CISC 4900 – The Unity Solo Game Project

Contact Information:

Alexander Rodriguez

Email: alexanderrrodri@gmail.com

Role – Developer & Designer

Supervisor:

Murray Gross

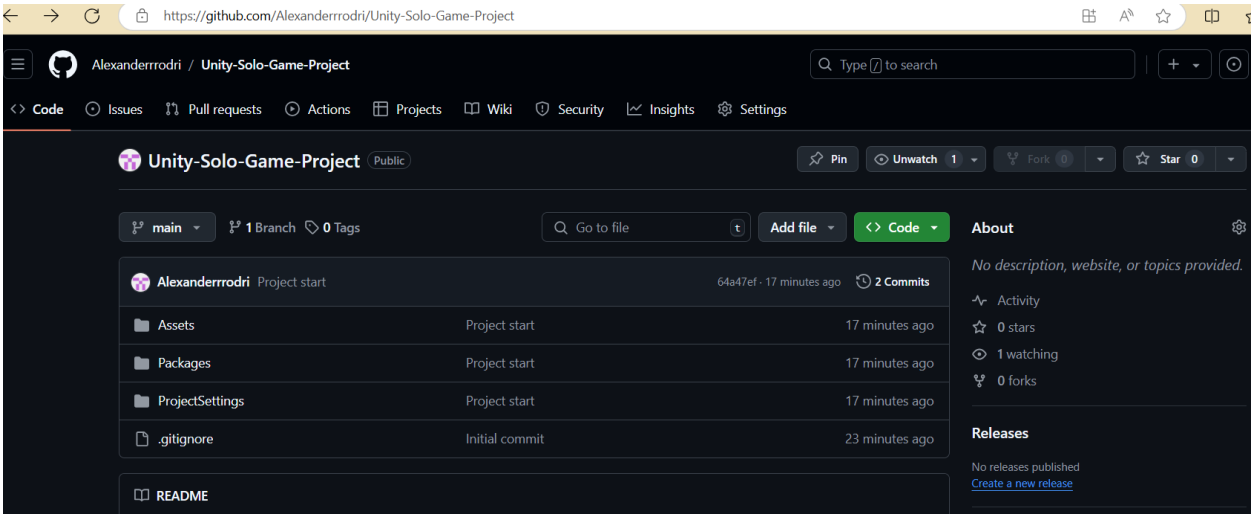
Email: magross@sci.brooklyn.cuny.edu

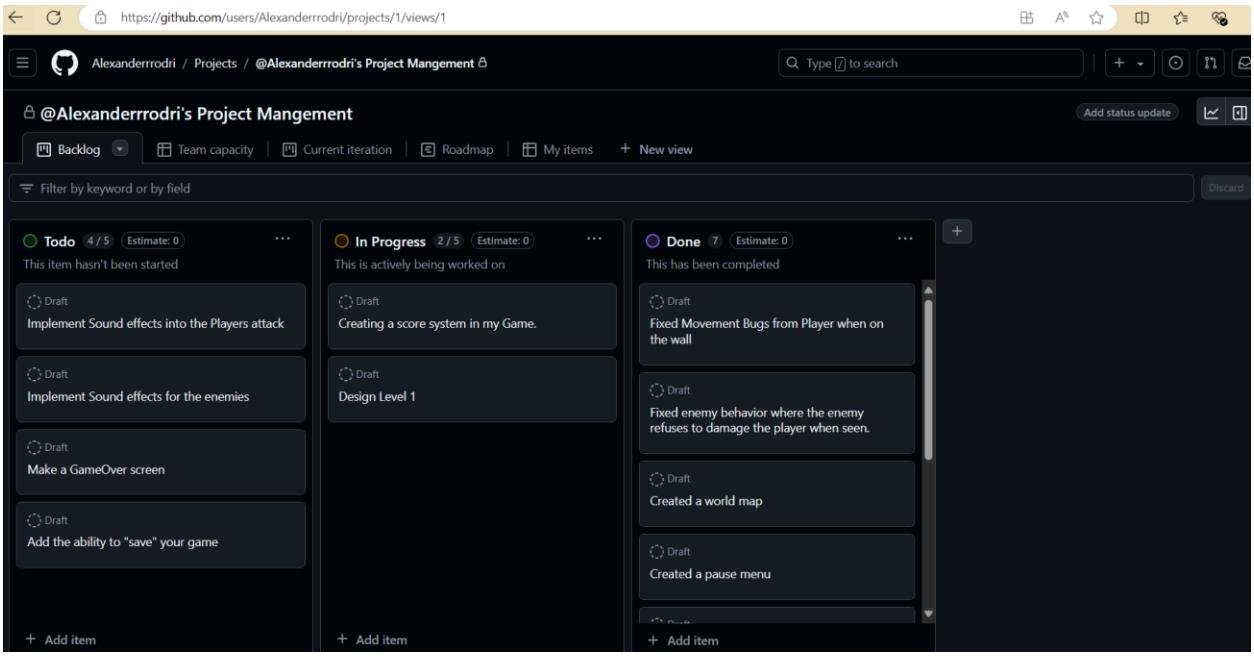
Abstract: My game will be an action-packed 2D platformer that combines dynamic combat, fluid movement, and strategic level design, immersing players in the role of a skilled female warrior on a quest through perilous landscapes. Players will engage in fast-paced battles against a variety of enemies, utilizing a blend of sword attacks, dodges, and special moves that require timing and skill. Each level presents unique challenges, from treacherous platforms to hidden secrets, encouraging exploration and mastery of the game's mechanics. The project aims to deliver a fun gameplay experience with intuitive controls, smooth animations, and retro style of art, all brought to life using the Unity engine. By focusing on finely-tuned combat systems, intelligent enemy AI, intricate level design, this game would surely be appreciated to people who like the platformer genre.

Tools: *Unity Game Engine (version 3.5.2), C# (for writing scripts), Windows 11(Version 23H2), GitHub*

Repository: [Alexanderrrodri/Unity-Solo-Game-Project (github.com)]

Project Management: [Backlog · @Alexanderrrodri's Project Mangement (github.com)]





Game Flow:

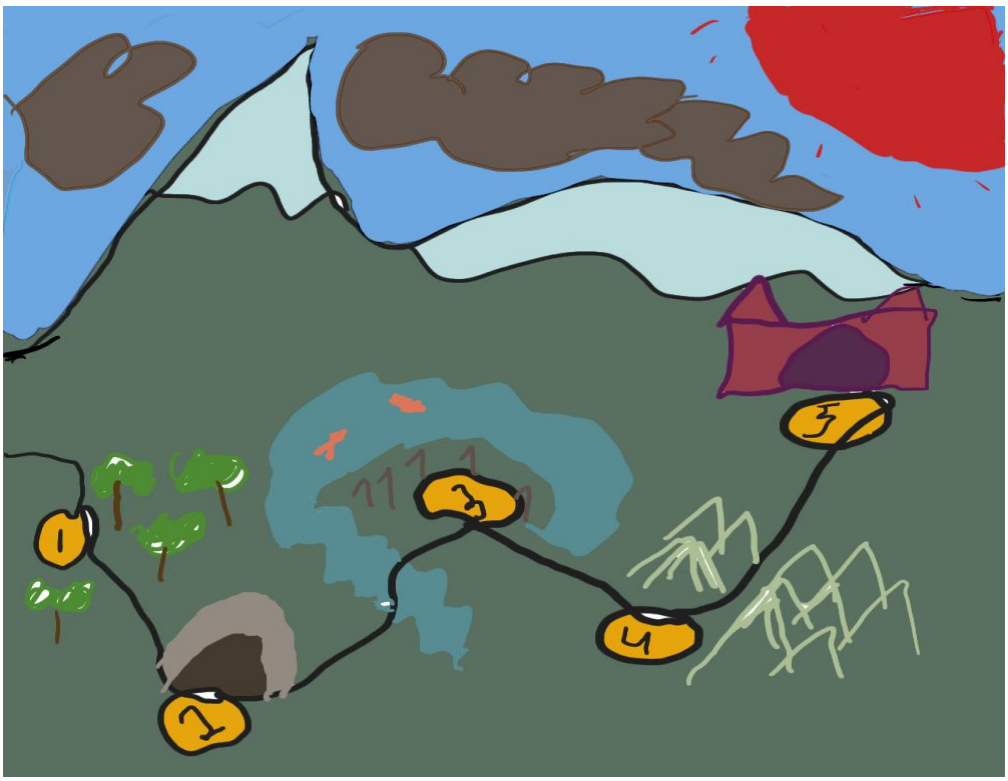
Title Screen → World Map → Stages → Goal

Then it repeats itself from the world map till you beat the game.

There will be a settings menu that will make players optimize their gaming experience.

World Map → Level One → Level Two → Level Three → Credits

World Map Concept Art:

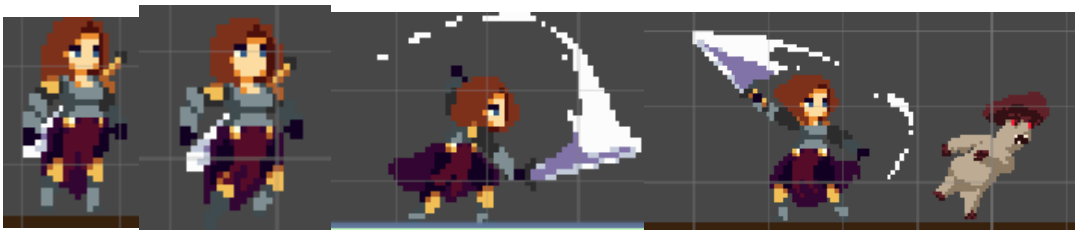


Updated World Map Design:

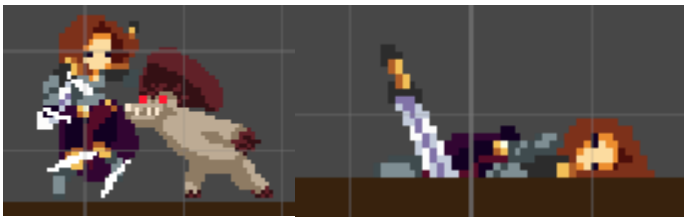
At least 3 stages with different environments to traverse and enemies to fight. The forest will be a basic level to help players get used to the game,

with some secrets in the level. I would like to create a cave level that has dynamic lighting, and the player needs to find light sources to find a way out of the level, would like to make a mountain level that tests the player's platforming capabilities. Would like to make a water level where the player either must swim or go up waterfalls to proceed through the level. Finally, the last level will be the most challenging level the player will ever face. With many enemies to fight, difficult platforming, and the end a final boss to end the game. In doing this, I will show that I have the capabilities of making a complex, well developed 2D platformer with unique gimmicks and fun level design.

Concept on how the game will look when in a stage:



Idle → Jumping → Attacking → Fighting Enemies



Taking Damage → Game Over (Hp = 0)

The player will have full control of your character's movement, and they will have multiple ways on how they may complete the stage. I would also like to create a scoring system when the game is nearly finished. This would make players want to play some stages multiple times and have the motivation to get better scores by beating enemies, not taking

damage, or how fast you complete the stage. Especially if I give out good rewards for better play in the levels.

Will add a way to save your game so your progress doesn't all get lost when you get a Game Over or leave the game.

Tentative Schedule for the Remainder of the Semester

- ***September**: Continue refining core mechanics, finish basic enemy AI, and implement basic levels. Including some player feedback systems like health and damage.*
- ***October**: Focus on level design, art integration, and additional enemy types.*
- ***November**: Continue to work on level design, polish gameplay, fix bugs, and conduct playtesting sessions.*
- ***December**: Finalize game, prepare documentation, and submit my project. Prepare for publishing on itch.io.*

Use Cases

- ***Use Case 1: Combat with Enemy** - Player encounters an enemy, engages using combat mechanics, and either defeats the enemy or takes damage.*
- ***Use Case 2: Level Navigation** - Player progresses through the level, interacts with obstacles, and reaches the exit.*

- ***Use Case 3: Saving Progress*** *- Player saves their high score in the game for each level. So when returning back to the game, it will remember their high score.*

September Log: I have mostly completed all the basic mechanics so far. I still have work to do by making a score system for the game and collectables in each stage. I want to do this so that there is reason to replay the stages to get a higher score by beating enemies, losing the least amount of health, collecting all the collectables, etc. I am also making my first level while testing all these features out. It will be basic but a great way to teach the player the basic controls of the game. Starting October, I will have done the level 1 stage by now and basic mechanics done which gives me time to finally start creating more complex dynamic stages.

October Log: I have completed 2 levels for my game that provides different gameplay mechanics and different enemy type encounters. I also provided powerups in the levels to help the player through the levels. I will start making the final level next month and try to make a final boss for the player to defeat. Once defeated, they beat the game and show the credits. Progress has slowed down a bit because of the other classes I need to take care so I need to manage my time better next month to get more things done for my project.

Closing Thoughts:

I will continue and dedicate my project for the whole semester. I will take responsibility for making progress in my project every day and make something that I can be proud of. This is my first time making a game like this and all by myself. But I believe by the end of the semester, I'll be able to make a finished project that will meet the expectations for my CISC 4900 class.