PROJECT PROPOSAL

Michael Elrod

Part 1: Game Design

- The game I am creating is a continuation of the game I started for the initial engine project. It's a 2D platformer
- The premise is the main character must traverse the platforms to get to the next level
- Dr. Adkins said it was ok for the gimmick to be something like, my game has a lot of assets and animations in it so I will be using this
- Game mechanics will be left to right movement, jumping and bouncing off enemies
- Collectable coins will be in the game for the player to get

Part 2: Development Design

- The "modules" for the game are discussed in the README on GitHub for the project here: https://github.com/Michael-Elrod-dev/Zombie-Lab
  Its quite long so I will try to summarize here.
- The Level is the core of the project. It handles object initialization, terrain setup, asset paths, camera view, and controls the state of the game and player. This file also tracks all collisions between the player sprite, the terrain, enemies, and coins. The update function within Level.py which runs every frame and updates all sprites based on user input and camera movement.
- The Player class handles the player's status by getting user input and updating the status of the player character based on its velocity. This file also calls the classes that import game assets.
- The Enemy class handles enemy movement and animation changes
- The Tile class handles the movement of tiles with the screen as well ass animating tiles spacecs like the coins
- I have also added a UI class for the interface, a Decorations class for background effects and reworked the Settings file
- Added an Overworld class that generates and overworld where the player can choose a level to play and see their progression
- I also have a few more classes that act as support for the engine. For example a Settings class that stores information about the game like the level map design (CSV) and the screen size.
- A visual diagram can be viewed on GitHub as well https://github.com/Michael-Elrod-dev/Zombie-Lab
- The user interface will be a health bar, coin collection, and a possible pausing option.
- The main technical challenge I anticipate for this project is creating/using the assets I have. For example I can easily create a csv file to generate a map with blank tiles but I intend to import tile assets for the map. To handle this I'll be using online tutorials for better understanding of the concept and using the application Tiled to generate a csv file that is tied to specific sprites
- A lot of the information I'll be using to help with this project will be from Clear Code on Youtube

Part 3: Division of Labor

- Michael Elrod – responsible for literally everything
- Rough time estimate of 40-50 hours
- Only conflict here is that I'm lazy and have a gaming addiction. No way to handle this other than motivating myself to actually get to work.

Part 4: Timeline

- Milestone 1: March 29th - All Completed
  - Michael – Get the map tiles generated in game by using a csv file generated from Tiled
  - Michael – implement Enemy and Coin sprites and their automated movements in game
  - Michael – Restrict enemy movements by using constraints in csv file
- Milestone 2: April 12th
  - Michael – Get my own sprites for enemies instead of stolen ones
  - Michael – Configure a UI system of some sort (Coins, Health, etc.)
  - Michael – Implement death animations for enemies
  - Michael – Implement player damage system
  - Michael – Implement a win/death state
- Milestone 3: April 26th
  - Michael – Implement SFX/Music
  - Michael - Complete game documentation
  - Michael - Check for bugs
- Game Presentation: May 4th
  - Michael - Learn how to make an executable
  - Michael - The game will be in a playable state for presentation

Milestone 1 Notes: https://github.com/Michael-Elrod-dev/Zombie-Lab

During this milestone I added the following implementations:

Enemies: I've imported some stolen sprites as a place holder for enemies. I also implemented their movement system to have them walking back and forth. This movement is restricted by checking for collision with invisible tiles that are located in my CSV file

Tiles: Using the Tiled software I created a 2D map and generated a csv file for that map. To decorate it I used some stolen sprites. This implementation takes a PNG of the tile set and splits the image into 64x64 sized pieces and places them where their respective numbers are according to the CSV file
Coins: I added some rotating coin sprites for fun. Currently they just spin and cant be interacted with but soon the player will be able to collect them!

Updated Timeline:
Nothing got moved up or down in the timeline but I did add some new features because I had some extra time. For example, I added some more in-depth UI stuff and added a possible earlier timeline for SFX. Also I added a death/win state condition to the third milestone.

Challenges:
Was massively challenged by trying to figure out how to work the Tiled software and get it to be configured properly for importing it to my game. To simplify this process instead of making several CSV file layers for different objects I have everything compressed into one singular CSV file. Definitely isn't ideal because this means no items can be stacked on game initial game state. If I wanted to add decorations like grass or trees or something, then my current implementation wouldn't allow that. The CSV file would need to be separated into layers. If I don't change this at a later time then the timeline wont be affected but the final "look" of the game will suffer as it will have a more plain appearance.

Milestone 2 Notes: https://github.com/Michael-Elrod-dev/Zombie-Lab

During this milestone I added the following features to my game:

- A UI system with a health bar and a coin count
- An overworld where the player can choose which level to play and see their progression
- A damage system for the player
- A death state that triggers respawning
- A win state that progresses you to the next level
- Reorganized the file structure to better group implementations with their respective classes for animations and importing
- Reformatted the way I take in a CSV file to allow for multiple layers of object on top of each other
- Created new assets for the games overworld, end goal and enemies using Aesprite
- More to come 😊

Updated Timeline: Nothing necessarily got moved in the timeline, but I was able to add more details to it for the current milestone

Challenges: The biggest challenge for me this time around was the death animations for the enemies. So much so that the bug I encountered is still present and probably will be by the time the game is finished. I have an implementation that is supposed to run a set of PNG files on screen looping just one time when an enemy dies to simulate their death. However, this animation plays several times before stopping. To be honest I'm starting to believe that only God knows why and a mere mortal like myself is not worthy of this knowledge and I am meant to be ignorant forever.

Milestone 3 Notes: https://github.com/Michael-Elrod-dev/Zombie-Lab

During this milestone I added the following features to my game:

- SFX and music
- Finally fixed the bug that was plaguing all my animations
- Learned how to create an executable file

Updated Timeline: I had to remove the "create new levels" portion of my timeline for this milestone because of the amount of time it took me to fix the animations bug

Challenges: The biggest challenge for me this time around was again, fixing the animations bug. This alone took me roughly a whole week and was fixed by mere coincidence.