Dev log of “Most Boring Racing Game Ever”

Most Boring Racing Game Ever (MBRGE) is a 2D local multiplayer game written in Python using PyGame. It features two players, drawn on screen as two colored dots (red and blue), racing from left to right across a flat horizontal track. As with any racing game, first one across the finish line wins. The game was inspired by the old Sonic the Hedgehog games on the Sega Mega Drive that released between 1992-1994. I am a Sonic fan, so this was honestly my first thought for the game. I wanted to also keep the game simple so it’s easy to grasp.

The core features here are the two playable characters, the static left to right track “design”, restart functionality and the win message at the top of the screen. The racers are controlled by WASD (Player 1) and the arrow keys (Player 2). The game is rendered in a 1910x1080 resolution because I didn’t want to change the level width from 2000 pixels as it felt like the perfect length. I actually started this game at a resolution of 640x480 but that’s just too small for a race like that’s pathetically short.

I used pygame.draw.circle() to create the two main racers. I also prevented the playable characters from going off screen to remove unnecessary jank. Just because its heavily inspired by Sonic doesn’t mean MBRGE needs to be just as janky as Sonic :)

The black rectangle on the floor is the ground and the green strip on the right of the screen is the finish line.

When it came to the camera, I had originally implemented a system that followed both players positions, but I scrapped this because while testing, I found that if one person got too far, the camera started to “drift” and it looked like one player was controlling both characters which looked really strange and it was just too confusing and just unnecessary for this type of game anyway.

I implemented a rudimentary check that detects if either player crosses the finish line with (player\_pos[0] <= finish\_x) which displays a “Player X Wins” and the movement freezes to determine the race is over. Also implemented a restart function which calls main() again so you don’t have to constantly close and reopen the app which is annoying.

Wasn’t all smooth sailing, at one point I got an error (ModuleNotFoundError: No module named 'pygame') because my Mac was running Python from its janky and quite frankly bad system version. Had to manually switch to the correct interpreter which is the version of Python you must download.

Actually, running PyGame stuff on MacOS is pretty miserable I have learned during this experience. I was writing the code in Visual Studio Code as usual but then when it came time to run the code, the renderer in-built into the system software always broke opening in a 1x1 square then when manually readjusted, there’s no game. Didn’t matter what was in the code, it just never worked. I was there stuck for hours until I downloaded IDLE and the code ran seamlessly. Apple is how rich, and their Python is just broken.

Now for an issue that was my fault, I had a black screen bug where the game was launching but nothing was happening. Yeah the game logic code was just not properly indented under the game loop. Just fixed that with indenting everything under “while running” in the code.

The restart button also didn’t work for a while because the full game loop was outside main(), relatively simple fix here just move the logic inside main().