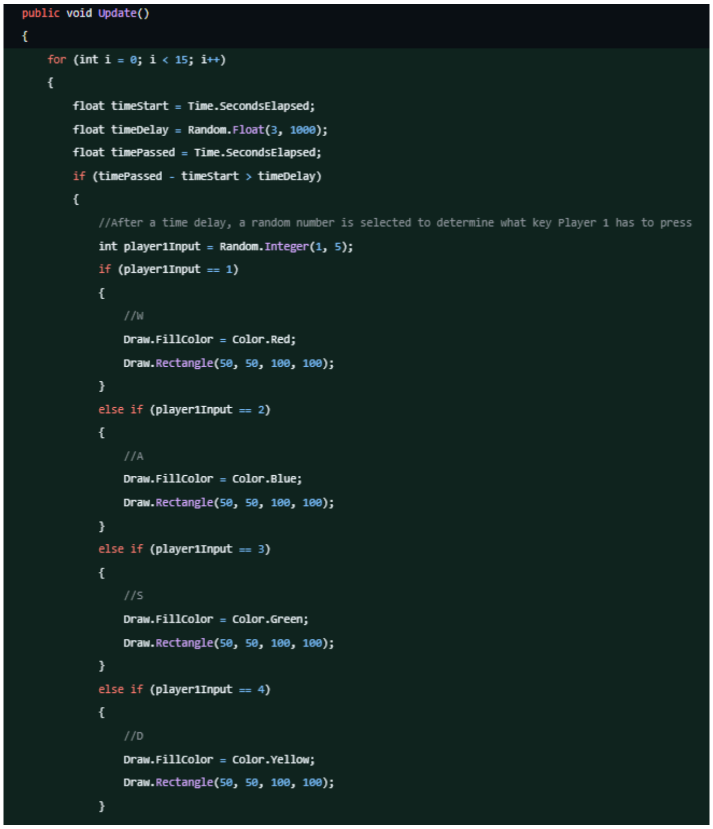
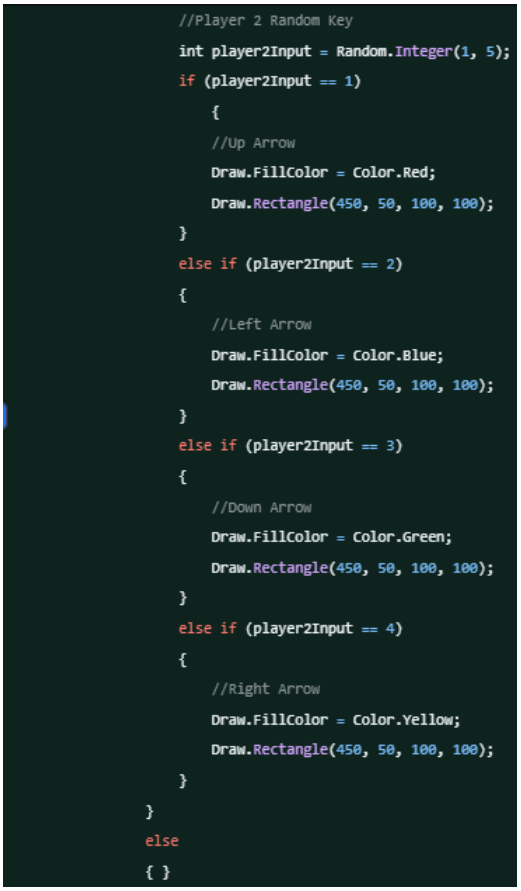
Brock Hutter Process Work - PVP Mode Branch
Initial Commit:



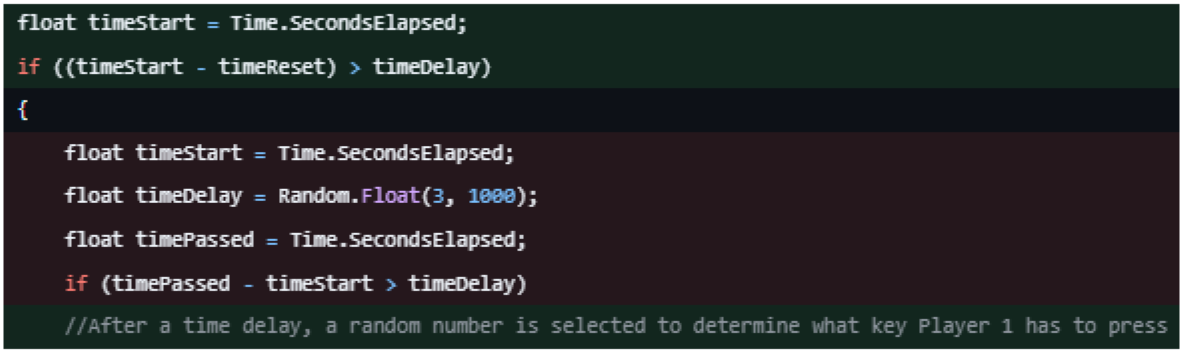
```
public void Update()
{
    for (int i = 0; i < 15; i++)
    {
        float timeStart = Time.SecondsElapsed;
        float timeDelay = Random.Float(3, 1000);
        float timePassed = Time.SecondsElapsed;
        if (timePassed - timeStart > timeDelay)
        {
            //After a time delay, a random number is selected to determine what key Player 1 has to press
            int player1Input = Random.Integer(1, 5);
            if (player1Input == 1)
            {
                //W
                Draw.FillColor = Color.Red;
                Draw.Rectangle(50, 50, 100, 100);
            }
            else if (player1Input == 2)
            {
                //A
                Draw.FillColor = Color.Blue;
                Draw.Rectangle(50, 50, 100, 100);
            }
            else if (player1Input == 3)
            {
                //S
                Draw.FillColor = Color.Green;
                Draw.Rectangle(50, 50, 100, 100);
            }
            else if (player1Input == 4)
            {
                //D
                Draw.FillColor = Color.Yellow;
                Draw.Rectangle(50, 50, 100, 100);
            }
```



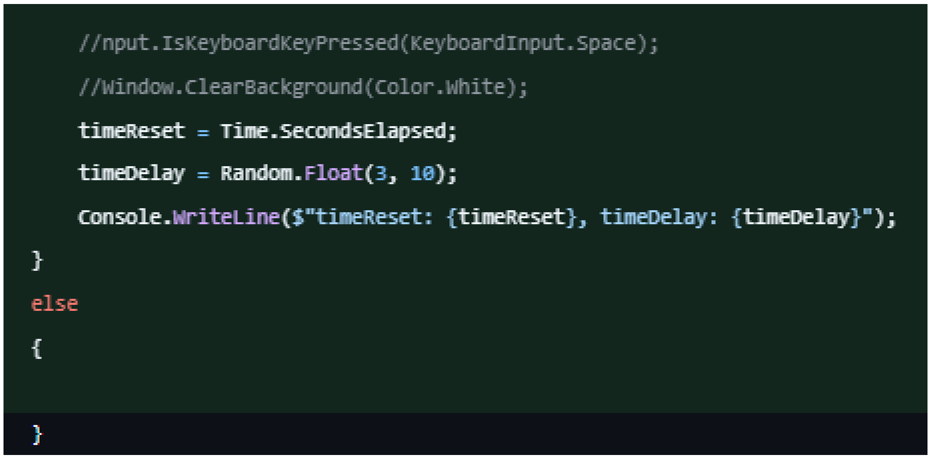
```
            //Player 2 Random Key
            int player2Input = Random.Integer(1, 5);
            if (player2Input == 1)
                {
                //Up Arrow
                Draw.FillColor = Color.Red;
                Draw.Rectangle(450, 50, 100, 100);
            }
            else if (player2Input == 2)
            {
                //Left Arrow
                Draw.FillColor = Color.Blue;
                Draw.Rectangle(450, 50, 100, 100);
            }
            else if (player2Input == 3)
            {
                //Down Arrow
                Draw.FillColor = Color.Green;
                Draw.Rectangle(450, 50, 100, 100);
            }
            else if (player2Input == 4)
            {
                //Right Arrow
                Draw.FillColor = Color.Yellow;
                Draw.Rectangle(450, 50, 100, 100);
            }
        }
        else
        { }
```

This first commit was a short start on the code, and was quite rough as I had lacked practice with a few of the elements and spent some time dabbling with the code a bit to create the game’s random input from each player and a colored box to go with it as a display of what button to press.

Second Commit:



```
float timeStart = Time.SecondsElapsed;
if ((timeStart - timeReset) > timeDelay)
{
    float timeStart = Time.SecondsElapsed;
    float timeDelay = Random.Float(3, 1000);
    float timePassed = Time.SecondsElapsed;
    if (timePassed - timeStart > timeDelay)
    //After a time delay, a random number is selected to determine what key Player 1 has to press
```



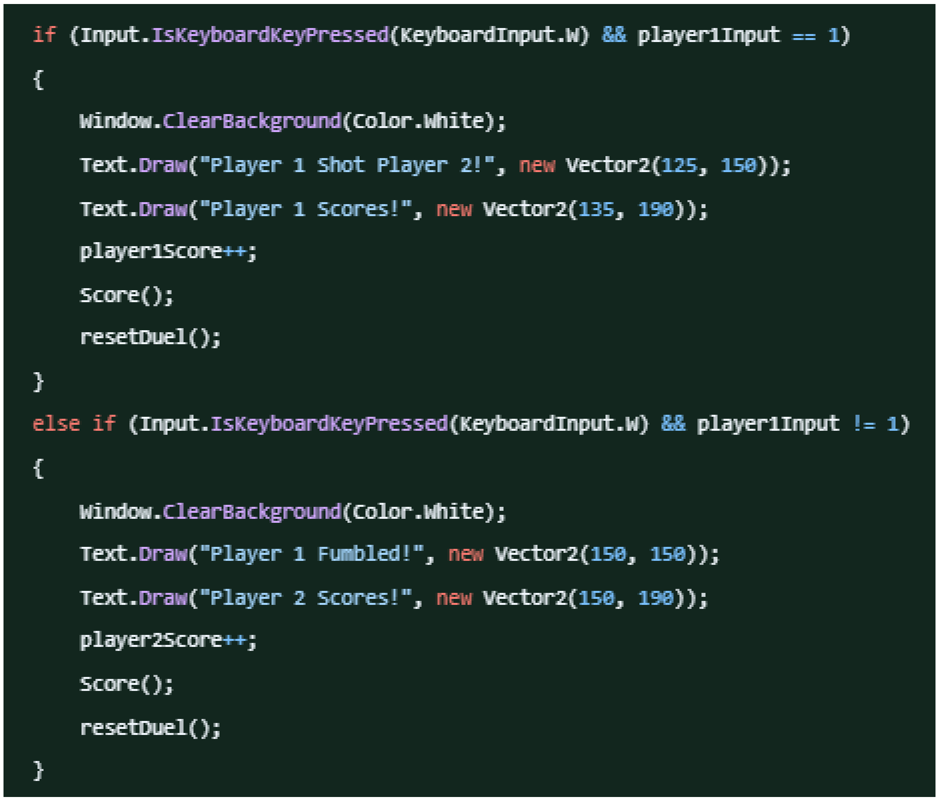
```
    //nput.IsKeyboardKeyPressed(KeyboardInput.Space);
    //Window.ClearBackground(Color.White);
    timeReset = Time.SecondsElapsed;
    timeDelay = Random.Float(3, 10);
    Console.WriteLine($"timeReset: {timeReset}, timeDelay: {timeDelay}");
}
else
{

}
```

In hindsight, many of the mistakes I made seem pretty silly and start to explain themselves as to why they didn't work, but for this commit I struggled with and managed to fix up the random time delay I wanted for each round before the players had to draw.

I was still left with the task of creating a pause between rounds, which ended up being much tougher for me than I'd expect.

Final Commit (for my branch)

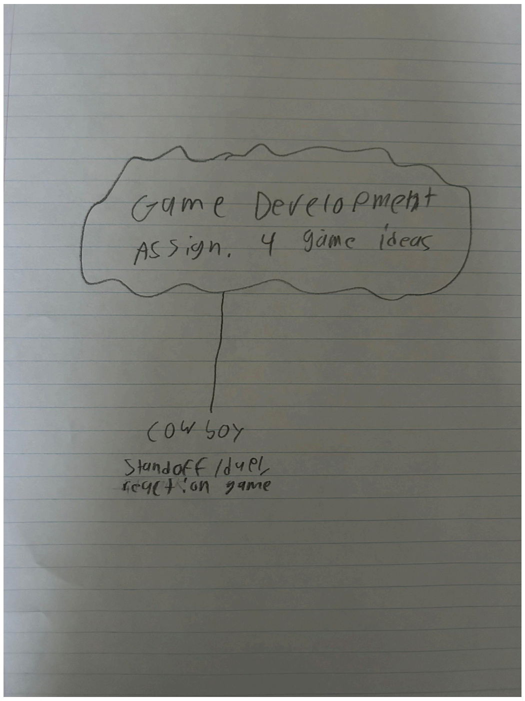


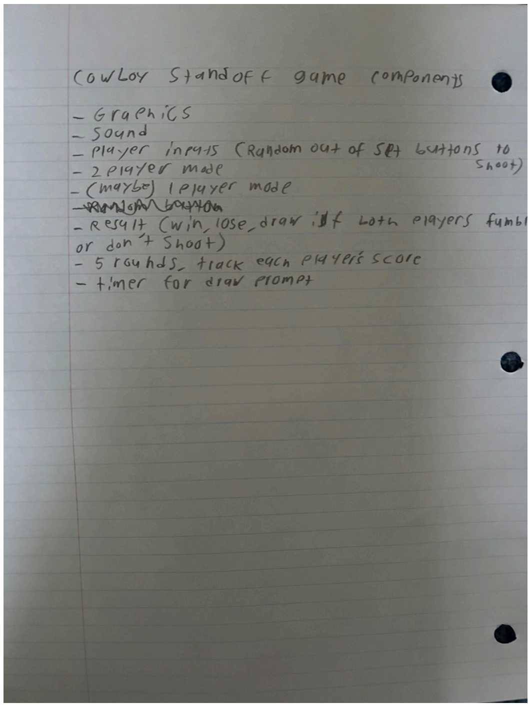
```
if (Input.IsKeyboardKeyPressed(KeyboardInput.W) && player1Input == 1)
{
    Window.ClearBackground(Color.White);
    Text.Draw("Player 1 Shot Player 2!", new Vector2(125, 150));
    Text.Draw("Player 1 Scores!", new Vector2(135, 190));
    player1Score++;
    Score();
    resetDuel();
}
else if (Input.IsKeyboardKeyPressed(KeyboardInput.W) && player1Input != 1)
{
    Window.ClearBackground(Color.White);
    Text.Draw("Player 1 Fumbled!", new Vector2(150, 150));
    Text.Draw("Player 2 Scores!", new Vector2(150, 190));
    player2Score++;
    Score();
    resetDuel();
}
```

When struggling with all the issues I was facing, I eventually took a long look at my code and realized how messy it got with my experimentation. I ended up rewriting the code from scratch and it turned out much better, a lot cleaner, and most importantly, it worked exactly the way I wanted it to!

In addition to fixing everything up, I ended up adding the paused intermission between rounds, the tracked score, text explaining the results of each round and so on.

Group Initial Plan:





cowboy standoff game components

- Graphics
- sound
- player inputs (Random out of set buttons to shoot)
- 2 player mode
- (maybe) 1 player mode
- ~~Random button~~
- Result (win, lose, draw if both players fumbl
or don't shoot)
- 5 rounds, track each player's score
- timer for draw prompt

After a very expansive brainstorm filled to the brim with ideas, we decided to settle on a western themed game, where I took inspiration from the Kirby Subgame: Samurai Kirby to make a game about dueling based on reaction times, but to put our own spin on it we made 2 major changes: one, it was a 2-player game where people could duel against each other as opposed to an ai opponent, and two, instead of just pressing one button, we scattered the controls to WASD and the arrow keys for each player, so when the prompt pops up, you have to press the correct button, otherwise, pressing the wrong button will cause you to fumble and the other player will score instead.

We ended up cutting a few ideas we wanted to implement to save time such as a single player mode and having a draw as a result.