

Ryan Sheffler

Software Development I

Milestone Writeup

Abstract

My goal is to create a working RPG-style game using only Java. It will include a full-fledged battle system, complete with unique enemies for the player to fight and different abilities for the player to learn as they progress through the game.

Motivation

My dream job is working as a game developer, so when I'm assigned a freeform project, I try to work in game development somehow. RPGs in particular are a major favorite of mine and it seemed like a simple turn-based RPG wouldn't be an overly complex project to code in Java.

Detailed System Description

The game operates on around a turn-based system. Both the player and the enemy have set statistics. For the enemy, these are simply taken from a list of possible enemies and their attacks. For the player, these are randomly rolled at the start of the game and slowly increase with each won battle. Each turn the game takes in a command from the user and generates a command for the enemy. It uses these commands to calculate the amount of damage the two deal to each other. Different values are used for different abilities. For example, a normal attack will calculate its damage from the user's attack stat and reduce the damage based on the target's defense stat, whereas an offensive magic spell would use the user and target's mental stats. It repeats this process until either the player's or the enemy's health stat drops below zero. If the player dies, it's game over. If the enemy dies, the player levels up, increasing their stats and learning new skills based on their new stats, and moves on to the next enemy on the list.

Attached at the end are my current UML diagrams.

Literature Survey

I'm drawing inspiration from a few different games, but the biggest influences as follows:

Dragon Quest (Dragon Warrior): As one of the first turn-based RPGs, and certainly the most popular of them, most of *Dragon Quest*'s features are still shown in RPGs today. For example, the common overworld roaming followed by random encounters in a separate UI was one of

Dragon Quest's major features. If I get to the point of working with pictures, that's the look and feel I want to mimic.

Final Fantasy: Probably the single most popular RPG franchise of all time. The original took what was done in *Dragon Quest* and added more ideas into the mix. Though I don't want to replicate the feel of *Final Fantasy* as much, the additional mechanics will most likely find their way into my game

User Manual

Currently the game can only be played using the console. The player will type in a command (for example, "attack" to do a normal attack or "spell1" to use whichever spell is in their first slot), the game will record that command and carry out a turn with the AI enemy. The game will provide feedback in the form of messages describing what took place and how much damage was done (e.g. "[Player name]'s blade rended [enemy name]'s armor. [Enemy name] took [damage] damage!"). In case the user forgets their current possible actions, the game will also accept a "help" and return all of the user's commands. In the future, the player will also be able to input "save" to restart from roughly where they left off at a later point. Every necessary input has a prompt before it to let the player know their options.

Conclusion

Currently, the project is not in a working state. I'm working on implementing more methods, mostly in the Player and Enemy classes, to allow the necessary math and transmission of values to take place. Once that's done, the project will be working, but will not provide much feedback. My next goal will be to add that feedback and get data files in place, allowing the player to save and enemy data to be modified. At this point, I will have completed all of my primary goals, so I'll move onto making the game run in a window with buttons instead as well as aesthetic changes.

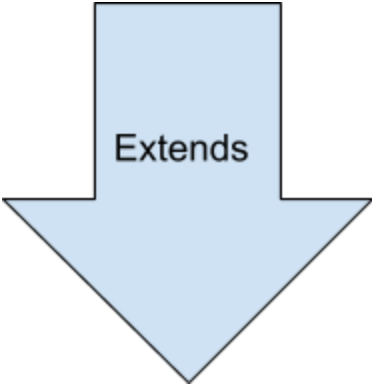
References/Bibliography

No outside sources were directly used

UML Diagrams and Additional Resources

MainRunner
+player: Player
+enemy: Enemy
+startGame()
+mainTurn()
+playerDecision(): String
+playerTurn(String: act)
+enemyTurn()
+save()

GameEntity
+name: String
+maxhealth: int
+health: int
+attack: int
+defense: int
+mental: int
+speed: int
+precision: int
+getName(): String
+getMaxHP(): int
+getHP(): int
+getAttack(): int
+getDefense(): int
+getMental(): int
+getSpeed(): int
+getPrecision(): int



Player	Enemy
+maxmp: int	+enemynum: int
+mp: int	+Enemy()
+Player(name: String)	+Enemy(int: enemynum)
+Player(String: name, int: maxhealth, int: attack, int: defense, int: mental, int: maxmp, int: speed, int: precision)	+calcDamage()
+getMaxMP(): int	+takeDamage()
+getMP(): int	
+calcDamage(boolean: isMagic, boolean: isCrit): int	
+takeDamage(int: damage)	
+levelUp()	



A picture of *Dragon Quest*'s user interface, for reference