

Jack Wagner

## Artist Statement

The rise corporate sponsorship in video games how has now reached a point where it can be seen in almost every AAA game being released today. Fortnite has skins that range from Rick and Morty, to Darth Vader. My animation is an art piece centered around mocking microtransactions in Video Games, posing the main question of, what if corporate sponsorship occurred much earlier in video game history. I did this by making a “skin selection” scene in popular game Super Mario World 2 on the GBA. The video shows the “player” picking a skin and then starting the game. The video is then just normal Super Mario World 2 but replaced with Naruto then the course ends and a Loot box is given to the player. I focused on trying to make this as realistic as possible so I implemented animations for specific game objects and character models from other games.

How I created my video was by pulling **many assets** from Mario, and other GBA games. In total I had around 54 individual character or item frames which included Apples, Coins, Lucky Blocks, Yoshi Coins, Chun Li, Spiderman, and Naruto. I cut and scaled the individual animation images from the normal 300 x 140 GBA screen size to 1080 for testing then to 4K for production. For the super mario world background I recorded a video of the game on my emulator then used VLC media to go frame by frame screen grabbing each one then I placed them side by side in mspaint until they formed the full course, which I scaled to 4k.

There are two main files that run my code the first one called player which has two objects in it the first is the animation object which can apply an animation at any given pygame rect. The animation class allows me to create an object with the frames, and time in between frames as array arguments. The update method would blit the current frame onto a given rect and change the frame depending on the FrameTime array. The other object in my player file was my player class which controlled the three animations I needed for the player, Idle, Run, Jump. The update method takes in an animation num either “idle”, “run”, “jump” and would blit that animation to the player rect. The other file was the main skel.py this controlled the background as well as the jumps from the player. This works by just changing the rectX value over time. I also tried making a camera system that could follow the player as they jump, so it again mimics the actual game. Overall, I think that I did well on this assignment, and I am proud of

the work. I just wish that I was able to get it in on time.

