Directions:

Must already have Steam overlay working with valid Steam ID and leaderboard created. There are some video tutorial on youtube explaining how to do this or you can visit: <https://docs.unrealengine.com/latest/INT/Programming/Online/Steam/>

The first class CSteamLeaderboards is almost a direct copy from <https://partner.steamgames.com/documentation/bootstrap_leaderboards>

Do not drag and drop these files into your project. I would make a new C++ class from unreal engine and copy the CSteamLeaderboards.

Next make a new Game Mode C++ class from the engine again. Copy the CFGameMode into those and rename them appropriately. My game is Crystal Flux hence the “CF.”

You’ll also need a structure class for “LeaderboardData.” This just creates a struct that can be used anywhere in your game if needed.

Once everything is in place and compiled you can start with the Blueprints. Derive Blueprint class based upon your C++ game mode. On whatever level you wish you get or set leaderboard information you need to set the game mode to the Blueprint version of that class. Mine is MyCFGameMode.

In this class there is a custom event “Leaderboard Scores Found” if you do not see it, uncheck “contact sensitive” when adding that node. It will be under “Steam Functions.” This will eventually output the top scores.

In your high score widget of whatever Blueprint class you wish to use, you are going to want to:

“get game mode” > cast to that game mode > “set leaderboard name” > “set leaderboard score” > “get leaderboard”

These are asynchronous calls that, in time, output the top scores. It will only work when you run the game in “standalone” mode. Because of this, you can’t really use breakpoints so printing to the screen is needed for debugging. You will want to delete them or comment them out when everything is working.

Its just the start and there are many more things the Steam API can do! Feel free to upload your branch and I also have a few more functions coming.

-Ryan Gadz