

Code Documentation – Team Olive

Generated by Doxygen 1.8.20

1 Namespace Index	1
1.1 Packages	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List.....	5
4 File Index	7
4.1 File List.....	7
5 Namespace Documentation	9
5.1 Roguelike_Dungeon Namespace Reference.....	9
6 Class Documentation	11
6.1 breakables Class Reference.....	11
6.1.1 Detailed Description	11
6.1.2 Member Data Documentation	11
6.1.2.1 brokenPieces	12
6.1.2.2 itemDropPercent	12
6.1.2.3 itemsToDrop	12
6.1.2.4 maxPieces	12
6.1.2.5 shouldDropItem.....	12
6.2 brokenPieces Class Reference.....	12
6.2.1 Detailed Description	13
6.2.2 Member Data Documentation	13
6.2.2.1 deceleration	13
6.2.2.2 fadeSpeed	13
6.2.2.3 lifetime	13
6.2.2.4 moveSpeed.....	13
6.2.2.5 theSR	13
6.3 CameraController Class Reference	14
6.3.1 Detailed Description	14
6.3.2 Member Function Documentation.....	14
6.3.2.1 ChangeTarget().....	14
6.3.3 Member Data Documentation	14
6.3.3.1 instance	15
6.3.3.2 moveSpeed.....	15
6.3.3.3 target	15
6.4 CharacterSelectManager Class Reference	15
6.4.1 Detailed Description	15
6.4.2 Member Data Documentation	16
6.4.2.1 activeCharSelect	16

6.4.2.2 activePlayer	16
6.4.2.3 instance	16
6.5 CharacterSelector Class Reference	16
6.5.1 Detailed Description	16
6.5.2 Member Data Documentation	16
6.5.2.1 message	17
6.5.2.2 playerToSpawn	17
6.6 DamagePlayer Class Reference	17
6.6.1 Detailed Description	17
6.7 EnemyBullet Class Reference	17
6.7.1 Detailed Description	18
6.7.2 Member Data Documentation	18
6.7.2.1 speed	18
6.8 EnemyController Class Reference	18
6.8.1 Detailed Description	19
6.8.2 Member Function Documentation	19
6.8.2.1 DamageEnemy()	19
6.8.3 Member Data Documentation	20
6.8.3.1 anim	20
6.8.3.2 bullet	20
6.8.3.3 canMove	20
6.8.3.4 deathSplatters	20
6.8.3.5 firePoint	20
6.8.3.6 fireRate	20
6.8.3.7 health	21
6.8.3.8 hitEffect	21
6.8.3.9 instance	21
6.8.3.10 moveSpeed	21
6.8.3.11 patrolPoints	21
6.8.3.12 pauseLength	21
6.8.3.13 rangeToChaseBot	21
6.8.3.14 rangeToChasePlayer	22
6.8.3.15 runAwayRange	22
6.8.3.16 shootRange	22
6.8.3.17 shouldChaseBot	22
6.8.3.18 shouldChasePlayer	22
6.8.3.19 shouldPatrol	22
6.8.3.20 shouldRunAway	22
6.8.3.21 shouldShoot	23
6.8.3.22 shouldWander	23
6.8.3.23 theBody	23
6.8.3.24 theRB	23

6.8.3.25 wanderLength	23
6.8.3.26 Zzzz	23
6.9 enemySleepAnimation Class Reference	23
6.9.1 Detailed Description	24
6.9.2 Member Function Documentation.....	24
6.9.2.1 enemyToSleep()	24
6.9.3 Member Data Documentation	24
6.9.3.1 instance	24
6.9.3.2 sleepingEnemy.....	24
6.9.3.3 theSR	25
6.10 essayPageController Class Reference.....	25
6.10.1 Detailed Description.....	25
6.10.2 Member Function Documentation.....	25
6.10.2.1 pickupPage()	25
6.10.3 Member Data Documentation.....	26
6.10.3.1 currentNumPages.....	26
6.10.3.2 essayPageUI.....	26
6.10.3.3 instance	26
6.10.3.4 maxNumPages.....	26
6.11 essayPages Class Reference.....	26
6.11.1 Detailed Description.....	27
6.11.2 Member Data Documentation.....	27
6.11.2.1 instance	27
6.11.2.2 test.....	27
6.12 Gun Class Reference	27
6.12.1 Detailed Description.....	27
6.12.2 Member Data Documentation.....	28
6.12.2.1 bulletToFire	28
6.12.2.2 firePoint.....	28
6.12.2.3 gunUI	28
6.12.2.4 timeBetweenShots.....	28
6.12.2.5 weaponName	28
6.13 GunChest Class Reference	28
6.13.1 Detailed Description.....	29
6.13.2 Member Data Documentation.....	29
6.13.2.1 chestOpen.....	29
6.13.2.2 notification.....	29
6.13.2.3 potentialGuns	29
6.13.2.4 scaleSpeed	29
6.13.2.5 spawnPoint	30
6.13.2.6 theSR.....	30
6.14 GunPickup Class Reference.....	30

6.14.1 Detailed Description.....	30
6.14.2 Member Data Documentation.....	30
6.14.2.1 theGun.....	30
6.14.2.2 waitToBeCollected.....	31
6.15 healthPickup Class Reference.....	31
6.15.1 Detailed Description.....	31
6.15.2 Member Data Documentation.....	31
6.15.2.1 healAmount.....	31
6.15.2.2 waitToBeCollected.....	31
6.16 LevelExit Class Reference.....	32
6.16.1 Detailed Description.....	32
6.16.2 Member Data Documentation.....	32
6.16.2.1 levelToLoad.....	32
6.17 LevelGenerator Class Reference.....	32
6.17.1 Detailed Description.....	33
6.17.2 Member Enumeration Documentation	33
6.17.2.1 Direction.....	33
6.17.3 Member Function Documentation.....	34
6.17.3.1 CreateRoomOutline().....	34
6.17.3.2 MoveGenerationPoint()	34
6.17.4 Member Data Documentation.....	34
6.17.4.1 allRooms.....	34
6.17.4.2 centreStart	34
6.17.4.3 distanceToEnd	34
6.17.4.4 generatorPoint.....	34
6.17.4.5 instance	35
6.17.4.6 layoutRoom.....	35
6.17.4.7 maxNumPagesInLevel.....	35
6.17.4.8 noEnemies	35
6.17.4.9 numEnemiesInLevel	35
6.17.4.10 numPagesInLevel.....	35
6.17.4.11 potentialCentres	35
6.17.4.12 rooms.....	36
6.17.4.13 selectedDirection.....	36
6.17.4.14 startColor	36
6.17.4.15 whatIsRoom	36
6.17.4.16 xOffset	36
6.18 LevelManager Class Reference.....	36
6.18.1 Detailed Description.....	37
6.18.2 Member Function Documentation.....	37
6.18.2.1 LevelEnd().....	37
6.18.2.2 PauseUnpause().....	37

6.18.3 Member Data Documentation.....	37
6.18.3.1 artCutscene.....	38
6.18.3.2 historyCutscene.....	38
6.18.3.3 instance	38
6.18.3.4 isPaused	38
6.18.3.5 musicCutscene.....	38
6.18.3.6 nextLevel	38
6.18.3.7 scienceCutscene	38
6.18.3.8 startPoint.....	39
6.18.3.9 thisPlayer	39
6.18.3.10 waitToLoad.....	39
6.19 LoadSceneAfterCutscene Class Reference.....	39
6.19.1 Detailed Description.....	39
6.19.2 Member Data Documentation.....	39
6.19.2.1 nextLevel	39
6.19.2.2 videoPlayer	40
6.20 MainMenu Class Reference	40
6.20.1 Detailed Description.....	40
6.20.2 Member Function Documentation.....	40
6.20.2.1 ExitGame()	40
6.20.2.2 GotoTutorial()	41
6.20.2.3 StartGame()	41
6.20.3 Member Data Documentation.....	41
6.20.3.1 instance	41
6.20.3.2 levelToLoad.....	41
6.20.3.3 tutorial.....	41
6.21 PauseMenu Class Reference	41
6.21.1 Detailed Description.....	42
6.21.2 Member Function Documentation.....	42
6.21.2.1 openTutorial()	42
6.21.2.2 PauseUnpause().....	42
6.21.2.3 Resume()	43
6.21.2.4 ReturnToMainMenu().....	43
6.21.3 Member Data Documentation.....	43
6.21.3.1 instance	43
6.21.3.2 isPaused	43
6.21.3.3 mainMenuScene.....	43
6.21.3.4 pauseMenu	43
6.21.3.5 tutorialScene	43
6.22 PlayerBullet Class Reference	44
6.22.1 Detailed Description.....	44
6.22.2 Member Data Documentation.....	44

6.22.2.1 damageToGive.....	44
6.22.2.2 impactEffect	44
6.22.2.3 speed.....	45
6.22.2.4 theRB.....	45
6.23 playerController Class Reference	45
6.23.1 Detailed Description.....	46
6.23.2 Member Function Documentation.....	46
6.23.2.1 switchGun()	46
6.23.3 Member Data Documentation.....	46
6.23.3.1 anim.....	46
6.23.3.2 availableWeapons	46
6.23.3.3 bodySR	46
6.23.3.4 canMove	46
6.23.3.5 currentGun	47
6.23.3.6 gunArm	47
6.23.3.7 instance	47
6.23.3.8 moveSpeed.....	47
6.23.3.9 theRB.....	47
6.24 PlayerHealthController Class Reference	47
6.24.1 Detailed Description.....	48
6.24.2 Member Function Documentation.....	48
6.24.2.1 DamagePlayer()	48
6.24.2.2 healPlayer().....	48
6.24.3 Member Data Documentation.....	48
6.24.3.1 currentHealth.....	49
6.24.3.2 damageInvincLength	49
6.24.3.3 instance	49
6.24.3.4 maxHealth.....	49
6.25 Roguelike_Dungeon.Program Class Reference	49
6.26 Room Class Reference	49
6.26.1 Detailed Description.....	50
6.26.2 Member Function Documentation.....	50
6.26.2.1 OpenDoors()	50
6.26.3 Member Data Documentation.....	50
6.26.3.1 closeWhenEntered	50
6.26.3.2 doors.....	50
6.26.3.3 levelExit	50
6.26.3.4 roomActive	51
6.27 RoomCentre Class Reference	51
6.27.1 Detailed Description.....	51
6.27.2 Member Data Documentation.....	51
6.27.2.1 enemies	52

6.27.2.2	essayPagePoints.....	52
6.27.2.3	essayPages.....	52
6.27.2.4	essayPagesInRoom.....	52
6.27.2.5	instance	52
6.27.2.6	levelExit	52
6.27.2.7	newPageLocation	52
6.27.2.8	numPagesInRoom.....	53
6.27.2.9	openWhenEnemiesCleared	53
6.27.2.10	pageLocation.....	53
6.27.2.11	theRoom	53
6.28	RoomPrefabs Class Reference.....	53
6.28.1	Member Data Documentation.....	53
6.28.1.1	singleUp.....	53
6.29	spriteSortOrder Class Reference	54
6.29.1	Detailed Description.....	54
6.30	Tutorial Class Reference	54
6.30.1	Detailed Description.....	54
6.30.2	Member Function Documentation.....	55
6.30.2.1	ReturnToMainMenu().....	55
6.30.2.2	ReturnToPauseMenu()	55
6.30.3	Member Data Documentation.....	55
6.30.3.1	mainMenuScene.....	55
6.30.3.2	pauseMenuScene.....	55
6.31	UIController Class Reference	55
6.31.1	Detailed Description.....	56
6.31.2	Member Function Documentation.....	56
6.31.2.1	Resume()	56
6.31.2.2	ReturnToMainMenu().....	56
6.31.2.3	StartFadeToBlack().....	57
6.31.3	Member Data Documentation.....	57
6.31.3.1	currentGun	57
6.31.3.2	essayPageSlider.....	57
6.31.3.3	essayPageText.....	57
6.31.3.4	fadeScreen.....	57
6.31.3.5	fadeSpeed.....	57
6.31.3.6	gunText	57
6.31.3.7	healthSlider	58
6.31.3.8	healthText.....	58
6.31.3.9	instance	58
6.31.3.10	mainMenuScene.....	58
6.31.3.11	numPages.....	58
6.31.3.12	pauseMenu	58

7 File Documentation	59
7.1 breakables.cs File Reference	59
7.2 brokenPieces.cs File Reference	59
7.3 CameraController.cs File Reference	59
7.4 CharacterSelectManager.cs File Reference	59
7.5 CharacterSelector.cs File Reference	60
7.6 DamagePlayer.cs File Reference	60
7.7 EnemyBullet.cs File Reference.....	60
7.8 EnemyController.cs File Reference.....	60
7.9 enemySleepAnimation.cs File Reference.....	60
7.10 essayPageController.cs File Reference	60
7.11 essayPages.cs File Reference	61
7.12 Gun.cs File Reference.....	61
7.13 GunChest.cs File Reference.....	61
7.14 GunPickup.cs File Reference	61
7.15 healthPickup.cs File Reference.....	61
7.16 LevelExit.cs File Reference	61
7.17 LevelGenerator.cs File Reference.....	62
7.18 LevelManager.cs File Reference	62
7.19 LoadSceneAfterCutscene.cs File Reference	62
7.20 MainMenu.cs File Reference	62
7.21 PauseMenu.cs File Reference.....	62
7.22 PlayerBullet.cs File Reference.....	62
7.23 playerController.cs File Reference	63
7.24 PlayerHealthController.cs File Reference.....	63
7.25 Program.cs File Reference	63
7.26 Room.cs File Reference	63
7.27 RoomCentre.cs File Reference.....	63
7.27.1 Typedef Documentation	64
7.27.1.1 Random	64
7.28 spriteSortOrder.cs File Reference.....	64
7.29 Tutorial.cs File Reference.....	64
7.30 UIController.cs File Reference.....	64

Chapter

Namespace Index

1.1 Packages

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

Roguelike_Dungeon	9
---	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

MonoBehaviour	
breakables	11
brokenPieces	12
CameraController	14
CharacterSelectManager	15
CharacterSelector.....	16
DamagePlayer	17
EnemyBullet.....	17
EnemyController.....	18
enemySleepAnimation.....	23
essayPageController	25
essayPages	26
Gun.....	27
GunChest.....	28
GunPickup	30
healthPickup	31
LevelExit	32
LevelGenerator.....	32
LevelManager	36
LoadSceneAfterCutscene	39
MainMenu	40
PauseMenu.....	41
PlayerBullet.....	44
playerController	45
PlayerHealthController.....	47
Room	49
RoomCentre.....	51
spriteSortOrder.....	54
Tutorial	54
UIController.....	55
Roguelike_Dungeon.Program	49
RoomPrefabs.....	53

Chapter

Class Index

3.1 Class List

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

breakables	This class controls the breakable objects such as the chests.....	11
brokenPieces	This class controls the broken pieces released by the breakable object.....	12
CameraController	This class controls the camera	14
CharacterSelectManager	Class to control the selection process of the game	15
CharacterSelector	This class controls the character selection process in the game.....	16
DamagePlayer	This class manages the player's health.....	17
EnemyBullet	This class controls the bullets fired by the enemies	17
EnemyController	This class controls the enemies.....	18
enemySleepAnimation	This class controls the sleep animation of enemies when their health drops to zero.....	23
essayPageController	Controller class for the essay page collectibles	25
essayPages	Class controls the collection of essay pages	26
Gun	This class controls the guns used by the player	27
GunChest	This class controls the chests where players can get new weapons	28
GunPickup	This class enables the player pickup guns.....	30
healthPickup	Class controls health pickups to restore player health.....	31
LevelExit	This class selects the next level to load.....	32
LevelGenerator	This class generated levels randomly	32
LevelManager	This class manages each level.....	36

LoadSceneAfterCutscene	
This class decides the scene to load after cutscenes.....	39
MainMenu	
This class creates and controls the main menu	40
PauseMenu	
Class to control pause menu scene	41
PlayerBullet	
This class controls bullets fired by the player.....	44
playerController	
This class controls the player	45
PlayerHealthController	
Class controls the health of the player and updates UI	47
Roguelike_Dungeon.Program	49
Room	
Class that controls the operation of rooms.....	49
RoomCentre	
This class controls the room centres.....	51
RoomPrefabs.....	53
spriteSortOrder	
This class controls the ordering of the sprites.....	54
Tutorial	
Class to control tutorial scene	54
UIController	
This class is tasked with managing the User Interface	55

Chapter

File Index

4.1 File List

Here is a list of all files with brief descriptions:

breakables.cs	59
brokenPieces.cs	59
CameraController.cs	59
CharacterSelectManager.cs	59
CharacterSelector.cs	60
DamagePlayer.cs	60
EnemyBullet.cs	60
EnemyController.cs	60
enemySleepAnimation.cs	60
essayPageController.cs	60
essayPages.cs	61
Gun.cs	61
GunChest.cs	61
GunPickup.cs	61
healthPickup.cs	61
LevelExit.cs	61
LevelGenerator.cs	62
LevelManager.cs	62
LoadSceneAfterCutscene.cs	62
MainMenu.cs	62
PauseMenu.cs	62
PlayerBullet.cs	62
playerController.cs	63
PlayerHealthController.cs	63
Program.cs	63
Room.cs	63
RoomCentre.cs	63
spriteSortOrder.cs	64
Tutorial.cs	64
UIController.cs	64

Chapter 5

Namespace Documentation

5.1 Roguelike_Dungeon Namespace Reference

Classes

- class [Program](#)

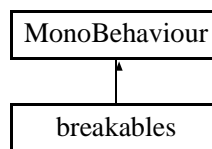
Chapter

Class Documentation

6.1 breakables Class Reference

this class controls the breakable objects such as the chests.

Inheritance diagram for breakables:



Public Attributes

- `GameObject[] brokenPieces`
list of broken pieces
- `int maxPieces = 5`
- `bool shouldDropItem`
list of pickups to drop
- `GameObject[] itemsToDrop`
- `float itemDropPercent`

6.1.1 Detailed Description

this class controls the breakable objects such as the chests.

it will break the object and release broken pieces and pickups when triggered by the player

6.1.2 Member Data Documentation

6.1.2.1 brokenPieces

```
GameObject [] breakables.brokenPieces
```

list of broken pieces

6.1.2.2 itemDropPercent

```
float breakables.itemDropPercent
```

6.1.2.3 itemsToDrop

```
GameObject [] breakables.itemsToDrop
```

6.1.2.4 maxPieces

```
int breakables.maxPieces = 5
```

6.1.2.5 shouldDropItem

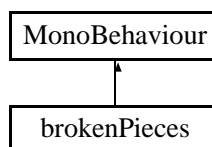
```
bool breakables.shouldDropItem
```

list of pickups to drop

6.2 brokenPieces Class Reference

this class controls the broken pieces released by the breakable object.

Inheritance diagram for brokenPieces:



Public Attributes

- float `moveSpeed`= 3f
- float `deceleration`= 5f
- float `lifetime`= 3f
- SpriteRenderer `theSR`
- float `fadeSpeed`= 2.5f

6.2.1 Detailed Description

this class controls the broken pieces released by the breakable object.

6.2.2 Member Data Documentation

6.2.2.1 deceleration

```
float brokenPieces.deceleration = 5f
```

6.2.2.2 fadeSpeed

```
float brokenPieces.fadeSpeed = 2.5f
```

6.2.2.3 lifetime

```
float brokenPieces.lifetime = 3f
```

6.2.2.4 moveSpeed

```
float brokenPieces.moveSpeed = 3f
```

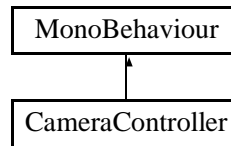
6.2.2.5 theSR

```
SpriteRenderer brokenPieces.theSR
```

6.3 CameraController Class Reference

This class controls the camera.

Inheritance diagram for CameraController:



Public Member Functions

- void [ChangeTarget](#)(Transform newTarget)

Public Attributes

- float [moveSpeed](#)
Measures the speed of the player to assess how quickly to move the camera.
- Transform [target](#)

Static Public Attributes

- static [CameraControllerinstance](#)

6.3.1 Detailed Description

This class controls the camera.

6.3.2 Member Function Documentation

6.3.2.1 ChangeTarget()

```
void CameraController.ChangeTarget (  
    Transform newTarget )
```

6.3.3 Member Data Documentation

6.3.3.1 instance

`CameraController` `CameraController.instance` [static]

6.3.3.2 moveSpeed

`float` `CameraController.moveSpeed`

Measures the speed of the player to assess how quickly to move the camera.

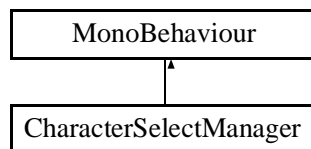
6.3.3.3 target

`Transform` `CameraController.target`

6.4 CharacterSelectManager Class Reference

Class to control the selection process of the game.

Inheritance diagram for `CharacterSelectManager`:



Public Attributes

- `playerController` `activePlayer`
- `CharacterSelector` `activeCharSelect`

Static Public Attributes

- `static` `CharacterSelectManager` `instance`

6.4.1 Detailed Description

Class to control the selection process of the game.

6.4.2 Member Data Documentation

6.4.2.1 activeCharSelect

`CharacterSelector``CharacterSelectManager.activeCharSelect`

6.4.2.2 activePlayer

`playerController``CharacterSelectManager.activePlayer`

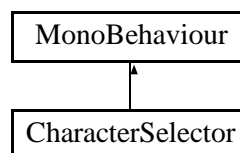
6.4.2.3 instance

`CharacterSelectManager``CharacterSelectManager.instance` [static]

6.5 CharacterSelector Class Reference

This class controls the character selection process in the game.

Inheritance diagram for CharacterSelector:



Public Attributes

- `GameObject``message`
- `playerController``playerToSpawn`

The character to be used in the game.

6.5.1 Detailed Description

This class controls the character selection process in the game.

6.5.2 Member Data Documentation

6.5.2.1 message

`GameObject CharacterSelector.message`

6.5.2.2 playerToSpawn

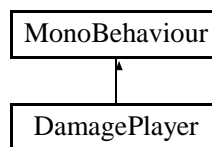
`playerControllerCharacterSelector.playerToSpawn`

The character to be used in the game.

6.6 DamagePlayer Class Reference

This class manages the player's health.

Inheritance diagram for DamagePlayer:



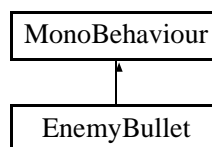
6.6.1 Detailed Description

This class manages the player's health.

6.7 EnemyBullet Class Reference

This class controls the bullets fired by the enemies.

Inheritance diagram for EnemyBullet:



Public Attributes

- `float speed`

enables the user to change the speed of the bullet in Unity

6.7.1 Detailed Description

This class controls the bullets fired by the enemies.

6.7.2 Member Data Documentation

6.7.2.1 speed

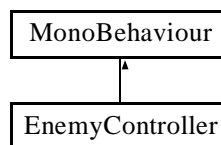
```
float EnemyBullet.speed
```

enables the user to change the speed of the bullet in Unity

6.8 EnemyController Class Reference

This class controls the enemies.

Inheritance diagram for EnemyController:



Public Member Functions

- void[DamageEnemy](#)(int damage)

Public Attributes

- Rigidbody2D[theRB](#)
- float[moveSpeed](#)
- bool[shouldChasePlayer](#)
- float[rangeToChasePlayer](#)
- bool[shouldChaseBot](#)
- float[rangeToChaseBot](#)
- variables for chase bot enemy*
- bool[shouldPatrol](#)
- Transform[] [patrolPoints](#)
- bool[shouldRunAway](#)
- float[runAwayRange](#)
- variables for the coward enemy*
- bool[shouldWander](#)
- float[wanderLength](#)
- float[pauseLength](#)

- how long the enemy should pause for*
- bool [shouldShoot](#)
- whether or not the enemy should shoot the player*
- GameObject [bullet](#)
- Transform [firePoint](#)
- float [fireRate](#)
- float [shootRange](#)
- SpriteRenderer [theBody](#)
- allows us to change the sprite of enemies*
- Animator [anim](#)
- allows us to animate the enemies*
- int [health](#) = 150
- enemies default have 150 health - this can be changed in Unity*
- GameObject[] [deathSplatters](#)
- included just in case the user wants to have death splatters to make it more violent*
- GameObject [hitEffect](#)
- hit effect for when the bullets hit the enemy*
- bool [canMove](#) = true
- GameObject [Zzzz](#)
- the speech bubble to show that the enemy is sleeping and hence inactive*

Static Public Attributes

- static [EnemyControllerinstance](#)

6.8.1 Detailed Description

This class controls the enemies.

6.8.2 Member Function Documentation

6.8.2.1 DamageEnemy()

```
void EnemyController.DamageEnemy (
    int damage )
```

when the enemy is hit by a piece of fruit, trigger the enemy's hit effect

when the enemy's health goes below zero, set its death sprite: sleeping

enemy sleeping sprite Destroy(gameObject); //don't make the enemy disappear ensure enemy can't move or shoot

disables animation

add the zzz to show that the enemy is sleeping

replaces sprite

6.8.3 Member Data Documentation

6.8.3.1 anim

`Animator EnemyController.anim`

allows us to animate the enemies

6.8.3.2 bullet

`GameObject EnemyController.bullet`

6.8.3.3 canMove

`bool EnemyController.canMove = true`

6.8.3.4 deathSplatters

`GameObject [] EnemyController.deathSplatters`

included just in case the user wants to have death splatters to make it more violent

6.8.3.5 firePoint

`Transform EnemyController.firePoint`

6.8.3.6 fireRate

`float EnemyController.fireRate`

6.8.3.7 health

```
int EnemyController.health = 150
```

enemies default have 150 health - this can be changed in Unity

6.8.3.8 hitEffect

```
GameObject EnemyController.hitEffect
```

hit effect for when the bullets hit the enemy

6.8.3.9 instance

```
EnemyController EnemyController.instance [static]
```

6.8.3.10 moveSpeed

```
float EnemyController.moveSpeed
```

6.8.3.11 patrolPoints

```
Transform [] EnemyController.patrolPoints
```

6.8.3.12 pauseLength

```
float EnemyController.pauseLength
```

how long the enemy should pause for

6.8.3.13 rangeToChaseBot

```
float EnemyController.rangeToChaseBot
```

variables for chase bot enemy

6.8.3.14 rangeToChasePlayer

```
float EnemyController.rangeToChasePlayer
```

6.8.3.15 runAwayRange

```
float EnemyController.runAwayRange
```

variables for the coward enemy

6.8.3.16 shootRange

```
float EnemyController.shootRange
```

6.8.3.17 shouldChaseBot

```
bool EnemyController.shouldChaseBot
```

6.8.3.18 shouldChasePlayer

```
bool EnemyController.shouldChasePlayer
```

6.8.3.19 shouldPatrol

```
bool EnemyController.shouldPatrol
```

6.8.3.20 shouldRunAway

```
bool EnemyController.shouldRunAway
```

6.9.1.1 shouldShoot

```
bool EnemyController.shouldShoot
```

whether or not the enemy should shoot the player

6.9.1.2 shouldWander

```
bool EnemyController.shouldWander
```

6.9.1.3 theBody

```
SpriteRenderer EnemyController.theBody
```

allows us to change the sprite of enemies

6.9.1.4 theRB

```
Rigidbody2D EnemyController.theRB
```

6.9.1.5 wanderLength

```
float EnemyController.wanderLength
```

6.9.1.6 Zzzz

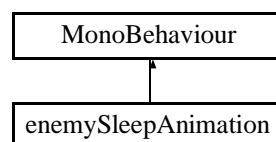
```
GameObject EnemyController.Zzzz
```

the speech bubble to show that the enemy is sleeping and hence inactive

6.9 enemySleepAnimation Class Reference

this class controls the sleep animation of enemies when their health drops to zero.

Inheritance diagram for enemySleepAnimation:



Public Member Functions

- `void enemyToSleep()`
replaces the enemy sprite with the sleeping sprite

Public Attributes

- `SpriteRenderer theSR`
- `Sprite sleepingEnemy`

Static Public Attributes

- `static enemySleepAnimation instance`

6.9.1 Detailed Description

this class controls the sleep animation of enemies when their health drops to zero.

less violent alternative to being killed

6.9.2 Member Function Documentation

6.9.2.1 enemyToSleep()

```
void enemySleepAnimation.enemyToSleep ( )
```

replaces the enemy sprite with the sleeping sprite

6.9.3 Member Data Documentation

6.9.3.1 instance

```
enemySleepAnimation enemySleepAnimation.instance [static]
```

6.9.3.2 sleepingEnemy

```
Sprite enemySleepAnimation.sleepingEnemy
```

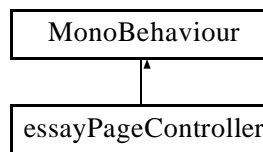
6.10.1.1 theSR

```
SpriteRenderer enemySleepAnimation.theSR
```

6.10 essayPageController Class Reference

controller class for the essay page collectibles.

Inheritance diagram for essayPageController:



Public Member Functions

- void [pickupPage\(\)](#)
method run when player walks over essay page increments current number of pages by 1 and updates UI

Public Attributes

- int [currentNumPages](#)
- int [maxNumPages](#)
- Sprite[] [essayPageUI](#)

Static Public Attributes

- static [essayPageControllerinstance](#)

6.10.1 Detailed Description

controller class for the essay page collectibles.

6.10.2 Member Function Documentation

6.10.2.1 pickupPage()

```
void essayPageController.pickupPage ( )
```

method run when player walks over essay page increments current number of pages by 1 and updates UI

6.10.3 Member Data Documentation

6.10.3.1 currentNumPages

```
int essayPageController.currentNumPages
```

6.10.3.2 essayPageUI

```
Sprite [] essayPageController.essayPageUI
```

6.10.3.3 instance

```
essayPageControlleressayPageController.instance [static]
```

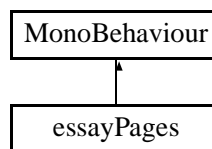
6.10.3.4 maxNumPages

```
int essayPageController.maxNumPages
```

6.11 essayPages Class Reference

class controls the collection of essay pages.

Inheritance diagram for essayPages:



Public Attributes

- string[test](#)

Static Public Attributes

- static[essayPagesinstance](#)

6.11.1 Detailed Description

class controls the collection of essay pages.

6.11.2 Member Data Documentation

6.11.2.1 instance

```
essayPagesessayPages.instance [static]
```

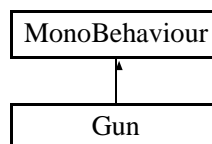
6.11.2.2 test

```
string essayPages.test
```

6.12 Gun Class Reference

This class controls the guns used by the player.

Inheritance diagram for Gun:



Public Attributes

- GameObject**bulletToFire**
differentiates between bullets (fruit)
- Transform**firePoint**
creates a fire point for the fruit to come from for each weapon
- float**timeBetweenShots**
- string**weaponName**
- Sprite**gunUI**
allows for a gun UI in the bottom left of the screen

6.12.1 Detailed Description

This class controls the guns used by the player.

6.12.2 Member Data Documentation

6.12.2.1 bulletToFire

`GameObject Gun.bulletToFire`

differentiates between bullets (fruit)

6.12.2.2 firePoint

`Transform Gun.firePoint`

creates a fire point for the fruit to come from for each weapon

6.12.2.3 gunUI

`Sprite Gun.gunUI`

allows for a gun UI in the bottom left of the screen

6.12.2.4 timeBetweenShots

`float Gun.timeBetweenShots`

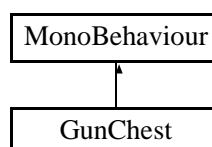
6.12.2.5 weaponName

`string Gun.weaponName`

6.13 GunChest Class Reference

This class controls the chests where players can get new weapons.

Inheritance diagram for GunChest:



Public Attributes

- [GunPickup\[\]](#) [potentialGuns](#)
create a list of potential guns for the user to change within Unity
- [SpriteRenderer](#) [theSR](#)
- [Sprite](#) [chestOpen](#)
allows the chest to change to an open sprite
- [GameObject](#) [notification](#)
- [Transform](#) [spawnPoint](#)
point at which the gun will spawn having been deployed from chest
- [float](#) [scaleSpeed](#) = 2f

6.13.1 Detailed Description

This class controls the chests where players can get new weapons.

6.13.2 Member Data Documentation

6.13.2.1 chestOpen

```
Sprite GunChest.chestOpen
```

allows the chest to change to an open sprite

6.13.2.2 notification

```
GameObject GunChest.notification
```

6.13.2.3 potentialGuns

```
GunPickup[] GunChest.potentialGuns
```

create a list of potential guns for the user to change within Unity

6.13.2.4 scaleSpeed

```
float GunChest.scaleSpeed = 2f
```

6.13.2.5 spawnPoint

`Transform GunChest.spawnPoint`

point at which the gun will spawn having been deployed from chest

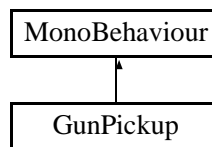
6.13.2.6 theSR

`SpriteRenderer GunChest.theSR`

6.14 GunPickup Class Reference

This class enables the player pick up guns.

Inheritance diagram for GunPickup:



Public Attributes

- [GuntheGun](#)
- `floatwaitToBeCollected= .5f`
sets a brief moment of time that the weapon can't be collected for

6.14.2 Detailed Description

This class enables the player pick up guns.

6.14.3 Member Data Documentation

6.14.3.1 theGun

`GunGunPickup.theGun`

6.15.1.1 waitToBeCollected

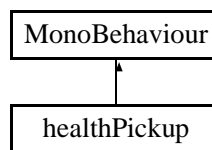
```
float GunPickup.waitToBeCollected = .5f
```

sets a brief moment of time that the weapon can't be collected for

6.15 healthPickup Class Reference

class controls health pickups to restore player health.

Inheritance diagram for healthPickup:



Public Attributes

- `int` `healAmount` = 1
- `float` `waitToBeCollected` = .5f

6.15.1 Detailed Description

class controls health pickups to restore player health.

6.15.2 Member Data Documentation

6.15.2.1 healAmount

```
int healthPickup.healAmount = 1
```

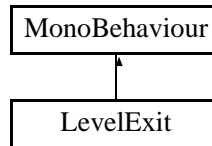
6.15.2.2 waitToBeCollected

```
float healthPickup.waitToBeCollected = .5f
```

6.16 LevelExit Class Reference

This class selects the next level to load.

Inheritance diagram for LevelExit:



Public Attributes

•string [levelToLoad](#)

6.16.1 Detailed Description

This class selects the next level to load.

6.16.2 Member Data Documentation

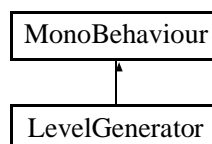
6.16.2.1 levelToLoad

```
string LevelExit.levelToLoad
```

6.17 LevelGenerator Class Reference

This class generated levels randomly.

Inheritance diagram for LevelGenerator:



Public Types

•enum [Direction](#){[Direction.up](#),[Direction.right](#),[Direction.down](#),[Direction.left](#)}

Public Member Functions

- void [MoveGenerationPoint](#)()
- void [CreateRoomOutline](#)(Vector3 roomPosition)

Public Attributes

- GameObject [layoutRoom](#)
- Color [startColor](#)
- int [distanceToEnd](#)
- Transform [generatorPoint](#)
- Direction [selectedDirection](#)
- float [xOffset](#) = 18f
- LayerMask [whatIsRoom](#)
- List < [RoomCentre](#) > [allRooms](#) = new List <[RoomCentre](#)>()
list of all room centres used in a given level
- int [numEnemiesInLevel](#) = 0
number of enemies in a given level
- bool [noEnemies](#) = false
- int [numPagesInLevel](#) = 0
- RoomPrefab [rooms](#)
- RoomCentre [centreStart](#)
- RoomCentre[] [potentialCentres](#)
- int [maxNumPagesInLevel](#)

Static Public Attributes

- static [LevelGeneratorinstance](#)

6.17.1 Detailed Description

This class generated levels randomly.

6.17.2 Member Enumeration Documentation

6.17.2.1 Direction

```
enum LevelGenerator.Direction [strong]
```

Enumerator

up	
right	
down	
left	

6.17.3 Member Function Documentation

6.17.3.1 CreateRoomOutline()

```
void LevelGenerator.CreateRoomOutline (
    Vector3 roomPosition )
```

6.17.3.2 MoveGenerationPoint()

```
void LevelGenerator.MoveGenerationPoint ( )
```

6.17.4 Member Data Documentation

6.17.4.1 allRooms

```
List<RoomCentre>LevelGenerator.allRooms = new List<RoomCentre>()
```

list of all room centres used in a given level

6.17.4.2 centreStart

```
RoomCentreLevelGenerator.centreStart
```

6.17.4.3 distanceToEnd

```
int LevelGenerator.distanceToEnd
```

6.17.4.4 generatorPoint

```
Transform LevelGenerator.generatorPoint
```

6.17.4.5 instance

`LevelGenerator` `LevelGenerator.instance` [static]

6.17.4.6 layoutRoom

`GameObject` `LevelGenerator.layoutRoom`

6.17.4.7 maxNumPagesInLevel

`int` `LevelGenerator.maxNumPagesInLevel`

6.17.4.8 noEnemies

`bool` `LevelGenerator.noEnemies` = false

6.17.4.9 numEnemiesInLevel

`int` `LevelGenerator.numEnemiesInLevel` = 0

number of enemies in a given level

6.17.4.10 numPagesInLevel

`int` `LevelGenerator.numPagesInLevel` = 0

6.17.4.11 potentialCentres

`RoomCentre`[] `LevelGenerator.potentialCentres`

6.17.4.12 rooms

`RoomPrefabsLevelGenerator.rooms`

6.17.4.13 selectedDirection

`DirectionLevelGenerator.selectedDirection`

6.17.4.14 startColor

`Color LevelGenerator.startColor`

6.17.4.15 whatIsRoom

`LayerMask LevelGenerator.whatIsRoom`

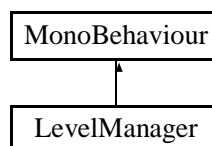
6.17.4.16 xOffset

`float LevelGenerator.xOffset = 18f`

6.18 LevelManager Class Reference

This class manages each level.

Inheritance diagram for LevelManager:



Public Member Functions

- `IEnumeratorLevelEnd()`
- `voidPauseUnpause()`

When false, the pause menu is displayed.

Public Attributes

- float `waitToLoad` = 4f
- string `nextLevel`
- string `artCutscene`
- string `historyCutscene`
- string `musicCutscene`
- string `scienceCutscene`
- bool `isPaused`
Is it a pause menu.
- Transform `startPoint`
- string `thisPlayer`

Static Public Attributes

- static `LevelManagerinstance`

6.18.1 Detailed Description

This class manages each level.

6.18.2 Member Function Documentation

6.18.2.1 LevelEnd()

```
IEnumerator LevelManager.LevelEnd ( )
```

if current scene is character select screen load the correct cutscene depending on character selected

6.18.2.2 PauseUnpause()

```
void LevelManager.PauseUnpause ( )
```

When false, the pause menu is displayed.

When true, the pause menu is hidden.

6.18.3 Member Data Documentation

6.18.3.1 artCutscene

```
string LevelManager.artCutscene
```

6.18.3.2 historyCutscene

```
string LevelManager.historyCutscene
```

6.18.3.3 instance

```
LevelManagerLevelManager.instance [static]
```

6.18.3.4 isPaused

```
bool LevelManager.isPaused
```

Is it a pause menu.

6.18.3.5 musicCutscene

```
string LevelManager.musicCutscene
```

6.18.3.6 nextLevel

```
string LevelManager.nextLevel
```

6.18.3.7 scienceCutscene

```
string LevelManager.scienceCutscene
```

6.19.1.1 startPoint

```
Transform LevelManager.startPoint
```

6.19.1.2 thisPlayer

```
string LevelManager.thisPlayer
```

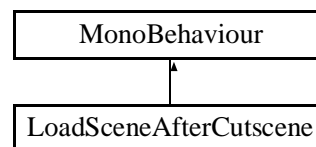
6.19.1.3 waitToLoad

```
float LevelManager.waitToLoad = 4f
```

6.19 LoadSceneAfterCutscene Class Reference

This class decides the scene to load after cutscenes.

Inheritance diagram for LoadSceneAfterCutscene:



Public Attributes

- string `nextLevel`
- VideoPlayer `videoPlayer`

6.19.1 Detailed Description

This class decides the scene to load after cutscenes.

6.19.2 Member Data Documentation

6.19.2.1 nextLevel

```
string LoadSceneAfterCutscene.nextLevel
```

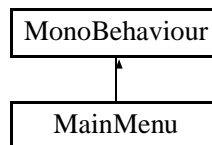
6.19.2.2 videoPlayer

VideoPlayer LoadSceneAfterCutscene.videoPlayer

6.20 MainMenu Class Reference

This class creates and controls the main menu.

Inheritance diagram for MainMenu:



Public Member Functions

- void[StartGame\(\)](#)
- void[GotoTutorial\(\)](#)
- void[ExitGame\(\)](#)

Public Attributes

- string[levelToLoad](#)

Static Public Attributes

- static[MainMenuinstance](#)
- static int[tutorial](#)

6.20.1 Detailed Description

This class creates and controls the main menu.

6.20.2 Member Function Documentation

6.20.2.1 ExitGame()

```
void MainMenu.ExitGame ( )
```

this line ONLY works when you're not in the Unity editor

6.20.2.2 GotoTutorial()

```
void MainMenu.GotoTutorial ( )
```

6.20.2.3 StartGame()

```
void MainMenu.StartGame ( )
```

6.20.3 Member Data Documentation

6.20.3.1 instance

```
MainMenuMainMenu.instance [static]
```

6.20.3.2 levelToLoad

```
string MainMenu.levelToLoad
```

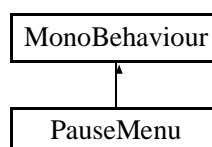
6.20.3.3 tutorial

```
int MainMenu.tutorial [static]
```

6.21 PauseMenu Class Reference

class to control pause menu scene

Inheritance diagram for PauseMenu:



Public Member Functions

- void [PauseUnpause\(\)](#)
- void [Resume\(\)](#)
when button clicked, will resume game
- void [ReturnToMainMenu\(\)](#)
when button clicked, will return user to main menu
- void [openTutorial\(\)](#)
when button clicked, will load tutorial scene

Public Attributes

- GameObject [pauseMenu](#)
- string [mainMenuScene](#)
- string [tutorialScene](#)

Static Public Attributes

- static [PauseMenuinstance](#)
- static bool [isPaused](#)=false

6.21.1 Detailed Description

class to control pause menu scene

6.21.2 Member Function Documentation

6.21.2.1 [openTutorial\(\)](#)

```
void PauseMenu.openTutorial ( )
```

when button clicked, will load tutorial scene

6.21.2.2 [PauseUnpause\(\)](#)

```
void PauseMenu.PauseUnpause ( )
```

6.21.2.3 Resume()

```
void PauseMenu.Resume ( )
```

when button clicked, will resume game

6.21.2.4 ReturnToMainMenu()

```
void PauseMenu.ReturnToMainMenu ( )
```

when button clicked, will return user to main menu

6.21.3 Member Data Documentation

6.21.3.1 instance

```
PauseMenuPauseMenu.instance [static]
```

6.21.3.2 isPaused

```
bool PauseMenu.isPaused = false [static]
```

6.21.3.3 mainMenuScene

```
string PauseMenu.mainMenuScene
```

6.21.3.4 pauseMenu

```
GameObject PauseMenu.pauseMenu
```

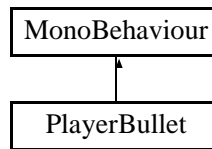
6.21.3.5 tutorialScene

```
string PauseMenu.tutorialScene
```


6.22 PlayerBullet Class Reference

This class controls bullets fired by the player.

Inheritance diagram for PlayerBullet:



Public Attributes

- float `speed` = 7.5f
bullet speed is 7.5 by default, can be changed in Unity.
- Rigidbody2D `theRB`
- GameObject `impactEffect`
- int `damageToGive` = 50
bullets do 50 damage by default. can be changed in Unity.

6.22.1 Detailed Description

This class controls bullets fired by the player.

6.22.2 Member Data Documentation

6.22.2.1 `damageToGive`

```
int PlayerBullet.damageToGive = 50
```

bullets do 50 damage by default. can be changed in Unity.

6.22.2.2 `impactEffect`

```
GameObject PlayerBullet.impactEffect
```

6.23.1.1 speed

```
float PlayerBullet.speed = 7.5f
```

bullet speed is 7.5 by default, can be changed in Unity.

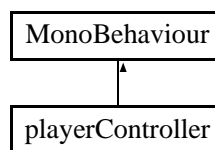
6.23.1.2 theRB

```
Rigidbody2D PlayerBullet.theRB
```

6.23 playerController Class Reference

This class controls the player.

Inheritance diagram for playerController:



Public Member Functions

- voidswitchGun()

when the user switches gun, it is reflected in the UI on the bottom left.

Public Attributes

- floatmoveSpeed
- Rigidbody2DtheRB
- TransformgunArm
- Animatoranim
- SpriteRendererbodySR
- List < Gun > availableWeapons= new List <Gun>()
- intcurrentGun
- boolcanMove= true

Static Public Attributes

- staticplayerControllerinstance

public variables can be accessed in the Inspector pane in Unity

6.23.1 Detailed Description

This class controls the player.

6.23.2 Member Function Documentation

6.23.2.1 switchGun()

```
void playerController.switchGun ( )
```

when the user switches gun, it is reflected in the UI on the bottom left.

6.23.3 Member Data Documentation

6.23.3.1 anim

```
Animator playerController.anim
```

6.23.3.2 availableWeapons

```
List<Gun>playerController.availableWeapons = new List<Gun>()
```

6.23.3.3 bodySR

```
SpriteRenderer playerController.bodySR
```

6.23.3.4 canMove

```
bool playerController.canMove = true
```

6.24.1.1 currentGun

```
int playerController.currentGun
```

6.24.1.2 gunArm

```
Transform playerController.gunArm
```

6.24.1.3 instance

```
playerController playerController.instance [static]
```

public variables can be accessed in the Inspector pane in Unity

6.24.1.4 moveSpeed

```
float playerController.moveSpeed
```

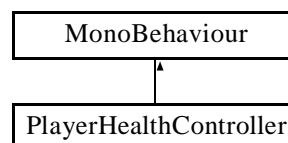
6.24.1.5 theRB

```
Rigidbody2D playerController.theRB
```

6.24 PlayerHealthController Class Reference

class controls the health of the player and updates UI.

Inheritance diagram for PlayerHealthController:



Public Member Functions

- void [DamagePlayer\(\)](#)
method damages player when player is hit by a bullet or runs over a spike
- void [healPlayer](#)(int healAmount)
method heals player when health pickup collected

Public Attributes

- `int` `currentHealth`
- `int` `maxHealth`
sets the current and max health. Can be changed in Unity
- `float` `damageInvincLength` = 1f
sets the invincibility length (after being damaged) to one second

Static Public Attributes

- `static` `PlayerHealthControllerinstance`

6.24.1 Detailed Description

class controls the health of the player and updates UI.

6.24.2 Member Function Documentation

6.24.2.1 DamagePlayer()

```
void PlayerHealthController.DamagePlayer ( )
```

method damages player when player is hit by a bullet or runs over a spike

health goes down by 1 if the player is not currently invincible

when the player is invincible, the user can see this because the player becomes slightly transparent

when the player dies, the death screen appears

UI shows health on top left

6.24.2.2 healPlayer()

```
void PlayerHealthController.healPlayer (
    int healAmount )
```

method heals player when health pickup collected

if the player picks up a health pack, the current health increases

but it can't go above the max health

6.24.3 Member Data Documentation

6.25.1.1 currentHealth

```
int PlayerHealthController.currentHealth
```

6.25.1.2 damageInvincLength

```
float PlayerHealthController.damageInvincLength = 1f
```

sets the invincibility length (after being damaged) to one second

6.25.1.3 instance

```
PlayerHealthController PlayerHealthController.instance [static]
```

6.25.1.4 maxHealth

```
int PlayerHealthController.maxHealth
```

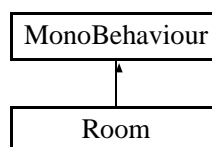
sets the current and max health. Can be changed in Unity

6.25 Roguelike_Dungeon.Program Class Reference

6.26 Room Class Reference

Class that controls the operation of rooms.

Inheritance diagram for Room:



Public Member Functions

- void [OpenDoors\(\)](#)

Method to open doors for the player.

Public Attributes

- bool `closeWhenEntered`
openWhenEnemiesCleared
- GameObject[] `doors`
- bool `roomActive`
- GameObject `levelExit`

6.26.1 Detailed Description

Class that controls the operation of rooms.

6.26.2 Member Function Documentation

6.26.2.1 OpenDoors()

```
void Room.OpenDoors ( )
```

Method to open doors for the player.

6.26.3 Member Data Documentation

6.26.3.1 closeWhenEntered

```
bool Room.closeWhenEntered
```

`openWhenEnemiesCleared`

6.26.3.2 doors

```
GameObject [ ] Room.doors
```

6.26.3.3 levelExit

```
GameObject Room.levelExit
```

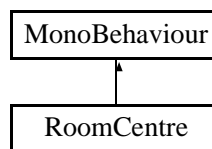
6.26.3.4 roomActive

```
bool Room.roomActive
```

6.27 RoomCentre Class Reference

This class controls the room centres.

Inheritance diagram for RoomCentre:



Public Attributes

- List < GameObject > [enemies](#) = new List <GameObject>()
list of all enemies in that room used by [LevelGenerator](#) class to calculate total number of enemies in level
- bool [openWhenEnemiesCleared](#)
- [RoomtheRoom](#)
- GameObject [levelExit](#)
- List < Transform > [essayPagePoints](#) = new List <Transform>()
list of locations where an essay could be located
- List < GameObject > [essayPages](#) = new List <GameObject>()
list of essay page prefabs
- List < GameObject > [essayPagesInRoom](#) = new List <GameObject>()
list of actual essay pages in that room used by [LevelGenerator](#) class to calculate total number of pages in level
- List < int > [pageLocation](#) = new List <int>()
list of actual page locations used in that room
- int [numPagesInRoom](#)
- int [newPageLocation](#)

Static Public Attributes

- static [RoomCentreinstance](#)

6.27.1 Detailed Description

This class controls the room centres.

6.27.2 Member Data Documentation

6.27.2.1 enemies

```
List<GameObject>RoomCentre.enemies = new List<GameObject>()
```

list of all enemies in that room used by [LevelGenerator](#) class to calculate total number of enemies in level

6.27.2.2 essayPagePoints

```
List<Transform>RoomCentre.essayPagePoints = new List<Transform>()
```

list of locations where an essay could be located

6.27.2.3 essayPages

```
List<GameObject>RoomCentre.essayPages = new List<GameObject>()
```

list of essay page prefabs

6.27.2.4 essayPagesInRoom

```
List<GameObject>RoomCentre.essayPagesInRoom = new List<GameObject>()
```

list of actual essay pages in that room used by [LevelGenerator](#) class to calculate total number of pages in level

6.27.2.5 instance

```
RoomCentreRoomCentre.instance [static]
```

6.27.2.6 levelExit

```
GameObject RoomCentre.levelExit
```

6.27.2.7 newPageLocation

```
int RoomCentre.newPageLocation
```

6.28.1.1 numPagesInRoom

```
int RoomCentre.numPagesInRoom
```

6.28.1.2 openWhenEnemiesCleared

```
bool RoomCentre.openWhenEnemiesCleared
```

6.28.1.3 pageLocation

```
List<int>RoomCentre.pageLocation = new List<int>()
```

list of actual page locations used in that room

6.28.1.4 theRoom

```
RoomRoomCentre.theRoom
```

6.28 RoomPrefabs Class Reference

Public Attributes

•GameObjects[singleUp](#)

6.28.1 Member Data Documentation

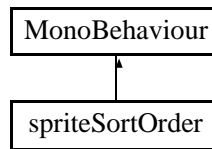
6.28.1.1 singleUp

```
GameObject RoomPrefabs.singleUp
```

6.29 spriteSortOrder Class Reference

This class controls the ordering of the sprites.

Inheritance diagram for spriteSortOrder:



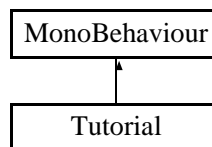
6.29.1 Detailed Description

This class controls the ordering of the sprites.

6.30 Tutorial Class Reference

class to control tutorial scene.

Inheritance diagram for Tutorial:



Public Member Functions

- void [ReturnToMainMenu\(\)](#)
when button clicked, will return user to main menu
- void [ReturnToPauseMenu\(\)](#)
when button clicked, will return user to pause menu

Public Attributes

- string [mainMenuScene](#)
- string [pauseMenuScene](#)

6.30.1 Detailed Description

class to control tutorial scene.

6.30.2 Member Function Documentation

6.30.2.1 ReturnToMainMenu()

```
void Tutorial.ReturnToMainMenu ( )
```

when button clicked, will return user to main menu

6.30.2.2 ReturnToPauseMenu()

```
void Tutorial.ReturnToPauseMenu ( )
```

when button clicked, will return user to pause menu

6.30.3 Member Data Documentation

6.30.3.1 mainMenuScene

```
string Tutorial.mainMenuScene
```

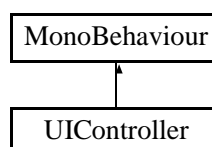
6.30.3.2 pauseMenuScene

```
string Tutorial.pauseMenuScene
```

6.31 UIController Class Reference

This class is tasked with managing the User Interface.

Inheritance diagram for UIController:



Public Member Functions

- void[StartFadeToBlack\(\)](#)
- void[ReturnToMainMenu\(\)](#)
when the users clicks to return to main menu, this happens
- void[Resume\(\)](#)
when the user clicks resume, the gameplay is resumed

Public Attributes

- Slider[healthSlider](#)
- Text[healthText](#)
- GameObject[pauseMenu](#)
- string[mainMenuScene](#)
- Image[fadeScreen](#)
- float[fadeSpeed](#)
- Image[currentGun](#)
- Text[gunText](#)
- Slider[essayPageSlider](#)
- Text[essayPageText](#)
- Image[numPages](#)

Static Public Attributes

- static[UIControllerinstance](#)

6.31.1 Detailed Description

This class is tasked with managing the User Interface.

6.31.2 Member Function Documentation

6.31.2.1 Resume()

```
void UIController.Resume ( )
```

when the user clicks resume, the gameplay is resumed

6.31.2.2 ReturnToMainMenu()

```
void UIController.ReturnToMainMenu ( )
```

when the users clicks to return to main menu, this happens

6.31.2.3 StartFadeToBlack()

```
void UIController.StartFadeToBlack ( )
```

6.31.3 Member Data Documentation

6.31.3.1 currentGun

```
Image UIController.currentGun
```

6.31.3.2 essayPageSlider

```
Slider UIController.essayPageSlider
```

6.31.3.3 essayPageText

```
Text UIController.essayPageText
```

6.31.3.4 fadeScreen

```
Image UIController.fadeScreen
```

6.31.3.5 fadeSpeed

```
float UIController.fadeSpeed
```

6.31.3.6 gunText

```
Text UIController.gunText
```

6.31.3.7 healthSlider

Slider UIController.healthSlider

6.31.3.8 healthText

Text UIController.healthText

6.31.3.9 instance

[UIController](#)UIController.instance [static]

6.31.3.10 mainMenuScene

string UIController.mainMenuScene

6.31.3.11 numPages

Image UIController.numPages

6.31.3.12 pauseMenu

GameObject UIController.pauseMenu

Chapter 7

File Documentation

7.1 breakables.cs File Reference

Classes

- class [breakables](#)

this class controls the breakable objects such as the chests.

7.2 brokenPieces.cs File Reference

Classes

- class [brokenPieces](#)

this class controls the broken pieces released by the breakable object.

7.3 CameraController.cs File Reference

Classes

- class [CameraController](#)

This class controls the camera.

7.4 CharacterSelectManager.cs File Reference

Classes

- class [CharacterSelectManager](#)

Class to control the selection process of the game.

7.5 CharacterSelector.cs File Reference

Classes

- class[CharacterSelector](#)

This class controls the character selection process in the game.

7.6 DamagePlayer.cs FileReference

Classes

- class[DamagePlayer](#)

This class manages the player's health.

7.7 EnemyBullet.cs File Reference

Classes

- class[EnemyBullet](#)

This class controls the bullets fired by the enemies.

7.8 EnemyController.cs File Reference

Classes

- class[EnemyController](#)

This class controls the enemies.

7.9 enemySleepAnimation.cs File Reference

Classes

- class[enemySleepAnimation](#)

this class controls the sleep animation of enemies when their health drops to zero.

7.10 essayPageController.cs File Reference

Classes

- class[essayPageController](#)

controller class for the essay page collectibles.

7.11 essayPages.cs File Reference

Classes

- class [essayPages](#)

class controls the collection of essay pages.

7.12 Gun.cs File Reference

Classes

- class [Gun](#)

This class controls the guns used by the player.

7.13 GunChest.cs File Reference

Classes

- class [GunChest](#)

This class controls the chests where players can get new weapons.

7.14 GunPickup.cs File Reference

Classes

- class [GunPickup](#)

This class enables the player pick up guns.

7.15 healthPickup.cs File Reference

Classes

- class [healthPickup](#)

class controls health pickups to restore player health.

7.16 LevelExit.cs File Reference

Classes

- class [LevelExit](#)

This class selects the next level to load.

7.17 LevelGenerator.cs File Reference

Classes

- class[LevelGenerator](#)
This class generated levels randomly.
- class[RoomPrefabs](#)

7.18 LevelManager.cs File Reference

Classes

- class[LevelManager](#)
This class manages each level.

7.19 LoadSceneAfterCutscene.cs File Reference

Classes

- class[LoadSceneAfterCutscene](#)
This class decides the scene to load after cutscenes.

7.20 MainMenu.cs File Reference

Classes

- class[MainMenu](#)
This class creates and controls the main menu.

7.21 PauseMenu.cs File Reference

Classes

- class[PauseMenu](#)
class to control pause menu scene

7.22 PlayerBullet.cs File Reference

Classes

- class[PlayerBullet](#)
This class controls bullets fired by the player.

7.23 ~~playerController.cs~~ File Reference

Classes

- class [playerController](#)

This class controls the player.

7.24 PlayerHealthController.cs File Reference

Classes

- class [PlayerHealthController](#)

class controls the health of the player and updates UI.

7.25 Program.cs File Reference

Classes

- class [Roguelike_Dungeon.Program](#)

Namespaces

- namespace [Roguelike_Dungeon](#)

7.26 Room.cs File Reference

Classes

- class [Room](#)

Class that controls the operation of rooms.

7.27 RoomCentre.cs File Reference

Classes

- class [RoomCentre](#)

This class controls the room centres.

Typedefs

- using [Random](#) = UnityEngine.Random

7.27.1 Typedef Documentation

7.27.1.1 Random

```
usingRandom= UnityEngine.Random
```

7.28 spriteSortOrder.cs File Reference

Classes

- class[spriteSortOrder](#)

This class controls the ordering of the sprites.

7.29 Tutorial.cs File Reference

Classes

- class[Tutorial](#)

class to control tutorial scene.

7.30 UIController.cs File Reference

Classes

- class[UIController](#)

This class is tasked with managing the User Interface.