

Proeve

Generated by Doxygen 1.8.11



# 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      | Events Namespace Reference . . . . .             | 7        |
| 4.2      | Menus Namespace Reference . . . . .              | 7        |
| 4.3      | Util Namespace Reference . . . . .               | 8        |
| <b>5</b> | <b>Class Documentation</b>                       | <b>9</b> |
| 5.1      | BallControler Class Reference . . . . .          | 9        |
| 5.1.1    | Member Function Documentation . . . . .          | 9        |
| 5.1.1.1  | OnTriggerEnter2D(Collider2D collision) . . . . . | 9        |
| 5.2      | Menus.BaseMenu Class Reference . . . . .         | 10       |
| 5.2.1    | Member Function Documentation . . . . .          | 10       |
| 5.2.1.1  | Close() . . . . .                                | 10       |
| 5.2.1.2  | Open() . . . . .                                 | 10       |
| 5.3      | CustomGrid Class Reference . . . . .             | 11       |
| 5.3.1    | Detailed Description . . . . .                   | 11       |
| 5.3.2    | Member Function Documentation . . . . .          | 12       |

|          |  |    |
|----------|--|----|
| 5.3.2.1  | ForceUpdate()                                  | 12 |
| 5.3.3    | Member Data Documentation                      | 12 |
| 5.3.3.1  | AnchorPoint                                    | 12 |
| 5.3.3.2  | continuesUpdates                               | 12 |
| 5.3.3.3  | CurrentSpacing                                 | 12 |
| 5.3.3.4  | maxRows  | 12 |
| 5.3.3.5  | maxSpacing                                     | 12 |
| 5.3.3.6  | minNeededForFirstRow                           | 12 |
| 5.3.3.7  | ObjSize  | 12 |
| 5.3.3.8  | padding  | 12 |
| 5.3.3.9  | useAnimations                                  | 13 |
| 5.3.3.10 | yoffset  | 13 |
| 5.4      | Events.EventDispatcher Class Reference         | 13 |
| 5.4.1    | Detailed Description                           | 13 |
| 5.4.2    | Member Function Documentation                  | 13 |
| 5.4.2.1  | AddEventListener(Type type, Action handler)    | 13 |
| 5.4.2.2  | AddEventListener< T >(Action< T > handler)     | 14 |
| 5.4.2.3  | Invoke(Type type, object evt)                  | 14 |
| 5.4.2.4  | Invoke< T >(T evt)                             | 14 |
| 5.4.2.5  | RemoveAllListeners()                           | 15 |
| 5.4.2.6  | RemoveEventListener(Type type, Action handler) | 15 |
| 5.4.2.7  | RemoveEventListener< T >(Action< T > handler)  | 15 |
| 5.5      | GameManager Class Reference                    | 16 |
| 5.5.1    | Member Function Documentation                  | 16 |
| 5.5.1.1  | ContinueGame()                                 | 16 |
| 5.5.1.2  | PauseGame()                                    | 16 |
| 5.5.2    | Property Documentation                         | 16 |
| 5.5.2.1  | GamePaused                                     | 16 |
| 5.5.2.2  | Score  | 17 |
| 5.6      | Util.GridLayoutHeightSetter Class Reference    | 17 |

|          |   |    |
|----------|---|----|
| 5.7      | Util.GridlayoutWidthSetter Class Reference . . . . .                  | 17 |
| 5.8      | Menus.HighScoreDisplayObject Class Reference . . . . .                | 18 |
| 5.8.1    | Detailed Description . . . . .  | 18 |
| 5.8.2    | Member Function Documentation . . . . .                               | 18 |
| 5.8.2.1  | SetValues(string Name, string Score, Color backgroundColor) . . . . . | 18 |
| 5.9      | Menus.HighScoreMenu Class Reference . . . . .                         | 19 |
| 5.10     | Menus.HighScoreSubmitScreen Class Reference . . . . .                 | 19 |
| 5.11     | Events.IEvent Interface Reference . . . . .                           | 20 |
| 5.11.1   | Detailed Description . . . . .  | 20 |
| 5.12     | Events.IEventDispatcher Interface Reference . . . . .                 | 20 |
| 5.12.1   | Detailed Description . . . . .  | 21 |
| 5.12.2   | Member Function Documentation . . . . .                               | 21 |
| 5.12.2.1 | AddEventListener(Type type, Action handler) . . . . .                 | 21 |
| 5.12.2.2 | AddEventListener< T >(Action< T > handler) . . . . .                  | 21 |
| 5.12.2.3 | Invoke(Type type, object evt) . . . . .                               | 21 |
| 5.12.2.4 | Invoke< T >(T evt) . . . . .  | 22 |
| 5.12.2.5 | RemoveEventListener(Type type, Action handler) . . . . .              | 22 |
| 5.12.2.6 | RemoveEventListener< T >(Action< T > handler) . . . . .               | 22 |
| 5.13     | InputManager Class Reference . . . . .                                | 23 |
| 5.13.1   | Event Documentation . . . . .   | 23 |
| 5.13.1.1 | onClick . . . . .   | 23 |
| 5.13.1.2 | onEscapePress . . . . .   | 23 |
| 5.14     | Events.IPause Class Reference . . . . .                               | 24 |
| 5.14.1   | Constructor & Destructor Documentation . . . . .                      | 24 |
| 5.14.1.1 | IPause(bool State) . . . . .  | 24 |
| 5.14.2   | Member Data Documentation . . . . .                                   | 24 |
| 5.14.2.1 | State . . . . .   | 24 |
| 5.15     | Events.IPlayerHitBottom Class Reference . . . . .                     | 25 |
| 5.15.1   | Detailed Description . . . . .  | 25 |
| 5.16     | Events.IResetGameState Class Reference . . . . .                      | 25 |

|   |    |
|---|----|
| 5.16.1 Detailed Description . . . . .                             | 25 |
| 5.17 Events.IScore Class Reference . . . . .                      | 25 |
| 5.17.1 Detailed Description . . . . .                             | 26 |
| 5.18 Events.LocalEvents Class Reference . . . . .                 | 26 |
| 5.18.1 Detailed Description . . . . .                             | 26 |
| 5.18.2 Member Function Documentation . . . . .                    | 26 |
| 5.18.2.1 AddEventListener(Type type, Action handler) . . . . .    | 26 |
| 5.18.2.2 AddEventListener< T >(Action< T > handler) . . . . .     | 27 |
| 5.18.2.3 Invoke(Type type, object evt) . . . . .                  | 27 |
| 5.18.2.4 Invoke< T >(T evt) . . . . .                             | 27 |
| 5.18.2.5 RemoveEventListener(Type type, Action handler) . . . . . | 28 |
| 5.18.2.6 RemoveEventListener< T >(Action< T > handler) . . . . .  | 28 |
| 5.19 Menus.MainMenu Class Reference . . . . .                     | 29 |
| 5.20 PauseMenu Class Reference . . . . .                          | 29 |
| 5.20.1 Detailed Description . . . . .                             | 29 |
| 5.20.2 Member Function Documentation . . . . .                    | 30 |
| 5.20.2.1 Close() . . . . .  | 30 |
| 5.21 SaveData Class Reference . . . . .                           | 30 |
| 5.21.1 Constructor & Destructor Documentation . . . . .           | 30 |
| 5.21.1.1 SaveData() . . . . .                                     | 30 |
| 5.21.2 Member Data Documentation . . . . .                        | 31 |
| 5.21.2.1 highScores . . . . .                                     | 31 |
| 5.21.2.2 SelectedBackground . . . . .                             | 31 |
| 5.21.2.3 SelectedCharacter . . . . .                              | 31 |
| 5.21.2.4 StorePoints . . . . .                                    | 31 |
| 5.21.2.5 UnlockedBackgrounds . . . . .                            | 31 |
| 5.21.2.6 UnlockedCharacters . . . . .                             | 31 |
| 5.22 ScaleToCameraView Class Reference . . . . .                  | 31 |
| 5.22.1 Detailed Description . . . . .                             | 32 |
| 5.23 Util.ScaleToScreenSize Class Reference . . . . .             | 32 |

|   |    |
|---|----|
| 5.23.1 Detailed Description . . . . .   | 32 |
| 5.24 Util.SceneUtils Class Reference . . . . .                                    | 32 |
| 5.24.1 Detailed Description . . . . .   | 33 |
| 5.25 SaveData.ScoreBlock Struct Reference . . . . .                               | 33 |
| 5.25.1 Detailed Description . . . . .   | 33 |
| 5.25.2 Constructor & Destructor Documentation . . . . .                           | 33 |
| 5.25.2.1 ScoreBlock(int score, string name) . . . . .                             | 33 |
| 5.26 Util.SerializableDictionary< TKey, TVal > Class Template Reference . . . . . | 34 |
| 5.27 Menus.ShopMenu Class Reference . . . . .                                     | 34 |
| 5.27.1 Member Function Documentation . . . . .                                    | 35 |
| 5.27.1.1 ClickBackground(int Index) . . . . .                                     | 35 |
| 5.27.1.2 ClickCharacter(int Index) . . . . .                                      | 35 |
| 5.27.1.3 Open() . . . . .   | 35 |
| 5.27.1.4 OpenBackgrounds() . . . . .  | 35 |
| 5.27.1.5 OpenCharacter() . . . . .  | 36 |
| 5.28 Menus.ShopMenuData Class Reference . . . . .                                 | 36 |
| 5.29 Util.SliderSetter Class Reference . . . . .                                  | 36 |
| 5.29.1 Detailed Description . . . . .   | 37 |
| 5.30 SnapToScreenPoint Class Reference . . . . .                                  | 37 |
| 5.30.1 Detailed Description . . . . .   | 37 |
| 5.30.2 Member Data Documentation . . . . .  | 37 |
| 5.30.2.1 screenPosition . . . . .   | 37 |
| 5.30.2.2 StartSize . . . . .  | 38 |
| 5.31 SplashScreen Class Reference . . . . .                                       | 38 |
| 5.32 Menus.ShopMenuData.StoreObject Struct Reference . . . . .                    | 38 |
| 5.33 TargetControler Class Reference . . . . .                                    | 39 |
| 5.34 UIManager Class Reference . . . . .  | 39 |
| 5.34.1 Detailed Description . . . . .   | 39 |
| 5.35 Util.ValueDebugger Class Reference . . . . .                                 | 40 |
| 5.35.1 Detailed Description . . . . .   | 40 |
| 5.35.2 Member Function Documentation . . . . .                                    | 40 |
| 5.35.2.1 ValueLog(string name, object value) . . . . .                            | 40 |
| 5.36 Util.ValueWrapper< T > Class Template Reference . . . . .                    | 41 |





# Chapter 1

## Namespace Index

### 1.1 Namespace List

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

|                        |   |
|------------------------|---|
| <a href="#">Events</a> | 7 |
| <a href="#">Menus</a>  | 7 |
| <a href="#">Util</a>   | 8 |



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

|   |    |
|---|----|
| Dictionary  |    |
| Util.SerializableDictionary< TKey, TVal > . . . . . | 34 |
| Events.IEvent . . . . .                             | 20 |
| Events.IPause . . . . .                             | 24 |
| Events.IPlayerHitBottom . . . . .                   | 25 |
| Events.IResetGameState . . . . .                    | 25 |
| Events.IScore . . . . .                             | 25 |
| Events.IEventDispatcher . . . . .                   | 20 |
| Events.EventDispatcher . . . . .                    | 13 |
| Events.LocalEvents . . . . .                        | 26 |
| ISerializable                                       |    |
| Util.SerializableDictionary< TKey, TVal > . . . . . | 34 |
| IXmlSerializable                                    |    |
| Util.SerializableDictionary< TKey, TVal > . . . . . | 34 |
| MonoBehaviour                                       |    |
| BallControler . . . . .                             | 9  |
| CustomGrid . . . . .                                | 11 |
| Events.LocalEvents . . . . .                        | 26 |
| GameManager . . . . .                               | 16 |
| InputManager . . . . .                              | 23 |
| Menus.BaseMenu . . . . .                            | 10 |
| Menus.HighScoreMenu . . . . .                       | 19 |
| Menus.ShopMenu . . . . .                            | 34 |
| Menus.HighScoreDisplayObject . . . . .              | 18 |
| Menus.HighScoreSubmitScreen . . . . .               | 19 |
| Menus.MainMenu . . . . .                            | 29 |
| Menus.ShopMenuData . . . . .                        | 36 |
| PauseMenu . . . . .                                 | 29 |
| ScaleToCameraView . . . . .                         | 31 |
| SnapToScreenPoint . . . . .                         | 37 |
| SplashScreen . . . . .                              | 38 |
| TargetControler . . . . .                           | 39 |
| UIManager . . . . .                                 | 39 |
| Util.GridLayoutHeightSetter . . . . .               | 17 |
| Util.GridlayoutWidthSetter . . . . .                | 17 |

|  |    |
|--|----|
| Util.ScaleToScreenSize . . . . .         | 32 |
| Util.SceneUtils . . . . .                | 32 |
| Util.SliderSetter . . . . .              | 36 |
| Util.ValueDebugger . . . . .             | 40 |
| SaveData . . . . .                       | 30 |
| SaveData.ScoreBlock . . . . .            | 33 |
| Menus.ShopMenuData.StoreObject . . . . . | 38 |
| Util.ValueWrapper< T > . . . . .         | 41 |

## Chapter 3

# Class Index

### 3.1 Class List

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

|   |    |
|---|----|
| <a href="#">BallControler</a>   | 9  |
| <a href="#">Menus.BaseMenu</a>  | 10 |
| <a href="#">CustomGrid</a>  |    |
| A grid sorting method that has the ability create simple soothend animations. It also only does things when it detects changes making it quite light; | 11 |
| <a href="#">Events.EventDispatcher</a>  |    |
| Event dispatcher class.   | 13 |
| <a href="#">GameManager</a>   | 16 |
| <a href="#">Util.GridLayoutHeightSetter</a>   | 17 |
| <a href="#">Util.GridlayoutWidthSetter</a>  | 17 |
| <a href="#">Menus.HighScoreDisplayObject</a>  |    |
| A display object for in the highscore screen  | 18 |
| <a href="#">Menus.HighScoreMenu</a>   | 19 |
| <a href="#">Menus.HighScoreSubmitScreen</a>   | 19 |
| <a href="#">Events.IEvent</a>   |    |
| The base interface for <a href="#">Events</a> .   | 20 |
| <a href="#">Events.IEventDispatcher</a>   |    |
| Event dispatcher interface.   | 20 |
| <a href="#">InputManager</a>  | 23 |
| <a href="#">Events.IPause</a>   | 24 |
| <a href="#">Events.IPlayerHitBottom</a>   |    |
| Empty event class that is called when the player hits the ground  | 25 |
| <a href="#">Events.IResetGameState</a>  |    |
| Called when the game requires a restart like when the player goes game over   | 25 |
| <a href="#">Events.IScore</a>   |    |
| Called when the player hits the target  | 25 |
| <a href="#">Events.LocalEvents</a>  |    |
| Use this for local event handling.  | 26 |
| <a href="#">Menus.MainMenu</a>  | 29 |
| <a href="#">PauseMenu</a>   |    |
| Pause Menu Opens when the player pauses the game  | 29 |
| <a href="#">SaveData</a>  | 30 |
| <a href="#">ScaleToCameraView</a>   |    |
| 3d version of Scale to ScreenSize   | 31 |
| <a href="#">Util.ScaleToScreenSize</a>  |    |
| Scales with screen dimensions   | 32 |

|  |    |
|--|----|
| <a href="#">Util.SceneUtils</a>  |    |
| Used in quick prototyping of buttons for the UI sytem  | 32 |
| <a href="#">SaveData.ScoreBlock</a>  |    |
| Struct that represents a single score in the highscore list                                      | 33 |
| <a href="#">Util.SerializableDictionary&lt; TKey, TVal &gt;</a>                                  | 34 |
| <a href="#">Menus.ShopMenu</a>   | 34 |
| <a href="#">Menus.ShopMenuData</a>   | 36 |
| <a href="#">Util.SliderSetter</a>  |    |
| A classes that is used to create a slider for the Scrollrect thatdoes not change size            | 36 |
| <a href="#">SnapToScreenPoint</a>  |    |
| Snaps a Object to a position on screen   | 37 |
| <a href="#">SplashScreen</a>   | 38 |
| <a href="#">Menus.ShopMenuData.StoreObject</a>   | 38 |
| <a href="#">TargetControler</a>  | 39 |
| <a href="#">UIManager</a>  |    |
| Controles the ingame UI  | 39 |
| <a href="#">Util.ValueDebugger</a>   |    |
| On screen debugger usefull when working with a game build but you want to do some error tracking | 40 |
| <a href="#">Util.ValueWrapper&lt; T &gt;</a>   | 41 |

## Chapter 4

# Namespace Documentation

### 4.1 Events Namespace Reference

#### Classes

- class [EventDispatcher](#)  
*Event dispatcher class.*
- class **GlobalEvents**  
*Use this for global event handling.*
- interface [IEvent](#)  
*The base interface for [Events](#).*
- interface [IEventDispatcher](#)  
*Event dispatcher interface.*
- class [IPause](#)
- class [IPlayerHitBottom](#)  
*Empty event class that is called when the player hits the ground*
- class [IResetGameState](#)  
*Called when the game requires a restart like when the player goes game over*
- class [IScore](#)  
*Called when the player hits the target*
- class [LocalEvents](#)  
*Use this for local event handling.*

### 4.2 Menus Namespace Reference

#### Classes

- class [BaseMenu](#)
- class [HighScoreDisplayObject](#)  
*A display object for in the highscore screen*
- class [HighScoreMenu](#)
- class [HighScoreSubmitScreen](#)
- class [MainMenu](#)
- class [ShopMenu](#)
- class [ShopMenuData](#)

## 4.3 Util Namespace Reference

### Classes

- class **Common**

*Common Utily library. It contains fuctions that I used regularly or where very hard to figure out.*

- class **Debugger**

*On screen debugger call class*

- class [GridLayoutHeightSetter](#)

- class [GridlayoutWidthSetter](#)

- class [ScaleToScreenSize](#)

*Scales with screen dimensions*

- class **SceneControler**

*Controlles scence changes this was write because Application.load has been marked as legacy*

- class [SceneUtils](#)

*Used in quick prototyping of buttons for the UI sytem*

- class [SerializableDictionary](#)

- class **Serialization**

*class that handels serialization and writing to disk*

- class [SliderSetter](#)

*A classes that is used to create a slider for the Scrollrect thatdoes not change size*

- class [ValueDebugger](#)

*On screen debugger usefull when working with a game build but you want to do some error tracking*

- class [ValueWrapper](#)

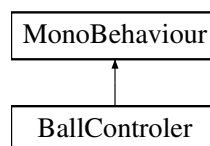


## Chapter 5

# Class Documentation

### 5.1 BallControler Class Reference

Inheritance diagram for BallControler:



#### Public Member Functions

- void **ChangeLooks** ()
- void **OnDestroy** ()
- void **OnTriggerEnter2D** (Collider2D collision)  
*Unity Function*
- void **OnCollisionEnter2D** (Collision2D collision)

#### 5.1.1 Member Function Documentation

5.1.1.1 void BallControler.OnTriggerEnter2D ( Collider2D *collision* ) [inline]

Unity Function

##### Parameters

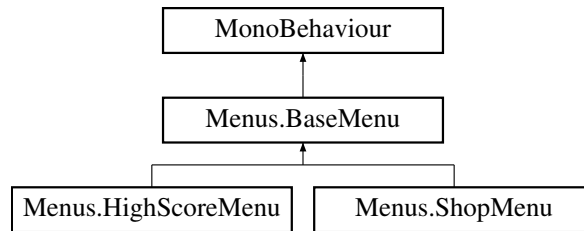
|                  |  |
|------------------|--|
| <i>collision</i> |  |
|------------------|--|

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

- Assets/Scripts/Controlers/BallControler.cs

## 5.2 Menus.BaseMenu Class Reference

Inheritance diagram for Menus.BaseMenu:



### Public Member Functions

- virtual void [Open](#) ()  
*Open menu by default it just enables and disables the game object*
- virtual void [Close](#) ()  
*Closes menu*

### Events

- VoidDelegate **onClose**

### 5.2.1 Member Function Documentation

#### 5.2.1.1 virtual void Menus.BaseMenu.Close ( ) [inline],[virtual]

Closes menu

by default it sends a event when the menu is closed the menu is closed by disabling the game object

#### 5.2.1.2 virtual void Menus.BaseMenu.Open ( ) [inline],[virtual]

Open menu by default it just enables and disables the game object

Reimplemented in [Menus.ShopMenu](#).

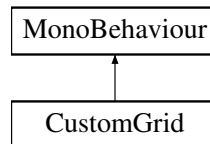
The documentation for this class was generated from the following file:

- Assets/Scripts/UI/BaseMenu.cs

## 5.3 CustomGrid Class Reference

A grid sorting method that has the ability to create simple smooth animations. It also only does things when it detects changes making it quite light;

Inheritance diagram for CustomGrid:



### Public Member Functions

- void `ForceUpdate ()`  
*Used for force call a update*

### Public Attributes

- Vector2 `ObjSize`  
*Size that each object will have*
- Vector2 `padding`  
*forced space between each object*
- Vector2 `maxSpacing`  
*the maximum space between two objects*
- Vector2 `AnchorPoint` = Vector2.zero  
*The anchor point or from which the object will be centered*
- Vector2 `CurrentSpacing` = Vector2.zero  
*The current space between the objects*
- int `maxRows` = 2  
*The maximum number of rows*
- int `minNeededForFirstRow` = 2  
*The minimum objects on a single row needed before it creates a second row*
- float `yoffset`  
*The y offset used for when the objects are not well centered themselves*
- bool `continuesUpdates` = false  
*Does the grid update continuously. useful when debugging the grid or seeing if everything works correctly. Recommended to have it turned off when you are building the game because it saves performance*
- bool `useAnimations` = false  
*Enable the animations so that the elements move towards the new points instead of teleporting*

### 5.3.1 Detailed Description

A grid sorting method that has the ability to create simple smooth animations. It also only does things when it detects changes making it quite light;

### 5.3.2 Member Function Documentation

#### 5.3.2.1 void CustomGrid.ForceUpdate ( ) [inline]

Used for force call a update

### 5.3.3 Member Data Documentation

#### 5.3.3.1 Vector2 CustomGrid.AnchorPoint = Vector2.zero

The anchor point or from wich the object wil be centered

#### 5.3.3.2 bool CustomGrid.continuesUpdates = false

Does the grid update continuesly. usefull when debugging the grid or seeing if everything works correctly. Recommended to have it turned of when you are building the game because it saves preformance

#### 5.3.3.3 Vector2 CustomGrid.CurrentSpacing = Vector2.zero

The current space between the objects

#### 5.3.3.4 int CustomGrid.maxRows = 2

The maxium number of rows

#### 5.3.3.5 Vector2 CustomGrid.maxSpacing

the maxium space between two objects

#### 5.3.3.6 int CustomGrid.minNeededForFirstRow = 2

The minum objects on a single row needed before it creates a second row

#### 5.3.3.7 Vector2 CustomGrid.ObjSize

Size that each object will have

#### 5.3.3.8 Vector2 CustomGrid.padding

forced space between each object

### 5.3.3.9 bool CustomGrid.useAnimations = false

Enable the animations so that the elements move towards the new points instead of teleporting

### 5.3.3.10 float CustomGrid.yoffset

The y offset used for when the objects are not well centered them selfs

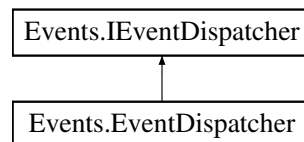
The documentation for this class was generated from the following file:

- Assets/Scripts/Util/CustomGrid.cs

## 5.4 Events.EventDispatcher Class Reference

Event dispatcher class.

Inheritance diagram for Events.EventDispatcher:



### Public Member Functions

- void [AddEventListener](#) (Type type, Action handler)  
*Implementation of [IEventDispatcher.AddEventListener\(Type, Action\)](#)*
- void [AddEventListener< T >](#) (Action< T > handler)  
*Implementation of [IEventDispatcher.AddEventListener< T >\(Action< T >\)](#)*
- void [RemoveEventListener](#) (Type type, Action handler)  
*Implementation of [IEventDispatcher.RemoveEventListener\(Type, Action\)](#)*
- void [RemoveEventListener< T >](#) (Action< T > handler)  
*Implementation of [IEventDispatcher.RemoveEventListener< T >\(Action< T >\)](#)*
- void [RemoveAllListeners](#) ()  
*Remove all registered Listeners.*
- void [Invoke](#) (Type type, object evt)  
*Implementation of [IEventDispatcher.Invoke\(Type, object\)](#)*
- void [Invoke< T >](#) (T evt)  
*Implementation of [IEventDispatcher.Invoke< T >\(T\)](#)*

### 5.4.1 Detailed Description

Event dispatcher class.

### 5.4.2 Member Function Documentation

#### 5.4.2.1 void Events.EventDispatcher.AddEventListener ( Type type, Action handler ) [inline]

Implementation of [IEventDispatcher.AddEventListener\(Type, Action\)](#)

## Parameters

|                |                                |
|----------------|--------------------------------|
| <i>type</i>    | The type of event to be added. |
| <i>handler</i> | The handler.                   |

Implements [Events.IEventDispatcher](#).

5.4.2.2 `void Events.EventDispatcher.AddEventListener<T>( Action<T> handler ) [inline]`

Implementation of [IEventDispatcher.AddEventListener<T>\(Action<T>\)](#)

## Template Parameters

|          |                                |
|----------|--------------------------------|
| <i>T</i> | The type of event to be added. |
|----------|--------------------------------|

## Parameters

|                |              |
|----------------|--------------|
| <i>handler</i> | The handler. |
|----------------|--------------|

Implements [Events.IEventDispatcher](#).

## Type Constraints

***T : IEvent***

5.4.2.3 `void Events.EventDispatcher.Invoke ( Type type, object evt ) [inline]`

Implementation of [IEventDispatcher.Invoke\(Type, object\)](#)

## Parameters

|             |                             |
|-------------|-----------------------------|
| <i>type</i> | Type of event to be evoked. |
| <i>evt</i>  | The event to be evoked.     |

Implements [Events.IEventDispatcher](#).

5.4.2.4 `void Events.EventDispatcher.Invoke<T>( T evt ) [inline]`

Implementation of [IEventDispatcher.Invoke<T>\(T\)](#)

## Template Parameters

|          |                             |
|----------|-----------------------------|
| <i>T</i> | Type of event to be evoked. |
|----------|-----------------------------|

## Parameters

|            |                         |
|------------|-------------------------|
| <i>evt</i> | The event to be evoked. |
|------------|-------------------------|

Implements [Events.IEventDispatcher](#).

## Type Constraints

***T : IEvent***

5.4.2.5 void [Events.EventDispatcher.RemoveAllListeners](#) ( ) [inline]

Remove all registered Listeners.

5.4.2.6 void [Events.EventDispatcher.RemoveEventListener](#) ( Type *type*, Action *handler* ) [inline]

Implementation of [IEventDispatcher.RemoveEventListener\(Type, Action\)](#)

## Parameters

|                |                                  |
|----------------|----------------------------------|
| <i>type</i>    | The type of event to be removed. |
| <i>handler</i> | The handler.                     |

Implements [Events.IEventDispatcher](#).

5.4.2.7 void [Events.EventDispatcher.RemoveEventListener<T>](#) ( Action<T> *handler* ) [inline]

Implementation of [IEventDispatcher.RemoveEventListener<T>\(Action<T>\)](#)

## Template Parameters

|          |                                  |
|----------|----------------------------------|
| <i>T</i> | The type of event to be removed. |
|----------|----------------------------------|

## Parameters

|                |              |
|----------------|--------------|
| <i>handler</i> | The handler. |
|----------------|--------------|

Implements [Events.IEventDispatcher](#).

## Type Constraints

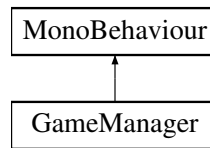
***T : IEvent***

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

- Assets/Scripts/Util/Events/EventDispatcher.cs

## 5.5 GameManager Class Reference

Inheritance diagram for GameManager:



### Public Member Functions

- void **OnDestroy** ()
- void **ContinueGame** ()  
*Call to unpause game*
- void **PauseGame** ()  
*Call to pause game*

### Properties

- static int **Score** [get]  
*The Points earned of the player*
- static bool **GamePaused** [get]  
*Show that game is paused*

### Events

- static VoidDelegate **OnScoreUpdate**

#### 5.5.1 Member Function Documentation

##### 5.5.1.1 void GameManager.ContinueGame ( ) [inline]

Call to unpause game

##### 5.5.1.2 void GameManager.PauseGame ( ) [inline]

Call to pause game

#### 5.5.2 Property Documentation

##### 5.5.2.1 bool GameManager.GamePaused [static],[get]

Show that game is paused



### 5.5.2.2 int GameManager.Score [static],[get]

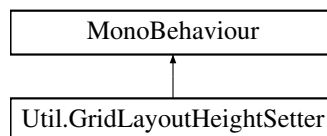
The Points earned of the player

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

- Assets/Scripts/Managers/GameManager.cs

## 5.6 Util.GridLayoutHeightSetter Class Reference

Inheritance diagram for Util.GridLayoutHeightSetter:



### Public Attributes

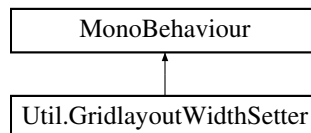
- bool **onlyUseActive**

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

- Assets/Scripts/Util/GridLayoutHeightSetter.cs

## 5.7 Util.GridlayoutWidthSetter Class Reference

Inheritance diagram for Util.GridlayoutWidthSetter:



### Public Member Functions

- void **ForceUpdate** ()

## Public Attributes

- int **ChildrenNeededToScroll** = 8
- bool **onlySetContainerHeight** = true
- bool **onlyUseActiveChildren** = true

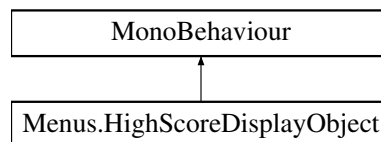
The documentation for this class was generated from the following file:

- Assets/Scripts/Util/GridlayoutWidthSetter.cs

## 5.8 Menus.HighScoreDisplayObject Class Reference

A display object for in the highscore screen

Inheritance diagram for Menus.HighScoreDisplayObject:



## Public Member Functions

- void **SetValues** (string Name, string Score, Color backgroundColor)  
*Sets the value that will be displayed*

## Properties

- UnityEngine.UI.Text **Name** [get]
- UnityEngine.UI.Text **Score** [get]

### 5.8.1 Detailed Description

A display object for in the highscore screen

### 5.8.2 Member Function Documentation

5.8.2.1 void Menus.HighScoreDisplayObject.SetValues ( string Name, string Score, Color backgroundColor ) [inline]

Sets the value that will be displayed

#### Parameters

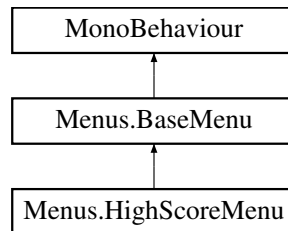
|                        |   |
|------------------------|---|
| <i>Name</i>            | Name of the player that submitted the score |
| <i>Score</i>           | Score the player got                        |
| <i>backgroundColor</i> | The color the background will be having     |

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

- Assets/Scripts/UI/HighScore/HighScoreDisplayObject.cs

## 5.9 Menus.HighScoreMenu Class Reference

Inheritance diagram for Menus.HighScoreMenu:



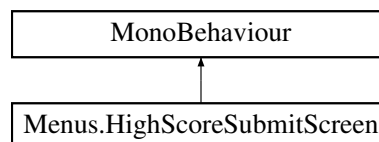
### Additional Inherited Members

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

- Assets/Scripts/UI/HighScore/HighScoreMenu.cs

## 5.10 Menus.HighScoreSubmitScreen Class Reference

Inheritance diagram for Menus.HighScoreSubmitScreen:



### Public Member Functions

- void **Open** ()
- void **Submit** ()
- void **Cancel** ()
- void **Close** ()

### Events

- VoidDelegate **onClose**

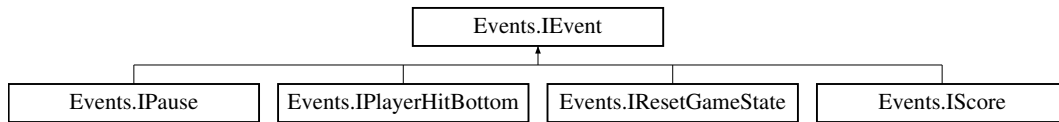
The documentation for this class was generated from the following file:

- Assets/Scripts/UI/HighScore/HighScoreSubmitScreen.cs

## 5.11 Events.IEvent Interface Reference

The base interface for [Events](#).

Inheritance diagram for Events.IEvent:



### 5.11.1 Detailed Description

The base interface for [Events](#).

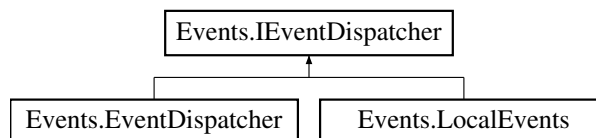
The documentation for this interface was generated from the following file:

- Assets/Scripts/Util/Events/IEvent.cs

## 5.12 Events.IEventDispatcher Interface Reference

Event dispatcher interface.

Inheritance diagram for Events.IEventDispatcher:



### Public Member Functions

- void [AddEventListener](#) (Type type, Action handler)  
*Add an event listener.*
- void [AddEventListener< T >](#) (Action< T > handler)  
*Add an event listener.*
- void [RemoveEventListener](#) (Type type, Action handler)  
*Remove an event listener.*
- void [RemoveEventListener< T >](#) (Action< T > handler)  
*Remove an event listener.*
- void [Invoke](#) (Type type, object evt)  
*Invoke an event.*
- void [Invoke< T >](#) (T evt)  
*Invoke an event.*

### 5.12.1 Detailed Description

Event dispatcher interface.

### 5.12.2 Member Function Documentation

#### 5.12.2.1 void Events.IEventDispatcher.AddEventListener ( Type *type*, Action *handler* )

Add an event listener.

Parameters

|                |                            |
|----------------|----------------------------|
| <i>type</i>    | The type of the event.     |
| <i>handler</i> | The handler for the event. |

Implemented in [Events.LocalEvents](#), and [Events.EventDispatcher](#).

#### 5.12.2.2 void Events.IEventDispatcher.AddEventListener< T > ( Action< T > *handler* )

Add an event listener.

Template Parameters

|          |                        |
|----------|------------------------|
| <i>T</i> | The type of the event. |
|----------|------------------------|

Parameters

|                |                            |
|----------------|----------------------------|
| <i>handler</i> | The handler for the event. |
|----------------|----------------------------|

Implemented in [Events.EventDispatcher](#), and [Events.LocalEvents](#).

Type Constraints

***T* : IEvent**

#### 5.12.2.3 void Events.IEventDispatcher.Invoke ( Type *type*, object *evt* )

Invoke an event.

Parameters

|             |                                  |
|-------------|----------------------------------|
| <i>type</i> | The type of event to be invoked. |
| <i>evt</i>  | The event to be invoked.         |

Implemented in [Events.EventDispatcher](#), and [Events.LocalEvents](#).

#### 5.12.2.4 void Events.IEventDispatcher.Invoke< T > ( T *evt* )

Invoke an event.

##### Template Parameters

|          |                                  |
|----------|----------------------------------|
| <i>T</i> | The type of event to be invoked. |
|----------|----------------------------------|

##### Parameters

|            |                          |
|------------|--------------------------|
| <i>evt</i> | The event to be invoked. |
|------------|--------------------------|

Implemented in [Events.EventDispatcher](#), and [Events.LocalEvents](#).

##### Type Constraints

***T : IEvent***

#### 5.12.2.5 void Events.IEventDispatcher.RemoveEventListener ( Type *type*, Action *handler* )

Remove an event listener.

##### Parameters

|                |                        |
|----------------|------------------------|
| <i>type</i>    | The type of the event. |
| <i>handler</i> | The handler.           |

Implemented in [Events.EventDispatcher](#), and [Events.LocalEvents](#).

#### 5.12.2.6 void Events.IEventDispatcher.RemoveEventListener< T > ( Action< T > *handler* )

Remove an event listener.

##### Template Parameters

|          |                        |
|----------|------------------------|
| <i>T</i> | The type of the event. |
|----------|------------------------|

##### Parameters

|                |              |
|----------------|--------------|
| <i>handler</i> | The handler. |
|----------------|--------------|

Implemented in [Events.EventDispatcher](#), and [Events.LocalEvents](#).

##### Type Constraints

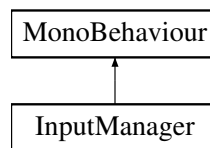
***T : IEvent***

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

- Assets/Scripts/Util/Events/IEventDispatcher.cs

## 5.13 InputManager Class Reference

Inheritance diagram for InputManager:



### Public Member Functions

- void **OnDestroy** ()

### Events

- static ClickDelegate [onClick](#)  
*Called when a click is registered on the screen*
- static VoidDelegate [onEscapePress](#)  
*Called when escape is pressed*

#### 5.13.1 Event Documentation

##### 5.13.1.1 ClickDelegate InputManager.onClick [static]

Called when a click is registered on the screen

##### 5.13.1.2 VoidDelegate InputManager.onEscapePress [static]

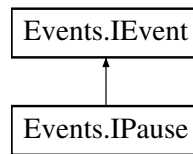
Called when escape is pressed

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

- Assets/Scripts/Managers/InputManager.cs

## 5.14 Events.IPause Class Reference

Inheritance diagram for Events.IPause:



### Public Member Functions

- [IPause](#) (bool [State](#))  
*Event Class that sends pause state changes to all listeners*

### Public Attributes

- bool [State](#)  
*Pause state of the game*

### 5.14.1 Constructor & Destructor Documentation

#### 5.14.1.1 Events.IPause.IPause ( bool [State](#) ) [[inline](#)]

Event Class that sends pause state changes to all listeners

#### Parameters

|                       |  |
|-----------------------|--|
| <a href="#">State</a> |  |
|-----------------------|--|

### 5.14.2 Member Data Documentation

#### 5.14.2.1 bool Events.IPause.State

Pause state of the game

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

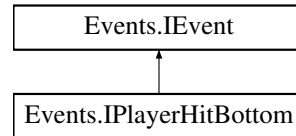
- Assets/Scripts/Events/IPause.cs



## 5.15 Events.IPlayerHitBottom Class Reference

Empty event class that is called when the player hits the ground

Inheritance diagram for Events.IPlayerHitBottom:



### 5.15.1 Detailed Description

Empty event class that is called when the player hits the ground

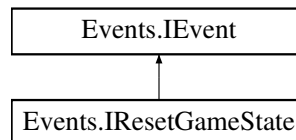
The documentation for this class was generated from the following file:

- Assets/Scripts/Events/IPlayerHitBottom.cs

## 5.16 Events.IResetGameState Class Reference

Called when the game requires a restart like when the player goes game over

Inheritance diagram for Events.IResetGameState:



### 5.16.1 Detailed Description

Called when the game requires a restart like when the player goes game over

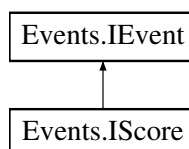
The documentation for this class was generated from the following file:

- Assets/Scripts/Events/IResetGameState.cs

## 5.17 Events.IScore Class Reference

Called when the player hits the target

Inheritance diagram for Events.IScore:



### 5.17.1 Detailed Description

Called when the player hits the target

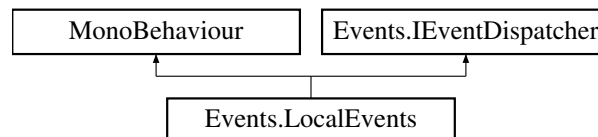
The documentation for this class was generated from the following file:

- Assets/Scripts/Events/IScore.cs

## 5.18 Events.LocalEvents Class Reference

Use this for local event handling.

Inheritance diagram for Events.LocalEvents:



### Public Member Functions

- void [AddEventListener](#) (Type type, Action handler)  
*Implementation of [IEventDispatcher.AddEventListener\(Type, Action\)](#).*
- void [AddEventListener< T >](#) (Action< T > handler)  
*Implementation of [IEventDispatcher.AddEventListener< T >\(Action< T >\)](#).*
- void [RemoveEventListener](#) (Type type, Action handler)  
*Implementation of [IEventDispatcher.RemoveEventListener\(Type, Action\)](#).*
- void [RemoveEventListener< T >](#) (Action< T > handler)  
*Implementation of [IEventDispatcher.RemoveEventListener< T >\(Action< T >\)](#).*
- void [Invoke](#) (Type type, object evt)  
*Implementation of [IEventDispatcher.Invoke\(Type, Action\)](#).*
- void [Invoke< T >](#) (T evt)  
*Implementation of [IEventDispatcher.Invoke< T >\(Action< T >\)](#).*

### 5.18.1 Detailed Description

Use this for local event handling.

### 5.18.2 Member Function Documentation

#### 5.18.2.1 void Events.LocalEvents.AddEventListener ( Type type, Action handler ) [inline]

Implementation of [IEventDispatcher.AddEventListener\(Type, Action\)](#).

## Parameters

|                |                    |
|----------------|--------------------|
| <i>type</i>    | The type of event. |
| <i>handler</i> | The handler.       |

Implements [Events.IEventDispatcher](#).

**5.18.2.2** `void Events.LocalEvents.AddEventListener< T > ( Action< T > handler )` `[inline]`

Implementation of [IEventDispatcher.AddEventListener<T>\(Action<T>\)](#).

## Template Parameters

|          |                    |
|----------|--------------------|
| <i>T</i> | The type of event. |
|----------|--------------------|

## Parameters

|                |              |
|----------------|--------------|
| <i>handler</i> | The handler. |
|----------------|--------------|

Implements [Events.IEventDispatcher](#).

## Type Constraints

***T : IEvent***

**5.18.2.3** `void Events.LocalEvents.Invoke ( Type type, object evt )` `[inline]`

Implementation of [IEventDispatcher.Invoke\(Type, Action\)](#).

## Parameters

|             |                    |
|-------------|--------------------|
| <i>type</i> | The type of event. |
| <i>evt</i>  | The event.         |

Implements [Events.IEventDispatcher](#).

**5.18.2.4** `void Events.LocalEvents.Invoke< T > ( T evt )` `[inline]`

Implementation of [IEventDispatcher.Invoke<T>\(Action<T>\)](#).

## Template Parameters

|          |                    |
|----------|--------------------|
| <i>T</i> | The type of event. |
|----------|--------------------|

## Parameters

|            |            |
|------------|------------|
| <i>evt</i> | The event. |
|------------|------------|

Implements [Events.IEventDispatcher](#).

## Type Constraints

***T : IEvent***

5.18.2.5 `void Events.LocalEvents.RemoveEventListener ( Type type, Action handler )` `[inline]`

Implementation of [IEventDispatcher.RemoveEventListener\(Type, Action\)](#).

## Parameters

|                |                    |
|----------------|--------------------|
| <i>type</i>    | The type of event. |
| <i>handler</i> | The handler.       |

Implements [Events.IEventDispatcher](#).

5.18.2.6 `void Events.LocalEvents.RemoveEventListener< T > ( Action< T > handler )` `[inline]`

Implementation of [IEventDispatcher.RemoveEventListener<T>\(Action<T>\)](#).

## Template Parameters

|          |                    |
|----------|--------------------|
| <i>T</i> | The type of event. |
|----------|--------------------|

## Parameters

|                |              |
|----------------|--------------|
| <i>handler</i> | The handler. |
|----------------|--------------|

Implements [Events.IEventDispatcher](#).

## Type Constraints

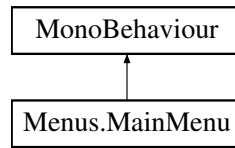
***T : IEvent***

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

- Assets/Scripts/Util/Events/LocalEvents.cs

## 5.19 Menus.MainMenu Class Reference

Inheritance diagram for Menus.MainMenu:



### Public Member Functions

- void **StartGame** ()
- void **OpenHighScore** ()
- void **OpenStore** ()
- void **Open** ()
- void **Close** ()

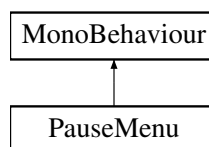
The documentation for this class was generated from the following file:

- Assets/Scripts/UI/MainMenu.cs

## 5.20 PauseMenu Class Reference

Pause Menu Opens when the player pauses the game

Inheritance diagram for PauseMenu:



### Public Member Functions

- void **Close** ()  
*Close the pause menu*
- void **OnDestroy** ()
- void **BackToMainMenu** ()

### 5.20.1 Detailed Description

Pause Menu Opens when the player pauses the game

## 5.20.2 Member Function Documentation

### 5.20.2.1 void PauseMenu.Close ( ) [inline]

Close the pause menu

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

- Assets/Scripts/UI/PauseMenu.cs

## 5.21 SaveData Class Reference

### Classes

- struct [ScoreBlock](#)  
*Struct that represents a single score in the highscore list*

### Public Member Functions

- [SaveData](#) ()  
*Used when creating a new blank savedata.*

### Public Attributes

- [ScoreBlock](#)[] [highScores](#)  
*Value that contains all the highscore players submitted should have a size of 10*
- bool[] [UnlockedCharacters](#) = new bool[0]  
*Bools that says if a Charater on that index is unlocked yes or no*
- bool[] [UnlockedBackgrounds](#) = new bool[0]  
*Bool that says if a Background on that index is unlocked yes or no*
- int [SelectedCharacter](#) = 0  
*Current selected character*
- int [StorePoints](#) = 0  
*The points the player can spent in the the store*
- int [SelectedBackground](#) = 0  
*That background that is currently selected*

## 5.21.1 Constructor & Destructor Documentation

### 5.21.1.1 SaveData.SaveData ( ) [inline]

Used when creating a new blank savedata.

## 5.21.2 Member Data Documentation

### 5.21.2.1 ScoreBlock [ ] SaveData.highScores

Value that contains all the highscore players submitted should have a size of 10

### 5.21.2.2 int SaveData.SelectedBackground = 0

That background that is currently selected

### 5.21.2.3 int SaveData.SelectedCharacter = 0

Current selected character

### 5.21.2.4 int SaveData.StorePoints = 0

The points the player can spent in the the store

### 5.21.2.5 bool [ ] SaveData.UnlockedBackgrounds = new bool[0]

Bool that says if a Background on that index is unlocked yes or no

### 5.21.2.6 bool [ ] SaveData.UnlockedCharacters = new bool[0]

Bools that says if a Charater on that index is unlocked yes or no

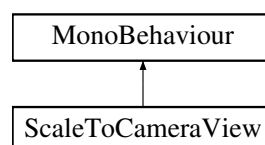
The documentation for this class was generated from the following file:

- Assets/Scripts/SaveData.cs

## 5.22 ScaleToCameraView Class Reference

3d version of Scale to ScreenSize

Inheritance diagram for ScaleToCameraView:



### 5.22.1 Detailed Description

3d version of Scale to ScreenSize

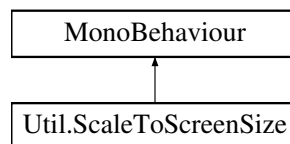
The documentation for this class was generated from the following file:

- Assets/Scripts/Util/ScaleToCameraView.cs

## 5.23 Util.ScaleToScreenSize Class Reference

Scales with screen dimensions

Inheritance diagram for Util.ScaleToScreenSize:



### Public Attributes

- Vector3 **screenSize** = Vector3.zero

### 5.23.1 Detailed Description

Scales with screen dimensions

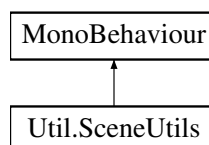
The documentation for this class was generated from the following file:

- Assets/Scripts/Util/ScaleToScreenSize.cs

## 5.24 Util.SceneUtils Class Reference

Used in quick prototyping of buttons for the UI sytem

Inheritance diagram for Util.SceneUtils:





## Public Member Functions

- void **OpenScene** (string name)
- void **CloseGame** ()

### 5.24.1 Detailed Description

Used in quick prototyping of buttons for the UI sytem

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

- Assets/Scripts/Util/SceneUtils.cs

## 5.25 SaveData.ScoreBlock Struct Reference

Struct that represents a single score in the highscore list

## Public Member Functions

- [ScoreBlock](#) (int score, string name)  
*Constructor for Scoreblock*

## Public Attributes

- int **score**
- string **name**

### 5.25.1 Detailed Description

Struct that represents a single score in the highscore list

### 5.25.2 Constructor & Destructor Documentation

#### 5.25.2.1 SaveData.ScoreBlock.ScoreBlock ( int *score*, string *name* ) `[inline]`

Constructor for Scoreblock

#### Parameters

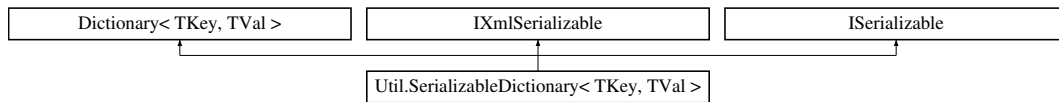
|              |  |
|--------------|--|
| <i>score</i> | Points scored that the player got durring there run                        |
| <i>name</i>  | Name the player gave up when submitting there score to the highscore table |

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

- Assets/Scripts/SaveData.cs

## 5.26 Util.SerializableDictionary< TKey, TVal > Class Template Reference

Inheritance diagram for Util.SerializableDictionary< TKey, TVal >:



### Public Member Functions

- **SerializableDictionary** (IDictionary< TKey, TVal > dictionary)
- **SerializableDictionary** (IEqualityComparer< TKey > comparer)
- **SerializableDictionary** (int capacity)
- **SerializableDictionary** (IDictionary< TKey, TVal > dictionary, IEqualityComparer< TKey > comparer)
- **SerializableDictionary** (int capacity, IEqualityComparer< TKey > comparer)

### Protected Member Functions

- **SerializableDictionary** (SerializationInfo info, StreamingContext context)

### Properties

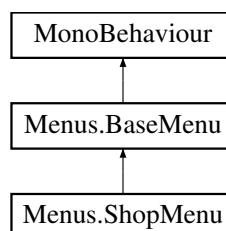
- XmlSerializer **ValueSerializer** [get]

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

- Assets/Scripts/Util/Saving/SerializableDictionary.cs

## 5.27 Menus.ShopMenu Class Reference

Inheritance diagram for Menus.ShopMenu:



## Public Member Functions

- override void [Open](#) ()  
*Open menu by default it just enables and disables the game object*
- void [OpenCharacter](#) ()  
*Opens character selection menu*
- void [OpenBackgrounds](#) ()  
*Opens background selection menu*
- void [ClickCharacter](#) (int Index)  
*Handels clicks on a character and checks if it has been unlocked and if it is then it will select that character Else*
- void [ClickBackground](#) (int Index)  
*Handels Clicks for the Backgrounds. Also checks if a background as been unlocked*

## Additional Inherited Members

### 5.27.1 Member Function Documentation

#### 5.27.1.1 void Menus.ShopMenu.ClickBackground ( int Index ) [inline]

Handels Clicks for the Backgrounds. Also checks if a background as been unlocked

##### Parameters

|              |                                       |
|--------------|---------------------------------------|
| <i>Index</i> | Index of the item that was clicked on |
|--------------|---------------------------------------|

#### 5.27.1.2 void Menus.ShopMenu.ClickCharacter ( int Index ) [inline]

Handels clicks on a character and checks if it has been unlocked and if it is then it will select that character Else

##### Parameters

|              |                                |
|--------------|--------------------------------|
| <i>Index</i> | Index of the item that clicked |
|--------------|--------------------------------|

#### 5.27.1.3 override void Menus.ShopMenu.Open ( ) [inline],[virtual]

Open menu by default it just enables and disables the game object

Reimplemented from [Menus.BaseMenu](#).

#### 5.27.1.4 void Menus.ShopMenu.OpenBackgrounds ( ) [inline]

Opens background selection menu

#### 5.27.1.5 void MenuShopMenu.OpenCharacter ( ) [inline]

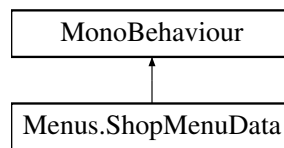
Opens character selection menu

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

- Assets/Scripts/UI/ShopMenu/ShopMenu.cs

## 5.28 MenuShopMenuData Class Reference

Inheritance diagram for MenuShopMenuData:



### Classes

- struct [StoreObject](#)

### Public Attributes

- const string **ResourceName** = "ShopData"

### Properties

- [StoreObject\[\]](#) **Characters** [get]
- [StoreObject\[\]](#) **Backgrounds** [get]

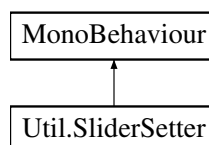
The documentation for this class was generated from the following file:

- Assets/Scripts/UI/ShopMenu/ShopMenuData.cs

## 5.29 Util.SliderSetter Class Reference

A classes that is used to create a slider for the Scrollrect thatdoes not change size

Inheritance diagram for Util.SliderSetter:



## Public Attributes

- Slider **slider**
- ScrollRect **scrollRect**
- float **StartValue**

### 5.29.1 Detailed Description

A classes that is used to create a slider for the Scrollrect thatdoes not change size

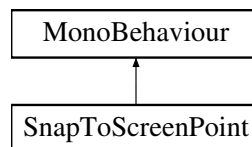
The documentation for this class was generated from the following file:

- Assets/Scripts/Util/SliderSetter.cs

## 5.30 SnapToScreenPoint Class Reference

Snaps a Object to a position on screen

Inheritance diagram for SnapToScreenPoint:



## Public Attributes

- Vector3 **screenPosition** = Vector3.zero  
*Position on screen Value should be between -100 and 100*
- Vector2 **StartSize** = Vector2.one  
*Sprite size in pixels to make sure it's does not get stretched*

### 5.30.1 Detailed Description

Snaps a Object to a position on screen

### 5.30.2 Member Data Documentation

#### 5.30.2.1 Vector3 SnapToScreenPoint.screenPosition = Vector3.zero

Position on screen Value should be between -100 and 100

#### 5.30.2.2 Vector2 SnapToScreenPoint.StartSize = Vector2.one

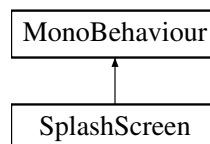
Sprite size in pixels to make sure it's does not get streched

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

- Assets/Scripts/Util/SnapToScreenPoint.cs

### 5.31 SplashScreen Class Reference

Inheritance diagram for SplashScreen:



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

- Assets/Scripts/SplashScreen.cs

### 5.32 Menus.ShopMenuData.StoreObject Struct Reference

#### Public Attributes

- int **Cost**
- string **Name**
- Sprite **highRes**
- Sprite **lowRes**

#### Properties

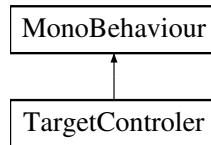
- Sprite **HighRes** [get]
- Sprite **LowRes** [get]

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

- Assets/Scripts/UI/ShopMenu/ShopMenuData.cs

## 5.33 TargetControler Class Reference

Inheritance diagram for TargetControler:



### Public Member Functions

- void **OnDestroy** ()

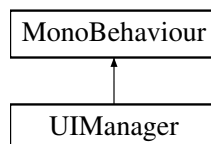
The documentation for this class was generated from the following file:

- Assets/Scripts/Controlers/TargetControler.cs

## 5.34 UIManager Class Reference

Controles the ingame UI

Inheritance diagram for UIManager:



### Public Member Functions

- void **OnDestroy** ()

### 5.34.1 Detailed Description

Controles the ingame UI

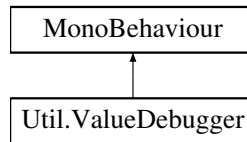
The documentation for this class was generated from the following file:

- Assets/Scripts/Managers/UIManager.cs

## 5.35 Util.ValueDebugger Class Reference

On screen debugger usefull when working with a game build but you want to do some error tracking

Inheritance diagram for Util.ValueDebugger:



### Static Public Member Functions

- static void [ValueLog](#) (string name, object value)  
*Value that will be logged. Also create the object that will be rendered onscreen completely by code so it does not need a prefab*

### Protected Attributes

- Dictionary< string, object > **Values**
- Text **t**

#### 5.35.1 Detailed Description

On screen debugger usefull when working with a game build but you want to do some error tracking

#### 5.35.2 Member Function Documentation

**5.35.2.1** static void Util.ValueDebugger.ValueLog ( string *name*, object *value* ) [inline],[static]

Value that will be logged. Also create the object that will be rendered onscreen completely by code so it does not need a prefab

##### Parameters

|              |                                      |
|--------------|--------------------------------------|
| <i>name</i>  | Value's name so you can find it back |
| <i>value</i> | Value of the object                  |

Make object if it does not exist

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

- Assets/Scripts/Util/ValueDebugger.cs



## 5.36 Util.ValueWrapper< T > Class Template Reference

### Properties

- **T value** [get, set]

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

- Assets/Scripts/Util/ValueWrapper.cs



# Index

- AddEventListener
  - Events::EventDispatcher, [13](#)
  - Events::IEventDispatcher, [21](#)
  - Events::LocalEvents, [26](#)
- AddEventListener< T >
  - Events::EventDispatcher, [14](#)
  - Events::IEventDispatcher, [21](#)
  - Events::LocalEvents, [27](#)
- AnchorPoint
  - CustomGrid, [12](#)
- BallControler, [9](#)
  - OnTriggerEnter2D, [9](#)
- ClickBackground
  - Menus::ShopMenu, [35](#)
- ClickCharacter
  - Menus::ShopMenu, [35](#)
- Close
  - Menus::BaseMenu, [10](#)
  - PauseMenu, [30](#)
- ContinueGame
  - GameManager, [16](#)
- continuesUpdates
  - CustomGrid, [12](#)
- CurrentSpacing
  - CustomGrid, [12](#)
- CustomGrid, [11](#)
  - AnchorPoint, [12](#)
  - continuesUpdates, [12](#)
  - CurrentSpacing, [12](#)
  - ForceUpdate, [12](#)
  - maxRows, [12](#)
  - maxSpacing, [12](#)
  - minNeededForFirstRow, [12](#)
  - ObjSize, [12](#)
  - padding, [12](#)
  - useAnimations, [12](#)
  - yoffset, [13](#)
- Events, [7](#)
- Events.EventDispatcher, [13](#)
- Events.IEvent, [20](#)
- Events.IEventDispatcher, [20](#)
- Events.IPause, [24](#)
- Events.IPlayerHitBottom, [25](#)
- Events.IResetGameState, [25](#)
- Events.IScore, [25](#)
- Events.LocalEvents, [26](#)
- Events::EventDispatcher
  - AddEventListener, [13](#)
  - AddEventListener< T >, [14](#)
  - Invoke, [14](#)
  - Invoke< T >, [14](#)
  - RemoveAllListeners, [15](#)
  - RemoveEventListener, [15](#)
  - RemoveEventListener< T >, [15](#)
- Events::IEventDispatcher
  - AddEventListener, [21](#)
  - AddEventListener< T >, [21](#)
  - Invoke, [21](#)
  - Invoke< T >, [21](#)
  - RemoveEventListener, [22](#)
  - RemoveEventListener< T >, [22](#)
- Events::IPause
  - IPause, [24](#)
  - State, [24](#)
- Events::LocalEvents
  - AddEventListener, [26](#)
  - AddEventListener< T >, [27](#)
  - Invoke, [27](#)
  - Invoke< T >, [27](#)
  - RemoveEventListener, [28](#)
  - RemoveEventListener< T >, [28](#)
- ForceUpdate
  - CustomGrid, [12](#)
- GameManager, [16](#)
  - ContinueGame, [16](#)
  - GamePaused, [16](#)
  - PauseGame, [16](#)
  - Score, [16](#)
- GamePaused
  - GameManager, [16](#)
- highScores
  - SaveData, [31](#)
- IPause
  - Events::IPause, [24](#)
- InputManager, [23](#)
  - onClick, [23](#)
  - onEscapePress, [23](#)
- Invoke
  - Events::EventDispatcher, [14](#)
  - Events::IEventDispatcher, [21](#)
  - Events::LocalEvents, [27](#)
- Invoke< T >
  - Events::EventDispatcher, [14](#)

- Events::IEventDispatcher, 21
- Events::LocalEvents, 27
- maxRows
  - CustomGrid, 12
- maxSpacing
  - CustomGrid, 12
- Menus, 7
- Menus.BaseMenu, 10
- Menus.HighScoreDisplayObject, 18
- Menus.HighScoreMenu, 19
- Menus.HighScoreSubmitScreen, 19
- Menus.MainMenu, 29
- Menus.ShopMenu, 34
- Menus.ShopMenuData, 36
- Menus.ShopMenuData.StoreObject, 38
- Menus::BaseMenu
  - Close, 10
  - Open, 10
- Menus::HighScoreDisplayObject
  - SetValues, 18
- Menus::ShopMenu
  - ClickBackground, 35
  - ClickCharacter, 35
  - Open, 35
  - OpenBackgrounds, 35
  - OpenCharacter, 35
- minNeededForFirstRow
  - CustomGrid, 12
- ObjSize
  - CustomGrid, 12
- onClick
  - InputManager, 23
- onEscapePress
  - InputManager, 23
- OnTriggerEnter2D
  - BallControler, 9
- Open
  - Menus::BaseMenu, 10
  - Menus::ShopMenu, 35
- OpenBackgrounds
  - Menus::ShopMenu, 35
- OpenCharacter
  - Menus::ShopMenu, 35
- padding
  - CustomGrid, 12
- PauseGame
  - GameManager, 16
- PauseMenu, 29
  - Close, 30
- RemoveAllListeners
  - Events::EventDispatcher, 15
- RemoveEventListener
  - Events::EventDispatcher, 15
  - Events::IEventDispatcher, 22
  - Events::LocalEvents, 28
- RemoveEventListener< T >
  - Events::EventDispatcher, 15
  - Events::IEventDispatcher, 22
  - Events::LocalEvents, 28
- SaveData, 30
  - highScores, 31
  - SaveData, 30
  - SelectedBackground, 31
  - SelectedCharacter, 31
  - StorePoints, 31
  - UnlockedBackgrounds, 31
  - UnlockedCharacters, 31
- SaveData.ScoreBlock, 33
- SaveData::ScoreBlock
  - ScoreBlock, 33
- ScaleToCameraView, 31
- Score
  - GameManager, 16
- ScoreBlock
  - SaveData::ScoreBlock, 33
- screenPosition
  - SnapToScreenPoint, 37
- SelectedBackground
  - SaveData, 31
- SelectedCharacter
  - SaveData, 31
- SetValues
  - Menus::HighScoreDisplayObject, 18
- SnapToScreenPoint, 37
  - screenPosition, 37
  - StartSize, 37
- SplashScreen, 38
- StartSize
  - SnapToScreenPoint, 37
- State
  - Events::IPause, 24
- StorePoints
  - SaveData, 31
- TargetControler, 39
- UIManager, 39
- UnlockedBackgrounds
  - SaveData, 31
- UnlockedCharacters
  - SaveData, 31
- useAnimations
  - CustomGrid, 12
- Util, 8
- Util.GridLayoutHeightSetter, 17
- Util.GridlayoutWidthSetter, 17
- Util.ScaleToScreenSize, 32
- Util.SceneUtils, 32
- Util.SerializableDictionary< TKey, TVal >, 34
- Util.SliderSetter, 36
- Util.ValueDebugger, 40
- Util.ValueWrapper< T >, 41
- Util::ValueDebugger

ValueLog, [40](#)

ValueLog

Util::ValueDebugger, [40](#)

yoffset

CustomGrid, [13](#)