BaysideBalloons
Jason Kim and Derek Song

APCS2 pd8
2018-05-28

Final Project Proposal

4 Years at Stuy: Mini RPG Adventure

The Idea:

This is a bit of an ambitious task for quite a few reasons. We are trying to make a full RPG adventure with characters, story and a typical turn-based battle system. The reason why we call it "4 Years at Stuy" is because there will be 3 bosses and 1 final boss, representing the 4 years that a student should spend at stuy.

The enemies you will encounter during the adventure include: teachers, homework, exams, quizzes, internship applications and possibly more. All of these enemies will get stronger as you go through the 4 years. These enemies will also be in the form of random encounters.

There will be a final boss for each grade as well. Freshmen will have to defeat a security guard. Sophomores will have to defeat a dean. Juniors will have to defeat the SATs. And Seniors will need to defeat a particular teacher (It's a secret for now). After each boss is defeated, the player's stats will be increased.

You will also let the player move through a map. We will be using tiles to help us generate the map. The tiles will have special properties that classify them as walls or floors.

Regarding term 2 concepts, we will be utilizing queues for controlling the random enemy encounters. This is so we can control the concentration of the different enemies we have. Because each enemy varies in strength, we don't want to bombard the player with too many of the same strong or weak enemies. We plan using RandomQueue to help us with this. Everytime, we encounter an enemy, we will dequeue, getting a random enemy. Once the queue is empty, we will make a new one filled with the same concentration of enemies once again.

To-Do List

1) Look into tilesets that we could use to help us with the graphics.
2) Plan out the little things such as a level up system, stat balancing, etc.
3) Start coding the very basics.
    a) Get the player class finished.
    b) Get one enemy finished.
    c) Get one area finished (Ex: 5th floor).
4) Test and see if it runs smoothly.
5) Implement more and more and expand from there.
6) Once one year is done, rinse and repeat.

Implementation of Classes

Player:

The Player will be its own separate class as it has many attributes special to it. Specifically, movement in the open area. Something we have yet to figure out, is how we will animate that but it is not completely necessary. The Player will have attack(), and magicAttackOne() & magicAttackTwo(). It will also have the typical instance variables, HP, Attack, Defense, MagicAtk, MagicDef, Mana, and Speed. Within the move() method, there will be chance of activating battle(). The Player will also have a RandomQueue filled with enemies. Everytime there is a battle, we will dequeue from this and once the queue is empty, we will refill it. This is helpful as it allows us to control the concentration of enemies.

Enemies:

Enemies will fall under the superclass Enemy. Enemy will be an abstract class with methods such as attack() and magicAttack(). Of course, it will also have instance variables such as HP, Attack, Defense, Magic and Speed. Certain enemies, like the ones we are making bosses, will also have the method increaseStats(). This method will increase the stats of the PLAYER, as a reward for defeating such a hard enemy.

Tiles:

We are currently looking at different PNGs that we can use that will make this job easier. We will make it so certain tiles are passable, but others or not. For example, you would be able to walk on a floor tile, but not a wall tile. A portal tile would load a new map. There will be three subclasses: PortalTile, WallTile, and PassTile. These subclasses will then have many, many subclasses within them, each holding different pngs.

Map:

Map will have an instance variable in which it holds a 2D Array of Tiles. This is how we will be forming our areas. The player will be able to traverse through this area and using some math, we will be able to tell what tile the player is on. That will allow us to implement the proper interactions necessary for that tile.