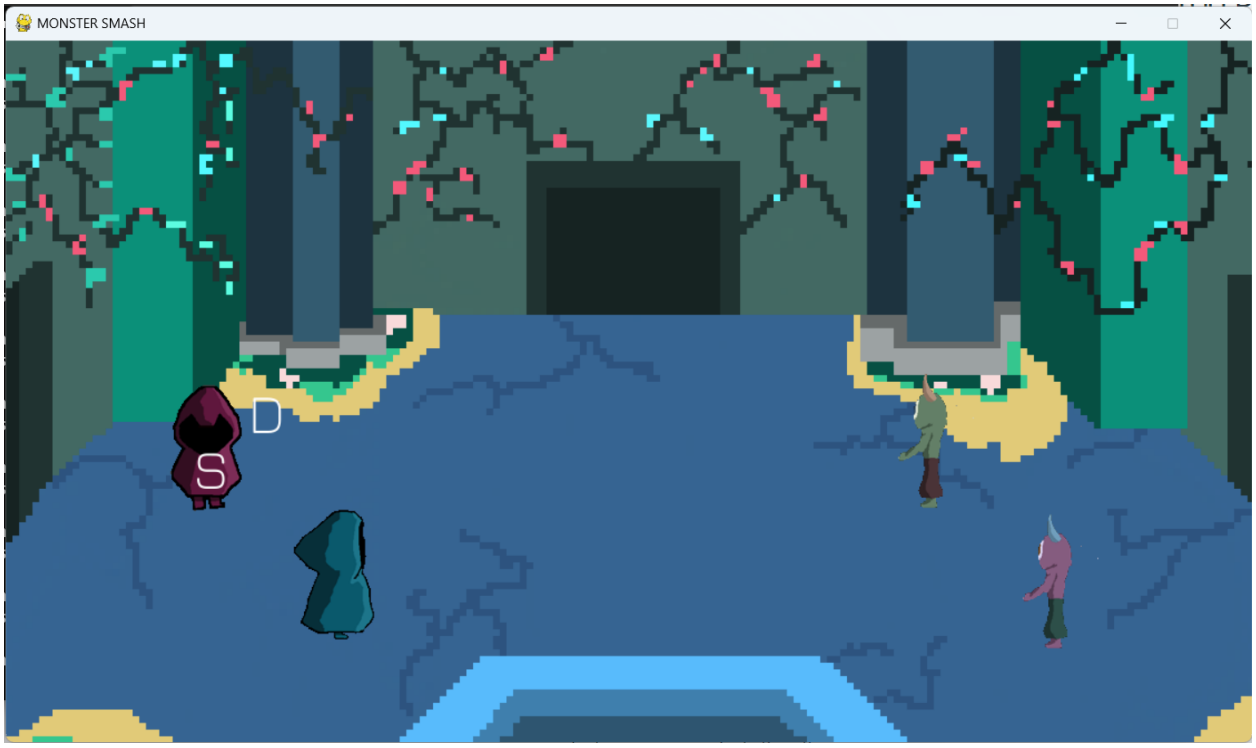
Monster Smash

By Sean Shipley and Sydney Fleming

**PART 1: INTRODUCTION**

Link: https://github.com/sshiple/RPG-Project

The name of our game is Monster Smash. Below is a picture of it.




DOWNLOAD INSTRUCTIONS:

1. Download the zip folder
2. Extract it
3. Open a terminal in the extracted folder
4. Type “python3 rpg.py”
5. The game should launch

VERSIONS:

Python: 3.11.3

Pygame: 2.3.0

CONTROLS:

Space: starts the game

W: makes the upper character jump during enemy phase

D: makes the upper character attack the upper enemy during player phase

S: makes the upper character attack the lower enemy during player phase

Up-Arrow: makes the lower character jump during enemy phase. Also makes the lower character attack the upper enemy during player phase.

Right-Arrow: Makes the lower character attack the lower enemy during player phase

**PART 2: GAME DESIGN**

- The gameplay loop has enemies coming at you and the players must evade them by jumping out of the way, while also fighting them. If the enemies hit you then you lose health. If the player gets hit too many times and health runs out then it is GAME OVER.
- The core mechanics would be the evade mechanics that were put into the game.
- The gimmick of the game is how it is an rpg-like game with dodging involved.
- This game differs from other rpgs as the attacks are simplified and you now have the ability to dodge them.
- The game uses its game engine well with each screen serving its purpose.

STORY

- Two travelers come up to the fortress to slay monsters for attacking an innocent village.
- The main characters are two mysterious cloaked figures trying to avenge the village.
- The antagonists of the game are the monsters themselves. They see the main characters as threats, so they attack them.

PLAYER EXPERIENCE

- The most prevalent emotion that they should feel is stress as the enemies can change their attack patterns, so they must dodge in time.
- The players will face confusion at first, but it will be overcome with recognizing the attack pattern.
- As the players win the battle, they will start to see enemies defeated, giving satisfaction that you're almost there.
- While playing the game, there's a positive feedback loop. The calmer you are, the easier the game will be.

- The battle background is detailed, giving the fortress some history.

**PART 3: GAME DESIGN CHANGES**

Originally, our game was going to be grander with an overworld and many types of enemies with completely separate attacks. We essentially ran over time and ultimately suffered from feature creep. We had to scrap most of the ideas we had so that we could get the game out on time. However, the game mechanics with dodging and the gimmick of the game being an action-like rpg mostly stayed the same. We just ran into major time constraints since grad work for other classes took time away from this project.

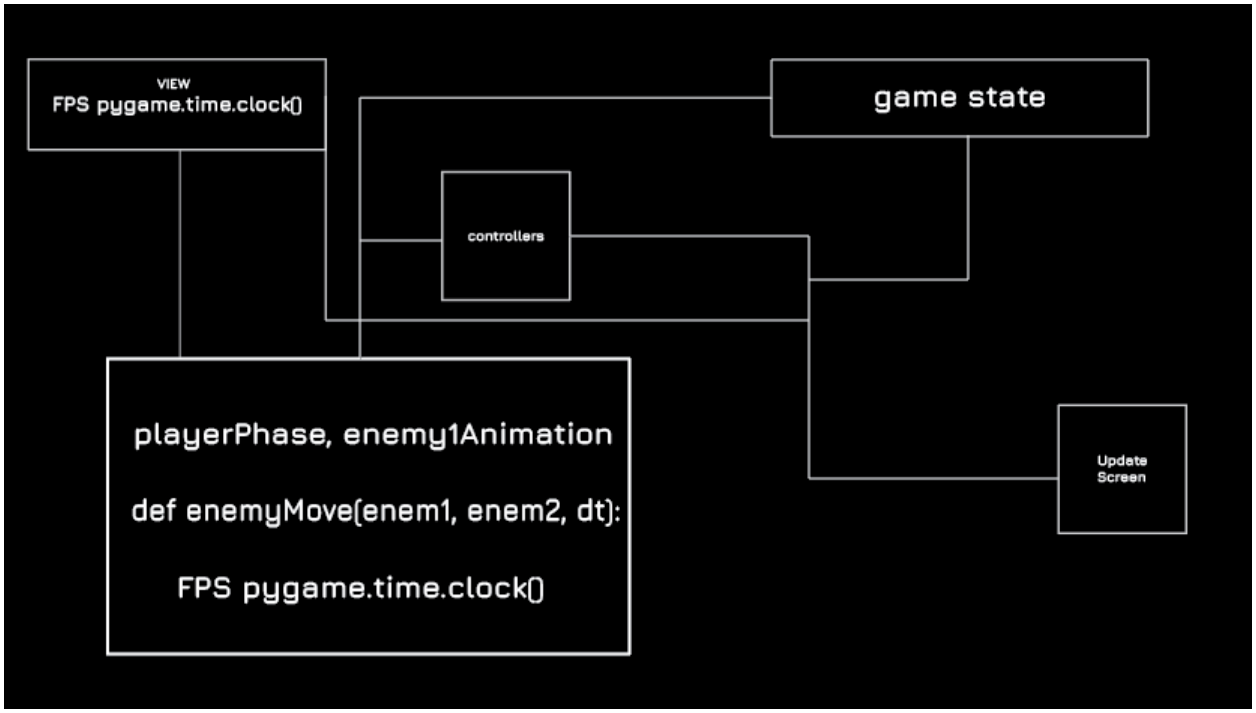
**PART 4: GAME DEVELOPMENT & DOCUMENTATION**

Some major functionality methods are the player movement classes, which have player phases, and character jumps. This also includes enemy movement.

The game receives input for the keyboard that allows the characters to move when prompted.

State of the game is determined by the game, and whether or not the game is over or needs to be restarted.

The updating of the board is handles at the end of the loop



Originally, there was a bug where if the player jumps right at the end of the enemy phase, then their jump stopped and they were floating. This was fixed to just teleport the players to their default locations at the end of enemy phase. Also, there was a bug where the defeated enemy sprite would happen too late, so I made it happen right after the player's attack and not at the start of the enemy's attack.

**Part 5: GROUP MEMBER ROLES, TASKS, AND PERFORMANCE**

The division of labor remained the same throughout the duration of the project.

Timeline of workload and development

Milestone 1: March 30

- Sydney: I collected some reference for the environment and created sketches for us to choose from to then be quickly made in a pixel art creation software
- Sean: Mostly worked on the battle scene. The game mechanics will remain the same, but I need to make more progress. Some things might be removed from the final version, such as the overworld.

Milestone 2: April 18

- Sydney: All level designs have been implemented and finalized, all artwork is on its final iteration ready for final playtesting and debugging. The artwork still needs hammering down, with coinciding color palettes with other characters in the game.
- Sean: Still worked on the battle scene, but I still need to make more progress. However, I implemented the jump and added a second character to be controlled.

Final Game Submission: May 3

- Sydney: The character sprites and backgrounds are all finished.
- Sean: The battle scene is done with a good amount of attack combinations between the characters and enemies.

Final Exam Presentation: May 4