GDD Homework 1

Out: Monday, September 25th at Noon Due: Monday, October 2nd (10/02/17) at Noon (12 PM)

Windows Users: Make sure to unzip the assignment (extract all) and delete the now useless Homework1.zip before getting started.

This assignment focuses a little more on the topics covered in the previous homework. We'll be looking at user input as well at the very basics of Unity's UI elements.

Part 1: Input

So far, the platform follows the mouse back and forth, but it's pretty hard to stack boxes and do well. Let's give the player a little more control.

Replace the body of the following functions (in *PlayerController.cs*) with the Unity functions that indicate whether either key is *currently* being pressed.

- `TiltLeft`: "A" or the Left Arrow Key
- `TiltRight`: "D" or the Right Arrow Key
- `PushUp`: "W" or the Up Arrow Key
- `PushDown`: "S" or the Down Arrow Key

Part 2: Unity UI

Let's spruce up the User Interface. We're going to add a score counter and the ability to quit by clicking on a button.

We'll be working in *UserInterface.cs*.

Score Counter

The game *is* keeping track of the player's performance under the hood, but the player has no way of knowing this. Rather than silently judging their actions, we should let the player know how they're doing.

- Fill in the body of `UpdateScore` to set the text stored in `_scoreText` to display the score in the format "Score: $Score" (that is, the word "Score" followed by the current score).

Quitting

It's good form to let the players quit the game without having to resort to Operating System-level commands (Cmd+Q, Alt+F4, etc.).

There is a function named `OnQuitClick` which will gracefully stop the game (whether in editor or in a standalone build).

- In the `InitializeQuit` function, make it so that clicking on the `_quit` button calls `OnQuitClick`

Part 3: A Tutorial

As our final trick, we're going to give the player a bit of instruction. In the wild, you generally spend a lot time on your tutorial. We've whipped up a really simple one that just displays text on the screen and advances whenever the player does certain things.

We'll be working in *Tutorial.cs*.

- Fill in the body of `UpdateText` so that it displays the correct message as determined by the value of `_tutorialState`.

Submission

As with the last homework, zip up **only** your "Assets" and "ProjectSettings" folders into a new folder named "NETID-Homework1".

For instance, my .zip file would contain a folder named "Assets" and a folder named "ProjectSettings", and the .zip file would be named "ecr867-Homework1".

Reminder: Unlike the previous homework, this one will be peer-graded. That means that you should *definitely* double-check that the .zip file that you submit is correct.

Upload it to Canvas and you're all set!