



Leaderboard Manager Offline Documentation

[Introduction](#)

[Unsupported platforms](#)

[Getting Started](#)

[Importing the package](#)

[Setup](#)

[TextMeshPro Support](#)

[Example Scene](#)

[Inspector Reference](#)

[Leaderboard Display](#)

[Leaderboard Row Prefab](#)

[Scripting Reference](#)

[Namespaces](#)

[Scripts](#)

[LeaderboardManager.cs](#)

[CreateLeaderboard](#)

[DeleteLeaderboard](#)

[ClearLeaderboard](#)

[BoardExists](#)

[GetLeaderboard](#)

[AddEntryToBoard](#)

[DeleteEntryFromBoard](#)

[Save](#)

[Load](#)

[LeaderboardDataStore.cs](#)

[Constructors](#)

[Leaderboards](#)

[LeaderboardData.cs](#)

[Constructors](#)

[BoardID](#)

[BoardData](#)

[AddEntry](#)

[DeleteEntry](#)

[ClearBoard](#)

[GetEntry](#)

[GetTop3Ascending](#)

[GetTopXAscending](#)

[GetAllAscending](#)

[GetPositionAscending](#)

[GetTop3Descending](#)

[GetTopXDescending](#)

[GetAllDescending](#)

[GetPositionDescending](#)

[LeaderboardEntry.cs](#)

[Constructors](#)

[EntryID](#)

[Name](#)

[Score](#)

[ConvertScoreToTime](#)

[LeaderboardDisplay.cs](#)

[BoardID](#)

[DisplayOption](#)

[EntriesToShow](#)

[UpdateDisplay](#)

[ClearDisplay](#)

[DisplayOption.cs](#)

[Unassigned](#)

[AsWritten](#)

[Ascending](#)

[Descending](#)

[Top3Ascending](#)

[Top3Descending](#)

[TopXAscending](#)

[TopXDescending](#)

[DisplayTimeFormat.cs](#)

[Unassigned](#)

[MillisecondsOnly](#)
[SecondsOnly](#)
[SecondsMilliseconds](#)
[MinutesOnly](#)
[MinutesSeconds](#)
[MinutesSecondsMilliseconds](#)
[HoursOnly](#)
[HoursMinutes](#)
[HoursMinutesSeconds](#)
[HoursMinutesSecondsMilliseconds](#)

[LeaderboardDisplayTMP.cs](#)

[ExampleManager.cs](#)

[AddToBoard](#)

[RemoveFromBoard](#)

[ClearBoard](#)

F.A.Q

[Can I use leaderboard from version 1.x.x?](#)

[Can I use Leaderboard Data to add entries instead of Leaderboard Manager?](#)

[The example scene shows no entries, is there something wrong?](#)

Support

[Email](#)

[Discord](#)

[Report Form](#)

[Contact Form](#)

Introduction



Firstly, thank you for deciding to use our asset for your project. If you like our asset, feel free to leave us a review! If you find that our asset is not up to scratch or find an issue please do let us know either via our email: support@carter.games and we will do our best to help you with the issues you are facing.

The Leaderboard Manager is an asset designed to help implement simple local leaderboards into your game projects. The system uses a binary file to save all data for the board to the user's local machine. These can be accessed through the manager class to enable you to add/remove entries, make new boards and get the data in a format you can display on your game's UI.





This documentation is for version 2.0.0 or higher

Unsupported platforms


Sadly, due to technical reasons, the asset doesn't support WebGL games. As saving to a local file requires cloud saving to function, which we don't support in the scope of this asset.

Getting Started


Importing the package




Leaderboard Manager | CG



 Carter Games (not enough ratings) | ❤️ (20)

FREE


 **46 views** in the past week

[Add to My Assets](#)







Leaderboard Manager | CG

 Carter Games (not enough ratings) | ❤️ (20)

FREE

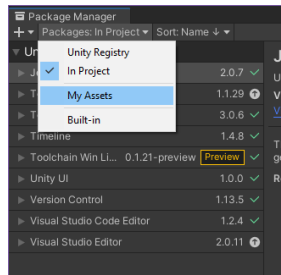
 **46 views** in the past week

[Open in Unity](#)

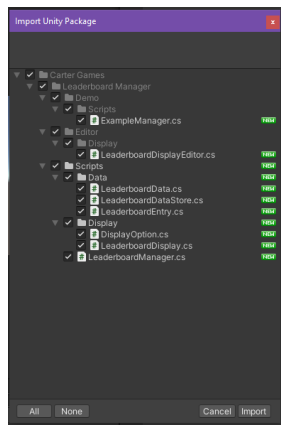
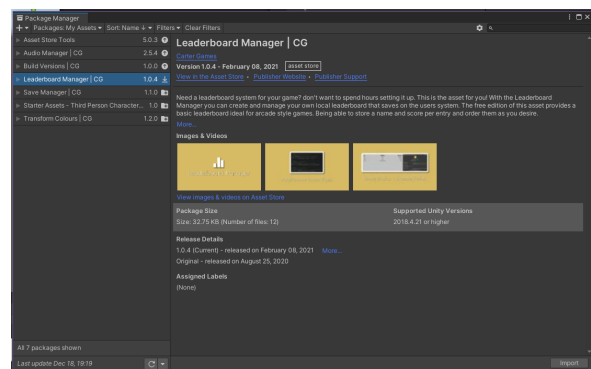
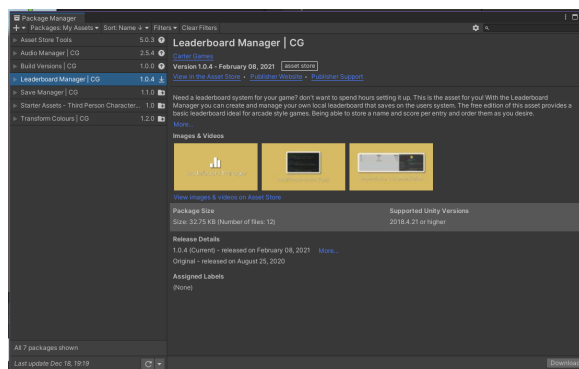


To get the asset you'll have to press the **Add to My Assets** button. Doing so will add the asset your account so you can access it in which ever version of Unity you want that the asset is supported in.

From the asset store you can then press the **Open in Unity** button. Doing so will open Unity on your system with the package to use.



When in Unity you can import the asset via the package manager, found under **Window → Package Manager**. The window will have to option to filter to the packages, from here you can select **My Assets**. You may need to login to your UnityID to see your assets. Once the packages appear, the Leaderboard Manager will appear on the list. Here you can download the latest version of the asset and then import it via the button in the bottom right of the Package Manager panel.



Once you press the import button, a package window may appear with information about what the package contains and options for you to choose what to import. You should import the full package and then remove and demo elements if you don't want them. You can see an example of the package import window to the left. When ready, just press import and the package will import into project, the editor will reload and you will be good to go.

Setup

The asset doesn't need any setup by you to function. By default there are no leaderboards in the save. The save will automatically create itself when you first create a leaderboard with the system. The leaderboard save goes to the users computer persistent datapath, under the **/Leaderboard** directory. For more on this please and to get the location for your system, please see the Unity documentation below:

Application.persistentDataPath

This value is a directory path where you can store data that you want to be kept between runs. When you publish on iOS and Android, persistentDataPath points to a public directory on the device. Files in this location are not erased by app updates. The files can still be erased by users directly.

<https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html>



TextMeshPro Support

The asset has an extra package that adds a version of the leaderboard display that uses a TextMeshPro text component instead of the normal unity component. This version is not in the package as standard in-case more beginner users get are unsure of the difference or don't have TextMeshPro setup. To import the package, just navigate to [Carter Games/Leaderboard Manager/TMP](#) and double click the package and press import. Once the editor has reloaded you'll have the Leaderboard Display (TextMeshPro) script to attach in to objects and use just like the normal display script.

Example Scene



The asset comes with an example scene which is in the package to allow you to see the asset in action without needing to do any work on your end. You can use the buttons to perform actions for the manager and display at the same time so you can see the results in real-time. The buttons are labelled with what they do and are self explanatory. The fields define when is changed, a breakdown is below:

Board ID

Defines the board ID to use for entries, Example is the boardID for the example scene, changing this will break the display unless you update it there as well.

Name

Defines the name of any entry to add or remove dependant on the buttons you press.

Score

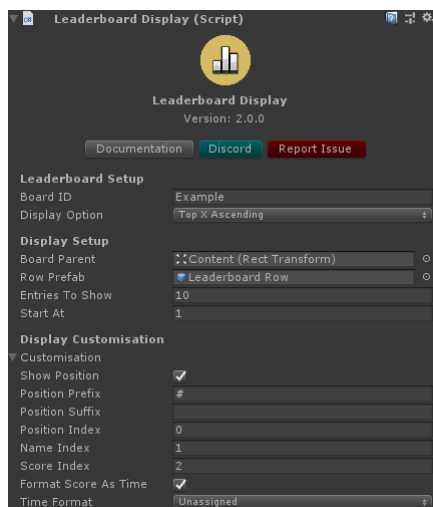
Defines the score of any entry to add or remove dependant on the buttons you press.

Inspector Reference



Here you can find detailed breakdowns of all the custom inspectors this asset uses.

Leaderboard Display



This display inspector applies to both the normal and TextMeshPro variants of the display script. This script is designed to help you display the leaderboard data, however you may want to create your own for a custom display. The header section shows the asset logo if imported along with the version number for the asset and some buttons.

- **Documentation** - Links to the online copy of the asset documentation.
- **Discord** - Links to the Carter Games Discord server, where you chat about games, show your work and get additional support.
- **Report Issue** - Links to the bug reporting form on the [carter.games](#) website where issues with the asset can be reported directly.

Leaderboard Setup Section:

- **Board ID** - The ID of the board to display, Example will display the example board info.
- **Display Option** - Defines how the data from the board is displayed.

Display Setup Section:

- **Board Parent** - The object that is the parent for the leaderboard rows to spawn under, normally a scroll rect or layout group object.
- **Row Prefab** - The prefab to use for spawning rows in the board display, we provide one with the asset, it just has 3 text objects in a line.
- **Entries To Show** - Defines how many entries to show on the board, this option is disabled on display options that define this, like top 3 options.
- **Start At** - Defines where in the parent the leaderboard starts, so if you have objects that are children of the parent already, you can offset where the board starts writing. In the example scene we set it to 1 to allow the leaderboard row headers to be at the top of the display. Set it to 0 if you don't need an offset.

Display Customisation Section, this section can be hidden by a dropdown if desired.

- **Show Position** - Defines whether or not a position number should be show on the display.
- **Position Prefix** - Defines the prefix used before the position value on the display.
- **Position Suffix** - Defines the suffix used after the position value on the display.
- **Position Index** - Defines the index in the row prefab where the position text element is.
- **Name Index** - Defines the index in the row prefab where the name text element is.
- **Score Index** - Defines the index in the row prefab where the name text element is.
- **Format Score As Time** - Defines whether or not the score value of the entries are read as time values.
- **Time Format** - Defines the way the time is formatted.

Leaderboard Row Prefab

The leaderboard row prefab is a prefab with 3 text elements attached, if you have imported the TextMeshPro extension there will also be a TextMeshPro version of the prefab. You don't need to use these, but you will need at-least 2 text UI elements to display the name & score with our display. If needed you can always make your own display by inheriting the LeaderboardDisplay class and make an override to the `UpdateDataOnRows` method.

Scripting Reference



Here you can find detailed breakdowns of all methods, properties, actions and events for the leaderboard manager.

Namespaces

All the leaderboard manager asset code is under the following namespaces:

```
// General Code
CarterGames.Assets.LeadersboardManager

// Editor Code (Custom Inspectors)
CarterGames.Assets.LeadersboardManager.Editor

// Example Code (Example Scene Only)
CarterGames.Assets.LeadersboardManager.Demo
```

To access code from the asset you will need to be using the asset namespace in your script or use the full path of the asset script in order to access it.

Example Usage

```
// Using statement
using CarterGames.Assets.LeadersboardManager;

LeadersboardManager.CreateLeaderboard("Example");

// Full path usage
CarterGames.Assets.LeadersboardManager.LeadersboardManager.CreateLeaderboard("Example");
```

Scripts

Full breakdowns of all methods in each class can be found here. Each script has its own page listed below:

LeadersboardManager.cs




All methods in class are static and can be accessed by using class name before the method without needing a reference to an instance of the script.

```
// Example of usage... replacing "MyMethodHere" with the method you want to call.
LeadersboardManager.MyMethodHere
```

CreateLeaderboard

Creates a new leaderboard with the id entered and adds it to the leaderboard file for use in your project.

Arguments

 Argument	 Type	 Description
<u>boardID</u>	string	The ID that the board should be accessed by, this is CaSe SeNsItivE.




Example Usage

```
LeadersboardManager.CreateLeaderboard("Level1");
```

DeleteLeaderboard

Deletes the leaderboard of the id entered. This is an instant action and cannot be undone when called.

Arguments

 Argument	 Type	 Description
<u>boardID</u>	string	The ID that the board should be used, this is CaSe SeNsItIvE.




Example Usage

```
LeaderboardManager.DeleteLeaderboard("Level1");
```

ClearLeaderboard

Clears the leaderboard of the id entered of all entries. This is an instant action and cannot be undone when called.

Arguments

 Argument	 Type	 Description
<u>boardID</u>	string	The ID that the board should be used, this is CaSe SeNsItIvE.




Example Usage

```
LeaderboardManager.ClearLeaderboard("Level1");
```

BoardExists

Gets whether or not a board of the id entered exists in the system.

Arguments

 Argument	 Type	 Description
<u>boardID</u>	string	The ID that the board should be used, this is CaSe SeNsItIvE.

Returns

bool - Does the board exists, true if it does, false if not.

Example Usage




```
// If Statement
if (LeaderboardManager.BoardExists("Level1"))
{
    // Do Stuff here
}

// Variable
var _hasBoard = LeaderboardManager.BoardExists("Level1");
```

GetLeaderboard

Gets the data for the leaderboard of the id entered.

Arguments

 Argument	 Type	 Description
<u>boardID</u>	string	The ID that the board should be used, this is CaSe SeNsItIvE.

Returns

LeaderboardData - The data on the leaderboard requested or null if no data was found.




Example Usage




```
var _data = LeaderboardManager.GetLeaderboard("Level1");
```

AddEntryToBoard

Adds an entry of the values defined to the board of the defined id. There are 2 versions of this method.

Arguments

 Argument	 Type	 Description
<u>boardID</u>	string	The ID that the board should be used, this is CaSe SeNsItIvE.
<u>entry</u>	LeaderboardEntry	The entry data to add to the leaderboard.

 Argument	 Type	 Description
<u>boardID</u>	string	The ID that the board should be used, this is CaSe SeNsItIvE.

Aa Argument	Type	Description
<u>name</u>	string	The name to enter into the board entry.
<u>score</u>	double	The score to enter into the board entry.

Example Usage

```
// LeaderboardEntry version...
LeaderboardManager.AddEntryToBoard("Level1", new LeaderboardEntry("John", 100));

// Standard version...
LeaderboardManager.AddEntryToBoard("Level1", "John", 100);
```

DeleteEntryFromBoard

Removes an entry of the values defined from the board of the defined id. There are 2 versions of this method.

Arguments

Aa Argument	Type	Description
<u>boardID</u>	string	The ID that the board should be used, this is CaSe SeNsItIvE.
<u>entry</u>	LeaderboardEntry	The entry data to add to the leaderboard.

Aa Argument	Type	Description
<u>boardID</u>	string	The ID that the board should be used, this is CaSe SeNsItIvE.
<u>name</u>	string	The name to find in the board to remove.
<u>score</u>	double	The score to find in the board to remove.

Example Usage

```
// LeaderboardEntry version...
LeaderboardManager.DeleteEntryFromBoard("Level1", new LeaderboardEntry("John", 100));

// Standard version...
LeaderboardManager.DeleteEntryFromBoard("Level1", "John", 100);
```

Save

Updates the save file for the leaderboard data with the latest data in the system when called.

Example Usage

```
LeaderboardManager.Save();
```

Load

Updates the system with the latest data from the save file for the leaderboard data when called.

Example Usage

```
LeaderboardManager.Load();
```

LeaderboardDataStore.cs

The leaderboard data class holds all the data for 1 leaderboard in the system. It has a variety of methods and properties to get information from the leaderboard to use in your projects.

Constructors

The data script just has a blank constructor that sets up the leaderboards list to not be null.

```
// Blank Constructor for the data class...  
public LeaderboardDataStore()
```

Example Usage

```
// Using blank constructor  
var _board = new LeaderboardDataStore();
```

Leaderboards

Gets/sets the leaderboards in the store.

Returns

`List<LeaderboardData>` - A list of all the leaderboards in the data store.

Example Usage

```
LeaderboardDataStore dataStore;  
  
var _exampleBoard = dataStore.Leaderboards.FirstOrDefault(t => t.BoardID.Equals("Example"));
```

LeaderboardData.cs

The leaderboard data class holds all the data for 1 leaderboard in the system. It has a variety of methods and properties to get information from the leaderboard to use in your projects.

Constructors

The data script has the option to use a constructor to setup the board ID on creation as well as a blank one.

```
// Blank Constructor for the data class...
public LeaderboardData()

// Constructor that sets up the board ID when created...
public LeaderboardData(string id)
```

Example Usage

```
// Using blank constructor
var _board = new LeaderboardData();

// Using setup constructor
var _board = new LeaderboardData("Level1");
```

BoardID

Gets/sets the board ID defined for this leaderboard.

Returns

`string` - The ID of the board.

Example Usage

```
LeaderboardData data;

Debug.Log(data.BoardID);
```

BoardData

Gets/sets the board data for this leaderboard.

Returns

`List<LeaderboardEntry>` - All the entries in the leaderboard as `LeaderboardEntry` data.




Example Usage




```
LeaderboardData data;  
  
Debug.Log(data.BoardData.Count);
```

AddEntry

Adds an entry to the board. There are 2 versions of this method.

Arguments

 Arguments	 Type	 Description
<u>name</u>	string	The name of the entry to add.
<u>score</u>	double	The score of the entry to add.

 Arguments	 Type	 Description
<u>entry</u>	LeaderboardEntry	The entry data to add to the board.




Example Usage

```
LeaderboardData data;  
  
// Using name and score  
data.AddEntry("John", 100);  
  
// Using entry data  
data.AddEntry(new LeaderboardEntry("John", 100));
```

DeleteEntry

Removes an entry to the board. There are 2 versions of this method.

Arguments

 Arguments	 Type	 Description
<u>name</u>	string	The name of the entry to find and remove.
<u>score</u>	double	The score of the entry to find and remove.

Aa Arguments	Type	Description
<u>entry</u>	LeaderboardEntry	The entry data to find and remove from the board.

Example Usage

```
LeaderboardData data;

// Using name and score
data.DeleteEntry("John", 100);

// Using entry data
data.DeleteEntry(new LeaderboardEntry("John", 100));
```

ClearBoard



Note this action cannot be undone once called.

Clears the leaderboard of all data.

Example Usage

```
LeaderboardData data;

data.ClearBoard();
```

GetEntry

Gets the entry at the entry ID entered. This method has several overloads, arguments for each are displayed below

Arguments

Aa Arguments	Type	Description
<u>id</u>	double	The entry id of the entry to get.

Aa Arguments	Type	Description
<u>name</u>	string	The name of the entry to get.
<u>score</u>	double	The score of the entry to get.

Aa Arguments	Type	Description
<u>entry</u>	LeaderboardEntry	The entry data to match to get.

Returns

LeaderboardEntry - The leaderboard entry found in the board or null if no matching entries were found.

Example Usage

```
LeaderboardData data;
LeaderboardEntry entry;

// Using entry ID
entry = data.GetEntry(1);

// Using name and score
entry = data.GetEntry("John", 100);

// Using entry data
entry = data.GetEntry(new LeaderboardEntry("John", 100));
```

GetTop3Ascending

Gets the top 3 entries in the board with the lowest score.

Returns

LeaderboardEntry[] - The top 3 entries in the board in ascending order.

Example Usage

```
LeaderboardData data;

var _top3Ascending = data.GetTop3Ascending;
```

GetTopXAscending

Gets the top x number entries in the board with the lowest score.

Arguments

Aa Arguments	Type	Description
x	int	The amount of entries to get.

Returns

`LeaderboardEntry[]` - The top x number entries in the board in ascending order.

Example Usage

```
LeaderboardData data;  
  
// Gets the top 5 entries ascending...  
var _topXAscending = data.GetTopXAscending(5);
```

`GetAllAscending`

Gets all entries in the board in ascending order.

Returns

`LeaderboardEntry[]` - The top x number entries in the board in ascending order.

Example Usage

```
LeaderboardData data;  
  
var _allAscending = data.GetAllAscending;
```

`GetPositionAscending`

Gets the position of the entered entry in the leaderboard from an ascending score perspective. There are two versions of this method.

Arguments

Arguments	Type	Description
<u>entry</u> .	<code>LeaderboardEntry</code>	The entry to search for.

Arguments	Type	Description
<u>name</u>	<code>string</code>	The name to search for.
<u>score</u>	<code>double</code>	The score to search for.

Returns

Int - The position of the entry.

Example Usage

```
LeaderboardData data;
LeaderboardEntry entry;

// Using LeaderboardEntry...
var _positionAscending = data.GetPositionAscending(entry);

// Using Name & Score Values...
var _positionAscending = data.GetPositionAscending("John", 100);
```

GetTop3Descending

Gets the top 3 entries in the board with the highest score.

Returns

LeaderboardEntry[] - The top 3 entries in the board in descending order.

Example Usage




```
LeaderboardData data;

var _top3Decending = data.GetTop3Decending;
```

GetTopXDescending

Gets the top x number entries in the board with the highest score.

Arguments

 Arguments	 Type	 Description
x	int	The amount of entries to get.

Returns

LeaderboardEntry[] - The top x number entries in the board in descending order.

Example Usage


```
LeaderboardData data;

// Gets the top 5 entries descending...
var _topXDescending = data.GetTopXDescending(5);
```

GetAllDescending

Gets all entries in the board in descending order.

Returns

LeaderboardEntry[] - The top x number entries in the board in descending order.

Example Usage




```
LeaderboardData data;




var _allDescending = data.GetAllDescending;
```

GetPositionDescending

Gets the position of the entered entry in the leaderboard from an descending score perspective. There are two versions of this method.

Arguments

 Arguments	 Type	 Description
<u>entry</u>	LeaderboardEntry	The entry to search for.

 Arguments	 Type	 Description
<u>name</u>	string	The name to search for.
<u>score</u>	double	The score to search for.

Returns

Int - The position of the entry.

Example Usage

```
LeaderboardData data;
LeaderboardEntry entry;

// Using LeaderboardEntry...
var _positionDescending = data.GetPositionDescending(entry);

// Using Name & Score Values...
var _positionDescending = data.GetPositionDescending("John", 100);
```

LeaderboardEntry.cs

The leaderboard data class holds all the data for 1 leaderboard in the system. It has a variety of methods and properties to get information from the leaderboard to use in your projects.

Constructors

The data script has 2 constructors, a blank constructor that does nothing other than making a new instance of the class & a setup constructor that takes in a name and score.

```
// Blank Constructor for the data class...
public LeaderboardEntry()

// Setup Constructor
public LeaderboardEntry(string name, double score)
```

Example Usage

```
// Using blank constructor
var _entry = new LeaderboardEntry();

// Using setup constructor
var _entry = new LeaderboardEntry("John", 100);
```

EntryID

The entry number for this entry, not the position, but the order of when it was added.

Returns

double - A id for the entry.

Example Usage

```
LeaderboardEntry entry;

Debug.Log(entry.EntryID);
```

Name

The name for the entry to the board.

Returns

string - The name of the person this entry is for.

Example Usage

```
LeaderboardEntry entry;  
  
Debug.Log(entry.Name);
```

Score

The entry number for this entry, not the position, but the order of when it was added.

Returns

double - A score of the person this entry is for.

Example Usage

```
LeaderboardEntry entry;  
  
Debug.Log(entry.Score);
```

ConvertScoreToTime

Converts the score value to a timespan going from seconds. You can use the same code in this method to make your own for milliseconds, minutes etc.

Returns

TimeSpan - The timespan the score is equal to when converted to seconds.

Example Usage

```
LeaderboardEntry entry;  
  
Debug.Log(entry.ConvertScoreToTime(entry.score));
```

LeaderboardDisplay.cs

The leaderboard display class is a pre-built display system for a leaderboard. You can display a leaderboard without touching the code in the script at all. However here you'll find the public options you can edit via code should you wish.

If you are looking for information on the inspector for this script, please refer to the inspector reference section.

BoardID

The board id of the leaderboard to display. This is a property with a `get` and `set` that are public.

Returns

`string` - A id of the board that is being displayed.

Example Usage

```
LeaderboardDisplay display;

// Getter
var _id = display.BoardID;

// Setter
display.BoardID = "Example";
```

DisplayOption

The display option for the leaderboard display to use. This is a property with a `get` and `set` that are public.

Returns

`DisplayOption` - A display option that is currently in use.

Example Usage

```
LeaderboardDisplay display;

// Getter
var _option = display.DisplayOption;

// Setter
display.DisplayOption = DisplayOption.AsWritten;
```

EntriesToShow

The display option for the leaderboard display to use. This is a property with a `get` and `set` that are public.

Returns

`int` - The amount of entries to show.

Example Usage

```
LeaderboardDisplay display;

// Getter
var _entries = display.EntriesToShow;

// Setter
display.EntriesToShow = 5;
```

UpdateDisplay

Updates the display to show the latest entries using the settings defined in the script.

Example Usage

```
LeaderboardDisplay display;

display.UpdateDisplay();
```

ClearDisplay

Clears the display visuals of entries.

Example Usage

```
LeaderboardDisplay display;

display.ClearDisplay();
```

DisplayOption.cs

The leaderboard option enum holds all the display options for leaderboard display script.



Note: that there will only work with the display script we provide. You will have to implement your own usage

Unassigned

The none/null option for the display enum. This will not display the leaderboard if using the leaderboard display provided with the asset.

AsWritten

Display the leaderboard in the order it is written in the data. This is essentially unordered, but great if you want to show the latest entries in the board.

Ascending

Display the leaderboard in ascending order based on the score of each entry.

Descending

Display the leaderboard in descending order based on the score of each entry.

Top3Ascending

Displays the leaderboard in ascending order based on the score of each entry, but only shows the first 3 entries.

Top3Descending

Displays the leaderboard in descending order based on the score of each entry, but only shows the first 3 entries.

TopXAscending

Displays the leaderboard in ascending order based on the score of each entry, but only shows the amount of entries setup in the EntriesToShow value on the display script.

TopXDescending

Displays the leaderboard in descending order based on the score of each entry, but only shows the amount of entries setup in the EntriesToShow value on the display script.

DisplayTimeFormat.cs

The leaderboard time format enum holds all the display options for leaderboard display script to format the score value as a time value instead. This only take effect if the format as time bool is true.



Note: that there will only work with the display script we provide. You will have to implement your own usage

All examples below use the following values in their score values:

```
123
456
12345
```

Unassigned

The none/null option for the display enum. This will not display the leaderboard score values as their raw values with no formatting.

MillisecondsOnly

Display the leaderboard score as if the score value is in milliseconds only.

Example

```
123
456
345
```

SecondsOnly

Display the leaderboard score as if the score value is in seconds only.

Example

```
03
36
45
```

SecondsMilliseconds

Display the leaderboard score as if the score value is in seconds followed by milliseconds.

Example

```
00:123
00:456
12:345
```

MinutesOnly

Display the leaderboard score as if the score value is in minutes only.

Example

```
02
07
25
```

MinutesSeconds

Display the leaderboard score as if the score value is in minutes followed by seconds.

Example

```
02:03
07:36
25:45
```

MinutesSecondsMilliseconds

Display the leaderboard score as if the score value is in minutes followed by seconds and milliseconds.

Example

```
00:00:123
00:00:456
00:12:345
```

HoursOnly

Display the leaderboard score as if the score value is in hours only.

Example

```
00
00
03
```

HoursMinutes

Display the leaderboard score as if the score value is in hours followed by minutes.

Example

```
00:02
00:07
03:25
```

HoursMinutesSeconds

Display the leaderboard score as if the score value is in hours followed by minutes and seconds.

Example

```
00:02:03
00:07:36
03:25:45
```

HoursMinutesSecondsMilliseconds

Display the leaderboard score as if the score value is in hours followed by minutes, seconds & milliseconds.

Example


```
00:00:00:123
00:00:00:456
00:00:12:345
```

LeaderboardDisplayTMP.cs

This is an exact copy of the leaderboard display script but with text mesh pro instead of the unity text component. Please refer to the leaderboard display script section for scripting reference.

ExampleManager.cs

The example manager is the script that runs the example scene and nothing else. You don't need it for the asset but you can use it as a guide to help you should you need it.

AddToBoard

Calls the leaderboard manager to add an entry to the board. It uses the text fields in the example scene to get the fields to add.

RemoveFromBoard

Calls the leaderboard manager to remove an entry to the board. It uses the text fields in the example scene to get the fields to remove.

ClearBoard

Calls the leaderboard manager to clear the leaderboard data.

F.A.Q

Can I use leaderboard from version 1.x.x?

Sadly not, the backend change a lot in the upgrade to version 2.x.x, hence why we didn't make it a minor patch staying on the 1.x.x versioning. We advise that you only upgrade if you need as you will lose and stored entries beforehand.

Can I use Leaderboard Data to add entries instead of Leaderboard Manager?

Yep, the Leaderboard Manager just calls the methods on the data for the board you are adding to, so you are calling the same method but with 1 less step.

The example scene shows no entries, is there something wrong?

Nope, we can't ship the asset with entries in the board that easily, so by default there are no entries. The example scene is meant to show you that you can add and remove entries from an example board and display them an many ways.

Support

If you need any additional support or just have some questions you may get in touch via to following methods:

Email

You can send us an email to the following address support@carter.games this is monitored and you will get a response within 72 hours of your request, though we normally get back to you within the hour. Note we are UK based so if its the middle of the night for us we won't be able to get back to you until the morning.

Discord

We have a community discord server, note that this is not just for asset support so please you the **Leaderboard Manager** channel in the server for your request. You can access this via assigning yourself the **Assets** role on the server. This is monitored and you will get a response within 72 hours of your request, though we normally get back to you within the hour. Note we are UK based so if its the middle of the night for us we won't be able to get back to you until the morning. The server invite link is below should you wish to use this method:

Join the Carter Games Discord Server!

Check out the Carter Games community on Discord - hang out with 57 other members and enjoy free voice and text chat.

 <https://carter.games/discord>



Report Form

Should you find an issue with our asset, the best way to let us know is via the report form on our website. you can access this via the link below:

Carter Games: Report an Bug/Issue in our products

Use this page to report issues that you find in our products so we can fix them. People who report issues to us will get credited for the help.

 <https://carter.games/report/>

Contact Form

Should you want to, you can send a message via our website contact form which we will get and response within 72 hours of your request, though we normally get back to you within the hour. Note we are UK based so if its the middle of the night for us we won't be able to get back to you until the morning. You can access this via the link below:

Carter Games: Contact Us

Use the form below to send us a email with your message. Alternatively you can email us directly via hello@carter.games. Emailing us is the best way to get in touch, we can't guarantee a response to messages sent by other means.

 <https://carter.games/contact/>