***Angel Castillo’s CAP 4720 Project:***

***Rock Paper Scissor Simulator*IMPORTANT NOTE:**  
There’s a bug where the victory/defeat animation won’t play after the CPU makes its selection sometimes for reasons unknown. Just ***double click*** play again and try again.  
  
**RUNNING THE PROJECT:**  
1. Open an instance of a terminal window inside the project directory  
2. run "http-server . -p 8000"  
3. open "http://127.0.0.1:8000/index.html" in browser  
  
Alternatively, you can view it at <https://angelcastillo407.github.io/CrashRPSGame/>  
  
The original idea for this project was to have a character select screen displaying models as you hover over them and pit the 2 you choose in head-to-head competition in rock paper scissors. I couldn’t achieve this due to time constraints and an unforeseen learning curve trying to rig 3D models, to then giving up and using a pre-rigged model (in the process in learning to rig, I was able to fix some issues with the model I did use).   
  
The way the simulation now works is simple: you are presented with 3 buttons at the top of the screen representing your choice in either rock, paper, or scissors. Once you ***double click*** an option, you the user are pitted in head to head competition with the computer, represented by a 3D model of Crash Bandicoot. An animation will play representing what the computer has selected as their option (randomly assigned each time you play the game). If you both choose the same option, you tie, but are shown the loser animation and music (no victories are handed out here). If you emerge victorious, a new animation will play of the 3D model holding his arms out in agony of defeat as well as the title changing to a victory screen with triumphant music. If you are defeated, a new animation will play of the 3D model dancing to ridicule you as well as the title changing to announce your defeat with somber music. To play again, you can then ***double click*** the play again button at the top of the screen.   
  
As for the HOW this project was done, I used a program called Blender to animate 3D models and export them as gtlf files. Then, using Threejs’ gtlf loader, I passed in the file to display it in 3D space. To start the animations, I used a Three.AnimationMixer to handle everything. I ran into problems with attaching several animations to a single gtlf file, so I had to export each animation as its own separate file. The rest of the project is if-statements spread throughout to decide what animation to play based on the difference between the computer’s choice and the user’s choice in the rock paper scissor competition.