TP 1 Deliverable

Ryan Bao (ryanbao)

11/28/2020

Contents

1 Project Proposal **1**

1.1 Project Description 1

1.2 Competitive Analysis 1

1.3 Structural Plan 2

1.4 Algorithmic Plan 2

1.5 Timeline Plan 3

1.6 Version control plan 3

1.7 Module List 3

2 Storyboard **4**

List of Figures

1 Example character design 1

2 Image of Github page with version control 3

3 Storyboard showing basic enemy and player interactions . . . 4

4 Storyboard showing player ranged attack interactions 5

5 Storyboard showing player melee attack interactions 5

6 Storyboard showing planned crafting interface 6

7 Storyboard showing planned map creation interface 6

8 Storyboard showing planned character selection screen 7

1 Project Proposal

1.1 Project Description

My term project is called 'The Box Game' and is an open world game with random, seed-based map generation. The game will also include a grid-based crafting system, similar to that of Minecraft, but also with an RNG based system for the 'quality' of the items. The game movement will be based around the WASD keys, with side scrolling chunks when you get to the edge of the screen. Combat systems will be based around mouseclicks, with a combination of ranged and melee attacks. The enemies will pathfind based on slight modifications of the astar algorithm.

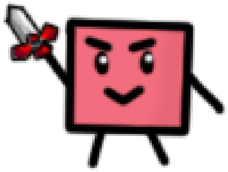


Figure 1: Example character design

1.2 Competitive Analysis

The closest games to this term project are Genshin Impact, a recently released RPG game, and Minecraft, a relatiely old open-world survival game. This game will take many of the element of Genshin Impact, such as a similar combat system and, if time permits, a similar art style, although it will be a simplified, top-down, 2 dimensional version of the game, similar to the Pokemon games. This game will also incorporate an RNG-based system for crafting, similar to Genshin Impact cooking mechanics.

This game will also be similar to Minecraft in the terrain modification, crafting system, and random seed-based generation. This game uses a heavily simplified version of Minecraft random chunk-based generation, where each chunk is generated separately and then placed into a 2-dimensional list that represents the map. Similarly, the crafting system will be grid-based, similar to that of Minecraft, although the exact details of such a system have yet to be hashed out fully. The game also incorporates somewhat modifiable terrain, with obstacle destruction, and, time permitting, a limited building system, similarly to that of Minecraft.

1.3 Structural Plan

The majority of the structure of the project has already been created. The code will be organized into different code documents, based on reasonable categories. For example, one such category is enemyCode.py, which includes the enemy class, an enemy movement function that can be placed into timerFired in the main code, as well as enemy generation functions that help map generation, as defined in mapCode. The rest of the project will be organized similarly. Most of the necessary objects have been created already. The last class before MVP would be a class for crafting recipes. The code is designed to have as little as possible in the main code class. Ideally, only the functions that are required for cmu_112_graphics, such as redrawAll, timerFired, appStarted, and such will be included, with all other functions on separate documents to promote readibility.

1.4 Algorithmic Plan

The central algorithmic complexity of the project will be in the enemy movement algorithm, which will be based on the A* pathfinding algorithm. In a simplified version of the algorithm, a list of moves that are an specified distance away from the enemy and separated by a specified angle theta will be evaluated, and the optimal move will be selected from those moves based on the distance traveled by the enemy, as well as the distance between the enemy and the player character. For ranged enemies, a slightly modified version of the algorithm will be used, where the enemy selects the move that keeps them within optimal distance with the player character while also looking for a clear line-of-sight between the enemy and the player character.

Another core algorithm of the project will be the crafting algorithm. Although the details of this algorithm have yet to be determined, crafting recipes will probably be stored in a 2-dimensional list of dimensions 3x3, with None being used to fill the empty spaces. Then, the player, in the crafting UI, the player will have the option of filling each of the spaces with an item of their choice. If the grid fits one of the defined crafting recipes, the completed item will show up in the UI and the player will have the option of claiming it. If the player does choose to claim it, an RNG system will be used to determine the quality of the item.

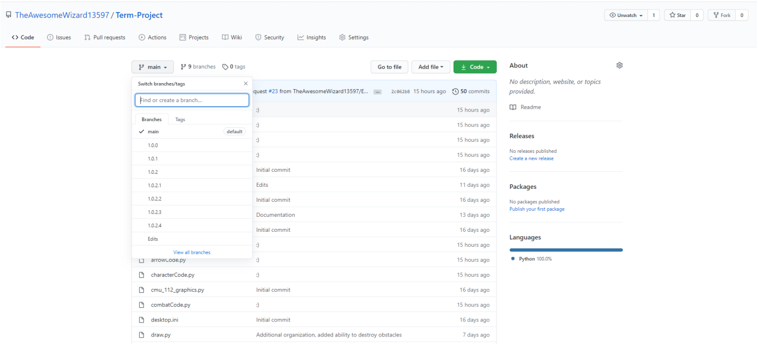
1.5 Timeline Plan

At the writing of this document, the majority of the combat system has been determined. The only components left are item drops when an obstacle or enemy is destroyed, as well as a few minor bugs in the A* algorithm. Therefore, the majority of next week will be devoted to updating the graphics of the program, as well as determining a crafting system.

1.6 Version control plan

The code for this project will be backed up using Github. Each version of code will be pushed to main, and then a new, numerical branch will be created to specifiy the version of the code.

The code submitted represents the core functionality of the project. A full version can be found on `https://github.com/TheAwesomeWizard13597/Term-Project`.



No description, website, or topics provided.

No releases published

No packages published

quest #23 from TheAwesomeWizard13597/E... 2c062b8 15 hours ago 50 commits

	:)	15 hours ago
	:)	15 hours ago
	:)	15 hours ago
	:)	15 hours ago
	Initial commit	16 days ago
	Edits	11 days ago
	Initial commit	16 days ago
	Documentation	13 days ago
	Initial commit	16 days ago
	:)	15 hours ago
arrowCode.py	:)	15 hours ago
characterCode.py	:)	15 hours ago
cmu_112_graphics.py	Initial commit	16 days ago
combatCode.py	:)	15 hours ago
desktop.ini	Initial commit	16 days ago
draw.py	Additional organization, added ability to destroy obstacles	7 days ago

Figure 2: Image of Github page with version control

1.7 Module List

The code for this project will include the pandas module, confirmed in my tech demo. It also includes a small function from numpy, which will be superseded in a later version.

2 Storyboard

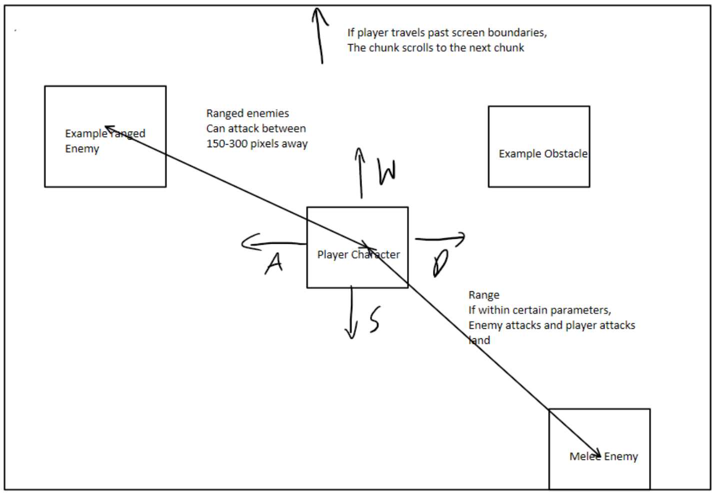

If player travels past screen boundaries,
The chunk scrolls to the next chunk
Example ranged
Enemy
Ranged enemies
Can attack between
150-300 pixels away
W
Example Obstacle
A
Player Character
D
S
Range
If within certain parameters,
Enemy attacks and player attacks
land
Melee Enemy

Figure 3: Storyboard showing basic enemy and player interactions

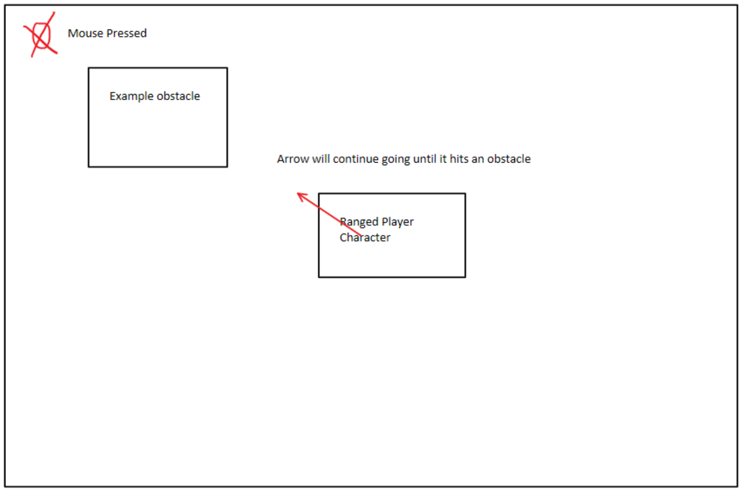


Figure 4: Storyboard showing player ranged attack interactions

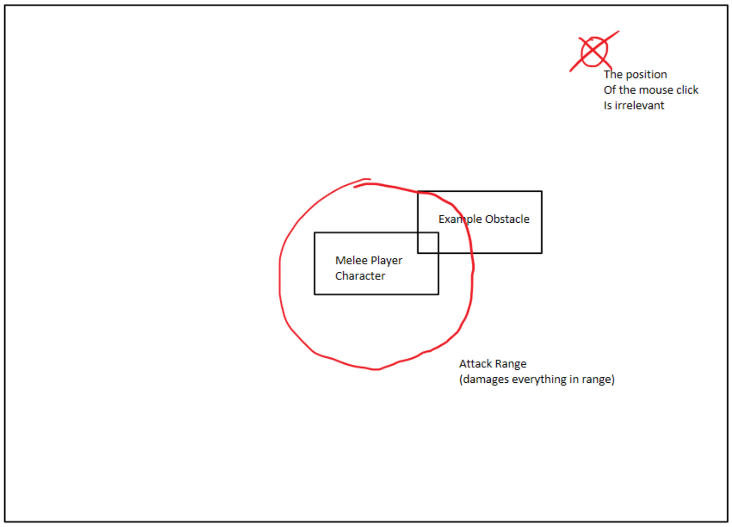


Figure 5: Storyboard showing player melee attack interactions

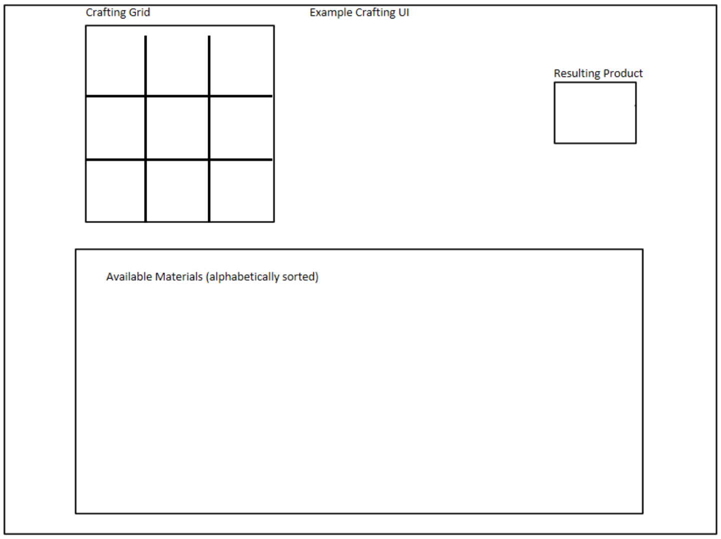


Figure 6: Storyboard showing planned crafting interface

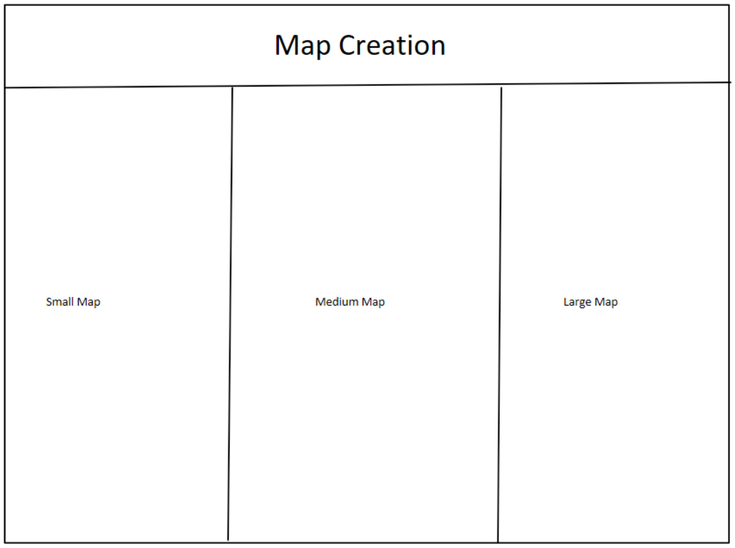


Figure 7: Storyboard showing planned map creation interface

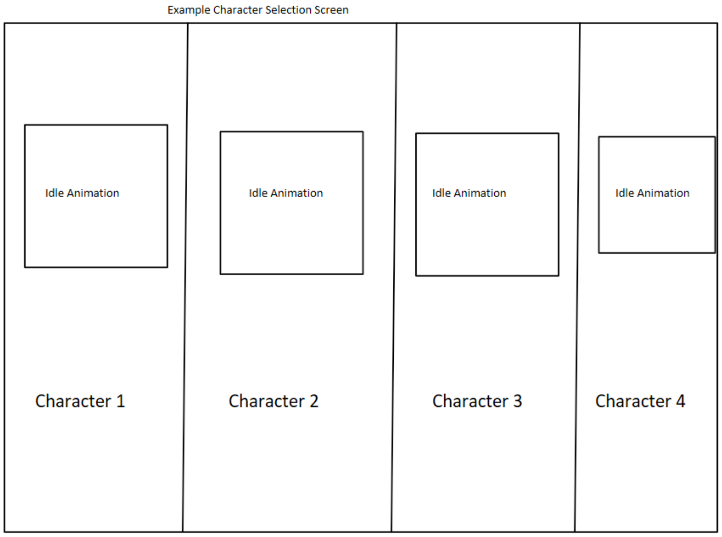


Figure 8: Storyboard showing planned character selection screen