

Faults discussed in this essay (Descriptions in the essay and numbered):

- 1) Fault where player could move left and right at wrong times
- 2) There was a fault where the player could get stuck when falling off a platform
- 3) Fault with how to call the animator for when the player should be immune

For our Code Review, we chose to go through the Player.cs script/class. This is the largest of the files that support our game, and would be the most likely to have errors. As we went through the file, one of the first things that we noticed was that all of the global variables we defined were somewhat unorganized. To make it look better, we could organize so that there is more space between some of those variables, do things like group the public variables, and then also make the comments so that they start on the same column to make it more readable and organized. Moving onto the Start() function, to reduce the lines of code, we noticed that we could have removed a variable and instead just added the arithmetic value generated into the next line. Not a huge deal but it would have made it more readable. We then moved onto the update function, and since it was just calling functions we didn't find anything we would want to change in it. We went down a couple more functions and we got to the Movement() function. **1)** In this one we had one issue where we were trying to figure out how to make it so the player could only move left and right when on the ground. So in combination with our onGround bool, Sam added a canMove bool such that if both of those were true, the player could move left and right. Going back through this code, we found that we wouldn't have changed this back since it worked well. **2)** Moving down into the Jumping() function, one fault that we found was due to the feetColliderSize variable. Without this variable added and set to the right size, when falling off a platform the player would get stuck. Sam helped make the variable, and Ben helped find a place to put it and set the value based on if the player was in the air or on the ground. When going through this code together, it was helpful to discuss which areas to place the variable, since the final spot to set the change of the variable was found to be best when the player released the jump button (space bar). This wouldn't have been possible without analyzing the code. **3)** Another issue that we found was the code would provide a fault when trying to define another animator object in another class. The first animator object for the player was defined in the player class, but the variable it needed to access was stuck in another class (the Game class). After analyzing the code and discussing options, we ended on the idea of keeping only one animator object and letting the variables in the other class be public for the Player class to see. This seemed obvious once we discussed the options, but it helped to discuss with the team to get to the result. The resulting code was placed in the CheckIfImmune() function, the fault was discussed by the team, and Ben was the one who wrote and pushed the code. Overall, going through parts of the code together was helpful for finding bugs that would have most likely taken much longer to find if doing it all alone. It was also helpful to discuss things like code organization at the beginning, and the discussion will help make our code better in the future.