

CS 246 Project Fall 2020 - Turn Based RPG

Yu Lu

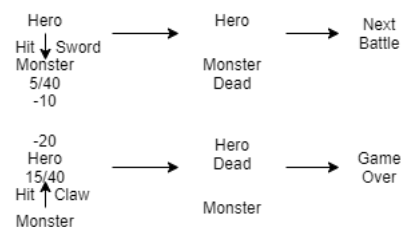
December 2, 2020

Abstract

The Project will create a simple turn based combat RPG, with at least two damage types, turn order, equipment, items and skills. The RPG would be playable using console via commands and outputs, and more intuitively via a graphical interface.

Goals:Basic

- Implement Stats including HP, SP, Physical Attack and Defense, Special Attack and Defense, Speed. (Dec 3, 2020)
- Implement A Text Based Stats Display of Character. (Dec 3, 2020)
- Implement A Basic Attack Command that takes the Stats of both Attacker and Defender Stats into Account. (Dec 4, 2020)
- Implement Basic Turn Order where entity with higher Speed act first. (Dec 5, 2020)



- Implement Trigger of Game Over Event. (Dec 5, 2020)
- Implement Basic Equipment that affect the Stats of Characters, with differentiation of True Stats and Affected Stats. (Dec 6, 2020)

- Implement Text IO that equip Items. (Dec 6, 2020)
- Implement Preparation Phase so battle do not begin immediately, so the character may choose to Change their Equipment. (Dec 6, 2020)
- Implement the second Special Attack Type and Combat Skills and implement SP consumption. (Dec 7, 2020)
- Change the combat display so when a skill is used, it is reflected. (Dec 7, 2020)
- Implement new turn order so a character with higher speed may move more often instead of just at the beginning of a turn. Make the original turn order optional. (Dec 8, 2020)
- Implement chains of events so multiple battle may occur. (Dec 8, 2020)
- Implement post battle effect such as level up, Stats change and item drops. (Dec 8, 2020)
- Implement additional events such as Recovery and Equipment Change. (Dec 9, 2020)
- Implement basic randomized combat such as miss, critical attacks, and variable damage. Make disabling randomized damage optional.(Dec 10, 2020)
- Implement basic timed status effect such as regen, poison, Stats boost. (Dec 10, 2020)
- Implement multiple enemies and party wide attack skills. Implement Multi Hit Skills where target Acquisition happens during skill Execution, such that further attack will always trigger on a valid target.(Dec 11, 2020)

	-5 mp	-5 mp	-5 mp
Hero	Hero	Hero	Hero
Hit Sword	Sword Sword	Sword Sword	Sword Sword
Monster	Monster	Monster1	Monster2
-10	-20	-10	-10

Goals:Advanced

- Enhancement: Implement Basic GUI with simple graphics. Console Output is automatically disabled.(Dec 12, 2020)
- Implement Optional GUI such that the player character's appearance is varied with the choice of equipment. (Dec 13, 2020)
- Implement Second Character. (Dec 13, 2020)
- Implement Class, Class Skills and Equipment Type with Class Restrictions. (Dec 14, 2020)

Implementation Request Change

- For Console Output change, only the Output Manger Need to be altered, as every other class such as equipment and enemy names are stored as basic word string.
- For Console based Input Change, only the Input Manager will be changed, which will further communicate with the internal state of the game.
- For Turn Based Order to change, there will first be an abstract class of Battle Manager, that will lead to either basic speed determines turn order or the specified rule.
- Alternative, Template Method may be implemented to avoid creating too many subclass of combat implementation.
- For attack property to change, there will be a list of Decorator associated with each action (within the battle manager instead of the entity) with coverage to things such as ignore defence that is added to the battle manager and drawn from the skill property. Similar passive skills and status effect will also be part of the list of Decorators. A good example of this is a charge attack.
- If a beginning/ending/credit is required, they will be treated as an event. Other such events include battles and a resting event with opportunity to change equipment, or a shop even if it ever happens.
- By altering the state of event manager, and calling all update via this event manager, events such as ending, game over or event end can be implemented such that the game knows to go over the next one, and Event Manager will then call event to finish. Either way, the top update must come from the event manager, and it is primarily here where Controller interact with.
- If GUI is implemented, then each entity such as equipment, character and enemy will be expanded to include their own graphical structures. Additionally, the graphic manager will cover displaying such objects. Alternatively, there could be a new abstract class of graphic entity whose subclass will each associate with one of equipment, character and enemy. With some effort, it is even possible to implement features such as texture packs.
- The capability for scripted battle and dialogue is likely uncovered by the limited design. However, some alteration to the battle manager could prove to be fruitful, such as creating a scripted object within the battle manager. Should this happen, it might be appropriate to add an abstract graphical object class.

- If Save/Load is required, one may use a file IO, which on each game load the event manager will handle loading in the information from the save file, such as the state of the player Character.