Project Proposal

Name: Ella Lee

Section: N

* **Project Description**
* Project name: Woodboy and Metalgirl
* The project is inspired by Fireboy and Watergirl that I used to play during childhood. Woodboy and Metalgirl is a modified version of the original game, where two game characters need to avoid obstacles while collecting gems and go through five different doors. In the end of the fifth door, they will reach to a room where they can trade their gems with their desired products. There will be four options for users to trade with the gems as the prize for completing all the stages.
* **Competitive Analysis**

A screenshot of a video game

Description automatically generated with medium confidence

* My project is a modified version of Fireboy and Watergirl, where two characters work through together to escape the maze. I have found two similar projects: 1) <https://github.com/RyanChakov/FireBoy-and-WaterGirl> from Github and 2) <https://www.pygame.org/project/4562> from Pygame.
* They are similar to my project in three main ways:
  + It is a collaborative game between two players who use different keys to move around the game characters. In order to enhance the collaboration between users, one player will use keys of ‘a’, ‘w’, ‘d’, and ‘s’ for ‘Left’, ‘Up’, ‘Right’, and ‘Down’ respectively, while the other will use conventional arrow keys to move around the character.
  + Common obstacles are hanging woods, boiling water, buttons, and switches.
  + Both of the players need to get to each corresponding door to reach the next level.
* Yet, it is different in several ways:
  + Whereas the original game requires two players, since I am going to implement game AI, I will have two options, which are 1) two players and 2) one player and a computer.
  + Also, characters’ traits are clearly distinguishable from the original characters.
  + Some obstacles (such as platform block that lights on and off depending on where characters are) are differentiated from the previous ones.
  + Also, I have incorporated the trading game at the end of the final level, where users can trade their gems (coins) with gold.
* **Structural Plan**
* All the objects in the game such as characters, obstacles, and platforms are created inside classes in order to organize and classify numerous attributes (Most of them will be called in Timerfired(app)).
* The functions will be organized in different files based on their purposes. For example, if some are to draw characters and obstacles, they will be put into a filename called “drawClasses.” In other example, if some are helper functions, they will be stored in a file called “helperFunctions.” These files will be imported to the main file where I call runapp.
* Also, I will have three major folders called TP1, TP2, TP3 deliverables in order to see the progress made during the term project.
* **Algorithmic Plan**
* The character-obstacle interaction is a key to this project. Hence, enhancing a user experience with various obstacles on different platforms with varying difficulties is a main concern.
* On that regard, for a single user, I will implement game AI, where the computer is working with the user simultaneously (as if another user is existing) and supporting the user to reach the final level. This is one of the trickiest parts I am dealing with in this project.
* However, most of algorithm complexity comes from a random map generation including random obstacle installation. I will incorporate a maze generation method to decide up possible paths and place objects (obstacles) smartly on each platform so that it is always solvable yet challenging to tackle.
* Possibly those algorithms can be considered: Prim’s Algorithm, DFS, Kruskals
* **Timeline Plan**
* By Nov 18th (TP1 deadline):
  + Design main components such as doors, gems, background, characters, etc.,
  + Move around characters
  + Create interactions between characters and obstacles
  + Consider how to incorporate maze generation to random map generation
  + Hardcode four-five levels to test out the difficulties and see if they are solvable.
  + Character’s movement (especially make them jump by using ‘up’ and ‘w’ keys)
* By Nov 23rd (TP2 deadline):
  + Implement random map generation
  + Figure out how to move characters smoothly (like an animation)
  + Debug minor issues
  + Polish characters and other elements into user-friendly design
* By Dec 1st (TP3 deadline):
  + Based on user-study-a-thon, change some codes and incorporate some music if possible
* **Version Control Plan**
* I’ll upload all the python files to Github as well as Googled drive and icloud, in case I lose data in my laptop.

**Google Drive & Icloud**

**Graphical user interface, application

Description automatically generatedGraphical user interface, website

Description automatically generated**

* **Module List**
* I currently do not plan to add any modules, but I am considering incorporating music with Pygame if available.
* **Storyboard (attached to the next page & present in CMU 15112 File as a pdf)**

Graphical user interface, application, Word

Description automatically generatedGraphical user interface, diagram, application

Description automatically generated

**TP2 UPDATE (Nov 23rd, 2021)**

After implementing the random maze generation, I decided to change the platform setting from three-level platforms to a maze-like platform.

Along with this platform, the functionality of obstacles changed: switches and buttons are for collecting coins, and the doors open when characters consume all the coins (gems).

If characters collide with walls, they die. (This is a new rule I’ve added in the game).

Also, I’m implementing the hint system on top of the maze generation for users to solve the game. I have created a side bar that indicates how many coins are consumed and whether if switches are open or not, and I have created a button for hint so that if users press it, they could see arrows pointing towards the solution. As an alternative, I could draw multiple rectangles by simply using a list of possible locations.

**TP3 UPDATE (Dec 1st, 2021)**

There are a lot of changes made since TP2 until TP3 submission:

1. I created a loop with reset(app) so that users can restart the game after characters die.
2. I used pygame modules to add background music and sound effects such as door open when finishing the game, collecting gems, pressing buttons, turning the switch to the right side, and characters’ deaths.
3. I have implemented the random level generation by increasing the difficulties of the maze and the number of coins and obstacles. Now, there are three different levels from easy, medium to hard.
4. Accordingly, on the instruction page, I have added three buttons for each level.
5. Also, I have revised some of the rules, after the user-a-thon, so that players get a clear idea of how to play the game.
6. After some peer feedback, I have adjusted the speed of the obstacles from fast to slow as well as made the difference between the normal button and the button pressed more distinguishable by drawing out the latter part.
7. I added a lift, which was absent until TP2, that basically works as a feature like hangingwoods.
8. I have fixed the bugs where an element is popped from the empty list by adding Boolean statements.