# Impossible Circle - Round The Balls DOCUMENTATION

Generated by Doxygen 1.8.10

Wed Apr 13 2016 14:54:24

# Contents


**1 Hierarchical Index** **1**

1.1 Class Hierarchy . . . 1

**2 Class Index** **3**

2.1 Class List . . . 3

**3 Class Documentation** **5**

3.1 ButtonLeaderboard Class Reference . . . 5

3.1.1 Detailed Description . . . 5

3.1.2 Member Function Documentation . . . 5

3.1.2.1 OnClickedOpenLeaderboard() . . . 5

3.2 CameraManager Class Reference . . . 6

3.2.1 Detailed Description . . . 6

3.3 CanvasManager Class Reference . . . 7

3.3.1 Detailed Description . . . 7

3.4 Circle Class Reference . . . 7

3.4.1 Detailed Description . . . 8

3.5 CollisionDetection Class Reference . . . 8

3.5.1 Detailed Description . . . 8

3.5.2 Member Function Documentation . . . 9

3.5.2.1 OnTriggerEnter2D(Collider2D other) . . . 9

3.6 GameManager Class Reference . . . 9

3.6.1 Detailed Description . . . 10

3.6.2 Member Function Documentation . . . 10

3.6.2.1 ReportScoreToLeaderboard(int p) . . . 10

3.7 InputTouch Class Reference . . . 10

3.7.1 Detailed Description . . . 11

3.7.2 Member Function Documentation . . . 11

3.7.2.1 OnTouchDown(TouchDirection td) . . . 11

3.7.2.2 OnTouchUp() . . . 11

3.7.2.3 Update() . . . 11

3.8 ObstacleLogic Class Reference . . . 11

3.8.1 Detailed Description . . . 12
3.9 ParticleLogic Class Reference . . . 12
3.9.1 Detailed Description . . . 12
3.10 Player Class Reference . . . 13
3.10.1 Detailed Description . . . 13
3.11 Singleton< T > Class Template Reference . . . 14
3.11.1 Detailed Description . . . 14
3.11.2 Member Function Documentation . . . 14
3.11.2.1 OnDestroy() . . . 14
3.12 SoundManager Class Reference . . . 15
3.12.1 Detailed Description . . . 15
3.13 SpawnManager Class Reference . . . 15
3.13.1 Detailed Description . . . 16
3.14 Util Class Reference . . . 16
3.14.1 Detailed Description . . . 16

**Index** **19**

# Chapter 1

# Hierarchical Index

## 1.1 Class Hierarchy

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


MonoBehaviour
- ButtonLeaderboard . . . 5
- CameraManager . . . 6
- CanvasManager . . . 7
- Circle . . . 7
- CollisionDetection . . . 8
- GameManager . . . 9
- InputTouch . . . 10
- ObstacleLogic . . . 11
- ParticleLogic . . . 12
- Player . . . 13
- Singleton< T > . . . 14
- SoundManager . . . 15

Singleton< SpawnManager > . . . 14
- SpawnManager . . . 15

Util . . . 16

# Chapter 2

# Class Index

## 2.1 Class List

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


ButtonLeaderboard
: Class attached to the leaderboard button. Works only on mobile (iOS & Android), with Very Simple Leaderboard : http://u3d.as/qxf . . . 5

CameraManager
: Class in charge to the camera animation: zoom in at start, shake when player jumps, and follow the player during the game . . . 6

CanvasManager
: Class in charge to handle UI elements. . . . 7

Circle
: Class in charge to draw the circles. . . . 7

CollisionDetection
: Class attached to the sprite child of the Player GameOBject, in charge to listen if the player collide with an obstacle . . . 8

GameManager
: Class in charge of the logic of the game. This class will restart the level at game over, handle and save the point, and call the Ads if you import the VERY SIMPLE ADS asset available here: http://u3d.as/oWD . . . 9

InputTouch
: Class in charge to listen the touch or click, and send event to subscribers . . . 10

ObstacleLogic
: Attached to ObstaclePrefab prefab in the prefab folder. In charge to initialize the spike and anim them in, and destroy them if ot of screen. . . . 11

ParticleLogic
: Class in charge to desactivate the particles. . . . 12

Player
: Class attached to the Player GameObject in the hierarchy. In charge to handle the Player the player controls, detect touch and collision. . . . 13

Singleton< T >
: Be aware this will not prevent a non singleton constructor such as `T myT = new T();` To prevent that, add `protected T () {}` to your singleton class . . . 14

SoundManager
: Class in charge to play sound in the game. . . . 15

SpawnManager
: Class in charge to pooled the particle to not have to instantiate and destroy them each time the player make a point (about 200 particles each time the player earns 1 point). . . . 15

Util
: Utility class. . . . 16

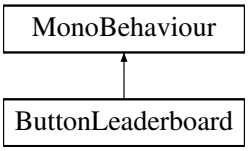
# Chapter 3

# Class Documentation

## 3.1 ButtonLeaderboard Class Reference

Class attached to the leaderboard button. Works only on mobile (iOS & Android), with Very Simple Leaderboard : http://u3d.as/qxf

Inheritance diagram for ButtonLeaderboard:




### Public Member Functions

- void OnClickedOpenLeaderboard ()

  *If player clics on the leaderbord button, we call this method. Works only on mobile (iOS & Android) if using Very Simple Leaderboard by App Advisory :* http://u3d.as/qxf

### Private Member Functions

- void **Awake** ()

### 3.1.1 Detailed Description

Class attached to the leaderboard button. Works only on mobile (iOS & Android), with Very Simple Leaderboard : http://u3d.as/qxf

### 3.1.2 Member Function Documentation

#### 3.1.2.1 void ButtonLeaderboard.OnClickedOpenLeaderboard ( ) [inline]

If player clics on the leaderbord button, we call this method. Works only on mobile (iOS & Android) if using Very Simple Leaderboard by App Advisory : http://u3d.as/qxf

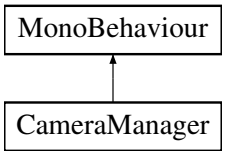
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/ButtonLeaderboard.cs

## 3.2 CameraManager Class Reference

Class in charge to the camera animation: zoom in at start, shake when player jumps, and follow the player during the game

Inheritance diagram for CameraManager:




### Public Member Functions

- void **DOStart** (Action callback)
- void **DOShake** ()

### Public Attributes

- Transform **toFollow**

### Private Member Functions

- void **Awake** ()
- IEnumerator **Start** ()
- IEnumerator **DOFollow** ()

### Private Attributes

- Camera **cam**
- GameManager **gameManager**
- float **orthoSize** = 2.3f

### 3.2.1 Detailed Description

Class in charge to the camera animation: zoom in at start, shake when player jumps, and follow the player during the game

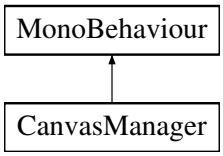
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/CameraManager.cs

## 3.3 CanvasManager Class Reference

Class in charge to handle UI elements.

Inheritance diagram for CanvasManager:




### Public Member Functions

- void **OnStartGame** (Action complete)
- void **OnGameOver** (Action complete)
- void **OnClickedStart** ()

### Public Attributes

- CanvasGroup **canvasGroupStart**
- CanvasGroup **canvasGroupInGame**
- Text **bestScoreText**
- Text **lastScoreText**

### Private Member Functions

- void **Awake** ()
- void **Start** ()
- void **UpdateText** ()

### 3.3.1 Detailed Description

Class in charge to handle UI elements.

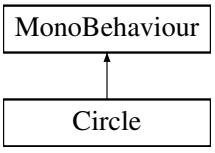
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/CanvasManager.cs

## 3.4 Circle Class Reference

Class in charge to draw the circles.

Inheritance diagram for Circle:



### Public Member Functions

- void **SetRadius** (float radius)
- float **GetRadius** ()
- float **GetWidth** ()
- void **DOParticle** ()
- void **DOStart** ()
- void **CreatePoints** ()

### Private Member Functions

- void **Awake** ()

### Private Attributes

- GameManager **gameManager**
- int **segments** = 200
- LineRenderer **line**
- float **radius**
- float **width** = 0.3f
- Transform **playerSpriteTransform**

#### 3.4.1 Detailed Description

Class in charge to draw the circles.

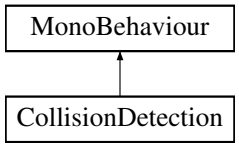
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/Circle.cs

## 3.5 CollisionDetection Class Reference

Class attached to the sprite child of the Player GameOBject, in charge to listen if the player collide with an obstacle

Inheritance diagram for CollisionDetection:




### Public Member Functions

- void OnTriggerEnter2D (Collider2D other)

  *Listen the collision. If collision: all the Player method DOOnTriggerEnter2D*

#### 3.5.1 Detailed Description

Class attached to the sprite child of the Player GameOBject, in charge to listen if the player collide with an obstacle

### 3.5.2 Member Function Documentation

#### 3.5.2.1 void CollisionDetection.OnTriggerEnter2D ( Collider2D *other* ) [inline]

Listen the collision. If collision: all the Player method DOOnTriggerEnter2D

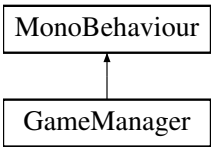
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/CollisionDetection.cs

## 3.6 GameManager Class Reference

Class in charge of the logic of the game. This class will restart the level at game over, handle and save the point, and call the Ads if you import the VERY SIMPLE ADS asset available here: http://u3d.as/oWD

Inheritance diagram for GameManager:




### Public Member Functions

- void **Start** ()
- void **Add1Point** ()
- void **InstantiateCircle** ()
- void **DOStart** ()
- void **ShowAds** ()
- void **GameOver** ()

### Public Attributes

- string **VerySimpleAdsURL** = "http://u3d.as/oWD"
- Color **backgroundColor** = Color.white
- Color **circleColor** = Color.black
- Color **playerColor** = Color.white
- bool **activateCameraShake** = false
- int **numberOfPlayToShowInterstitial** = 5
- bool **isStarted** = false
- bool **isGameOver** = false
- Text **pointText**
- GameObject **circlePrefab**
- GameObject **particle**
- GameObject **obstaclePrefab**
- float **radiusBorder**

### Properties

- Color **hazardColor** [get]

### Private Member Functions

- void **Awake** ()
- void **Init** ()
- IEnumerator **ObstacleInstantiator** ()
- void **DOInstantiateObstacle** ()
- void **SetNewGame** ()
- void ReportScoreToLeaderboard (int p)

  *If using Very Simple Leaderboard by App Advisory, report the score :* http://u3d.as/qxf

### Private Attributes

- CanvasScaler **canvasScaler**
- int **point** = 0
- Player **player**
- SoundManager **soundManager**

#### 3.6.1 Detailed Description

Class in charge of the logic of the game. This class will restart the level at game over, handle and save the point, and call the Ads if you import the VERY SIMPLE ADS asset available here: http://u3d.as/oWD

#### 3.6.2 Member Function Documentation

##### 3.6.2.1 void GameManager.ReportScoreToLeaderboard ( int *p* ) [inline], [private]

If using Very Simple Leaderboard by App Advisory, report the score : http://u3d.as/qxf

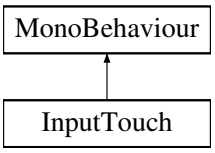
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/GameManager.cs

## 3.7 InputTouch Class Reference

Class in charge to listen the touch or click, and send event to subscribers

Inheritance diagram for InputTouch:




### Public Member Functions

- delegate void OnTouchDown (TouchDirection td)

  *Delegate to listen the touch down or click down, and send event to subscribers*
- delegate void OnTouchUp ()

  *Delegate to listen the touch up or click up, and send event to subscribers*

### Events

- static OnTouchDown **OnTouchedDown**
- static OnTouchUp **OnTouchedUp**

### Private Member Functions

- void Update ()

  *Listening for inputs*

### 3.7.1 Detailed Description

Class in charge to listen the touch or click, and send event to subscribers

### 3.7.2 Member Function Documentation

#### 3.7.2.1 delegate void InputTouch.OnTouchDown ( TouchDirection *td* )

Delegate to listen the touch down or click down, and send event to subscribers

#### 3.7.2.2 delegate void InputTouch.OnTouchUp ( )

Delegate to listen the touch up or click up, and send event to subscribers

#### 3.7.2.3 void InputTouch.Update ( ) [inline], [private]

Listening for inputs

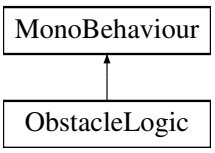
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/InputTouch.cs

## 3.8 ObstacleLogic Class Reference

Attached to ObstaclePrefab prefab in the prefab folder. In charge to initialize the spike and anim them in, and destroy them if ot of screen.

Inheritance diagram for ObstacleLogic:




### Public Member Functions

- void **Init** (float rot, bool isDown)
- void **DOPosition** (Vector3 rotation, bool isDown, Action callback)

### Public Attributes

- GameObject **obstacleSprite**
- bool **isVisible** = false

### Private Member Functions

- void **Awake** ()
- IEnumerator **DOStart** (float rot, bool isDown)
- IEnumerator **CheckVisibility** ()

### Private Attributes

- SpriteRenderer **sr**
- GameManager **gameManager**
- float **decal** = -1f

#### 3.8.1 Detailed Description

Attached to ObstaclePrefab prefab in the prefab folder. In charge to initialize the spike and anim them in, and destroy them if ot of screen.

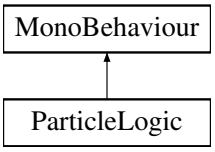
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/ObstacleLogic.cs

## 3.9 ParticleLogic Class Reference

Class in charge to desactivate the particles.

Inheritance diagram for ParticleLogic:




### Private Member Functions

- void **OnEnable** ()
- void **OnDisable** ()
- IEnumerator **waitForDesactivate** ()

#### 3.9.1 Detailed Description

Class in charge to desactivate the particles.

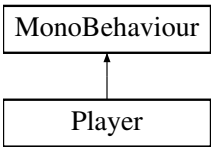
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/ParticleLogic.cs

## 3.10 Player Class Reference

Class attached to the Player GameObject in the hierarchy. In charge to handle the Player the player controls, detect touch and collision.

Inheritance diagram for Player:




### Public Member Functions

- float **GetRotation** ()
- void **DOPosition** (float x, float y)
- void **DOStart** ()
- void **ActivateTouchControl** ()
- void **DesactivateTouchControl** ()
- void **AnimPlayer** (float targetPos)
- IEnumerator **_AnimPlayer** (float targetPos)
- void **DOOnTriggerEnter2D** (Collider2D other)

### Public Attributes

- Transform **playerSprite**
- SpriteRenderer **sr**
- Vector3 **defaultPosition** = Vector3.zero

### Properties

- float **jumpWidth** `[get]`

### Private Member Functions

- void **Awake** ()
- void **Start** ()
- void **OnTouchDown** (TouchDirection td)
- void **OnCompleteJump** ()

### Private Attributes

- GameManager **gameManager**

### 3.10.1 Detailed Description

Class attached to the Player GameObject in the hierarchy. In charge to handle the Player the player controls, detect touch and collision.

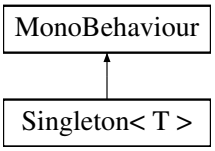
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/Player.cs

## 3.11 Singleton< T > Class Template Reference

Be aware this will not prevent a non singleton constructor such as `T myT = new T();` To prevent that, add `protected T () {}` to your singleton class.

Inheritance diagram for Singleton< T >:




### Public Member Functions

- void OnDestroy ()

  *When Unity quits, it destroys objects in a random order. In principle, a Singleton is only destroyed when application quits. If any script calls Instance after it have been destroyed, it will create a buggy ghost object that will stay on the Editor scene even after stopping playing the Application. Really bad! So, this was made to be sure we're not creating that buggy ghost object.*

### Properties

- static T **instance** `[get]`
- static T **Instance** `[get]`

### Static Private Attributes

- static T **_instance**
- static object **_lock** = new object()
- static bool **applicationIsQuitting** = false

### 3.11.1 Detailed Description

Be aware this will not prevent a non singleton constructor such as `T myT = new T();` To prevent that, add `protected T () {}` to your singleton class.

As a note, this is made as MonoBehaviour because we need Coroutines.

#### Type Constraints

***T* : *MonoBehaviour***

### 3.11.2 Member Function Documentation

#### 3.11.2.1 void Singleton< T >.OnDestroy ( ) [inline]

When Unity quits, it destroys objects in a random order. In principle, a Singleton is only destroyed when application quits. If any script calls Instance after it have been destroyed, it will create a buggy ghost object that will stay on the Editor scene even after stopping playing the Application. Really bad! So, this was made to be sure we're not creating that buggy ghost object.

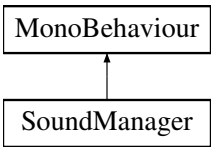
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/Singleton.cs

## 3.12 SoundManager Class Reference

Class in charge to play sound in the game.

Inheritance diagram for SoundManager:




### Public Member Functions

- void **PlayFail** ()
- void **PlayTouch** ()

### Properties

- AudioSource **audioSource** `[get]`

### Private Attributes

- AudioSource **_audioSource**
- AudioClip **soundFail**
- AudioClip **soundTouch**

#### 3.12.1 Detailed Description

Class in charge to play sound in the game.

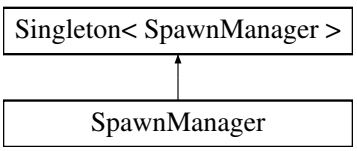
The documentation for this class was generated from the following file:

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/SoundManager.cs

## 3.13 SpawnManager Class Reference

Class in charge to pooled the particle to not have to instantiate and destroy them each time the player make a point (about 200 particles each time the player earns 1 point).

Inheritance diagram for SpawnManager:




### Public Member Functions

- void **Init** ()
- void **SpawnParticle** (Color c, Vector3 pos, Vector3 rot)

### Public Attributes

- GameObject **particle**

### Private Member Functions

- void **Awake** ()
- void **CreatePoolParticle** ()

### Private Attributes

- List< EllipsoidParticleEmitter > **listParticle** = new List<EllipsoidParticleEmitter>()

### Additional Inherited Members

#### 3.13.1 Detailed Description

Class in charge to pooled the particle to not have to instantiate and destroy them each time the player make a point (about 200 particles each time the player earns 1 point).

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

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/SpawnManager.cs

## 3.14 Util Class Reference

Utility class.

### Static Public Member Functions

- static double **GetRandomNumber** (double minimum, double maximum)
- static float **GetRandomNumber** (float minimum, float maximum)
- static void **SetLastScore** (int score)
- static int **GetBestScore** ()
- static int **GetLastScore** ()

### Static Private Member Functions

- static void **SetBestScore** (int score)

### Static Private Attributes

- static System.Random **random** = new System.Random()

#### 3.14.1 Detailed Description

Utility class.

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

- /Volumes/LaCie/Dropbox/Anthony/__AppAdvisory/Impossible Circle Round The Ball/Assets/Impossible↩ Circle_RoudTheBalls/Scripts/Util.cs

# Index


ButtonLeaderboard, 5
    OnClickedOpenLeaderboard, 5

CameraManager, 6
CanvasManager, 7
Circle, 7
CollisionDetection, 8
    OnTriggerEnter2D, 9

GameManager, 9
    ReportScoreToLeaderboard, 10

InputTouch, 10
    OnTouchDown, 11
    OnTouchUp, 11
    Update, 11

ObstacleLogic, 11
OnClickedOpenLeaderboard
    ButtonLeaderboard, 5
OnDestroy
    Singleton, 14
OnTouchDown
    InputTouch, 11
OnTouchUp
    InputTouch, 11
OnTriggerEnter2D
    CollisionDetection, 9

ParticleLogic, 12
Player, 13

ReportScoreToLeaderboard
    GameManager, 10

Singleton
    OnDestroy, 14
Singleton< T >, 14
SoundManager, 15
SpawnManager, 15

Update
    InputTouch, 11
Util, 16