot (
            Vector2 force )
```

Adds force to the projectile according to the given vector and the attack data.

Parameters

force	The force to add to the projectile

5.78.2.3 ShootAt()

```
void ProjectileController.ShootAt (
            Vector2 position )
```

For projectiles shot from the sky - spawns the projectile at the given position. Expects an Animator component to handle enabling the hitbox as well as the destruction of this object.

Parameters

position	The position to spawn at

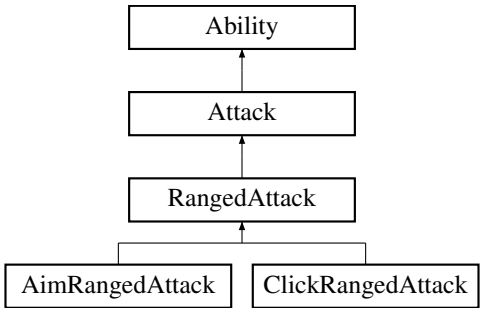
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/ProjectileController.cs

5.79 RangedAttack Class Reference

Base wrapper class for all ranged attacks.

Inheritance diagram for RangedAttack:



Protected Member Functions

- **RangedAttack** (CombatCharacter character, AttackSO data)
- void **SpawnProjectile** ()

Additional Inherited Members

5.79.1 Detailed Description

Base wrapper class for all ranged attacks.

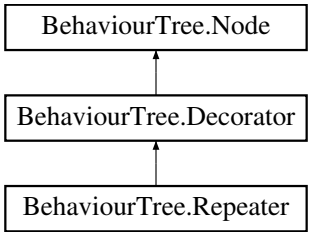
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/RangedAttack.cs

5.80 BehaviourTree.Repeater Class Reference

A decorator node which repeatedly processes its child after it returns a result (meaning either a success or a failure). Example usage: at the top of the tree to make the tree run continuously.

Inheritance diagram for BehaviourTree.Repeater:



Public Member Functions

- **Repeater** (Node child)
- override NodeStatus Update ()

  Updates the node according to the node's specific logic.

Additional Inherited Members

5.80.1 Detailed Description

A decorator node which repeatedly processes its child after it returns a result (meaning either a success or a failure). Example usage: at the top of the tree to make the tree run continuously.

5.80.2 Member Function Documentation

5.80.2.1 Update()

```
override NodeStatus BehaviourTree.Repeater.Update ( )  [virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

Implements BehaviourTree.Node.

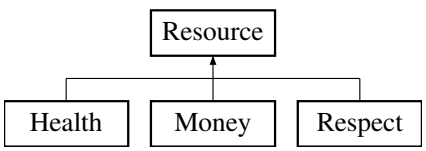
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.81 Resource Class Reference

Base wrapper class for all resources.

Inheritance diagram for Resource:



Public Member Functions

- **Resource** (int startingValue, int maxValue)
- int **GetCurrent** ()
- int **GetMax** ()
- bool **IsMaxed** ()
- int ChangeCurrent (int value)

  Adds the specified value to the current value of this resource, taking into account the min and max values of this resource.
- int **SetMax** (int value)
- void **CleanUp** ()

Public Attributes

- UnityEvent< int > **onChangedCurrent**
- UnityEvent< int > **onChangedMax**

5.81.1 Detailed Description

Base wrapper class for all resources.

5.81.2 Member Function Documentation

5.81.2.1 ChangeCurrent()

```
int Resource.ChangeCurrent (
            int value )
```

Adds the specified value to the current value of this resource, taking into account the min and max values of this resource.

Parameters

value	The value to add.

Returns

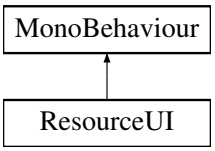
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Resources/Resource.cs

5.82 ResourceUI Class Reference

Handles the UI of any resource. Expects either a Slider or a Text component (or both) on one of this object's children.

Inheritance diagram for ResourceUI:



Public Member Functions

- void Init (Resource resource)

  Initializes this resource's UI. Should be called from a HUD manager for each resource's UI, or from a character spawner for world space UI.

5.82.1 Detailed Description

Handles the UI of any resource. Expects either a Slider or a Text component (or both) on one of this object's children.

5.82.2 Member Function Documentation

5.82.2.1 Init()

```
void ResourceUI.Init (
            Resource resource )
```

Initializes this resource's UI. Should be called from a HUD manager for each resource's UI, or from a character spawner for world space UI.

Parameters

resource	The resource whose UI we're initializing.

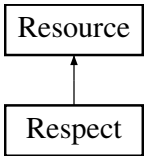
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/ResourceUI.cs

5.83 Respect Class Reference

Wrapper class for the respect resource. Exclusive to the player, represents the current respect value.

Inheritance diagram for Respect:



Public Member Functions

- **Respect** (int startingValue)

Additional Inherited Members

5.83.1 Detailed Description

Wrapper class for the respect resource. Exclusive to the player, represents the current respect value.

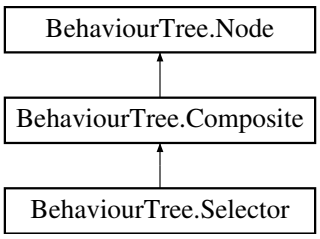
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Resources/Respect.cs

5.84 BehaviourTree.Selector Class Reference

A composite node which will process all its children in order - when a child reports success it reports success and does not process any further children. Equivalent to an OR operation - returns success if any of its children returned success.

Inheritance diagram for BehaviourTree.Selector:



Public Member Functions

- **Selector** (List< Node > children)
- override NodeStatus Update ()

  Updates the node according to the node's specific logic.

Additional Inherited Members

5.84.1 Detailed Description

A composite node which will process all its children in order - when a child reports success it reports success and does not process any further children. Equivalent to an OR operation - returns success if any of its children returned success.

5.84.2 Member Function Documentation

5.84.2.1 Update()

```
override NodeStatus BehaviourTree.Selector.Update ( )  [virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

Implements BehaviourTree.Node.

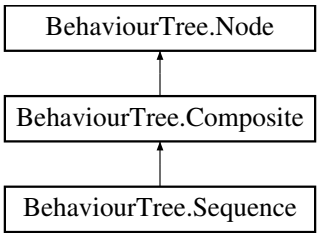
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.85 BehaviourTree.Sequence Class Reference

A composite node which will process all its children in order - it proceeds to the next child when the previous one reports success. Equivalent to an AND operation - returns success only if all of its children returned success.

Inheritance diagram for BehaviourTree.Sequence:



Public Member Functions

- **Sequence** (List< Node > children)
- override NodeStatus Update ()

  Updates the node according to the node's specific logic.

Additional Inherited Members

5.85.1 Detailed Description

A composite node which will process all its children in order - it proceeds to the next child when the previous one reports success. Equivalent to an AND operation - returns success only if all of its children returned success.

5.85.2 Member Function Documentation

5.85.2.1 Update()

```
override NodeStatus BehaviourTree.Sequence.Update ( )  [virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

Implements BehaviourTree.Node.

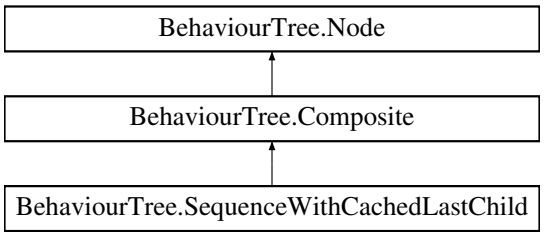
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.86 BehaviourTree.SequenceWithCachedLastChild Class Reference

A composite node which will process all its children in order - it proceeds to the next child when the previous one reports success. Equivalent to an AND operation - returns success only if all of its children returned success. When entering the sequence proceeds from the child processed in the previous tick - useful for a sequence of tasks that need to be completed one after another.

Inheritance diagram for BehaviourTree.SequenceWithCachedLastChild:



Public Member Functions

- **SequenceWithCachedLastChild** (List< Node > children)
- override NodeStatus Update ()

  Updates the node according to the node's specific logic.

Additional Inherited Members

5.86.1 Detailed Description

A composite node which will process all its children in order - it proceeds to the next child when the previous one reports success. Equivalent to an AND operation - returns success only if all of its children returned success. When entering the sequence proceeds from the child processed in the previous tick - useful for a sequence of tasks that need to be completed one after another.

5.86.2 Member Function Documentation

5.86.2.1 Update()

```
override NodeStatus BehaviourTree.SequenceWithCachedLastChild.Update ( )  [virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

Implements BehaviourTree.Node.

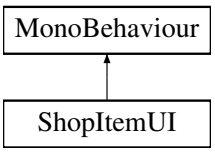
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.87 ShopItemUI Class Reference

Manager for the UI of items in the shop.

Inheritance diagram for ShopItemUI:



Public Member Functions

- void **Init** (ItemSO item, bool canInteract, UnityAction< ItemSO > onBuyPressed)
- void UpdateItem (PlayerCharacter playerCharacter)

  Updates the item UI according to the current player state.
- void **Hide** ()

  Hides this item from the shop UI.

5.87.1 Detailed Description

Manager for the UI of items in the shop.

5.87.2 Member Function Documentation

5.87.2.1 UpdateItem()

```
void ShopItemUI.UpdateItem (
            PlayerCharacter playerCharacter )
```

Updates the item UI according to the current player state.

Parameters

playerCharacter	The current player character instance

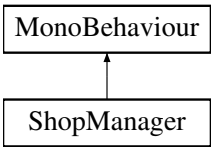
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/ShopItemUI.cs

5.88 ShopManager Class Reference

Class that manages the shop UI and connected events.

Inheritance diagram for ShopManager:



Public Member Functions

- void **Init** (GameManager gameManager, MerchantCharacter merchant)
- void ShowShop (ItemSO[ ] items)

  Opens the shop overlay.
- void **Close** ()

  Closes the shop overlay.

5.88.1 Detailed Description

Class that manages the shop UI and connected events.

5.88.2 Member Function Documentation

5.88.2.1 ShowShop()

```
void ShopManager.ShowShop (
            ItemSO[] items )
```

Opens the shop overlay.

Parameters

items	The data of the items to sell. (Expects an array of 3 items or less.)

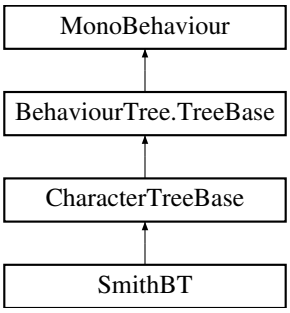
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/ShopManager.cs

5.89 SmithBT Class Reference

Behavioral tree for the boss of the first level.

Inheritance diagram for SmithBT:



Protected Member Functions

- override void Init ()

  Initiate every tree in this method - called from Start.

Additional Inherited Members

5.89.1 Detailed Description

Behavioral tree for the boss of the first level.

5.89.2 Member Function Documentation

5.89.2.1 Init()

```
override void SmithBT.Init ( )  [protected], [virtual]
```

Initiate every tree in this method - called from Start.

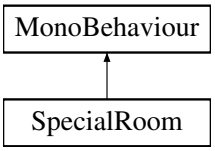
Implements BehaviourTree.TreeBase.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Trees/SmithBT.cs

5.90 SpecialRoom Class Reference

Inheritance diagram for SpecialRoom:



Properties

- Transform **Entrance** `[get]`
- Interactable **ExitTrigger** `[get]`
- Vector3 **EnemySpawn** `[get]`
- Vector3[ ] **EnemyPatrolPoints** `[get]`

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PCG/SpecialRoom.cs

5.91 SpriteImportSettings Struct Reference

Public Attributes

- int **ppu**
- Pivot **pivot**

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/SpriteSetUp.cs

5.92 SpriteOrganizer Class Reference

Used for general sprite organization.

Public Member Functions

- void Delete (string dir, string nameContains)

  Deletes all files with the specified parameters.
- void Rename (string dir, string renameFrom, string renameTo)

  Renames all files with the specified parameters.

5.92.1 Detailed Description

Used for general sprite organization.

5.92.2 Member Function Documentation

5.92.2.1 Delete()

```
void SpriteOrganizer.Delete (
            string dir,
            string nameContains )
```

Deletes all files with the specified parameters.

Parameters

dir	The directory with the files
nameContains	The substring the names of the files to be deleted have to contain

5.92.2.2 Rename()

```
void SpriteOrganizer.Rename (
            string dir,
            string renameFrom,
            string renameTo )
```

Renames all files with the specified parameters.

Parameters

dir	The director with the files
renameFrom	Substring to be replaced
renameTo	The replacement substring

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/SpriteOrganizer.cs

5.93 SpriteSetUp Class Reference

Class which handles setting up the sprites for use in the game.

Public Member Functions

- **SpriteSetUp** (SpriteImportSettings settings)
- void SetSpriteImportSettings (string dir, string nameContains)

  Sets import settings of all sprites in given directory, containing string specified in their file name.

5.93.1 Detailed Description

Class which handles setting up the sprites for use in the game.

5.93.2 Member Function Documentation

5.93.2.1 SetSpriteImportSettings()

```
void SpriteSetUp.SetSpriteImportSettings (
            string dir,
            string nameContains )
```

Sets import settings of all sprites in given directory, containing string specified in their file name.

Parameters

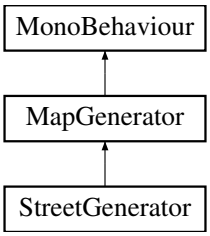
dir	Directory of sprites to modify
nameContains	Only modify sprites with this in their file name. Modifies all sprites if not specified.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Editor/SpriteSetUp.cs

5.94 StreetGenerator Class Reference

Inheritance diagram for StreetGenerator:



Public Member Functions

- override void Generate ()

  Generate a room for the game.

Public Attributes

- List< Tile > **grassTile**
- List< Tile > **pavementTile**
- List< Tile > **asphaltTile**
- List< GameObject > **greenery**
- List< Tile > **buildingsVert**
- List< Tile > **buildingsHor**
- int **greenTensity** = 1

Protected Member Functions

- override void SetUpParameters ()

Additional Inherited Members

5.94.1 Member Function Documentation

5.94.1.1 Generate()

```
override void StreetGenerator.Generate ( )  [virtual]
```

Generate a room for the game.

Reimplemented from MapGenerator.

5.94.1.2 SetUpParameters()

```
override void StreetGenerator.SetUpParameters ( )  [protected], [virtual]
```

Implements MapGenerator.

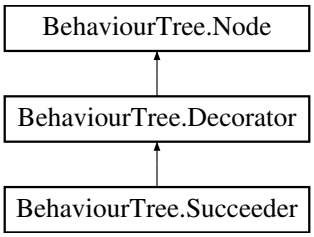
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PCG/StreetGenerator.cs

5.95 BehaviourTree.Succeeder Class Reference

A decorator node which always returns success.

Inheritance diagram for BehaviourTree.Succeeder:



Public Member Functions

- **Succeeder** (Node child)
- override NodeStatus Update ()

  Updates the node according to the node's specific logic.

Additional Inherited Members

5.95.1 Detailed Description

A decorator node which always returns success.

5.95.2 Member Function Documentation

5.95.2.1 Update()

```
override NodeStatus BehaviourTree.Succeeder.Update ( )  [virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

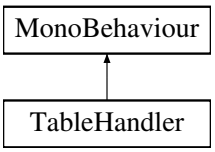
Implements BehaviourTree.Node.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/NodeBase.cs

5.96 TableHandler Class Reference

Inheritance diagram for TableHandler:



Public Member Functions

- void **SetTableVariant** (bool down, Sprite mainTable=null, Sprite supportTable=null, bool chair=true)

Public Attributes

- int **width** = 2
- int **height** = 1
- GameObject **chairUp**
- GameObject **chairDown**

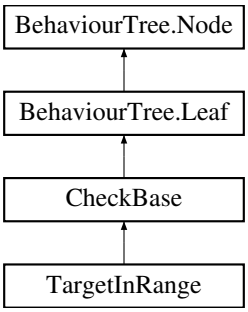
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/PCG/TableHandler.cs

5.97 TargetInRange Class Reference

Checks whether target is within specified range.

Inheritance diagram for TargetInRange:



Public Member Functions

- TargetInRange (CharacterTreeBase characterBT, string targetName, float range, EObstacleFilter obstacle↩ Filter=EObstacleFilter.None, string debugName="")

  Creates a task instance.

Protected Member Functions

- override bool Check ()

Additional Inherited Members

5.97.1 Detailed Description

Checks whether target is within specified range.

5.97.2 Constructor & Destructor Documentation

5.97.2.1 TargetInRange()

```
TargetInRange.TargetInRange (
            CharacterTreeBase characterBT,
            string targetName,
            float range,
            EObstacleFilter obstacleFilter = EObstacleFilter.None,
            string debugName = "" )
```

Creates a task instance.

Parameters

characterBT	The behavioral tree of this character
targetName	The name of the target - this task will try to retrieve the target from shared memory.
range	How far can the character see
obstacleFilter	Specifies which objects in the scene should obstruct this character's line of sight
debugName	Used in ToString() for debug purposes

5.97.3 Member Function Documentation

5.97.3.1 Check()

```
override bool TargetInRange.Check ( )  [protected], [virtual]
```

Implements CheckBase.

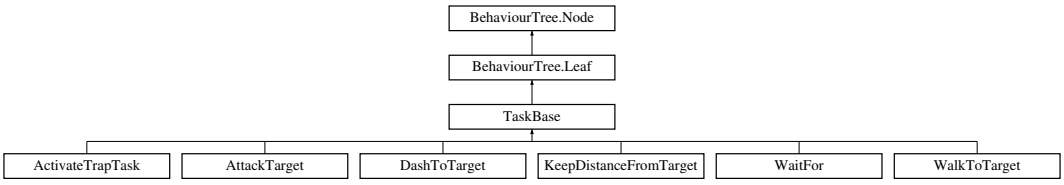
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/TargetInRange.cs

5.98 TaskBase Class Reference

Base class for all tasks. Tasks update the character's animation, physics, etc.

Inheritance diagram for TaskBase:



Public Member Functions

- override NodeStatus Update ()

  Updates the node according to the node's specific logic.

Protected Member Functions

- TaskBase (CharacterTreeBase characterBT, string debugName="")
- abstract void **OnBegin** ()
- abstract void **OnContinue** ()
- void OnEnd (bool taskSuccess)

  Every task has to call this after finishing !

Protected Attributes

- CharacterTreeBase **bt**

Additional Inherited Members

5.98.1 Detailed Description

Base class for all tasks. Tasks update the character's animation, physics, etc.

5.98.2 Constructor & Destructor Documentation

5.98.2.1 TaskBase()

```
TaskBase.TaskBase (
            CharacterTreeBase characterBT,
            string debugName = "" )  [protected]
```

Parameters

characterBT	The behavioral tree of this character
debugName	Used in ToString() for debug purposes

5.98.3 Member Function Documentation

5.98.3.1 OnEnd()

```
void TaskBase.OnEnd (
            bool taskSuccess )  [protected]
```

Every task has to call this after finishing !

Parameters

taskSuccess	

5.98.3.2 Update()

```
override NodeStatus TaskBase.Update ( )  [virtual]
```

Updates the node according to the node's specific logic.

Returns

The node status

Implements BehaviourTree.Node.

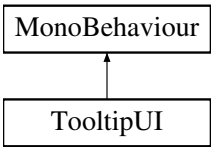
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/TaskBase.cs

5.99 TooltipUI Class Reference

Manager for UI tooltips.

Inheritance diagram for TooltipUI:



Public Member Functions

- void **Init** (string text="")
- void SetText (string text)

  Sets the text of the tooltip.
- void ShowToolTip (bool show)

  Show the tooltip.

5.99.1 Detailed Description

Manager for UI tooltips.

5.99.2 Member Function Documentation

5.99.2.1 SetText()

```
void TooltipUI.SetText (
            string text )
```

Sets the text of the tooltip.

Parameters

text	The string to set the text to

5.99.2.2 ShowToolTip()

```
void TooltipUI.ShowToolTip (
            bool show )
```

Show the tooltip.

Parameters

show	

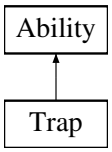
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/TooltipUI.cs

5.100 Trap Class Reference

Class for the trap ability.

Inheritance diagram for Trap:



Public Member Functions

- Trap (CombatCharacter character, AbilitySO data, EAbilityType type=EAbilityType.special)
  Creates a trap instance and instantiates a trap in the scene according to the specified parameters.
- override void OnBegin ()
  Begins performing the ability.
- override void OnEnd ()
  Ends the performing of the ability.

Properties

- TrapSO **Data** `[get]`

Additional Inherited Members

5.100.1 Detailed Description

Class for the trap ability.

5.100.2 Constructor & Destructor Documentation

5.100.2.1 Trap()

```
Trap.Trap (
            CombatCharacter character,
            AbilitySO data,
            EAbilityType type = EAbilityType.special )
```

Creates a trap instance and instantiates a trap in the scene according to the specified parameters.

Parameters

character	The character to which the attack belongs
data	The trap data
type	

5.100.3 Member Function Documentation

5.100.3.1 OnBegin()

```
override void Trap.OnBegin ( )  [virtual]
```

Begins performing the ability.

Reimplemented from Ability.

5.100.3.2 OnEnd()

```
override void Trap.OnEnd ( )  [virtual]
```

Ends the performing of the ability.

Reimplemented from Ability.

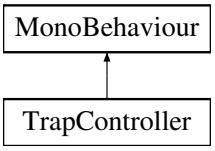
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/Trap.cs

5.101 TrapController Class Reference

Main controller for every trap in the scene.

Inheritance diagram for TrapController:



Public Member Functions

- void Init (TrapSO data)

  Initializes the controller.
- void **ActivateTrap** ()

  Plays an animation which activates the trap.
- void ActivateTrap (Vector3 target)

  Activates the trap while taking into account the target.

5.101.1 Detailed Description

Main controller for every trap in the scene.

5.101.2 Member Function Documentation

5.101.2.1 ActivateTrap()

```
void TrapController.ActivateTrap (
            Vector3 target )
```

Activates the trap while taking into account the target.

Parameters

target	The target of the trap

5.101.2.2 Init()

```
void TrapController.Init (
            TrapSO data )
```

Initializes the controller.

Parameters

data	The trap data

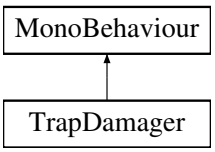
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/TrapController.cs

5.102 TrapDamager Class Reference

Handles the behaviour when the trap is triggered, ex. whether it should deal damage.

Inheritance diagram for TrapDamager:



Public Member Functions

- void Init (TrapSO data)
  Initializes the damager.

5.102.1 Detailed Description

Handles the behaviour when the trap is triggered, ex. whether it should deal damage.

5.102.2 Member Function Documentation

5.102.2.1 Init()

```
void TrapDamager.Init (
            TrapSO data )
```

Initializes the damager.

Parameters

data	The trap data

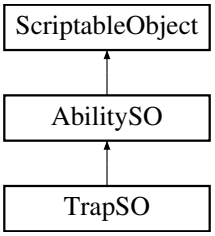
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/TrapDamager.cs

5.103 TrapSO Class Reference

Trap data class.

Inheritance diagram for TrapSO:



Public Member Functions

- Trap **GetTrap** (CombatCharacter character)

Public Attributes

- float **activationTime**
- Vector3 **spawnPosition**
- TrapController **trapControllerPrefab**
- ProjectileController **projectilePrefab**

5.103.1 Detailed Description

Trap data class.

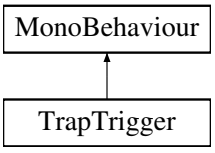
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/ScriptableObjects/TrapSO.cs

5.104 TrapTrigger Class Reference

Triggers the trap this is a child of when the player enters/stands in this collider.

Inheritance diagram for TrapTrigger:



5.104.1 Detailed Description

Triggers the trap this is a child of when the player enters/stands in this collider.

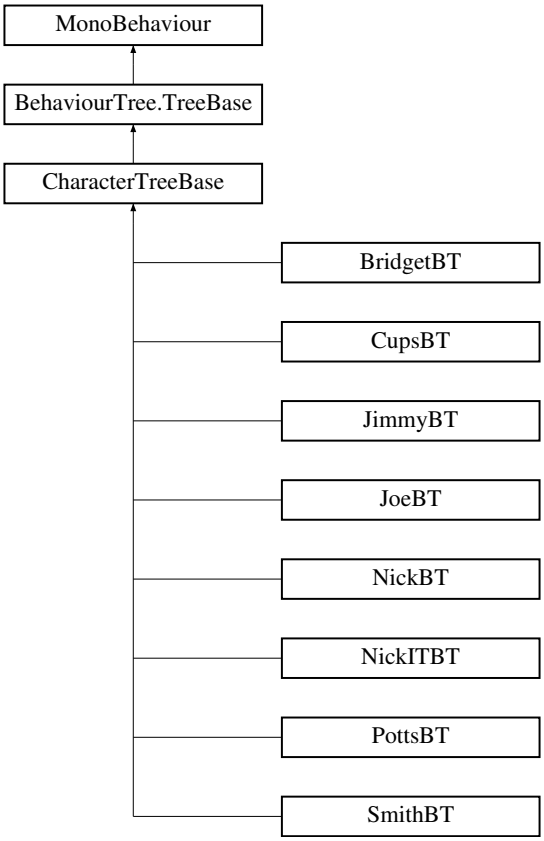
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Abilities/TrapTrigger.cs

5.105 BehaviourTree.TreeBase Class Reference

Base class for all behaviour trees.

Inheritance diagram for BehaviourTree.TreeBase:



Protected Member Functions

- abstract void Init ()

  Initiate every tree in this method - called from Start.

Properties

- Node **Root** `[get, protected set]`

5.105.1 Detailed Description

Base class for all behaviour trees.

5.105.2 Member Function Documentation

5.105.2.1 Init()

```
abstract void BehaviourTree.TreeBase.Init ( )  [protected], [pure virtual]
```

Initiate every tree in this method - called from Start.

Implemented in BridgetBT, CupsBT, JimmyBT, JoeBT, NickBT, NickITBT, PottsBT, and SmithBT.

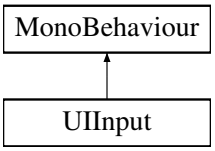
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/BehaviourTreeBase/TreeBase.cs

5.106 UIInput Class Reference

Class for input which should be independent of the player character.

Inheritance diagram for UIInput:



Public Member Functions

- void **Init** ()

  Initializes UI input events and starts detecting gamepad input.

Static Public Member Functions

- static void **TrySelectFirstButton** ()

  If a gamepad is connected, tries to select the first button it can find.

Public Attributes

- Dictionary< EUIButton, UnityEvent< EUIButton > > **buttonEvents**

Static Public Attributes

- static bool **GamepadConnected**

5.106.1 Detailed Description

Class for input which should be independent of the player character.

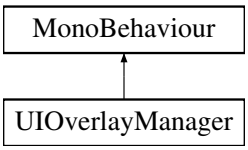
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/Input/UIInput.cs

5.107 UIOverlayManager Class Reference

Class for any UI overlay with buttons relating to high-level game logic - ex. menus, game over screen.

Inheritance diagram for UIOverlayManager:



Public Member Functions

- void **Init** (GameManager gameManager)
- void ChangeTitle (string text)

  Changes the title text of the overlay, if present.
- void **StartGame** ()

  Begin game - only call from main menu.
- void **ResumeGame** ()

  After pausing game - only call from pause menu.
- void **BackToMain** ()

  Return to main menu.
- void **ExitGame** ()
- void **Close** ()

5.107.1 Detailed Description

Class for any UI overlay with buttons relating to high-level game logic - ex. menus, game over screen.

5.107.2 Member Function Documentation

5.107.2.1 ChangeTitle()

```
void UIOverlayManager.ChangeTitle (
            string text )
```

Changes the title text of the overlay, if present.

Parameters

text	

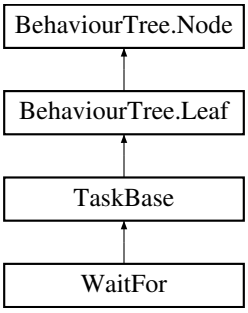
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/UI/UIOverlayManager.cs

5.108 WaitFor Class Reference

A task which makes the character wait for a specified time.

Inheritance diagram for WaitFor:



Public Member Functions

- WaitFor (CharacterTreeBase characterBT, float waitFor, string debugName="")
  Creates a new wait task.

Protected Member Functions

- override void OnBegin ()
- override void OnContinue ()

Additional Inherited Members

5.108.1 Detailed Description

A task which makes the character wait for a specified time.

5.108.2 Constructor & Destructor Documentation

5.108.2.1 WaitFor()

```
WaitFor.WaitFor (
            CharacterTreeBase characterBT,
            float waitFor,
            string debugName = "" )
```

Creates a new wait task.

Parameters

characterBT	The behavioral tree of this character
waitFor	How long do we want to wait for
debugName	Used in ToString() for debug purposes

5.108.3 Member Function Documentation

5.108.3.1 OnBegin()

```
override void WaitFor.OnBegin ( )  [protected], [virtual]
```

Implements TaskBase.

5.108.3.2 OnContinue()

```
override void WaitFor.OnContinue ( )  [protected], [virtual]
```

Implements TaskBase.

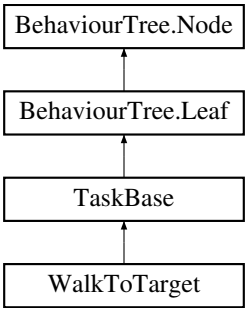
The documentation for this class was generated from the following file:

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/WaitFor.cs

5.109 WalkToTarget Class Reference

A task which makes the character walk towards a given target.

Inheritance diagram for WalkToTarget:



Public Member Functions

- **WalkToTarget** (CharacterTreeBase characterBT, Vector3 target, string debugName="")
- **WalkToTarget** (CharacterTreeBase characterBT, string targetName, string debugName="")
- override string **ToString** ()

Protected Member Functions

- override void OnBegin ()
- override void OnContinue ()

Additional Inherited Members

5.109.1 Detailed Description

A task which makes the character walk towards a given target.

5.109.2 Member Function Documentation

5.109.2.1 OnBegin()

```
override void WalkToTarget.OnBegin ( )  [protected], [virtual]
```

Implements TaskBase.

5.109.2.2 OnContinue()

```
override void WalkToTarget.OnContinue ( )  [protected], [virtual]
```

Implements TaskBase.

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

- D:/UnityProjects/Operation Okapi/Assets/Scripts/AI/Tasks/WalkToTarget.cs

Index

Ability, 15
    OnBegin, 16
    OnContinue, 16
    OnEnd, 16
AbilityOnCD, 17
    AbilityOnCD, 17
    Check, 18
AbilitySO, 18
ActivateTrap
    CombatCharacter, 41
    TrapController, 124
ActivateTrapTask, 19
    ActivateTrapTask, 19
    OnBegin, 20
    OnContinue, 20
AddItem
    PlayerInventory, 96
AIGenerator, 20
    GenerateEnemySpawnPoints, 21
    Init, 21
AimRangedAttack, 22
    AimRangedAttack, 22
    OnBegin, 23
    OnEnd, 23
AnimationClipGenerator, 23
    GenerateAllAnimations, 24
    GenerateAnimations, 24
AnimationClipProperties, 24
AnimatorGenerator, 25
    GenerateAnimator, 25
AnimatorTransitionProperties, 26
Attack, 27
    CombatCharacter, 41
    OnBegin, 27
    PlayerCharacter, 92
AttackFrames, 28
AttackSO, 28
AttackStateMachine, 29
AttackTarget, 30
    AttackTarget, 30
    CombatCharacter, 42
    OnBegin, 31
    OnContinue, 31
AudioManager, 31

BehaviourTree, 13
BehaviourTree.Composite, 44
BehaviourTree.Decorator, 50
BehaviourTree.Inverter, 65
    Update, 66
BehaviourTree.Leaf, 74
BehaviourTree.Node, 88
    Update, 89
BehaviourTree.Repeater, 102
    Update, 103
BehaviourTree.Selector, 106
    Update, 107
BehaviourTree.Sequence, 107
    Update, 108
BehaviourTree.SequenceWithCachedLastChild, 108
    Update, 109
BehaviourTree.Succeeder, 116
    Update, 116
BehaviourTree.TreeBase, 127
    Init, 128
BridgetBT, 32
    Init, 32

ChangeCurrent
    Resource, 104
ChangeTitle
    UIOverlayManager, 129
Character, 33
    Init, 33
CharacterSO, 34
CharacterTreeBase, 34
    GetDashBT, 36
Check
    AbilityOnCD, 18
    FindTargetInRange, 57
    TargetInRange, 119
CheckBase, 36
    CheckBase, 37
    Update, 37
ClickRangedAttack, 38
    ClickRangedAttack, 38
    OnBegin, 39
CollectItem
    PlayerCharacter, 92
CollectMoney
    PlayerCharacter, 92
ColorPaletteSO, 39
CombatCharacter, 40
    ActivateTrap, 41
    Attack, 41
    AttackTarget, 42
    Die, 42
    Init, 42
    Move, 43
    Rotate, 43

SetMovementSpeed, 43
TakeDamage, 44
CupsBT, 45
Init, 45

Dash, 46
Dash, 46
OnBegin, 47
OnContinue, 47
OnEnd, 47
DashSO, 48
DashToTarget, 48
DashToTarget, 49
OnBegin, 49
OnContinue, 49
DeathStateMachine, 50
Delete
SpriteOrganizer, 113
Die
CombatCharacter, 42
EnemyCharacter, 53
IDamagable, 64
DroppedItem, 51
Init, 51, 52

EnemyCharacter, 52
Die, 53
Freeze, 53
EnemyCharacterSO, 53
EnemySpawner, 54
FreezeEnemies, 54
Init, 55
SpawnEnemy, 55

FindTargetInRange, 55
Check, 57
FindTargetInRange, 56
FollowTarget, 57
Init, 57
Freeze
EnemyCharacter, 53
FreezeEnemies
EnemySpawner, 54
FreezeLevel
LevelManager, 75

GameEnd
GameManager, 59
GameManager, 58
GameEnd, 59
PauseGame, 59
Generate
MallGenerator, 79
MapGenerator, 81
OfficeRoomGenerator, 90
StreetGenerator, 115
GenerateAllAnimations
AnimationClipGenerator, 24
GenerateAnimations
AnimationClipGenerator, 24
GenerateAnimator
AnimatorGenerator, 25
GenerateEnemySpawnPoints
AIGenerator, 21
GetDashBT
CharacterTreeBase, 36
GetEnemiesToSpawn
LevelSO, 77, 78
GetGrid
MapGenerator, 81
GetGridTileWorldCoordinates
MapGenerator, 81
GetGridTileWorldCoordinatesMiddle
MapGenerator, 82
GetSmallGridTileWorldCoordinates
MapGenerator, 82
GFXSetUpWindow, 59

HandleCollision
HitBoxController, 62
HasSpace
PlayerInventory, 97
Health, 60
HealthUI, 61
HitBoxController, 61
HandleCollision, 62
Init, 62
HUDManager, 62
Init, 63

IDamagable, 63
Die, 64
TakeDamage, 64
Init
AIGenerator, 21
BehaviourTree.TreeBase, 128
BridgetBT, 32
Character, 33
CombatCharacter, 42
CupsBT, 45
DroppedItem, 51, 52
EnemySpawner, 55
FollowTarget, 57
HitBoxController, 62
HUDManager, 63
JimmyBT, 71
JoeBT, 72
MerchantCharacter, 85
NickBT, 87
NickITBT, 88
PlayerCharacter, 93
PottsBT, 100
ProjectileController, 101
ResourceUI, 105
SmithBT, 112
TrapController, 124
TrapDamager, 125
Interactable, 64

SetTooltip, 65
IPushable, 67
Push, 67
Item, 68
Item, 68
ReceiveDamage, 68
ItemEffect, 69
ItemEquipped
PlayerInventory, 97
ItemSetUpWindow, 69
ItemSO, 70
ItemUI, 70

JimmyBT, 71
Init, 71
JoeBT, 72
Init, 72

KeepDistanceFromTarget, 73
OnBegin, 73
OnContinue, 73

LevelManager, 75
FreezeLevel, 75
LoadFinalRoom, 76
LoadRoom, 76
LevelSO, 77
GetEnemiesToSpawn, 77, 78
LoadFinalRoom
LevelManager, 76
LoadRoom
LevelManager, 76

MallGenerator, 78
Generate, 79
SetUpParameters, 79
MapGenerator, 79
Generate, 81
GetGrid, 81
GetGridTileWorldCoordinates, 81
GetGridTileWorldCoordinatesMiddle, 82
GetSmallGridTileWorldCoordinates, 82
MapGenerator.GridTile, 60
MeleeAttack, 83
MeleeAttack, 83
OnBegin, 84
MerchantCharacter, 84
Init, 85
Sell, 85
MerchantCharacterSO, 85
Money, 86
Move
CombatCharacter, 43

NickBT, 86
Init, 87
NickITBT, 87
Init, 88

OfficeRoomGenerator, 89
Generate, 90
SetUpParameters, 90
UnityToScriptCoord, 90
OnBegin
Ability, 16
ActivateTrapTask, 20
AimRangedAttack, 23
Attack, 27
AttackTarget, 31
ClickRangedAttack, 39
Dash, 47
DashToTarget, 49
KeepDistanceFromTarget, 73
MeleeAttack, 84
Trap, 123
WaitFor, 131
WalkToTarget, 132
OnContinue
Ability, 16
ActivateTrapTask, 20
AttackTarget, 31
Dash, 47
DashToTarget, 49
KeepDistanceFromTarget, 73
WaitFor, 131
WalkToTarget, 132
OnEnd
Ability, 16
AimRangedAttack, 23
Dash, 47
TaskBase, 120
Trap, 123

PauseGame
GameManager, 59
Pivot, 91
PlacePlayer
PlayerSpawner, 98
PlayerCharacter, 91
Attack, 92
CollectItem, 92
CollectMoney, 92
Init, 93
Push, 93
TakeDamage, 93
PlayerCharacterSO, 94
PlayerController, 94
PlayerInput, 95
PlayerInventory, 96
AddItem, 96
HasSpace, 97
ItemEquipped, 97
ReceiveDamage, 97
PlayerSpawner, 98
PlacePlayer, 98
SpawnPlayerAndInit, 99
PottsBT, 99
Init, 100
ProjectileController, 100

Init, 101
Shoot, 101
ShootAt, 101
Push
IPushable, 67
PlayerCharacter, 93

RangedAttack, 102
ReceiveDamage
Item, 68
PlayerInventory, 97
Rename
SpriteOrganizer, 113
Resource, 103
ChangeCurrent, 104
ResourceUI, 105
Init, 105
Respect, 105
Rotate
CombatCharacter, 43

Sell
MerchantCharacter, 85
SetMovementSpeed
CombatCharacter, 43
SetSpriteImportSettings
SpriteSetUp, 114
SetText
TooltipUI, 121
SetTooltip
Interactable, 65
SetUpParameters
MallGenerator, 79
OfficeRoomGenerator, 90
StreetGenerator, 115
Shoot
ProjectileController, 101
ShootAt
ProjectileController, 101
ShopItemUI, 109
UpdateItem, 110
ShopManager, 110
ShowShop, 111
ShowShop
ShopManager, 111
ShowToolTip
TooltipUI, 122
SmithBT, 111
Init, 112
SpawnEnemy
EnemySpawner, 55
SpawnPlayerAndInit
PlayerSpawner, 99
SpecialRoom, 112
SpriteImportSettings, 112
SpriteOrganizer, 113
Delete, 113
Rename, 113
SpriteSetUp, 114
SetSpriteImportSettings, 114
StreetGenerator, 114
Generate, 115
SetUpParameters, 115

TableHandler, 117
TakeDamage
CombatCharacter, 44
IDamagable, 64
PlayerCharacter, 93
TargetInRange, 118
Check, 119
TargetInRange, 118
TaskBase, 119
OnEnd, 120
TaskBase, 120
Update, 120
TooltipUI, 121
SetText, 121
ShowToolTip, 122
Trap, 122
OnBegin, 123
OnEnd, 123
Trap, 123
TrapController, 124
ActivateTrap, 124
Init, 124
TrapDamager, 125
Init, 125
TrapSO, 126
TrapTrigger, 126

UIInput, 128
UIOverlayManager, 129
ChangeTitle, 129
UnityToScriptCoord
OfficeRoomGenerator, 90
Update
BehaviourTree.Inverter, 66
BehaviourTree.Node, 89
BehaviourTree.Repeater, 103
BehaviourTree.Selector, 107
BehaviourTree.Sequence, 108
BehaviourTree.SequenceWithCachedLastChild, 109
BehaviourTree.Succeeder, 116
CheckBase, 37
TaskBase, 120
UpdateItem
ShopItemUI, 110

WaitFor, 130
OnBegin, 131
OnContinue, 131
WaitFor, 130
WalkToTarget, 131
OnBegin, 132
OnContinue, 132