Code Documentation – Team Olive

Generated by Doxygen 1.8.20

Chapter

Namespace Index

1.1 Packages

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter

Class Index

3.1 Class List

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

Chapter

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 Roguelike_Dungeon Namespace Reference

Classes

- classProgram

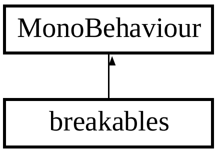
Chapter

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
- intmaxPieces= 5
- boolshouldDropItem

  list of pickups to drop
- GameObject[ ]itemsToDrop
- floatitemDropPercent

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

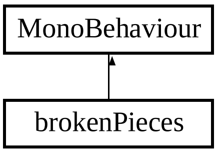
```
bool breakables.shouldDropItem
```

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

- floatmoveSpeed= 3f
- floatdeceleration= 5f
- floatlifetime= 3f
- SpriteRenderertheSR
- floatfadeSpeed= 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
```

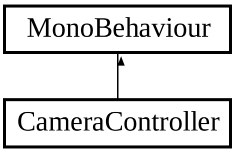
6.2.2.5 theSR

```
SpriteRenderer brokenPieces.theSR
```

6.3 CameraController Class Reference

This class controls the camera.

Inheritance diagram for CameraController:



Public Member Functions

- voidChangeTarget(Transform newTarget)

Public Attributes

- floatmoveSpeed

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

Static Public Attributes

- staticCameraControllerinstance

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

```
CameraControllerCameraController.instance [static]
```

6.3.3.2 moveSpeed

```
float CameraController.moveSpeed
```

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

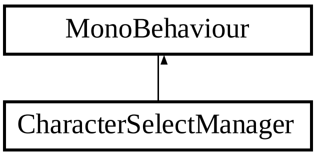
6.3.3.3 target

```
Transform CameraController.target
```

6.4 CharacterSelectManager Class Reference

Class to control the selection process of the game.

Inheritance diagram for CharacterSelectManager:



Public Attributes

- playerControlleractivePlayer
- CharacterSelectoractiveCharSelect

Static Public Attributes

- staticCharacterSelectManagerinstance

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

`CharacterSelectorCharacterSelectManager.activeCharSelect`

6.4.2.2 activePlayer

`playerControllerCharacterSelectManager.activePlayer`

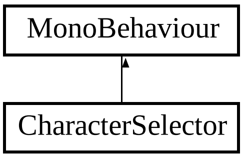
6.4.2.3 instance

`CharacterSelectManagerCharacterSelectManager.instance [static]`

6.5 CharacterSelector Class Reference

This class controls the character selection process in the game.

Inheritance diagram for CharacterSelector:



Public Attributes

- GameObjectmessage
- playerControllerplayerToSpawn

  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

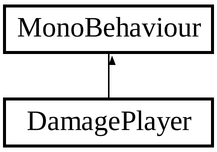
```
playerControllerCharacterSelector.playerToSpawn
```

The character to be used in the game.

6.6 DamagePlayer Class Reference

This class manages the player's health.

Inheritance diagram for DamagePlayer:



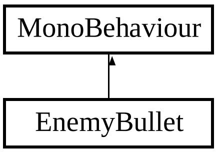
6.6.1 Detailed Description

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

- floatspeed

  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

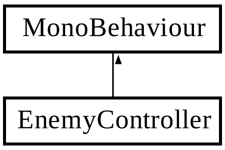
```
float EnemyBullet.speed
```

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

- voidDamageEnemy(int damage)

Public Attributes

- Rigidbody2DtheRB
- floatmoveSpeed
- boolshouldChasePlayer
- floatrangeToChasePlayer
- boolshouldChaseBot
- floatrangeToChaseBot

  variables for chase bot enemy
- boolshouldPatrol
- Transform[ ]patrolPoints
- boolshouldRunAway
- floatrunAwayRange

  variables for the coward enemy
- boolshouldWander
- floatwanderLength
- floatpauseLength

how long the enemy should pause for
- boolshouldShoot

  whether or not the enemy should shoot the player
- GameObjectbullet
- TransformfirePoint
- floatfireRate
- floatshootRange
- SpriteRenderertheBody

  allows us to change the sprite of enemies
- Animatoranim

  allows us to animate the enemies
- inthealth= 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
- GameObjecthitEffect

  hit effect for when the bullets hit the enemy
- boolcanMove= true
- GameObjectZzzz

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

Static Public Attributes

- staticEnemyControllerinstance

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

```
EnemyControllerEnemyController.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

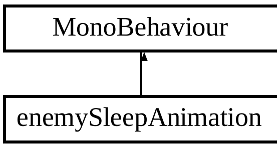
```
GameObject EnemyController.Zzzz
```

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

- voidenemyToSleep()

  replaces the enemy sprite with the sleeping sprite

Public Attributes

- SpriteRenderertheSR
- SpritesleepingEnemy

Static Public Attributes

- staticenemySleepAnimationinstance

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

```
enemySleepAnimationenemySleepAnimation.instance  [static]
```

6.9.3.2 sleepingEnemy

```
Sprite enemySleepAnimation.sleepingEnemy
```

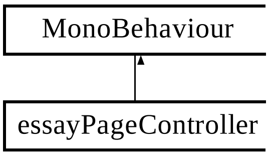
6.10.1.1 theSR

```
SpriteRenderer enemySleepAnimation.theSR
```

6.10 essayPageController Class Reference

controller class for the essay page collectibles.

Inheritance diagram for essayPageController:



Public Member Functions

- voidpickupPage()

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

Public Attributes

- intcurrentNumPages
- intmaxNumPages
- Sprite[ ]essayPageUI

Static Public Attributes

- staticessayPageControllerinstance

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]
```

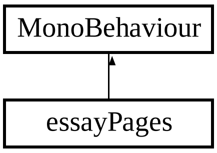
6.10.3.4 maxNumPages

```
int essayPageController.maxNumPages
```

6.11 essayPages Class Reference

class controls the collection of essay pages.

Inheritance diagram for essayPages:



Public Attributes

- stringtest

Static Public Attributes

- staticessayPagesinstance

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]
```

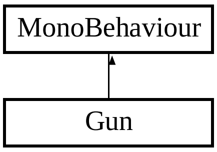
6.11.2.2 test

```
string essayPages.test
```

6.12 Gun Class Reference

This class controls the guns used by the player.

Inheritance diagram for Gun:



Public Attributes

- GameObjectbulletToFire

  differentiates between bullets (fruit)
- TransformfirePoint

  creates a fire point for the fruit to come from for each weapon
- floattimeBetweenShots
- stringweaponName
- SpritegunUI

  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
```

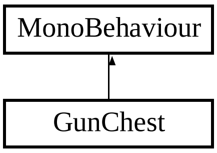
6.12.2.5 weaponName

```
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
- SpriteRenderertheSR
- SpritechestOpen

    allows the chest to change to an open sprite
- GameObjectnotification
- TransformspawnPoint

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

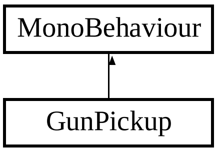
6.13.2.6 theSR

```
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

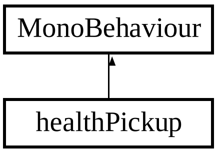
```
float GunPickup.waitToBeCollected = .5f
```

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

- inthealAmount= 1
- floatwaitToBeCollected= .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
```

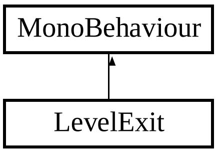
6.15.2.2 waitToBeCollected

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

6.16 LevelExit Class Reference

This class selects the next level to load.

Inheritance diagram for LevelExit:



Public Attributes

- stringlevelToLoad

6.16.1 Detailed Description

This class selects the next level to load.

6.16.2 Member Data Documentation

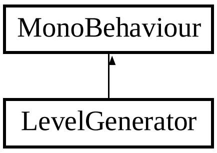
6.16.2.1 levelToLoad

```
string LevelExit.levelToLoad
```

6.17 LevelGenerator Class Reference

This class generated levels randomly.

Inheritance diagram for LevelGenerator:



Public Types

- enumDirection{Direction.up,Direction.right,Direction.down,Direction.left}

Public Member Functions

- voidMoveGenerationPoint()
- voidCreateRoomOutline(Vector3 roomPosition)

Public Attributes

- GameObjectlayoutRoom
- ColorstartColor
- intdistanceToEnd
- TransformgeneratorPoint
- DirectionselectedDirection
- floatxOffset= 18f
- LayerMaskwhatIsRoom
- List < RoomCentre > allRooms= new List <RoomCentre>()

  list of all room centres used in a given level
- intnumEnemiesInLevel= 0

  number of enemies in a given level
- boolnoEnemies= false
- intnumPagesInLevel= 0
- RoomPrefabsrooms
- RoomCentrecentreStart
- RoomCentre[ ]potentialCentres
- intmaxNumPagesInLevel

Static Public Attributes

- staticLevelGeneratorinstance

6.17.1 Detailed Description

This class generated levels randomly.

6.17.2 Member Enumeration Documentation

6.17.2.1 Direction

```
enumLevelGenerator.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

```
LevelGeneratorLevelGenerator.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
```

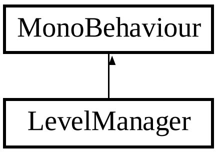
6.17.4.16 xOffset

```
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.

6.17 LevelGenerator Class Reference

Public Attributes

- floatwaitToLoad= 4f
- stringnextLevel
- stringartCutscene
- stringhistoryCutscene
- stringmusicCutscene
- stringscienceCutscene
- boolisPaused

    Is it a pause menu.
- TransformstartPoint
- stringthisPlayer

Static Public Attributes

- staticLevelManagerinstance

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
```

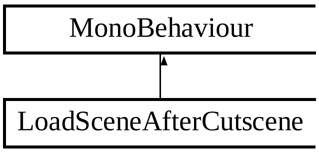
6.19.1.3 waitToLoad

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

6.19 LoadSceneAfterCutscene Class Reference

This class decides the scene to load after cutscenes.

Inheritance diagram for LoadSceneAfterCutscene:



Public Attributes

- stringnextLevel
- VideoPlayervideoPlayer

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
```

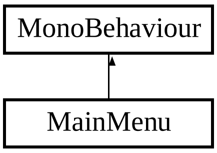
6.19.2.2 videoPlayer

```
VideoPlayer LoadSceneAfterCutscene.videoPlayer
```

6.20 MainMenu Class Reference

This class creates and controls the main menu.

Inheritance diagram for MainMenu:



Public Member Functions

- voidStartGame()
- voidGotoTutorial()
- voidExitGame()

Public Attributes

- stringlevelToLoad

Static Public Attributes

- staticMainMenuinstance
- static inttutorial

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
```

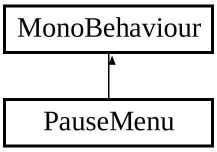
6.20.3.3 tutorial

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

6.21 PauseMenu Class Reference

class to control pause menu scene

Inheritance diagram for PauseMenu:



Public Member Functions

- voidPauseUnpause()
- voidResume()

  when button clicked, will resume game
- voidReturnToMainMenu()

  when button clicked, will return user to main menu
- voidopenTutorial()

  when button clicked, will load tutorial scene

Public Attributes

- GameObjectpauseMenu
- stringmainMenuScene
- stringtutorialScene

Static Public Attributes

- staticPauseMenuinstance
- static boolisPaused=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
```

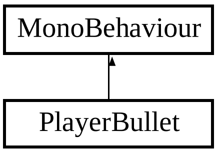
6.21.3.5 tutorialScene

```
string PauseMenu.tutorialScene
```

6.22 PlayerBullet Class Reference

This class controls bullets fired by the player.

Inheritance diagram for PlayerBullet:



Public Attributes

- floatspeed= 7.5f

  bullet speed is 7.5 by default, can be changed in Unity.
- Rigidbody2DtheRB
- GameObjectimpactEffect
- intdamageToGive= 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.

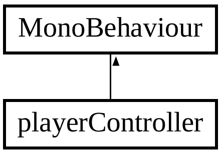
6.23.1.2 theRB

```
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

```
playerControllerplayerController.instance  [static]
```

public variables can be accessed in the Inspector pane in Unity

6.24.1.4 moveSpeed

```
float playerController.moveSpeed
```

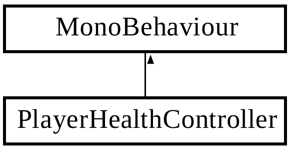
6.24.1.5 theRB

```
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

- voidDamagePlayer()

  method damages player when player is hit by a bullet or runs over a spike
- voidhealPlayer(int healAmount)

  method heals player when health pickup collected

Public Attributes

- intcurrentHealth
- intmaxHealth

  sets the current and max health. Can be changed in Unity
- floatdamageInvincLength= 1f

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

Static Public Attributes

- staticPlayerHealthControllerinstance

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

```
PlayerHealthControllerPlayerHealthController.instance  [static]
```

6.25.1.4 maxHealth

```
int PlayerHealthController.maxHealth
```

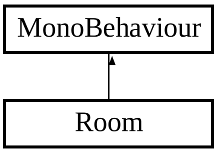
sets the current and max health. Can be changed in Unity

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

- boolcloseWhenEntered

  openWhenEnemiesCleared
- GameObject[ ]doors
- boolroomActive
- GameObjectlevelExit

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
```

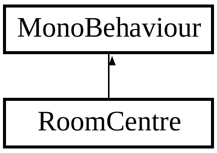
6.26.3.4 roomActive

```
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 byLevelGeneratorclass to calculate total number of enemies in level
- boolopenWhenEnemiesCleared
- RoomtheRoom
- GameObjectlevelExit
- 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 byLevelGeneratorclass to calculate total number of pages in level
- List < int > pageLocation= new List <int>()

  list of actual page locations used in that room
- intnumPagesInRoom
- intnewPageLocation

Static Public Attributes

- staticRoomCentreinstance

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 byLevelGeneratorclass 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 byLevelGeneratorclass 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

- GameObjectsingleUp

6.28.1 Member Data Documentation

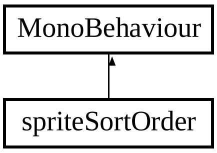
6.28.1.1 singleUp

```
GameObject RoomPrefabs.singleUp
```

6.29 spriteSortOrder Class Reference

This class controls the ordering of the sprites.

Inheritance diagram for spriteSortOrder:



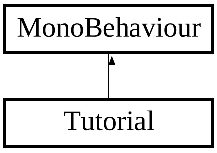
6.29.1 Detailed Description

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

- voidReturnToMainMenu()

  when button clicked, will return user to main menu
- voidReturnToPauseMenu()

  when button clicked, will return user to pause menu

Public Attributes

- stringmainMenuScene
- stringpauseMenuScene

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
```

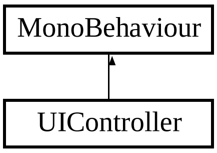
6.30.3.2 pauseMenuScene

```
string Tutorial.pauseMenuScene
```

6.31 UIController Class Reference

This class is tasked with managing the User Interface.

Inheritance diagram for UIController:



Public Member Functions

- voidStartFadeToBlack()
- voidReturnToMainMenu()

  when the users clicks to return to main menu, this happens
- voidResume()

  when the user clicks resume, the gameplay is resumed

Public Attributes

- SliderhealthSlider
- TexthealthText
- GameObjectpauseMenu
- stringmainMenuScene
- ImagefadeScreen
- floatfadeSpeed
- ImagecurrentGun
- TextgunText
- SlideressayPageSlider
- TextessayPageText
- ImagenumPages

Static Public Attributes

- staticUIControllerinstance

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

```
UIControllerUIController.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

- classbreakables

  this class controls the breakable objects such as the chests.

7.2 brokenPieces.cs File Reference

Classes

- classbrokenPieces

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

7.3 CameraController.cs File Reference

Classes

- classCameraController

  This class controls the camera.

7.4 CharacterSelectManager.cs File Reference

Classes

- classCharacterSelectManager

  Class to control the selection process of the game.

7.5 CharacterSelector.cs File Reference

Classes

- classCharacterSelector

  This class controls the character selection process in the game.

7.6 DamagePlayer.cs FileReference

Classes

- classDamagePlayer

  This class manages the player's health.

7.7 EnemyBullet.cs File Reference

Classes

- classEnemyBullet

  This class controls the bullets fired by the enemies.

7.8 EnemyController.cs File Reference

Classes

- classEnemyController

  This class controls the enemies.

7.9 enemySleepAnimation.cs File Reference

Classes

- classenemySleepAnimation

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

7.10 essayPageController.cs File Reference

Classes

- classessayPageController

  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

- classLevelGenerator

  This class generated levels randomly.
- classRoomPrefabs

7.18 LevelManager.cs File Reference

Classes

- classLevelManager

  This class manages each level.

7.19 LoadSceneAfterCutscene.cs File Reference

Classes

- classLoadSceneAfterCutscene

  This class decides the scene to load after cutscenes.

7.20 MainMenu.cs File Reference

Classes

- classMainMenu

  This class creates and controls the main menu.

7.21 PauseMenu.cs File Reference

Classes

- classPauseMenu

  class to control pause menu scene

7.22 PlayerBullet.cs File Reference

Classes

- classPlayerBullet

  This class controls bullets fired by the player.

7.23 playerController.cs File Reference

Classes

- classplayerController

  This class controls the player.

7.24 PlayerHealthController.cs File Reference

Classes

- classPlayerHealthController

  class controls the health of the player and updates UI.

7.25 Program.cs File Reference

Classes

- classRoguelike_Dungeon.Program

Namespaces

- namespaceRoguelike_Dungeon

7.26 Room.cs File Reference

Classes

- classRoom

  Class that controls the operation of rooms.

7.27 RoomCentre.cs File Reference

Classes

- classRoomCentre

  This class controls the room centres.

Typedefs

- usingRandom= UnityEngine.Random

7.27.1 Typedef Documentation

7.27.1.1 Random

```
usingRandom= UnityEngine.Random
```

7.28 spriteSortOrder.cs File Reference

Classes

- classspriteSortOrder

  This class controls the ordering of the sprites.

7.29 Tutorial.cs File Reference

Classes

- classTutorial

  class to control tutorial scene.

7.30 UIController.cs File Reference

Classes

- classUIController

  This class is tasked with managing the User Interface.