

## Looping Mania

Generated by Doxygen 1.8.10

Tue May 10 2016 18:41:57



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	AppAdvisory Namespace Reference	7
4.2	AppAdvisory.LoopMania Namespace Reference	7
4.2.1	Enumeration Type Documentation	8
4.2.1.1	TouchDirection	8
4.3	AppAdvisory.UI Namespace Reference	8
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	AppAdvisory.LoopMania.AnimPointText Class Reference	9
5.1.1	Detailed Description	9
5.1.2	Member Function Documentation	9
5.1.2.1	Awake()	9
5.1.2.2	DoAnimPointIn()	10
5.1.2.3	DoAnimPointOut()	10
5.1.2.4	Start()	10
5.2	AppAdvisory.LoopMania.Ball Class Reference	10
5.2.1	Detailed Description	11
5.2.2	Member Function Documentation	11
5.2.2.1	ActivateTouchControl()	11
5.2.2.2	Awake()	11
5.2.2.3	DeactivateTouchControl()	12
5.2.2.4	DOOnTriggerEnter2D(Collider2D other)	12
5.2.2.5	DOPosition(float x, float y)	12
5.2.2.6	DOSTart()	12

5.2.2.7	GetRotation()	12
5.2.2.8	OnDestroy()	12
5.2.2.9	OnDisable()	12
5.2.2.10	OnMainColorChanged(Color c)	12
5.2.2.11	OnPlayerJumpFinished()	12
5.2.2.12	OnTouchDown(TouchDirection td)	12
5.2.2.13	SpriteSize()	13
5.2.3	Member Data Documentation	13
5.2.3.1	gameManager	13
5.2.3.2	is_hazard	13
5.2.3.3	is_jumping	13
5.2.3.4	playerSprite	13
5.2.3.5	soundManager	13
5.2.3.6	sr	13
5.3	AppAdvisory.UI.ButtonLeaderboard Class Reference	13
5.3.1	Detailed Description	14
5.3.2	Member Function Documentation	14
5.3.2.1	OnClickedOpenLeaderboard()	14
5.4	AppAdvisory.LoopMania.ButtonLeaderboard Class Reference	14
5.4.1	Detailed Description	14
5.4.2	Member Function Documentation	15
5.4.2.1	OnClickedOpenLeaderboard()	15
5.5	AppAdvisory.UI.ButtonLike Class Reference	15
5.5.1	Detailed Description	15
5.5.2	Member Function Documentation	15
5.5.2.1	OnClickedFacebookLikeButton()	15
5.5.3	Member Data Documentation	15
5.5.3.1	facebookAddress	15
5.5.3.2	facebookApp	16
5.6	AppAdvisory.LoopMania.ButtonLike Class Reference	16
5.6.1	Detailed Description	16
5.7	AppAdvisory.UI.ButtonOpenUrl Class Reference	16
5.7.1	Detailed Description	17
5.8	AppAdvisory.UI.ButtonRate Class Reference	17
5.8.1	Detailed Description	17
5.8.2	Member Function Documentation	18
5.8.2.1	OnClickedRate()	18
5.8.3	Member Data Documentation	18
5.8.3.1	amazonRateURL	18
5.8.3.2	androidRateURL	18

5.8.3.3	iosRateURL	18
5.9	AppAdvisory.UI.ButtonWatchAd Class Reference	18
5.9.1	Detailed Description	18
5.9.2	Member Function Documentation	19
5.9.2.1	OnClickedWatchAd()	19
5.10	AppAdvisory.LoopMania.Circle Class Reference	19
5.10.1	Detailed Description	20
5.10.2	Member Function Documentation	20
5.10.2.1	Awake()	20
5.10.2.2	DOStart(float radius)	20
5.10.2.3	OnDestroy()	20
5.10.2.4	OnDisable()	20
5.10.2.5	OnMainColorChanged(Color c)	20
5.10.3	Member Data Documentation	20
5.10.3.1	gameManager	20
5.10.3.2	line	20
5.10.3.3	radius	20
5.10.3.4	real_width	21
5.10.3.5	segments	21
5.11	AppAdvisory.LoopMania.CollisionDetection Class Reference	21
5.11.1	Detailed Description	21
5.11.2	Member Function Documentation	21
5.11.2.1	OnTriggerEnter2D(Collider2D other)	21
5.12	DesactivatelfMobile Class Reference	21
5.12.1	Detailed Description	22
5.13	AppAdvisory.LoopMania.DotToCollect Class Reference	22
5.13.1	Detailed Description	23
5.14	AppAdvisory.LoopMania.GameManager Class Reference	23
5.14.1	Detailed Description	25
5.14.2	Member Function Documentation	25
5.14.2.1	Add1Point()	25
5.14.2.2	Awake()	25
5.14.2.3	DOInstantiateDotToCollect()	26
5.14.2.4	DOMainColorChange()	26
5.14.2.5	DoMainColorLerp(Color from, Color to, float time)	26
5.14.2.6	DOParticle(Vector3 position)	26
5.14.2.7	DOPlayerJumpFinished()	26
5.14.2.8	DOStart()	26
5.14.2.9	GameOver(Transform t1)	26
5.14.2.10	InstantiateHazard()	26

5.14.2.11 OnMainColorChange(Color c)	26
5.14.2.12 OnPlayerJumpFinish()	26
5.14.2.13 ReportScoreToLeaderboard(int p)	26
5.14.2.14 ShowAds()	27
5.14.2.15 Start()	27
5.14.2.16 StartDotColorChange()	27
5.14.3 Member Data Documentation	27
5.14.3.1 ballPrefab	27
5.14.3.2 cameraSize	27
5.14.3.3 circle	27
5.14.3.4 dotMaterial	27
5.14.3.5 dotToCollectPrefab	27
5.14.3.6 isGameOver	27
5.14.3.7 listColors	27
5.14.3.8 listDotColors	27
5.14.3.9 m_currentMainColor	28
5.14.3.10 numberOfDotsToCreate	28
5.14.3.11 numberOfPlayToShowInterstitial	28
5.14.3.12 particleExplosion	28
5.14.3.13 player	28
5.14.3.14 playerJumpSpeedInSeconds	28
5.14.3.15 point	28
5.14.3.16 pointText	28
5.14.3.17 radiusBorder	28
5.14.3.18 soundManager	28
5.14.3.19 timeAnimItemInDotToCollect	28
5.14.3.20 timeForCompleteCirclePlayer	28
5.14.3.21 VerySimpleAdsURL	29
5.14.4 Property Documentation	29
5.14.4.1 currentMainColor	29
5.14.5 Event Documentation	29
5.14.5.1 OnMainColorChanged	29
5.14.5.2 OnPlayerJumpFinished	29
5.15 AppAdvisory.LoopMania.InputTouch Class Reference	29
5.15.1 Detailed Description	30
5.15.2 Member Function Documentation	30
5.15.2.1 OnTouchDown(TouchDirection td)	30
5.15.2.2 OnTouchUp()	30
5.15.2.3 Update()	30
5.16 AppAdvisory.UI.UIController.OnUIAnimInEndHandler Class Reference	30

5.16.1 Detailed Description . . . . .	30
5.17 AppAdvisory.UI.UIController.OnUIAnimInStartHandler Class Reference . . . . .	30
5.17.1 Detailed Description . . . . .	31
5.18 AppAdvisory.UI.UIController.OnUIAnimOUTEndHandler Class Reference . . . . .	31
5.18.1 Detailed Description . . . . .	31
5.19 AppAdvisory.UI.UIController.OnUIAnimOUTStartHandler Class Reference . . . . .	31
5.19.1 Detailed Description . . . . .	32
5.20 AppAdvisory.LoopMania.ParticleAutoDestruct Class Reference . . . . .	32
5.20.1 Detailed Description . . . . .	32
5.21 AppAdvisory.LoopMania.ParticleLogic Class Reference . . . . .	32
5.21.1 Detailed Description . . . . .	33
5.22 AppAdvisory.LoopMania.Singleton< T > Class Template Reference . . . . .	33
5.22.1 Detailed Description . . . . .	33
5.22.2 Member Function Documentation . . . . .	34
5.22.2.1 OnDestroy() . . . . .	34
5.23 AppAdvisory.LoopMania.SoundManager Class Reference . . . . .	34
5.23.1 Detailed Description . . . . .	35
5.23.2 Member Data Documentation . . . . .	35
5.23.2.1 _audioSource . . . . .	35
5.23.2.2 soundGameOverVoice . . . . .	35
5.23.2.3 soundHit . . . . .	35
5.23.2.4 soundJump . . . . .	35
5.23.2.5 soundMetal . . . . .	35
5.23.2.6 soundPoint . . . . .	35
5.23.2.7 soundReloadVoice . . . . .	35
5.23.2.8 soundStartVoice . . . . .	35
5.23.3 Property Documentation . . . . .	36
5.23.3.1 audioSource . . . . .	36
5.24 AppAdvisory.UI.UIController Class Reference . . . . .	36
5.24.1 Detailed Description . . . . .	37
5.24.2 Member Function Documentation . . . . .	37
5.24.2.1 AnimateINFromTop(Action callback) . . . . .	37
5.24.2.2 AnimateINHorizontaly(Action callback) . . . . .	37
5.24.2.3 AnimateOUTFromTop(Action callback) . . . . .	37
5.24.2.4 AnimateOUTHorizontaly(Action callback) . . . . .	38
5.24.2.5 DOAnimIN() . . . . .	38
5.24.2.6 DOAnimOUT() . . . . .	38
5.24.2.7 SetBestText(int point) . . . . .	38
5.24.2.8 SetLastText(int point) . . . . .	38
5.24.3 Member Data Documentation . . . . .	38

5.24.3.1	OnUIAnimInEnd . . . . .	38
5.24.3.2	OnUIAnimInStart . . . . .	38
5.24.3.3	OnUIAnimOutEnd . . . . .	38
5.24.3.4	OnUIAnimOutStart . . . . .	38
5.24.3.5	textBest . . . . .	38
5.24.3.6	textLast . . . . .	38
5.24.3.7	toAnimateFromTop . . . . .	39
5.24.3.8	toAnimateHorizontal . . . . .	39
5.25	AppAdvisory.LoopMania.Util Class Reference . . . . .	39
5.25.1	Detailed Description . . . . .	39
5.25.2	Member Function Documentation . . . . .	39
5.25.2.1	CleanMemory() . . . . .	39
5.25.2.2	GetBestScore() . . . . .	40
5.25.2.3	GetLastScore() . . . . .	40
5.25.2.4	ReloadLevel() . . . . .	40
5.25.2.5	SetAlpha(this Text text, float alpha) . . . . .	40
5.25.2.6	SetLastScore(int score) . . . . .	40
5.25.2.7	SetScaleX(this RectTransform rect, float scale) . . . . .	40
5.25.2.8	Shuffle< T >(this IList< T > list) . . . . .	40
5.26	Welcome Class Reference . . . . .	40
<b>Index</b>		<b>43</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">AppAdvisory</a>	7
<a href="#">AppAdvisory.LoopMania</a>	7
<a href="#">AppAdvisory.UI</a>	8



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

EditorWindow	
Welcome . . . . .	40
MonoBehaviour	
AppAdvisory.LoopMania.AnimPointText . . . . .	9
AppAdvisory.LoopMania.Ball . . . . .	10
AppAdvisory.LoopMania.ButtonLeaderboard . . . . .	14
AppAdvisory.LoopMania.ButtonLike . . . . .	16
AppAdvisory.LoopMania.Circle . . . . .	19
AppAdvisory.LoopMania.CollisionDetection . . . . .	21
AppAdvisory.LoopMania.DofToCollect . . . . .	22
AppAdvisory.LoopMania.GameManager . . . . .	23
AppAdvisory.LoopMania.InputTouch . . . . .	29
AppAdvisory.LoopMania.ParticleAutoDestruct . . . . .	32
AppAdvisory.LoopMania.ParticleLogic . . . . .	32
AppAdvisory.LoopMania.Singleton< T > . . . . .	33
AppAdvisory.LoopMania.SoundManager . . . . .	34
AppAdvisory.UI.ButtonLeaderboard . . . . .	13
AppAdvisory.UI.ButtonLike . . . . .	15
AppAdvisory.UI.ButtonOpenUrl . . . . .	16
AppAdvisory.UI.ButtonRate . . . . .	17
AppAdvisory.UI.ButtonWatchAd . . . . .	18
AppAdvisory.UI.UIController . . . . .	36
DesactivateIfMobile . . . . .	21
UnityEvent	
AppAdvisory.UI.UIController.OnUIAnimInEndHandler . . . . .	30
AppAdvisory.UI.UIController.OnUIAnimInStartHandler . . . . .	30
AppAdvisory.UI.UIController.OnUIAnimOUTEndHandler . . . . .	31
AppAdvisory.UI.UIController.OnUIAnimOUTStartHandler . . . . .	31
AppAdvisory.LoopMania.Util . . . . .	39



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">AppAdvisory.LoopMania.AnimPointText</a>	Attached to the GameObject UIController/PointTextContainer/pointText. In charge of the animation of the point UI element. . . . .	9
<a href="#">AppAdvisory.LoopMania.Ball</a>	Class attached to the Player GameObject in the hierarchy, or to the hazards. In charge to handle the Player controls and detect touch and collision, or to move the hazards around the circle. . .	10
<a href="#">AppAdvisory.UI.ButtonLeaderboard</a>	Class attached to the leaderboard button. Works only on mobile (iOS & Android), with Very Simple Leaderboard : <a href="http://u3d.as/qxf">http://u3d.as/qxf</a> . . . . .	13
<a href="#">AppAdvisory.LoopMania.ButtonLeaderboard</a>	Class attached to the leaderboard button. Works only on mobile (iOS & Android), with Very Simple Leaderboard : <a href="http://u3d.as/qxf">http://u3d.as/qxf</a> . . . . .	14
<a href="#">AppAdvisory.UI.ButtonLike</a>	Attached to like button . . . . .	15
<a href="#">AppAdvisory.LoopMania.ButtonLike</a>	Attached to like button . . . . .	16
<a href="#">AppAdvisory.UI.ButtonOpenUrl</a>	Class attached to button to open an url. . . . .	16
<a href="#">AppAdvisory.UI.ButtonRate</a>	Attached to rate button . . . . .	17
<a href="#">AppAdvisory.UI.ButtonWatchAd</a>	Class attached to the watch ad button. Works only on mobile (iOS & Android), with Very Simple Ads : <a href="http://u3d.as/oWD">http://u3d.as/oWD</a> . . . . .	18
<a href="#">AppAdvisory.LoopMania.Circle</a>	Class in charge to draw the circle. . . . .	19
<a href="#">AppAdvisory.LoopMania.CollisionDetection</a>	Class attached to the sprite child of the Player GameOObject, in charge to listen if the player collide with an obstacle . . . . .	21
<a href="#">DesactivatelfMobile</a>	Attached to GameObject we want to deactivate at star if we run the game on mobile. . . . .	21
<a href="#">AppAdvisory.LoopMania.DotToCollect</a>	Attached to ObstaclePrefab prefab in the prefab folder. Represent the little dot the player (Who is a Ball) have to collect around the Circle. . . . .	22
<a href="#">AppAdvisory.LoopMania.GameManager</a>	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: <a href="http://u3d.as/oWD">http://u3d.as/oWD</a> . . . . .	23

<a href="#">AppAdvisory.LoopMania.InputTouch</a>	
Class in charge to listen the touch or click, and send event to subscribers	29
<a href="#">AppAdvisory.UI.UIController.OnUIAnimInEndHandler</a>	
Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is ended.	30
<a href="#">AppAdvisory.UI.UIController.OnUIAnimInStartHandler</a>	
Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is started.	30
<a href="#">AppAdvisory.UI.UIController.OnUIAnimOUTEndHandler</a>	
Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is ended.	31
<a href="#">AppAdvisory.UI.UIController.OnUIAnimOUTStartHandler</a>	
Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is started.	31
<a href="#">AppAdvisory.LoopMania.ParticleAutoDestruct</a>	
Attached to each particle. In charge to destroy particle when the particle doesn't emit anything.	32
<a href="#">AppAdvisory.LoopMania.ParticleLogic</a>	
Class in charge to deactivate the particles.	32
<a href="#">AppAdvisory.LoopMania.Singleton&lt; T &gt;</a>	
Be aware this will not prevent a non singleton constructor such as <code>T myT = new T();</code> To prevent that, add <code>protected T () {}</code> to your singleton class	33
<a href="#">AppAdvisory.LoopMania.SoundManager</a>	
Class in charge to play sound in the game.	34
<a href="#">AppAdvisory.UI.UIController</a>	
Class attached to the <a href="#">UIController</a> GameObject (who is a Canvas). In Charge of all the logics of the UI: animation, events...	36
<a href="#">AppAdvisory.LoopMania.Util</a>	
Utility class.	39
<a href="#">Welcome</a>	40

## Chapter 4

# Namespace Documentation

### 4.1 AppAdvisory Namespace Reference

#### Namespaces

- namespace [LoopMania](#)
- namespace [UI](#)

### 4.2 AppAdvisory.LoopMania Namespace Reference

#### Classes

- class [AnimPointText](#)  
*Attached to the GameObject UIController/PointTextContainer/pointText. In charge of the animation of the point [UI](#) element.*
- class [Ball](#)  
*Class attached to the Player GameObject in the hierarchy, or to the hazards. In charge to handle the Player controls and detect touch and collision, or to move the hazards around the circle.*
- class [ButtonLeaderboard](#)  
*Class attached to the leaderboard button. Works only on mobile (iOS & Android), with Very Simple Leaderboard : <http://u3d.as/qxf>*
- class [ButtonLike](#)  
*Attached to like button*
- class [Circle](#)  
*Class in charge to draw the circle.*
- class [CollisionDetection](#)  
*Class attached to the sprite child of the Player GameOBJect, in charge to listen if the player collide with an obstacle*
- class [DotToCollect](#)  
*Attached to ObstaclePrefab prefab in the prefab folder. Represent the little dot the player (Who is a [Ball](#)) have to collect around the [Circle](#).*
- class [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>*
- class [InputTouch](#)  
*Class in charge to listen the touch or click, and send event to subscribers*
- class [ParticleAutoDestruct](#)  
*Attached to each particle. In charge to destroy particle when the particle doesn't emit anything.*
- class [ParticleLogic](#)

*Class in charge to deactivate the particles.*

- class [Singleton](#)

*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.*

- class [SoundManager](#)

*Class in charge to play sound in the game.*

- class [Util](#)

*Utility class.*

## Enumerations

- enum [TouchDirection](#) { **none**, **left**, **right** }

*3 type of touch: left if on the left of the screen, right if on the right of the screen, or none*

### 4.2.1 Enumeration Type Documentation

#### 4.2.1.1 enum [AppAdvisory.LoopMania.TouchDirection](#) [strong]

3 type of touch: left if on the left of the screen, right if on the right of the screen, or none

## 4.3 AppAdvisory.UI Namespace Reference

### Classes

- class [ButtonLeaderboard](#)

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

- class [ButtonLike](#)

*Attached to like button*

- class [ButtonOpenUrl](#)

*Class attached to button to open an url.*

- class [ButtonRate](#)

*Attached to rate button*

- class [ButtonWatchAd](#)

*Class attached to the watch ad button. Works only on mobile (iOS & Android), with Very Simple Ads : <http://u3d.as/oWD>*

- class [UIController](#)

*Class attached to the [UIController](#) GameObject (who is a Canvas). In Charge of all the logics of the [UI](#): animation, events...*



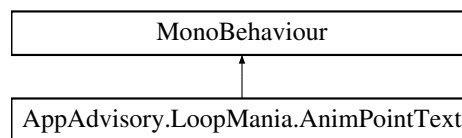
## Chapter 5

# Class Documentation

### 5.1 AppAdvisory.LoopMania.AnimationPoint Class Reference

Attached to the GameObject UIController/PointTextContainer/pointText. In charge of the animation of the point [UI](#) element.

Inheritance diagram for AppAdvisory.LoopMania.AnimationPoint:



#### Public Member Functions

- void [DoAnimPointIn](#) ()  
*Anim the point [UI](#) Text from out of the screen to in the screen.*
- void [DoAnimPointOut](#) ()  
*Anim the point [UI](#) Text from in the screen to out of the screen.*

#### Private Member Functions

- void [Awake](#) ()  
*Some initializations.*
- void [Start](#) ()  
*Some initializations.*

#### 5.1.1 Detailed Description

Attached to the GameObject UIController/PointTextContainer/pointText. In charge of the animation of the point [UI](#) element.

#### 5.1.2 Member Function Documentation

##### 5.1.2.1 void AppAdvisory.LoopMania.AnimationPoint.Awake ( ) [inline], [private]

Some initializations.

### 5.1.2.2 void AppAdvisory.LoopMania.AnimPointText.DoAnimPointIn ( ) [inline]

Anim the point [UI](#) Text from out of the screen to in the screen.

### 5.1.2.3 void AppAdvisory.LoopMania.AnimPointText.DoAnimPointOut ( ) [inline]

Anim the point [UI](#) Text from in the screen to out of the screen.

### 5.1.2.4 void AppAdvisory.LoopMania.AnimPointText.Start ( ) [inline],[private]

Some initializations.

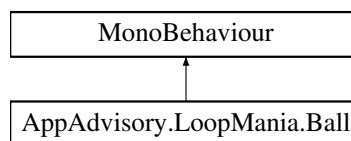
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/AnimPointText.cs

## 5.2 AppAdvisory.LoopMania.Ball Class Reference

Class attached to the Player GameObject in the hierarchy, or to the hazards. In charge to handle the Player controls and detect touch and collision, or to move the hazards around the circle.

Inheritance diagram for AppAdvisory.LoopMania.Ball:



### Public Member Functions

- float [GetRotation](#) ()  
*Return the current rotation of the [Ball](#), on the Z axis.*
- void [DOPosition](#) (float x, float y)  
*Apply a position to the [Ball](#).*
- void [DOStart](#) ()  
*Start the animation of the [Ball](#). The animation will not be the same if the [Ball](#) if the Player or an Hazard.*
- void [ActivateTouchControl](#) ()  
*Activate the touch detection if the [Ball](#) is a Player. Never call if the [Ball](#) is an Hazard.*
- void [DesactivateTouchControl](#) ()  
*Desactivate the touch detection (= unsubscribe to touch event) if the [Ball](#) is a Player. Never call if the [Ball](#) is an Hazard.*
- void [OnTouchDown](#) ([TouchDirection](#) td)  
*Method called when the player touch the screen.*
- void [DOOnTriggerEnter2D](#) (Collider2D other)  
*Method called when a trigger enter event happened in the collider (child of the [Ball](#)) If the [Ball](#) is an Hazard => Nothign happens! If the [Ball](#) is the player and triggered with an hazard:*

## Public Attributes

- bool [is\\_hazard](#) = false  
*True if it's an hazard (ie. spawned and can be destroy by the player, and can destroy the player). False if it's the player.*
- bool [is\\_jumping](#) = false  
*True if the ball is jumping.*
- Transform [playerSprite](#)  
*Reference to the parent of the SpriteRenderer.*
- SpriteRenderer [sr](#)  
*Reference to the SpriteRenderer.*

## Private Member Functions

- void [Awake](#) ()  
*Some initializations.*
- void [OnDisable](#) ()  
*When the [Ball](#) is disable, unsubscribe to the events subscribe before.*
- void [OnDestroy](#) ()  
*When the [Ball](#) is destroyed, unsubscribe to the events subscribe before.*
- void [OnPlayerJumpFinished](#) ()  
*Method called when the [Ball](#) player finished a Jump.*
- void [OnMainColorChanged](#) (Color c)  
*Method called when the main color is changed in the game. Please Refer to [GameManager OnMainColorChanged\(\)](#) method.*
- float [SpriteSize](#) ()  
*Return the sprite size in float (because it's a circle, so height = width).*

## Private Attributes

- [GameManager](#) [gameManager](#)  
*Reference to the [GameManager](#).*
- [SoundManager](#) [soundManager](#)  
*Reference to the [SoundManager](#).*

### 5.2.1 Detailed Description

Class attached to the Player GameObject in the hierarchy, or to the hazards. In charge to handle the Player controls and detect touch and collision, or to move the hazards around the circle.

### 5.2.2 Member Function Documentation

#### 5.2.2.1 void AppAdvisory.LoopMania.Ball.ActivateTouchControl ( ) [inline]

Activate the touch detection if the [Ball](#) is a Player. Never call if the [Ball](#) is an Hazard.

#### 5.2.2.2 void AppAdvisory.LoopMania.Ball.Awake ( ) [inline], [private]

Some initializations.

### 5.2.2.3 void AppAdvisory.LoopMania.Ball.DesactivateTouchControl ( ) [inline]

Desactivate the touch detection (= unsubscribe to touch event) if the [Ball](#) is a Player. Never call if the [Ball](#) is an Hazard.

### 5.2.2.4 void AppAdvisory.LoopMania.Ball.DOOnTriggerEnter2D ( Collider2D other ) [inline]

Method called when a trigger enter event happened in the collider (child of the [Ball](#)) If the [Ball](#) is an Hazard => Nothign happens! If the [Ball](#) is the player and triggered with an hazard:

- Game Over is the player is not jumping and touch an Hazard (who is a [Ball](#) too).
- Destroy an Hazard (who is a [Ball](#) too) if the player is jumping If the [Ball](#) is the player and triggered with a [DotToCollect](#):
- Collect the [DotToCollect](#) GameObject

### 5.2.2.5 void AppAdvisory.LoopMania.Ball.DOPosition ( float x, float y ) [inline]

Apply a position to the [Ball](#).

### 5.2.2.6 void AppAdvisory.LoopMania.Ball.DOStart ( ) [inline]

Start the animation of the [Ball](#). The animation will not be the same if the [Ball](#) if the Player or an Hazard.

### 5.2.2.7 float AppAdvisory.LoopMania.Ball.GetRotation ( ) [inline]

Return the current rotation of the [Ball](#), on the Z axis.

### 5.2.2.8 void AppAdvisory.LoopMania.Ball.OnDestroy ( ) [inline],[private]

When the [Ball](#) is destroyed, unsubscribe to the events subscribe before.

### 5.2.2.9 void AppAdvisory.LoopMania.Ball.OnDisable ( ) [inline],[private]

When the [Ball](#) is disable, unsubscribe to the events subscribe before.

### 5.2.2.10 void AppAdvisory.LoopMania.Ball.OnMainColorChanged ( Color c ) [inline],[private]

Method called when the main color is changed in the game. Please Refer to [GameManager OnMainColorChanged\(\)](#) method.

### 5.2.2.11 void AppAdvisory.LoopMania.Ball.OnPlayerJumpFinished ( ) [inline],[private]

Method called when the [Ball](#) player finished a Jump.

### 5.2.2.12 void AppAdvisory.LoopMania.Ball.OnTouchDown ( TouchDirection td ) [inline]

Method called when the player touch the screen.

5.2.2.13 `float AppAdvisory.LoopMania.Ball.SpriteSize ( ) [inline], [private]`

Return the sprite size in float (because it's a circle, so height = width).

### 5.2.3 Member Data Documentation

5.2.3.1 `GameManager AppAdvisory.LoopMania.Ball.gameManager [private]`

Reference to the [GameManager](#).

5.2.3.2 `bool AppAdvisory.LoopMania.Ball.is_hazard = false`

True if it's an hazard (ie. spawned and can be destroy by the player, and can destroy the player). False if it's the player.

5.2.3.3 `bool AppAdvisory.LoopMania.Ball.is_jumping = false`

True if the ball is jumping.

5.2.3.4 `Transform AppAdvisory.LoopMania.Ball.playerSprite`

Reference to the parent of the SpriteRenderer.

5.2.3.5 `SoundManager AppAdvisory.LoopMania.Ball.soundManager [private]`

Reference to the [SoundManager](#).

5.2.3.6 `SpriteRenderer AppAdvisory.LoopMania.Ball.sr`

Reference to the SpriteRenderer.

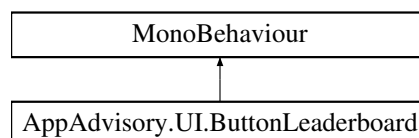
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/Ball.cs

## 5.3 AppAdvisory.UI.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 AppAdvisory.UI.ButtonLeaderboard:



## Public Member Functions

- void [OnClickedOpenLeaderboard](#) ()

*If player clicks 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>*

### 5.3.1 Detailed Description

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

### 5.3.2 Member Function Documentation

#### 5.3.2.1 void AppAdvisory.UI.ButtonLeaderboard.OnClickedOpenLeaderboard ( ) [inline]

If player clicks 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>

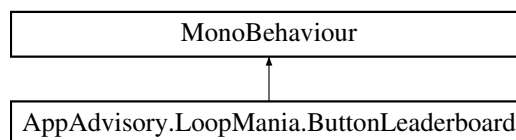
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/ButtonLeaderboard.cs

## 5.4 AppAdvisory.LoopMania.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 AppAdvisory.LoopMania.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** ()

### 5.4.1 Detailed Description

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

## 5.4.2 Member Function Documentation

### 5.4.2.1 void AppAdvisory.LoopingMania.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>

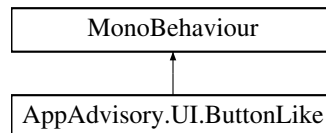
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/ButtonLeaderboard.cs

## 5.5 AppAdvisory.UI.ButtonLike Class Reference

Attached to like button

Inheritance diagram for AppAdvisory.UI.ButtonLike:



## Public Member Functions

- void [OnClickedFacebookLikeButton](#) ( )  
*If player clicks on the Facebook button, we call this method.*

## Public Attributes

- string [facebookApp](#) = "fb://profile/515431001924232"  
*URL use if the Facebook app is present in the mobile/tablet.*
- string [facebookAddress](#) = "https://www.facebook.com/appadvisory"  
*URL use if the Facebook app is not present or if we failed to call it quickly.*

### 5.5.1 Detailed Description

Attached to like button

## 5.5.2 Member Function Documentation

### 5.5.2.1 void AppAdvisory.UI.ButtonLike.OnClickedFacebookLikeButton ( ) [inline]

If player clicks on the Facebook button, we call this method.

## 5.5.3 Member Data Documentation

### 5.5.3.1 string AppAdvisory.UI.ButtonLike.facebookAddress = "https://www.facebook.com/appadvisory"

URL use if the Facebook app is not present or if we failed to call it quickly.

5.5.3.2 `string AppAdvisory.UI.ButtonLike.facebookApp = "fb://profile/515431001924232"`

URL use if the Facebook app is present in the mobile/tablet.

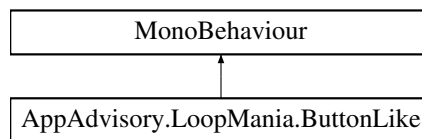
The documentation for this class was generated from the following file:

- `/Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/ButtonLike.cs`

## 5.6 AppAdvisory.LoopMania.ButtonLike Class Reference

Attached to like button

Inheritance diagram for AppAdvisory.LoopMania.ButtonLike:



### Public Member Functions

- void **OnClickedFacebookLikeButton** ()

### Public Attributes

- string **facebookApp** = "fb://profile/515431001924232"
- string **facebookAddress** = "https://www.facebook.com/appadvisory"

### 5.6.1 Detailed Description

Attached to like button

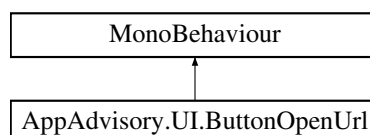
The documentation for this class was generated from the following file:

- `/Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/ButtonLike.cs`

## 5.7 AppAdvisory.UI.ButtonOpenUrl Class Reference

Class attached to button to open an url.

Inheritance diagram for AppAdvisory.UI.ButtonOpenUrl:



### Public Member Functions

- void **OnClickedOpenURL** ()



## Public Attributes

- string **URL** = "http://app-advisory.com"

## Private Member Functions

- void **Start** ()

### 5.7.1 Detailed Description

Class attached to button to open an url.

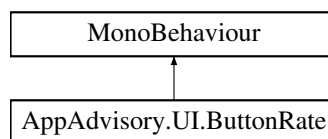
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/Button↔  
OpenUrl.cs

## 5.8 AppAdvisory.UI.ButtonRate Class Reference

Attached to rate button

Inheritance diagram for AppAdvisory.UI.ButtonRate:



## Public Member Functions

- void **OnClickedRate** ()  
*If player clicks on the rate button, we call this method.*

## Public Attributes

- bool **isAmazon** = false
- string **iosRateURL** = "fb://profile/515431001924232"  
*URL of the iOS game. Find it on iTunes Connect.*
- string **androidRateURL** = "https://www.facebook.com/appadvisory"  
*URL of the Android game. Find it on Google Play.*
- string **amazonRateURL** = "https://www.facebook.com/appadvisory"  
*URL of the Amazon game. Find it on the Amazon Developer Console.*

### 5.8.1 Detailed Description

Attached to rate button

## 5.8.2 Member Function Documentation

### 5.8.2.1 void AppAdvisory.UI.ButtonRate.OnClickedRate ( ) [inline]

If player clicks on the rate button, we call this method.

## 5.8.3 Member Data Documentation

### 5.8.3.1 string AppAdvisory.UI.ButtonRate.amazonRateURL = "https://www.facebook.com/appadvisory"

URL of the Amazon game. Find it on the Amazon Developer Console.

### 5.8.3.2 string AppAdvisory.UI.ButtonRate.androidRateURL = "https://www.facebook.com/appadvisory"

URL of the Android game. Find it on Google Play.

### 5.8.3.3 string AppAdvisory.UI.ButtonRate.iosRateURL = "fb://profile/515431001924232"

URL of the iOS game. Find it on iTunes Connect.

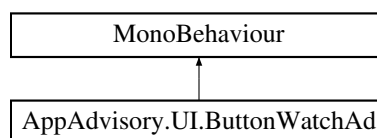
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/ButtonRate.cs

## 5.9 AppAdvisory.UI.ButtonWatchAd Class Reference

Class attached to the watch ad button. Works only on mobile (iOS & Android), with Very Simple Ads : <http://u3d.as/oWD>

Inheritance diagram for AppAdvisory.UI.ButtonWatchAd:



### Public Member Functions

- void [OnClickedWatchAd](#) ( )

*If player clicks on the watch ad button, we call this method. Works only on mobile (iOS & Android) if using Very Simple Ads by App Advisory : <http://u3d.as/oWD>*

### Public Attributes

- string **VerySimpleAdsURL** = "http://u3d.as/oWD"

### 5.9.1 Detailed Description

Class attached to the watch ad button. Works only on mobile (iOS & Android), with Very Simple Ads : <http://u3d.as/oWD>

## 5.9.2 Member Function Documentation

### 5.9.2.1 void AppAdvisory.UI.ButtonWatchAd.OnClickedWatchAd ( ) [inline]

If player clics on the watch ad button, we call this method. Works only on mobile (iOS & Android) if using Very Simple Ads by App Advisory : <http://u3d.as/oWD>

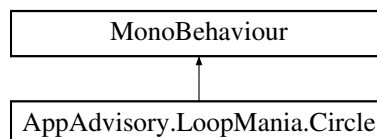
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/Button↔ WatchAd.cs

## 5.10 AppAdvisory.LoopMania.Circle Class Reference

Class in charge to draw the circle.

Inheritance diagram for AppAdvisory.LoopMania.Circle:



### Public Member Functions

- void [DOStart](#) (float [radius](#))  
*Draw the [Circle](#) and susbcribe to the [GameManager](#) event: OnMainColorChanged.*

### Public Attributes

- LineRenderer [line](#)  
*Reference to the LineRenderer.*
- float [real\\_width](#) = 0.1f  
*Width of the border of the [Circle](#).*

### Private Member Functions

- void [Awake](#) ()  
*Some initialization.*
- void [OnDisable](#) ()  
*Unsusbcribe to the [GameManager](#) event: OnMainColorChanged when the GameObject is disabled.*
- void [OnDestroy](#) ()  
*Unsusbcribe to the [GameManager](#) event: OnMainColorChanged when the GameObject is destroyed.*
- void [OnMainColorChanged](#) (Color c)  
*Called when the [GameManager](#) event: OnMainColorChanged is called.*

## Private Attributes

- [GameManager](#) `gameManager`  
*Reference to the [GameManager](#).*
- `int segments = 100`  
*Number of segment in the [LineRenderer](#) to draw the circle.*
- `float radius`  
*Radius of the [Circle](#).*

### 5.10.1 Detailed Description

Class in charge to draw the circle.

### 5.10.2 Member Function Documentation

#### 5.10.2.1 `void AppAdvisory.LoopMania.Circle.Awake ( ) [inline], [private]`

Some initialization.

#### 5.10.2.2 `void AppAdvisory.LoopMania.Circle.DOStart ( float radius ) [inline]`

Draw the [Circle](#) and susbcribe to the [GameManager](#) event: `OnMainColorChanged`.

#### 5.10.2.3 `void AppAdvisory.LoopMania.Circle.OnDestroy ( ) [inline], [private]`

Unsubscribe to the [GameManager](#) event: `OnMainColorChanged` when the `GameObject` is destroyed.

#### 5.10.2.4 `void AppAdvisory.LoopMania.Circle.OnDisable ( ) [inline], [private]`

Unsubcribe to the [GameManager](#) event: `OnMainColorChanged` when the `GameObject` is disabled.

#### 5.10.2.5 `void AppAdvisory.LoopMania.Circle.OnMainColorChanged ( Color c ) [inline], [private]`

Called when the [GameManager](#) event: `OnMainColorChanged` is called.

### 5.10.3 Member Data Documentation

#### 5.10.3.1 `GameManager AppAdvisory.LoopMania.Circle.gameManager [private]`

Reference to the [GameManager](#).

#### 5.10.3.2 `LineRenderer AppAdvisory.LoopMania.Circle.line`

Reference to the [LineRenderer](#).

#### 5.10.3.3 `float AppAdvisory.LoopMania.Circle.radius [private]`

Radius of the [Circle](#).

5.10.3.4 `float AppAdvisory.LoopMania.Circle.real_width = 0.1f`

Width of the border of the [Circle](#).

5.10.3.5 `int AppAdvisory.LoopMania.Circle.segments = 100` `[private]`

Number of segment in the LineRenderer to draw the circle.

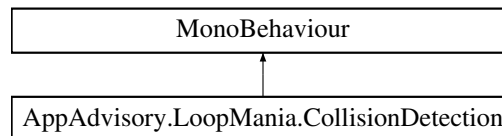
The documentation for this class was generated from the following file:

- `/Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/Circle.cs`

## 5.11 AppAdvisory.LoopMania.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 AppAdvisory.LoopMania.CollisionDetection:



### Public Member Functions

- `void OnTriggerEnter2D (Collider2D other)`  
*Listen the collision. If collision: all the Player method DOOnTriggerEnter2D*

### Public Attributes

- `Ball myPlayer`

#### 5.11.1 Detailed Description

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

#### 5.11.2 Member Function Documentation

5.11.2.1 `void AppAdvisory.LoopMania.CollisionDetection.OnTriggerEnter2D ( Collider2D other )` `[inline]`

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

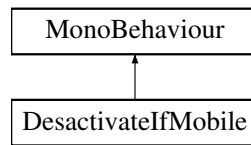
The documentation for this class was generated from the following file:

- `/Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/CollisionDetection.cs`

## 5.12 DesactivatelfMobile Class Reference

Attached to GameObject we want to deactivate at star if we run the game on mobile.

Inheritance diagram for DesactivateIfMobile:



### Private Member Functions

- void **OnEnable** ()

### 5.12.1 Detailed Description

Attached to GameObject we want to deactivate at star if we run the game on mobile.

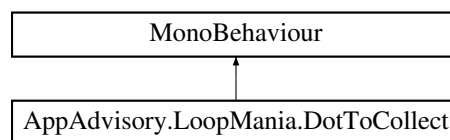
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/PROMO\_FACEBOOK/↔  
Scripts/DesactivateIfMobile.cs

## 5.13 AppAdvisory.LoopMania.DotToCollect Class Reference

Attached to ObstaclePrefab prefab in the prefab folder. Represent the little dot the player (Who is a [Ball](#)) have to collect around the [Circle](#).

Inheritance diagram for AppAdvisory.LoopMania.DotToCollect:



### Public Member Functions

- void **Init** (float rotation)
- void **DOCollect** ()

### Public Attributes

- bool **isItem** = true
- Collider2D **\_collider**
- Transform **obstacleSprite**
- SpriteRenderer **sr**
- Vector3 **defaultPosition** = Vector3.zero
- Vector3 **spritepos**

### Private Member Functions

- void **Awake** ()
- void **OnDotColorChanged** (Color c)
- float **SpriteSize** ()

## Private Attributes

- [GameManager](#) **gameManager**

### 5.13.1 Detailed Description

Attached to ObstaclePrefab prefab in the prefab folder. Represent the little dot the player (Who is a [Ball](#)) have to collect around the [Circle](#).

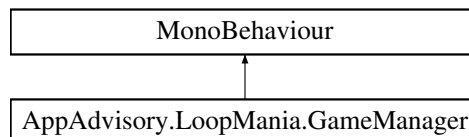
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/DotToCollect.cs

## 5.14 AppAdvisory.LoopMania.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 AppAdvisory.LoopMania.GameManager:



## Public Member Functions

- void [DOMainColorChange](#) ()  
*Change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.*
- IEnumerator [DoMainColorLerp](#) (Color from, Color to, float time)  
*Change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.*
- void [StartDotColorChange](#) ()  
*Change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.*
- delegate void [OnPlayerJumpFinish](#) ()  
*Event triggered when the [Ball](#) Player finishes a jump = touch the [Circle](#).*
- delegate void [OnMainColorChange](#) (Color c)  
*Event triggered when we change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.*
- void [DOParticle](#) (Vector3 position)  
*Spawn the particle explosion at a certain position.*
- void [DOStart](#) ()  
*Initialize the game. Set the point = 0. Initialize the radiusBorder. Set the [Ball](#) Player position. Start the different color change. Play start FX. Anim the MainCamera. At the end of the camera animation, enable the touch controll for the [Ball](#) Player (to jump) and start the [Ball](#) Player rotation. Spaxn the first [DotToCollect](#).*
- void [Add1Point](#) ()  
*Add 1 point, play point sound, update the [UI](#) Text point.*
- void [GameOver](#) (Transform t1)  
*Called at Game Over. Please have a look to the method DOOnTriggerEnter2D() in [Ball](#) : Game Over is triggered if, and only if: If the [Ball](#) is the player and triggered with an hazard and if the player is not jumping and touch an Hazard (who is a [Ball](#) too).*

- void [ShowAds](#) ()

If you want to monetize this game, get Very Simple Ads here: <http://u3d.as/oWD>.

## Static Public Member Functions

- static void [DOPlayerJumpFinished](#) ()

Call this method to called all the methods who subscribe to [OnPlayerJumpFinish](#).

## Public Attributes

- List< Color > [listColors](#)

Accessible in the editor, in the [GameObject \\_Managers](#) in the Hierarchy view. It's the list of colors use for the [Ball Hazards](#) and the [Circle](#).

- List< Color > [listDotColors](#)

Accessible in the editor, in the [GameObject \\_Managers](#) in the Hierarchy view. It's the list of colors use for the [Ball Player](#) and the [DotToCollect](#).

- Material [dotMaterial](#)

Reference to the dots Material, attached to each [DotToCollect](#) sprites. Usefull to change the color of each [DotToCollect](#) in one time.

- GameObject [particleExplosion](#)

Reference to the particle prefab.

- string [VerySimpleAdsURL](#) = "http://u3d.as/oWD"

If you want to monetize this game, get Very Simple Ads here: <http://u3d.as/oWD>.

- int [numberOfPlayToShowInterstitial](#) = 5

We will show an interstitial (need Very Simple Ads) each 5 Game Over. If you want to monetize this game, get Very Simple Ads here: <http://u3d.as/oWD>.

- int [numberOfDotsToCreate](#) = 15

Accessible in the editor, the number of [DotToCollect](#) we will create each turn.

- float [timeAnimItemInDotToCollect](#) = 0.3f

Accessible in the editor, the time in second to anim the [DotToCollect](#) at creation

- float [timeForCompleteCirclePlayer](#) = 2.5f

Accessible in the editor, the time for the [Ball Player](#) to do a complete turn.

- float [playerJumpSpeedInSeconds](#) = 0.12f

Accessible in the editor, the time for the [Ball Player](#) to do a complete jump.

- float [radiusBorder](#)

The radius of the circle.

- Text [pointText](#)

Reference to the [UI Text](#) who represents the point during the game.

- GameObject [dotToCollectPrefab](#)

Reference to the prefab [DotToCollect](#).

- GameObject [ballPrefab](#)

Reference to the prefab [Ball](#).

- [Ball player](#)

Reference to the [Ball Player](#).

## Properties

- Color [currentMainColor](#) [get, set]

The current main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.



## Events

- static [OnPlayerJumpFinish](#) [OnPlayerJumpFinished](#)  
Event triggered when the [Ball](#) Player finishes a jump = touch the [Circle](#).
- static [OnMainColorChange](#) [OnMainColorChanged](#)  
Event triggered when we change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.

## Private Member Functions

- void [Awake](#) ()  
Some initializations.
- void [Start](#) ()  
Some initializations. We will anim from far to close the MainCamera.
- void [InstantiateHazard](#) ()  
Instantiante a new [Ball](#) Hazard.
- void [DOInstantiateDotToCollect](#) ()  
Instantiante numberOfDotsToCreate = 15 by default [DotToCollect](#).
- void [ReportScoreToLeaderboard](#) (int p)  
If using Very Simple Leaderboard by App Advisory, report the score : <http://u3d.as/qxf>

## Private Attributes

- Color [m\\_currentMainColor](#)  
The current main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.
- float [cameraSize](#) = 5  
The orthographic camera size.
- bool [isGameOver](#) = false  
True if the game state is Game Over.
- int [point](#) = 0  
Current point in the game.
- [SoundManager](#) [soundManager](#)  
Reference to sound manager.
- [Circle](#) [circle](#) = null  
Reference to [Circle](#).

### 5.14.1 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>

### 5.14.2 Member Function Documentation

#### 5.14.2.1 void AppAdvisory.LoopMania.GameManager.Add1Point ( ) [inline]

Add 1 point, play point sound, update the [UI](#) Text point.

#### 5.14.2.2 void AppAdvisory.LoopMania.GameManager.Awake ( ) [inline],[private]

Some initializations.

**5.14.2.3** `void AppAdvisory.LoopMania.GameManager.DOInstantiateDotToCollect ( ) [inline],[private]`

Instantiante numberOfDotsToCreate = 15 by default [DotToCollect](#).

**5.14.2.4** `void AppAdvisory.LoopMania.GameManager.DOMainColorChange ( ) [inline]`

Change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.

**5.14.2.5** `IEnumerator AppAdvisory.LoopMania.GameManager.DoMainColorLerp ( Color from, Color to, float time ) [inline]`

Change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.

**5.14.2.6** `void AppAdvisory.LoopMania.GameManager.DOParticle ( Vector3 position ) [inline]`

Spawn the particle explosion at a certain position.

**5.14.2.7** `static void AppAdvisory.LoopMania.GameManager.DOPlayerJumpFinished ( ) [inline],[static]`

Call this method to called all the methods who subscribe to OnPlayerJumpFinish.

**5.14.2.8** `void AppAdvisory.LoopMania.GameManager.DOStart ( ) [inline]`

Initialize the game. Set the point = 0. Initialize the radiusBorder. Set the [Ball](#) Player position. Start the different color change. Play start FX. Anim the MainCamera. At the end of the camera animation, enable the touch controll for the [Ball](#) Player (to jump) and start the [Ball](#) Player rotation. Spaxn the first [DotToCollect](#).

**5.14.2.9** `void AppAdvisory.LoopMania.GameManager.GameOver ( Transform t1 ) [inline]`

Called at Game Over. Please have a look to the method DOOnTriggerEnter2D() in [Ball](#) : Game Over is triggered if, and only if: If the [Ball](#) is the player and triggered with an hazard and if the player is not jumping and touch an Hazard (who is a [Ball](#) too).

**5.14.2.10** `void AppAdvisory.LoopMania.GameManager.InstantiateHazard ( ) [inline],[private]`

Instantiante a new [Ball](#) Hazard.

**5.14.2.11** `delegate void AppAdvisory.LoopMania.GameManager.OnMainColorChange ( Color c )`

Event triggered when we change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.

**5.14.2.12** `delegate void AppAdvisory.LoopMania.GameManager.OnPlayerJumpFinish ( )`

Event triggered when the [Ball](#) Player finishes a jump = touch the [Circle](#).

**5.14.2.13** `void AppAdvisory.LoopMania.GameManager.ReportScoreToLeaderboard ( int p ) [inline],[private]`

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

5.14.2.14 `void AppAdvisory.LoopMania.GameManager.ShowAds ( ) [inline]`

If you want to monetize this game, get Very Simple Ads here: <http://u3d.as/oWD>.

5.14.2.15 `void AppAdvisory.LoopMania.GameManager.Start ( ) [inline],[private]`

Some initializations. We will anim from far to close the MainCamera.

5.14.2.16 `void AppAdvisory.LoopMania.GameManager.StartDotColorChange ( ) [inline]`

Change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.

### 5.14.3 Member Data Documentation

5.14.3.1 `GameObject AppAdvisory.LoopMania.GameManager.ballPrefab`

Reference to the prefab [Ball](#).

5.14.3.2 `float AppAdvisory.LoopMania.GameManager.cameraSize = 5 [private]`

The orthographic camera size.

5.14.3.3 `Circle AppAdvisory.LoopMania.GameManager.circle = null [private]`

Reference to [Circle](#).

5.14.3.4 `Material AppAdvisory.LoopMania.GameManager.dotMaterial`

Reference to the dots Material, attached to each [DotToCollect](#) sprites. Usefull to change the color of each [DotToCollect](#) in one time.

5.14.3.5 `GameObject AppAdvisory.LoopMania.GameManager.dotToCollectPrefab`

Reference to the prefab [DotToCollect](#).

5.14.3.6 `bool AppAdvisory.LoopMania.GameManager.isGameOver = false [private]`

True if the game state is Game Over.

5.14.3.7 `List<Color> AppAdvisory.LoopMania.GameManager.listColors`

Accessible in the editor, in the GameObject \_Managers in the Hierarchy view. It's the list of colors use for the [Ball](#) Hazards and the [Circle](#).

5.14.3.8 `List<Color> AppAdvisory.LoopMania.GameManager.listDotColors`

Accessible in the editor, in the GameObject \_Managers in the Hierarchy view. It's the list of colors use for the [Ball](#) Player and the [DotToCollect](#).

#### 5.14.3.9 Color AppAdvisory.LoopMania.GameManager.m\_currentMainColor [private]

The current main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.

#### 5.14.3.10 int AppAdvisory.LoopMania.GameManager.numberOfDotsToCreate = 15

Accessible in the editor, the number of [DotToCollect](#) we will create each turn.

#### 5.14.3.11 int AppAdvisory.LoopMania.GameManager.numberOfPlayToShowInterstitial = 5

We will show an interstitial (need Very Simple Ads) each 5 Game Over. If you want to monetize this game, get Very Simple Ads here: <http://u3d.as/oWD>.

#### 5.14.3.12 GameObject AppAdvisory.LoopMania.GameManager.particleExplosion

Reference to the particle prefab.

#### 5.14.3.13 Ball AppAdvisory.LoopMania.GameManager.player

Reference to the [Ball](#) Player.

#### 5.14.3.14 float AppAdvisory.LoopMania.GameManager.playerJumpSpeedInSeconds = 0.12f

Accessible in the editor, the time for the [Ball](#) Player to do a complete jump.

#### 5.14.3.15 int AppAdvisory.LoopMania.GameManager.point = 0 [private]

Current point in the game.

#### 5.14.3.16 Text AppAdvisory.LoopMania.GameManager.pointText

Reference to the [UI](#) Text who represents the point during the game.

#### 5.14.3.17 float AppAdvisory.LoopMania.GameManager.radiusBorder

The radius of the circle.

#### 5.14.3.18 SoundManager AppAdvisory.LoopMania.GameManager.soundManager [private]

Reference to sound manager.

#### 5.14.3.19 float AppAdvisory.LoopMania.GameManager.timeAnimItemInDotToCollect = 0.3f

Accessible in the editor, the time in second to anim the [DotToCollect](#) at creation

#### 5.14.3.20 float AppAdvisory.LoopMania.GameManager.timeForCompleteCirclePlayer = 2.5f

Accessible in the editor, the time for the [Ball](#) Player to do a complete turn.

5.14.3.21 string AppAdvisory.LoopMania.GameManager.VerySimpleAdsURL = "http://u3d.as/oWD"

If you want to monetize this game, get Very Simple Ads here: <http://u3d.as/oWD>.

## 5.14.4 Property Documentation

5.14.4.1 Color AppAdvisory.LoopMania.GameManager.currentMainColor [get], [set]

The current main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.

## 5.14.5 Event Documentation

5.14.5.1 OnMainColorChange AppAdvisory.LoopMania.GameManager.OnMainColorChanged [static]

Event triggered when we change the current main color main color = The color of the [Circle](#) + color of the [DotToCollect](#) + Particle explosion color.

5.14.5.2 OnPlayerJumpFinish AppAdvisory.LoopMania.GameManager.OnPlayerJumpFinished [static]

Event triggered when the [Ball](#) Player finishes a jump = touch the [Circle](#).

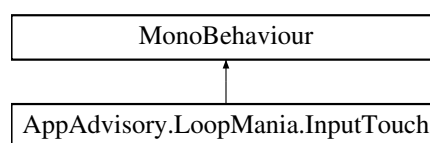
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/GameManager.cs

## 5.15 AppAdvisory.LoopMania.InputTouch Class Reference

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

Inheritance diagram for AppAdvisory.LoopMania.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*

### 5.15.1 Detailed Description

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

### 5.15.2 Member Function Documentation

#### 5.15.2.1 delegate void AppAdvisory.LoopMania.InputTouch.OnTouchDown ( TouchDirection td )

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

#### 5.15.2.2 delegate void AppAdvisory.LoopMania.InputTouch.OnTouchUp ( )

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

#### 5.15.2.3 void AppAdvisory.LoopMania.InputTouch.Update ( ) [inline],[private]

Listening for inputs

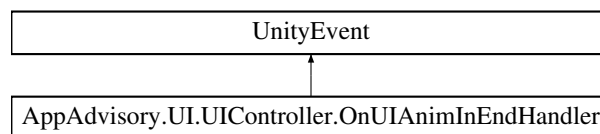
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/InputTouch.cs

## 5.16 AppAdvisory.UI.UIController.OnUIAnimInEndHandler Class Reference

Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is ended.

Inheritance diagram for AppAdvisory.UI.UIController.OnUIAnimInEndHandler:



### 5.16.1 Detailed Description

Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is ended.

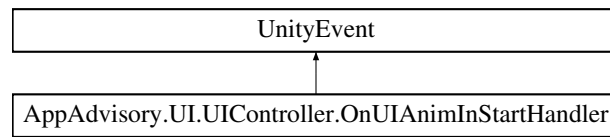
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/UIController.cs

## 5.17 AppAdvisory.UI.UIController.OnUIAnimInStartHandler Class Reference

Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is started.

Inheritance diagram for AppAdvisory.UI.UIController.OnUIAnimInStartHandler:



### 5.17.1 Detailed Description

Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is started.

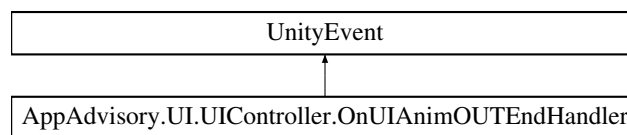
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/UIController.cs

## 5.18 AppAdvisory.UI.UIController.OnUIAnimOUTEndHandler Class Reference

Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is ended.

Inheritance diagram for AppAdvisory.UI.UIController.OnUIAnimOUTEndHandler:



### 5.18.1 Detailed Description

Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is ended.

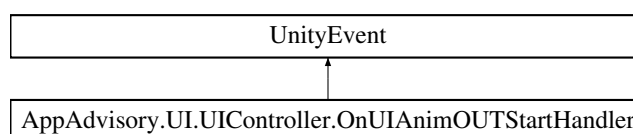
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/UIController.cs

## 5.19 AppAdvisory.UI.UIController.OnUIAnimOUTStartHandler Class Reference

Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is started.

Inheritance diagram for AppAdvisory.UI.UIController.OnUIAnimOUTStartHandler:



### 5.19.1 Detailed Description

Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is started.

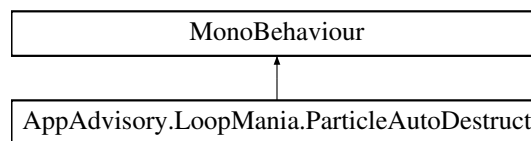
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/UIController.cs

## 5.20 AppAdvisory.LoopMania.ParticleAutoDestruct Class Reference

Attached to each particle. In charge to destroy particle when the particle doesn't emit anything.

Inheritance diagram for AppAdvisory.LoopMania.ParticleAutoDestruct:



### Public Attributes

- bool **OnlyDeactivate**

### Private Member Functions

- void **OnEnable** ()
- IEnumerator **CheckIfAlive** ()

### 5.20.1 Detailed Description

Attached to each particle. In charge to destroy particle when the particle doesn't emit anything.

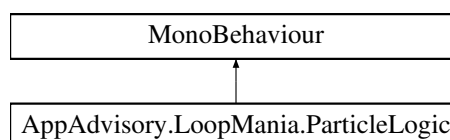
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/ParticleAutoDestruct.cs

## 5.21 AppAdvisory.LoopMania.ParticleLogic Class Reference

Class in charge to deactivate the particles.

Inheritance diagram for AppAdvisory.LoopMania.ParticleLogic:





### Private Member Functions

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

#### 5.21.1 Detailed Description

Class in charge to deactivate the particles.

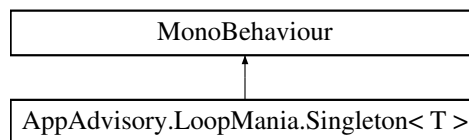
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/ParticleLogic.cs

## 5.22 AppAdvisory.LoopMania.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 AppAdvisory.LoopMania.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

#### 5.22.1 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***

## 5.22.2 Member Function Documentation

### 5.22.2.1 void AppAdvisory.LoopMania.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.

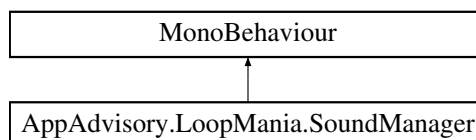
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/Singleton.cs

## 5.23 AppAdvisory.LoopMania.SoundManager Class Reference

Class in charge to play sound in the game.

Inheritance diagram for AppAdvisory.LoopMania.SoundManager:



### Public Member Functions

- void **PlayJump** ( )
- void **PlayPoint** ( )
- void **PlayMetal** ( )
- void **PlayHit** ( )
- void **PlayGameOverVoice** ( )
- void **PlayStartVoice** ( )
- void **PlayReloadVoice** ( )

### Properties

- AudioSource [audioSource](#) [get]  
*Reference to the AudioSource.*

### Private Attributes

- AudioSource [\\_audioSource](#)  
*Reference to the AudioSource.*
- AudioClip [soundJump](#)  
*Reference to the AudioClip played when the player jump.*
- AudioClip [soundHit](#)  
*Reference to the AudioClip played when the player hit the [Circle](#), ie. when the player ended his jump. (cf [Ball](#))*

- AudioClip [soundPoint](#)  
*Reference to the AudioClip played when the player get a point, ie. collect a [DotToCollect](#).*
- AudioClip [soundMetal](#)  
*Reference to the AudioClip played when the player hit the [Circle](#), ie. when the player ended his jump. (cf [Ball](#))*
- AudioClip [soundGameOverVoice](#)  
*Reference to the AudioClip played when the game is Game Over.*
- AudioClip [soundStartVoice](#)  
*Reference to the AudioClip played when the player start a new game.*
- AudioClip [soundReloadVoice](#)  
*Reference to the AudioClip when we spawned new [DotToCollect](#) elements.*

### 5.23.1 Detailed Description

Class in charge to play sound in the game.

### 5.23.2 Member Data Documentation

#### 5.23.2.1 AudioSource AppAdvisory.LoopMania.SoundManager.\_audioSource [private]

Reference to the AudioSource.

#### 5.23.2.2 AudioClip AppAdvisory.LoopMania.SoundManager.soundGameOverVoice [private]

Reference to the AudioClip played when the game is Game Over.

#### 5.23.2.3 AudioClip AppAdvisory.LoopMania.SoundManager.soundHit [private]

Reference to the AudioClip played when the player hit the [Circle](#), ie. when the player ended his jump. (cf [Ball](#))

#### 5.23.2.4 AudioClip AppAdvisory.LoopMania.SoundManager.soundJump [private]

Reference to the AudioClip played when the player jump.

#### 5.23.2.5 AudioClip AppAdvisory.LoopMania.SoundManager.soundMetal [private]

Reference to the AudioClip played when the player hit the [Circle](#), ie. when the player ended his jump. (cf [Ball](#))

#### 5.23.2.6 AudioClip AppAdvisory.LoopMania.SoundManager.soundPoint [private]

Reference to the AudioClip played when the player get a point, ie. collect a [DotToCollect](#).

#### 5.23.2.7 AudioClip AppAdvisory.LoopMania.SoundManager.soundReloadVoice [private]

Reference to the AudioClip when we spawned new [DotToCollect](#) elements.

#### 5.23.2.8 AudioClip AppAdvisory.LoopMania.SoundManager.soundStartVoice [private]

Reference to the AudioClip played when the player start a new game.

### 5.23.3 Property Documentation

#### 5.23.3.1 AudioSource AppAdvisory.LoopMania.SoundManager.audioSource [get],[private]

Reference to the AudioSource.

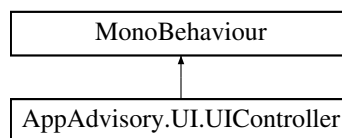
The documentation for this class was generated from the following file:

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/SoundManager.cs

## 5.24 AppAdvisory.UI.UIController Class Reference

Class attached to the [UIController](#) GameObject (who is a Canvas). In Charge of all the logics of the [UI](#): animation, events...

Inheritance diagram for AppAdvisory.UI.UIController:



### Classes

- class [OnUIAnimInEndHandler](#)  
*Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is ended.*
- class [OnUIAnimInStartHandler](#)  
*Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is started.*
- class [OnUIAnimOUTEndHandler](#)  
*Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is ended.*
- class [OnUIAnimOUTStartHandler](#)  
*Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is started.*

### Public Member Functions

- void [SetLastText](#) (int point)  
*Set the last score.*
- void [SetBestText](#) (int point)  
*Set the best score.*
- void [DOAnimIN](#) ()  
*Method called to do the animation IN, ie. from "out of the screen" to "in the screen". We will anim from top and horizontally.*
- void [DOAnimOUT](#) ()  
*Method called to do the animation OUT, ie. from "in the the screen" to "out of the screen". We will anim from top and horizontally.*

### Public Attributes

- LayoutGroup[] **layoutGroupToDesactivateAtStart**
- ContentSizeFitter[] **contentSizeFitterToDesactivateAtStart**
- LayoutElement[] **layoutElementToDesactivate**

- RectTransform[] [toAnimateFromTop](#)  
*Reference to all UI elements we will animate from the top of the screen.*
- RectTransform[] [toAnimateHorizontaly](#)  
*Reference to all UI elements we will animate horizontally.*
- Text [textLast](#)  
*Reference to UI Text for the last score.*
- Text [textBest](#)  
*Reference to UI Text for the best score.*
- [OnUIAnimInStartHandler OnUIAnimInStart](#)  
*Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is started.*
- [OnUIAnimInEndHandler OnUIAnimInEnd](#)  
*Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is ended.*
- [OnUIAnimOUTStartHandler OnUIAnimOutStart](#)  
*Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is started.*
- [OnUIAnimOUTEndHandler OnUIAnimOutEnd](#)  
*Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is ended.*

## Private Member Functions

- void **DesactivateUIFitter** ()
- void **Awake** ()
- void [AnimateINFromTop](#) (Action callback)  
*Do the animation IN, ie. from "out of the screen" to "in the screen", from top.*
- void [AnimateINHorizontaly](#) (Action callback)  
*Do the animation IN, ie. from "out of the screen" to "in the screen", horizontally.*
- void [AnimateOUTFromTop](#) (Action callback)  
*Do the animation OUT, ie. from "in the screen" to "out of the screen", from top.*
- void [AnimateOUTHorizontaly](#) (Action callback)  
*Do the animation OUT, ie. from "in the screen" to "out of the screen", horizontally.*

### 5.24.1 Detailed Description

Class attached to the [UIController](#) GameObject (who is a Canvas). In Charge of all the logics of the UI: animation, events...

### 5.24.2 Member Function Documentation

5.24.2.1 void AppAdvisory.UI.UIController.[AnimateINFromTop](#) ( Action *callback* ) [inline],[private]

Do the animation IN, ie. from "out of the screen" to "in the screen", from top.

5.24.2.2 void AppAdvisory.UI.UIController.[AnimateINHorizontaly](#) ( Action *callback* ) [inline],[private]

Do the animation IN, ie. from "out of the screen" to "in the screen", horizontally.

5.24.2.3 void AppAdvisory.UI.UIController.[AnimateOUTFromTop](#) ( Action *callback* ) [inline],[private]

Do the animation OUT, ie. from "in the screen" to "out of the screen", from top.

**5.24.2.4** `void AppAdvisory.UI.UIController.AnimateOUTHorizontal ( Action callback )` `[inline], [private]`

Do the animation OUT, ie. from "in the screen" to "out of the screen", horizontally.

**5.24.2.5** `void AppAdvisory.UI.UIController.DOAnimIN ( )` `[inline]`

Method called to do the animation IN, ie. from "out of the screen" to "in the screen". We will anim from top and horizontally.

**5.24.2.6** `void AppAdvisory.UI.UIController.DOAnimOUT ( )` `[inline]`

Method called to do the animation OUT, ie. from "in the the screen" to "out of the screen". We will anim from top and horizontally.

**5.24.2.7** `void AppAdvisory.UI.UIController.SetBestText ( int point )` `[inline]`

Set the best score.

**5.24.2.8** `void AppAdvisory.UI.UIController.SetLastText ( int point )` `[inline]`

Set the last score.

### 5.24.3 Member Data Documentation

**5.24.3.1** `OnUIAnimInEndHandler AppAdvisory.UI.UIController.OnUIAnimInEnd`

Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is ended.

**5.24.3.2** `OnUIAnimInStartHandler AppAdvisory.UI.UIController.OnUIAnimInStart`

Unity event triggered when the animation IN, ie. from "out of the screen" to "in the screen" is started.

**5.24.3.3** `OnUIAnimOUTEndHandler AppAdvisory.UI.UIController.OnUIAnimOutEnd`

Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is ended.

**5.24.3.4** `OnUIAnimOUTStartHandler AppAdvisory.UI.UIController.OnUIAnimOutStart`

Unity event triggered when the animation OUT, ie. from "in the the screen" to "out of screen" is started.

**5.24.3.5** `Text AppAdvisory.UI.UIController.textBest`

Reference to [UI Text](#) for the best score.

**5.24.3.6** `Text AppAdvisory.UI.UIController.textLast`

Reference to [UI Text](#) for the last score.

## 5.24.3.7 RectTransform [ ] AppAdvisory.UI.UIController.toAnimateFromTop

Reference to all [UI](#) elements we will animate from the top of the screen.

## 5.24.3.8 RectTransform [ ] AppAdvisory.UI.UIController.toAnimateHorizontally

Reference to all [UI](#) elements we will animate horizontally.

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

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Materials/UI/Scripts/UIScripts/UIController.cs

## 5.25 AppAdvisory.LoopMania.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 **Shuffle**< T > (this IList< T > list)  
*Real shuffle of List*
- static void **SetLastScore** (int score)  
*Set the last score, and eventually save a new best score in the PlayerPrefs.*
- static int **GetBestScore** ()  
*Get best score from PlayerPrefs.*
- static int **GetLastScore** ()  
*Get best score from PlayerPrefs.*
- static void **ReloadLevel** ()  
*Clean the memory and reload the scene*
- static void **CleanMemory** ()  
*Clean the memory*
- static void **SetAlpha** (this Text text, float alpha)  
*Set alpha of UI Text element.*
- static void **SetScaleX** (this RectTransform rect, float scale)  
*Set scale X of RectTransform element.*

### Static Private Attributes

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

### 5.25.1 Detailed Description

Utility class.

### 5.25.2 Member Function Documentation

#### 5.25.2.1 static void AppAdvisory.LoopMania.Util.CleanMemory ( ) [inline],[static]

Clean the memory

5.25.2.2 `static int AppAdvisory.LoopMania.Util.GetBestScore ( ) [inline],[static]`

Get best score from PlayerPrefs.

5.25.2.3 `static int AppAdvisory.LoopMania.Util.GetLastScore ( ) [inline],[static]`

Get best score from PlayerPrefs.

5.25.2.4 `static void AppAdvisory.LoopMania.Util.ReloadLevel ( ) [inline],[static]`

Clean the memory and reload the scene

5.25.2.5 `static void AppAdvisory.LoopMania.Util.SetAlpha ( this Text text, float alpha ) [inline],[static]`

Set alpha of [UI Text](#) element.

5.25.2.6 `static void AppAdvisory.LoopMania.Util.SetLastScore ( int score ) [inline],[static]`

Set the last score, and eventually save a new best score in the PlayerPrefs.

5.25.2.7 `static void AppAdvisory.LoopMania.Util.SetScaleX ( this RectTransform rect, float scale ) [inline],[static]`

Set scale X of RectTransform element.

5.25.2.8 `static void AppAdvisory.LoopMania.Util.Shuffle< T > ( this IList< T > list ) [inline],[static]`

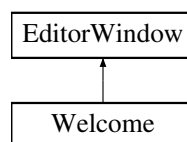
Real shuffle of List

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

- `/Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Scripts/Util.cs`

## 5.26 Welcome Class Reference

Inheritance diagram for Welcome:



### Public Member Functions

- void **OnGUI** ()

### Static Public Member Functions

- static void **OpenWelcomeWindow** ()



### Private Member Functions

- void **OnEnable** ()
- int **GetInt** (bool b)
- void **InitInterface** ()
- bool **Button** (Texture texture, string heading, string body, int space=10)

### Static Private Member Functions

- static void **OpenAtStartup** ()
- static void **OnPlayModeChanged** ()

### Private Attributes

- const string **ONLINE\_DOC\_URL** = "https://dl.dropboxusercontent.com/u/8289407/LoopingMania/Documentation/html/annotate.html"
- const string **RATEUS\_URL** = "http://u3d.as/tgs"
- const string **NAME\_OF\_THE\_GAME** = "Looping Mania"
- const string **VERYSIMPLEAD\_URL** = "http://u3d.as/oWD"
- const string **VERYSIMPLELEADERBOARD\_URL** = "http://u3d.as/qxf"
- const string **FACEBOOK\_URL** = "https://facebook.com/appadvisory"
- const string **REQUEST\_URL** = "https://appadvisory.zendesk.com/hc/en-us/requests/new"
- const string **APPADVISORY\_UNITY\_CATALOG\_URL** = "http://u3d.as/9cs"
- const string **COMMUNITY\_URL** = "https://appadvisory.zendesk.com/hc/en-us/community/topics"
- const float **width** = 500
- const float **height** = 710
- const string **PREFSHOWATSTARTUP** = "AppAdvisory.PREFSHOWATSTARTUP"

### Static Private Attributes

- static bool **showAtStartup**
- static GUIStyle **imgHeader**
- static bool **interfaceInitialized**
- static Texture **adslcon**
- static Texture **leaderboardIcon**
- static Texture **onlineDocIcon**
- static Texture **moreGamesIcon**
- static Texture **rateIcon**
- static Texture **communityIcon**
- static Texture **topicIcon**
- static Texture **questionIcon**
- static Texture **facebookIcon**

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

- /Users/anthonybarouch/Documents/LoopMania/Assets/LoopingMania/Editor/Welcome.cs



# Index

- [\\_audioSource](#)
      - [AppAdvisory::LoopMania::SoundManager, 35](#)
  - [ActivateTouchControl](#)
    - [AppAdvisory::LoopMania::Ball, 11](#)
  - [Add1Point](#)
    - [AppAdvisory::LoopMania::GameManager, 25](#)
  - [amazonRateURL](#)
    - [AppAdvisory::UI::ButtonRate, 18](#)
  - [androidRateURL](#)
    - [AppAdvisory::UI::ButtonRate, 18](#)
  - [AnimateINFromTop](#)
    - [AppAdvisory::UI::UIController, 37](#)
  - [AnimateINHorizontally](#)
    - [AppAdvisory::UI::UIController, 37](#)
  - [AnimateOUTFromTop](#)
    - [AppAdvisory::UI::UIController, 37](#)
  - [AnimateOUTHorizontally](#)
    - [AppAdvisory::UI::UIController, 37](#)
  - [AppAdvisory, 7](#)
  - [AppAdvisory.LoopMania, 7](#)
  - [AppAdvisory.LoopMania.AnimPointText, 9](#)
  - [AppAdvisory.LoopMania.Ball, 10](#)
  - [AppAdvisory.LoopMania.ButtonLeaderboard, 14](#)
  - [AppAdvisory.LoopMania.ButtonLike, 16](#)
  - [AppAdvisory.LoopMania.Circle, 19](#)
  - [AppAdvisory.LoopMania.CollisionDetection, 21](#)
  - [AppAdvisory.LoopMania.DotToCollect, 22](#)
  - [AppAdvisory.LoopMania.GameManager, 23](#)
  - [AppAdvisory.LoopMania.InputTouch, 29](#)
  - [AppAdvisory.LoopMania.ParticleAutoDestruct, 32](#)
  - [AppAdvisory.LoopMania.ParticleLogic, 32](#)
  - [AppAdvisory.LoopMania.Singleton< T >, 33](#)
  - [AppAdvisory.LoopMania.SoundManager, 34](#)
  - [AppAdvisory.LoopMania.Util, 39](#)
  - [AppAdvisory.UI, 8](#)
  - [AppAdvisory.UI.ButtonLeaderboard, 13](#)
  - [AppAdvisory.UI.ButtonLike, 15](#)
  - [AppAdvisory.UI.ButtonOpenUrl, 16](#)
  - [AppAdvisory.UI.ButtonRate, 17](#)
  - [AppAdvisory.UI.ButtonWatchAd, 18](#)
  - [AppAdvisory.UI.UIController, 36](#)
  - [AppAdvisory.UI.UIController.OnUIAnimInEndHandler, 30](#)
  - [AppAdvisory.UI.UIController.OnUIAnimInStartHandler, 30](#)
  - [AppAdvisory.UI.UIController.OnUIAnimOUTEnd↔Handler, 31](#)
  - [AppAdvisory.UI.UIController.OnUIAnimOUTStart↔Handler, 31](#)
- [AppAdvisory::LoopMania](#)
    - [TouchDirection, 8](#)
  - [AppAdvisory::LoopMania::AnimPointText](#)
    - [Awake, 9](#)
    - [DoAnimPointIn, 9](#)
    - [DoAnimPointOut, 10](#)
    - [Start, 10](#)
  - [AppAdvisory::LoopMania::Ball](#)
    - [ActivateTouchControl, 11](#)
    - [Awake, 11](#)
    - [DOOnTriggerEnter2D, 12](#)
    - [DOPosition, 12](#)
    - [DOStart, 12](#)
    - [DesactivateTouchControl, 11](#)
    - [gameManager, 13](#)
    - [GetRotation, 12](#)
    - [is\\_hazard, 13](#)
    - [is\\_jumping, 13](#)
    - [OnDestroy, 12](#)
    - [OnDisable, 12](#)
    - [OnMainColorChanged, 12](#)
    - [OnPlayerJumpFinished, 12](#)
    - [OnTouchDown, 12](#)
    - [playerSprite, 13](#)
    - [soundManager, 13](#)
    - [SpriteSize, 12](#)
    - [sr, 13](#)
  - [AppAdvisory::LoopMania::ButtonLeaderboard](#)
    - [OnClickedOpenLeaderboard, 15](#)
  - [AppAdvisory::LoopMania::Circle](#)
    - [Awake, 20](#)
    - [DOStart, 20](#)
    - [gameManager, 20](#)
    - [line, 20](#)
    - [OnDestroy, 20](#)
    - [OnDisable, 20](#)
    - [OnMainColorChanged, 20](#)
    - [radius, 20](#)
    - [real\\_width, 20](#)
    - [segments, 21](#)
  - [AppAdvisory::LoopMania::CollisionDetection](#)
    - [OnTriggerEnter2D, 21](#)
  - [AppAdvisory::LoopMania::GameManager](#)
    - [Add1Point, 25](#)
    - [Awake, 25](#)
    - [ballPrefab, 27](#)
    - [cameraSize, 27](#)
    - [circle, 27](#)
    - [currentMainColor, 29](#)

- DOInstantiateDotToCollect, 25
- DOMainColorChange, 26
- DOParticle, 26
- DOPlayerJumpFinished, 26
- DOStart, 26
- DoMainColorLerp, 26
- dotMaterial, 27
- dotToCollectPrefab, 27
- GameOver, 26
- InstantiateHazard, 26
- isGameOver, 27
- listColors, 27
- listDotColors, 27
- m\_currentMainColor, 27
- numberOfDotsToCreate, 28
- numberOfPlayToShowInterstitial, 28
- OnMainColorChange, 26
- OnMainColorChanged, 29
- OnPlayerJumpFinish, 26
- OnPlayerJumpFinished, 29
- particleExplosion, 28
- player, 28
- playerJumpSpeedInSeconds, 28
- point, 28
- pointText, 28
- radiusBorder, 28
- ReportScoreToLeaderboard, 26
- ShowAds, 26
- soundManager, 28
- Start, 27
- StartDotColorChange, 27
- timeAnimItemInDotToCollect, 28
- timeForCompleteCirclePlayer, 28
- VerySimpleAdsURL, 28
- AppAdvisory::LoopMania::InputTouch
  - OnTouchDown, 30
  - OnTouchUp, 30
  - Update, 30
- AppAdvisory::LoopMania::Singleton
  - OnDestroy, 34
- AppAdvisory::LoopMania::SoundManager
  - \_audioSource, 35
  - audioSource, 36
  - soundGameOverVoice, 35
  - soundHit, 35
  - soundJump, 35
  - soundMetal, 35
  - soundPoint, 35
  - soundReloadVoice, 35
  - soundStartVoice, 35
- AppAdvisory::LoopMania::Util
  - CleanMemory, 39
  - GetBestScore, 39
  - GetLastScore, 40
  - ReloadLevel, 40
  - SetAlpha, 40
  - SetLastScore, 40
  - SetScaleX, 40
  - Shuffle< T >, 40
- AppAdvisory::UI::ButtonLeaderboard
  - OnClickedOpenLeaderboard, 14
- AppAdvisory::UI::ButtonLike
  - facebookAddress, 15
  - facebookApp, 15
  - OnClickedFacebookLikeButton, 15
- AppAdvisory::UI::ButtonRate
  - amazonRateURL, 18
  - androidRateURL, 18
  - iosRateURL, 18
  - OnClickedRate, 18
- AppAdvisory::UI::ButtonWatchAd
  - OnClickedWatchAd, 19
- AppAdvisory::UI::UIController
  - AnimateINFromTop, 37
  - AnimateINHorizontally, 37
  - AnimateOUTFromTop, 37
  - AnimateOUTHorizontally, 37
  - DOAnimIN, 38
  - DOAnimOUT, 38
  - OnUIAnimInEnd, 38
  - OnUIAnimInStart, 38
  - OnUIAnimOutEnd, 38
  - OnUIAnimOutStart, 38
  - SetBestText, 38
  - SetLastText, 38
  - textBest, 38
  - textLast, 38
  - toAnimateFromTop, 38
  - toAnimateHorizontally, 39
- audioSource
  - AppAdvisory::LoopMania::SoundManager, 36
- Awake
  - AppAdvisory::LoopMania::AnimPointText, 9
  - AppAdvisory::LoopMania::Ball, 11
  - AppAdvisory::LoopMania::Circle, 20
  - AppAdvisory::LoopMania::GameManager, 25
- ballPrefab
  - AppAdvisory::LoopMania::GameManager, 27
- cameraSize
  - AppAdvisory::LoopMania::GameManager, 27
- circle
  - AppAdvisory::LoopMania::GameManager, 27
- CleanMemory
  - AppAdvisory::LoopMania::Util, 39
- currentMainColor
  - AppAdvisory::LoopMania::GameManager, 29
- DOAnimIN
  - AppAdvisory::UI::UIController, 38
- DOAnimOUT
  - AppAdvisory::UI::UIController, 38
- DOInstantiateDotToCollect
  - AppAdvisory::LoopMania::GameManager, 25
- DOMainColorChange
  - AppAdvisory::LoopMania::GameManager, 26

- DOOnTriggerEnter2D
  - AppAdvisory::LoopMania::Ball, [12](#)
- DOParticle
  - AppAdvisory::LoopMania::GameManager, [26](#)
- DOPlayerJumpFinished
  - AppAdvisory::LoopMania::GameManager, [26](#)
- DOPosition
  - AppAdvisory::LoopMania::Ball, [12](#)
- DOStart
  - AppAdvisory::LoopMania::Ball, [12](#)
  - AppAdvisory::LoopMania::Circle, [20](#)
  - AppAdvisory::LoopMania::GameManager, [26](#)
- DeactivateIfMobile, [21](#)
- DeactivateTouchControl
  - AppAdvisory::LoopMania::Ball, [11](#)
- DoAnimPointIn
  - AppAdvisory::LoopMania::AnimPointText, [9](#)
- DoAnimPointOut
  - AppAdvisory::LoopMania::AnimPointText, [10](#)
- DoMainColorLerp
  - AppAdvisory::LoopMania::GameManager, [26](#)
- dotMaterial
  - AppAdvisory::LoopMania::GameManager, [27](#)
- dotToCollectPrefab
  - AppAdvisory::LoopMania::GameManager, [27](#)
- facebookAddress
  - AppAdvisory::UI::ButtonLike, [15](#)
- facebookApp
  - AppAdvisory::UI::ButtonLike, [15](#)
- gameManager
  - AppAdvisory::LoopMania::Ball, [13](#)
  - AppAdvisory::LoopMania::Circle, [20](#)
- GameOver
  - AppAdvisory::LoopMania::GameManager, [26](#)
- GetBestScore
  - AppAdvisory::LoopMania::Util, [39](#)
- GetLastScore
  - AppAdvisory::LoopMania::Util, [40](#)
- GetRotation
  - AppAdvisory::LoopMania::Ball, [12](#)
- InstantiateHazard
  - AppAdvisory::LoopMania::GameManager, [26](#)
- iosRateURL
  - AppAdvisory::UI::ButtonRate, [18](#)
- is\_hazard
  - AppAdvisory::LoopMania::Ball, [13](#)
- is\_jumping
  - AppAdvisory::LoopMania::Ball, [13](#)
- isGameOver
  - AppAdvisory::LoopMania::GameManager, [27](#)
- line
  - AppAdvisory::LoopMania::Circle, [20](#)
- listColors
  - AppAdvisory::LoopMania::GameManager, [27](#)
- listDotColors
  - AppAdvisory::LoopMania::GameManager, [27](#)
- m\_currentMainColor
  - AppAdvisory::LoopMania::GameManager, [27](#)
- numberOfDotsToCreate
  - AppAdvisory::LoopMania::GameManager, [28](#)
- numberOfPlayToShowInterstitial
  - AppAdvisory::LoopMania::GameManager, [28](#)
- OnClickedFacebookLikeButton
  - AppAdvisory::UI::ButtonLike, [15](#)
- OnClickedOpenLeaderboard
  - AppAdvisory::LoopMania::ButtonLeaderboard, [15](#)
  - AppAdvisory::UI::ButtonLeaderboard, [14](#)
- OnClickedRate
  - AppAdvisory::UI::ButtonRate, [18](#)
- OnClickedWatchAd
  - AppAdvisory::UI::ButtonWatchAd, [19](#)
- OnDestroy
  - AppAdvisory::LoopMania::Ball, [12](#)
  - AppAdvisory::LoopMania::Circle, [20](#)
  - AppAdvisory::LoopMania::Singleton, [34](#)
- OnDisable
  - AppAdvisory::LoopMania::Ball, [12](#)
  - AppAdvisory::LoopMania::Circle, [20](#)
- OnMainColorChange
  - AppAdvisory::LoopMania::GameManager, [26](#)
- OnMainColorChanged
  - AppAdvisory::LoopMania::Ball, [12](#)
  - AppAdvisory::LoopMania::Circle, [20](#)
  - AppAdvisory::LoopMania::GameManager, [29](#)
- OnPlayerJumpFinish
  - AppAdvisory::LoopMania::GameManager, [26](#)
- OnPlayerJumpFinished
  - AppAdvisory::LoopMania::Ball, [12](#)
  - AppAdvisory::LoopMania::GameManager, [29](#)
- OnTouchDown
  - AppAdvisory::LoopMania::Ball, [12](#)
  - AppAdvisory::LoopMania::InputTouch, [30](#)
- OnTouchUp
  - AppAdvisory::LoopMania::InputTouch, [30](#)
- OnTriggerEnter2D
  - AppAdvisory::LoopMania::CollisionDetection, [21](#)
- OnUIAnimInEnd
  - AppAdvisory::UI::UIController, [38](#)
- OnUIAnimInStart
  - AppAdvisory::UI::UIController, [38](#)
- OnUIAnimOutEnd
  - AppAdvisory::UI::UIController, [38](#)
- OnUIAnimOutStart
  - AppAdvisory::UI::UIController, [38](#)
- particleExplosion
  - AppAdvisory::LoopMania::GameManager, [28](#)
- player
  - AppAdvisory::LoopMania::GameManager, [28](#)
- playerJumpSpeedInSeconds
  - AppAdvisory::LoopMania::GameManager, [28](#)

playerSprite  
     AppAdvisory::LoopMania::Ball, [13](#)  
 point  
     AppAdvisory::LoopMania::GameManager, [28](#)  
 pointText  
     AppAdvisory::LoopMania::GameManager, [28](#)  
 radius  
     AppAdvisory::LoopMania::Circle, [20](#)  
 radiusBorder  
     AppAdvisory::LoopMania::GameManager, [28](#)  
 real\_width  
     AppAdvisory::LoopMania::Circle, [20](#)  
 ReloadLevel  
     AppAdvisory::LoopMania::Util, [40](#)  
 ReportScoreToLeaderboard  
     AppAdvisory::LoopMania::GameManager, [26](#)  
 segments  
     AppAdvisory::LoopMania::Circle, [21](#)  
 SetAlpha  
     AppAdvisory::LoopMania::Util, [40](#)  
 SetBestText  
     AppAdvisory::UI::UIController, [38](#)  
 SetLastScore  
     AppAdvisory::LoopMania::Util, [40](#)  
 SetLastText  
     AppAdvisory::UI::UIController, [38](#)  
 SetScaleX  
     AppAdvisory::LoopMania::Util, [40](#)  
 ShowAds  
     AppAdvisory::LoopMania::GameManager, [26](#)  
 Shuffle< T >  
     AppAdvisory::LoopMania::Util, [40](#)  
 soundGameOverVoice  
     AppAdvisory::LoopMania::SoundManager, [35](#)  
 soundHit  
     AppAdvisory::LoopMania::SoundManager, [35](#)  
 soundJump  
     AppAdvisory::LoopMania::SoundManager, [35](#)  
 soundManager  
     AppAdvisory::LoopMania::Ball, [13](#)  
     AppAdvisory::LoopMania::GameManager, [28](#)  
 soundMetal  
     AppAdvisory::LoopMania::SoundManager, [35](#)  
 soundPoint  
     AppAdvisory::LoopMania::SoundManager, [35](#)  
 soundReloadVoice  
     AppAdvisory::LoopMania::SoundManager, [35](#)  
 soundStartVoice  
     AppAdvisory::LoopMania::SoundManager, [35](#)  
 SpriteSize  
     AppAdvisory::LoopMania::Ball, [12](#)  
 sr  
     AppAdvisory::LoopMania::Ball, [13](#)  
 Start  
     AppAdvisory::LoopMania::AnimPointText, [10](#)  
     AppAdvisory::LoopMania::GameManager, [27](#)  
 StartDotColorChange  
     AppAdvisory::LoopMania::GameManager, [27](#)  
 textBest  
     AppAdvisory::UI::UIController, [38](#)  
 textLast  
     AppAdvisory::UI::UIController, [38](#)  
 timeAnimItemInDotToCollect  
     AppAdvisory::LoopMania::GameManager, [28](#)  
 timeForCompleteCirclePlayer  
     AppAdvisory::LoopMania::GameManager, [28](#)  
 toAnimateFromTop  
     AppAdvisory::UI::UIController, [38](#)  
 toAnimateHorizontally  
     AppAdvisory::UI::UIController, [39](#)  
 TouchDirection  
     AppAdvisory::LoopMania, [8](#)  
 Update  
     AppAdvisory::LoopMania::InputTouch, [30](#)  
 VerySimpleAdsURL  
     AppAdvisory::LoopMania::GameManager, [28](#)  
 Welcome, [40](#)