Possible Enhancements

\* = Not so hard, shouldn’t take TOO long

\*\* = Not easy but not hard, could take some time

\*\*\* = Moderate difficulty, will take a while to complete

\*\*\*\* = Other enhancements may need to be finished before this, in for the long haul

\*\*\*\*\* = Very hard and time consuming, a semester long (or more) enhancement

Three stars or less are considered good choices for this class, however for the more ambitious types, four and five star difficulties do exist. Most of these higher starred difficulties can be broken down in to smaller tasks which, by themselves can be considered two or three star tasks.

1. \*\*\*\* NPCs need to be interacted with – Talking(Messaging System), damaging/attacking
2. \*\*\*\* Create the battle system - Monsters who die from collision with a sword, who have HP, attack, defense, special attacks, who drop experience points or collectible items.
3. \*\* Tiles should be accessible based on specific parameters like the player position or an NPC’s position
4. \*\*\* Create a better mapping system which can be read in to the engine from a text file instead of hard coded in to the files.
5. \*\*\* Design a more fluent map overlay system with an endless number of possible ‘Layers’ which are rendered one over another like a depth rating. Everything in each layer is rendered at once, then the next layer and so on, creating better depth. This should also be a variable which can be kept inside of all game objects, so depth can be changed in game using code.
6. \*\*\* Create more art for the city and surrounding areas. Think houses, doors, windows, flowers, paths, crates, tents, clothes, shops, garbage, barrels, pots, crates, walls, roofs, the list goes on and on. The goal is to make everything interchangeable, so one roof could be chosen from many which can be applied to numerous different walls and window or door types.
7. \*\* Create a day/night cycle system which allows for specific events to happen based on the time of the day, this could also be tied in to the computer’s system clock for accurate use of time.
8. \*\*\*\* Create an NPC which when talked to, opens up a shop where categories can be sifted through in order to find a wanted item like a potion or sword or apple or chair.
9. \* Create better UI, more appealing to the user.
10. \*\*\* Create a more immersive in game menu design, think outside of the box!
11. \*\*\* Find a way to make interactions happen based on their proximity to the player. For example, right now, in order to open a chest, the player must be touching it and hitting the open button at once; that can be cumbersome. This could be solved by completing enhancement #3.
12. \* Design a better control scheme, for example, space bar could be the action button, ‘E’ could be the menu, ‘esc’ could be pause, ‘shift’ for run.
13. \*\* Create a controls screen for new players.
14. \*\* Design the option of deleting files from the title screen.
15. \*\* Create music for the game and find a way to trigger the music in game. (Files already exist for this purpose)
16. \*\*\*\* Create a better event system which covers specific methods in a certain order, for example, the logic behind movement or attacking could be done in a ‘step’ event, which could be broken down in to begin or end step, afterwards the ‘Draw’ or ‘render’ event could trigger. These events should be easily accessible in all in game objects so the coder always knows when some bit of code will play.
17. \*\*\*\*\* Start work on a GUI for the engine itself so future designers can use this engine more intuitively instead of strictly lines of code.
18. \*\*\* Create more events which could be triggered in a tile Event, for example, toggle switches for something, like a light, general triggers which can be tied to starting cut scenes or traps.
19. \*\*\* Create a cut scene system so NPCs can be bossed around as actors in a play, along with the player, in order to form short narratives for the player.
20. \*\* Create a projectile system for use in magic or guns or bows/arrows. Some of this code already exists.
21. \*\*\* Create additional AI types in the Mob.java class.
22. \*\*\* Make the inGameMenu.java class savable so players can keep their stats and items.
23. \*\*\* Design a fatigue system which raises whenever an attack is done, like magic, but lowers whenever a food item is consumed. This leads to a cooking/gathering system as well as many additional items that need to be made. Also, make sure that when the player fatigue level gets too high, certain issues arise for him or her.