

MA: Final Python game project - finished REFLECTION

Saturday, June 4, 2022 10:13 AM

During this Python final, I created a pong style game, with a few more advanced aspects, like ball speed and different game modes. I created 2 game modes. The first game mode was a local 2 player game mode in which one player would use the buttons, AD on their keyboard to control the blue paddle and the other player would use the arrow keys to control the green paddle. Here, I encountered an issue regarding handling two keyboard inputs simultaneously. I tried to fix this by using other packages and libraries to handle keypresses but nothing seemed to work and I observed that it was likely due to some hardware limitations. After some time, I tried to change the overall movement of the game where, once the movement buttons (left/right arrow keys or A/D keys) were pressed, the paddle would move that direction indefinitely. However, this still didn't work because if one person wanted to move another direction and the other player did, one of the inputs would still be cancelled out. So, I reverted back to the original movement mechanism. However, along this process I learned how to use different libraries to handle keyboard inputs, like pygame but I did not apply it to my game because it did not fix my issue of handling two keyboard inputs simultaneously. Moving on, my second game mode was a solo game mode and here, I created a computer player to play against. At first, this was tricky because the computer player would not allow the ball to hit its side and thus, it was too hard to play against. But, I challenged myself and I played around with the code and I eventually found the perfect balance between difficulty and enjoyability with the computer player. I created the computer player by using the position of the ball and moving the paddle when the ball would approach it. Next, I wanted to add music and sound effects to my game. Here, it took me some time to find the perfect way to handle music/sound in my game and I spent lots of time researching possible ways to utilize sound effects. In the end, I decided to use the library, pygame. I used pygame for all the music handling in my game, like the bouncing effect when the ball hit the wall or the paddle, the startup/intro music when the game is launched, and the sound effect when buttons are clicked. I did this by importing all my audio clips into a folder, like I did with my images, and I used the following commands to play the sound:

```
pygame.mixer.music.load(root_path+"\\Audios\\IntroSound.mp3")
pygame.mixer.music.play(loops=0)
pygame.mixer.music.set_volume(0.01)
```

I made sure to use the third command, which set the volume for all the clips, because most of the audio files were loud and annoying. All in all, during this process I challenged and stretched myself by creating a computer player to play against and 2 unique game modes, solo and duo. I learned many new things throughout this process about programming, but, specifically, I learned how to download and utilize the pygame library for sound and, although I did not implement it, I learned how to handle keyboard inputs using pygame.