Steps for Project Increment 3

Overview

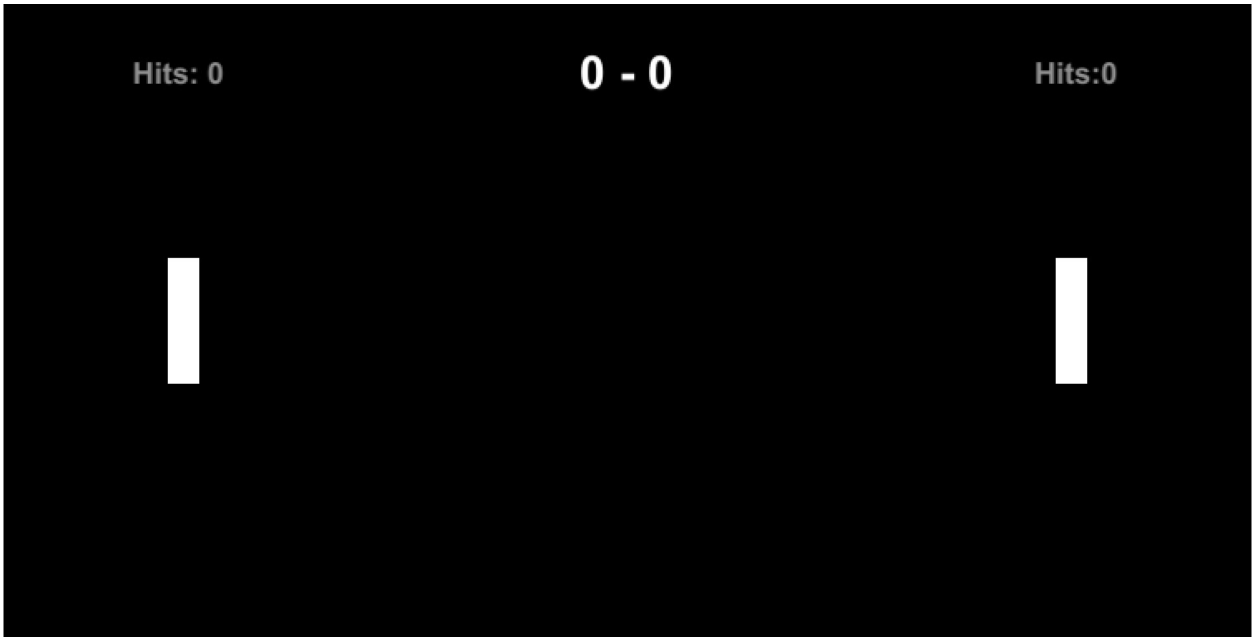
To give you some help in approaching your work for Increment 3, I've provided the steps I implemented when building my Increment 3 solution.

The required functionality for this increment is "the functionality from the previous increment and correct display of the score and hits for each player."

Step 1: Add HUD with scores and hits

For this step, you're adding a HUD with the scores and hit counts. These values won't be updated yet, that will come in the following steps.

Add a new HUD canvas to the scene and add left hits, right hits, and score Text objects as children of that game object. Put reasonable default text for those Text objects. Just as a guideline, I've provided a screen shot of my game with both hits and score displays included. You of course don't have to match my spacing exactly.



Be sure to select the Canvas in the HUD game object and change the UI Scale Mode in the Canvas Scaler component to Scale With Screen Size. This makes it so everything is placed and sized reasonably no matter what resolution the player uses when they run the game.

Step 2: Add Accurate Hit Counts

For this step, you're making the hits display accurate for each player. Remember, players score 1 hit for a standard ball and two hits for a bonus ball (which we don't have in our game yet). Players only score a hit if they bounce the ball off the front of their paddle.

Create a HUD script in the scripts/gameplay folder and attach the HUD script to the HUD game object. Add serialized fields to hold the left and right hits Text game objects on the canvas and populate those fields in the Inspector. Add static fields to hold the left and right hits Text scripts and any other information you think will be useful. Add a static `AddHits` method so a paddle can call it to add hits to its hits count. In the body of the new method, update the appropriate hits text that's displayed.

Add code to the `Start` method to populate the left and right Text script fields using the serialized fields. You need to do this because you need to actually get the Text components attached to the Text game objects.

Add a `hits` field and a public `Hits` property to the `Ball` class.

Add code to the Paddle script to call the new HUD method to add the appropriate number of hits when a ball hits the front of the paddle. You should access the `Hits` property in the ball that's hitting the paddle to find out how many hits the ball is worth. This isn't an optimal object-oriented solution because Paddle objects shouldn't really have to know about the existence of the HUD or the methods it exposes, but it's a reasonable solution before we start using delegates and event handling later in the course.

Add a property for the Standard ball hits to the `ConfigurationUtils` class so the `Ball` class (see next paragraph) will be able to access this value. Although you might think it would be better to just add a constant to the Ball class, in Project Increment 9 you'll be adding a more robust game configuration data structure to your game to make tuning easier. Putting all our configuration values into the `ConfigurationUtils` class as we move toward that increment will make things easier when we get there.

Add code to the `Ball Start` method to set the hits the ball is worth to the Standard hits value from `ConfigurationUtils`.

When you run your game, the appropriate hits text should increase by 1 each time you hit the ball with the front of a paddle.

Step 3: Add Accurate Scores

For this step, you're making the score accurate for each player. Remember, players score 1 point for a standard ball and two points for a bonus ball (which we don't have in our game yet). You should be able to complete this step by making changes that are similar to those you made in the previous step.

Of course, the `AddPoints` method you add to `HUD` script will need to have parameters for both the side that scored the points and how many points to add.

Turning In Your Assignment

This project increment is worth 10% of your overall course grade.

You're required to turn in ALL of the following by the beginning of the scheduled class time on the due date:

Electronic Copy

1. Zip up your entire assignment folder into a file named <your last name and first initial>.zip. Log into Canvas and submit the file into the appropriate assignment.

Use the default Windows compression utility for this; the grader may not own WinZip, 7Zip, or whatever other program you're using to build your zip file. If you use something other than the Windows compression utility and the grader can't unzip your submission, you'll get a 0 on the assignment.

Late Turn-ins

Turn-ins are due at the beginning of the scheduled class time on the specified due date. No late turn-ins will be accepted.