

Realm Games

Simple Local Leaderboard

Overview

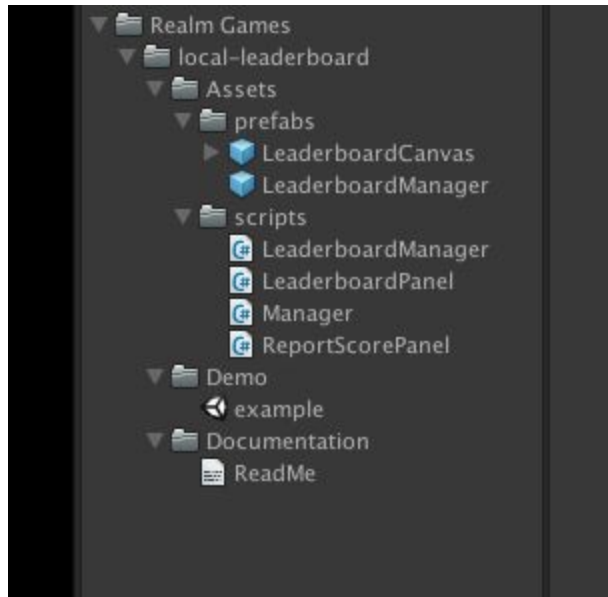
a simple leaderboard that uses Unity playerprefs to store leaderboard entries.



Features:

1. multiple user support
2. multiple sort mode
3. multiple display modes
4. example leaderboard ui
5. automatically overrides old scores

Prefab Usage



Drag **LeaderboardManager** prefab into any scene that uses the leaderboard, the Leaderboard Manager is the main entry point into the leaderboard API

Drag **LeaderboardCanvas** prefab into your scene to use the default leaderboard UI.

Report Score

Save a score to the leaderboard system

Long **score** : value of the score to save

String **leaderboardId** : name of the leaderboard to save score to

String **userId** : display name of the user

SortOrder **mode** : sorting mode for the leaderboard (HIGH_TO_LOW, LOW_TO_HIGH)

```
string leaderboardId = "default";
string userId = nameField.text;
long score = long.Parse(scoreField.text);

LeaderboardManager.Instance.ReportScore(score, leaderboardId, userId, SortOrder.HIGH_TO_LOW, (bool success) =>
{
    Debug.Log("Score Reported: " + success);

    if (success)
    {
        leaderboard.Refresh("default");
    }
});
```

Delete Scores

Deletes all scores in the leaderboard named "default"

```
LeaderboardManager.Instance.DeleteScores("default");
```

Load Scores

Loads all the scores in the leaderboard identified by the id 'leaderboardID'

String **leaderboardId** : name of the leaderboard

Action<IScore[]> **Callback** : callback method to return scores

SortOrder **mode** : sorting mode for the leaderboard (HIGH_TO_LOW, LOW_TO_HIGH)

```
LeaderboardManager.Instance.LoadScores(leaderboardID, Callback, sortMode);
```