Andey Ng

TP1 Write Up

* + **Project Description** [5 pts]: The name of the term project and a short description of what it will be.

Andey’s Arcade will consist of multiple games such as a “Streetfighter” version of *The Last Airbender* that the user can play/control with the Kinect.

* + **Competitive Analysis** [5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.

*The Last Airbender* is every millennial’s childhood favorite. *The Last Airbender* is currently developing their live-action animation; however, there are no interactive games to help hype their release to the public. Because there are currently no *Avatar: The Last Airbender* games that allow players to become their favorite character, this project will fulfill every child’s dream of waterbending in the showers or firebending in thin air. This project will mimic Streetfighter’s game; however, there will be different powers and players.

Many of the games we grew up playing are played through controllers; however, there is a lack of games which are virtually controlled by a user’s motions. By developing an arcade which players can choose their favorite games, they are able to interact with the games and *become* their players.

* + **Structural Plan** [5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or objects.

The game will require multiple objects, mainly a CPU which will have AI implemented to evaluate the player’s moves and become stronger and defeat the user. I will need to place parameters/rules of the game to assure that the AI can make proper calculations.

The player should be controlling a sprite displayed on Pygame through the player’s motions on the Kinect.

For other games in the arcade, the player should control the pieces with the Kinect through drag-and-drop motions.

The changing screens require me to implement a “splash screen” that can switch scenes.

* + **Algorithmic Plan** [5 pts]: A detailed algorithmic plan for how you will approach the trickiest part of the project.

The most complex part of my project is to understand how to use the Kinect to interact with my Pygame screen. Because the Kinect has been discontinued, understanding the documentation of how to make the player control the Pygame objects with the Kinect will be tedious.

Training the CPU to defeat the user through AI will be difficult as well.

* + **Timeline Plan** [5 pts]: A timeline for when you intend to complete the major features of the project.

11/24/18: Finish Pygame character controlled by keys and Kinect Motions

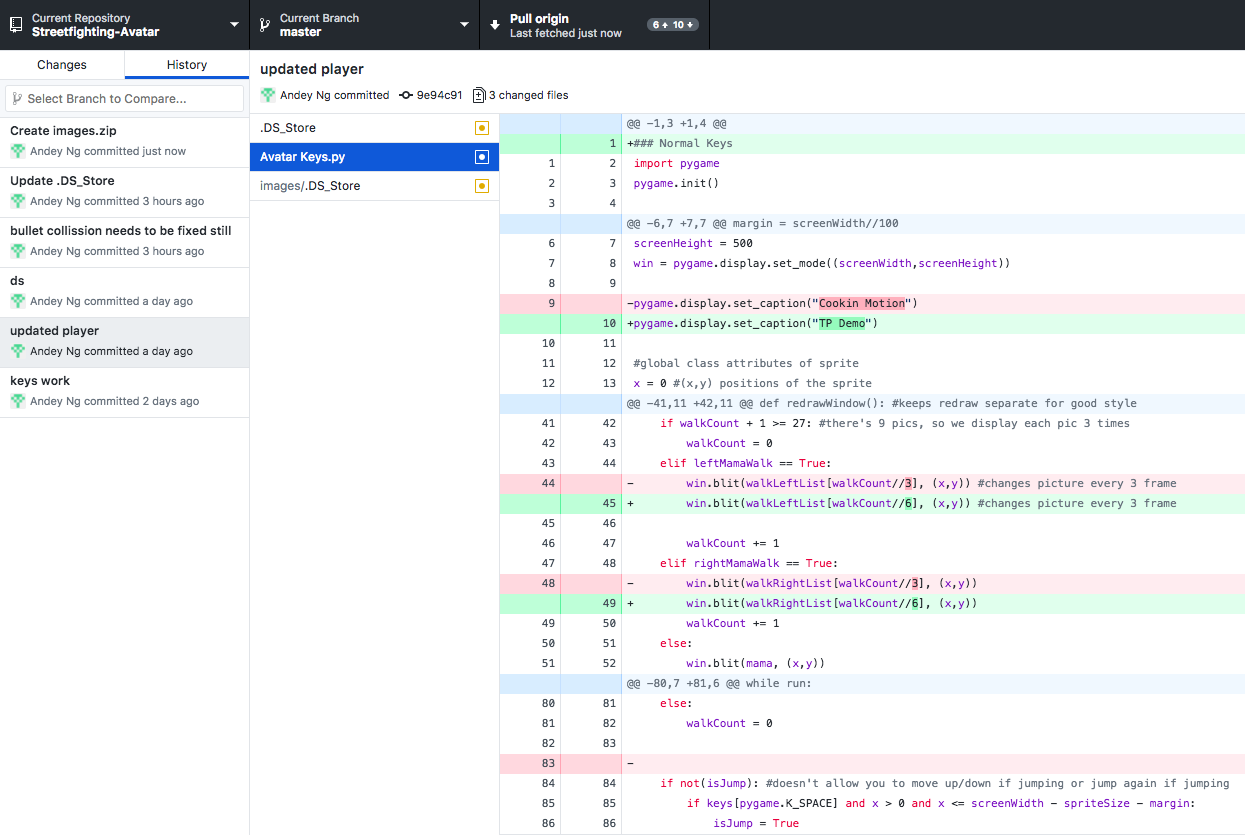
11/30/18: Kinect motions can control sprites on the screen of Pygame

12/07/18: Splash screens, multiplayer and AI half way done

12/12/18: AI completed, Video finishing up

* + **Version Control Plan** [3 pts]: A short description and image demonstrating how you are using version control to back up your code. **You must back up your code somehow!!!**

I am using GitHub to upload all my files.



* + **Module List** [2 pts]: A list of all external modules/hardware/technologies you are planning to use in your project. Note that any such modules must be approved by a tech demo. If you are not planning to use any additional modules, that's okay, just say so!
    - Microsoft Kinect sensor
    - PyKinect2
    - Pygame

**TP2 Update:**

* The Kinect is semi-glitchy and cannot detect motion fast enough, so the way the moves are controlled have been altered to simple claps
* The shields are not very feasible, because the Kinect cannot update in time for the bullet gets to the player and the player motions to shield.
* The opponent is currently using key pressed to verse the player

**TP3 Update:**

* The video is small so it’s difficult for the player to see large movements of what is happening in the game.
  + To solve this, an image will be projected on the player to signal what move he/she made.
* GUI/UX:
  + Created an interactive screen where the player can choose which character he/she wants to play as.
  + Placed the image of the avatar costume on top of the player so the player becomes the avatar.
* AI for the CPU:
  + Created game logic with the rules are stated below

**Game Moves:**

* Charge = **Free**
  + Serves as ammunition. If the player wants to shoot or mirror aka play offensive, they must charge.
  + Charging is a vulnerable state, as the player has the risk of being shot at, but also has the potential to shoot later.
* Jump = **Free**
  + Jumping avoids any bullets if the CPU shoots at the player.
  + Nothing is gained, so you won’t advance in the game if you continuously jump
* Shoot = **1 charge**
  + The player must choose to shoot strategically in the hopes that the opponent is vulnerable at that moment.
  + Offensive move
* Mirror = **1 charge**
  + If the player “mirrors” and is shot at, the player can shield themselves and cause the CPU to lose points.
  + The player must be strategic because the player must charge before using this.
  + This only benefits the player if the player is shot at.
* Big Bomb = **5 charges**
  + Must have 5 charges which is difficult to attain, as the player must be vulnerable 5 times to receive a “big bomb”
  + The big bomb causes double the amount of damage