Rapport ProjetJeuRPG

Pierre Durollet

June 2024

Contents

1 Introduction

I decided to take on a project about making a video game. For that I first did the video game without any graphic, nor any game engin. Therefore, the game is fully played on the terminal, with of course the display of what is going on.

This game is an RPG with levels and fighting mobs.

2 Functionnalities

In order to make the game more complete and to learn more about different aspects, I have decided to have more functionnalities other than simply fighting mobs.

2.1 Fighting

Of course the player still needs to be able to fight with against ennemny monsters, so there is a functionnality that allows a player to fight against mobs.

2.2 Winning a fight

It would not be fun without gains, so when you defeat a monster, you gain experience points, that allow you to level up. But monsters can also drop loot when defeated. So I Added the possibility of loot being dropped (each item has an independent probability to be dropped). Items dropped this way are added to the player inventory.

2.3 Progress save

In order to not restart from the beginning, I decided to create a class that creates or loads a player profile from a text file located in the src/PlayerFiles folder.

The goal is that when quitting the game, the profile is saved so that it can be picked up later. For now there is no protection and no way to select a certain file. I plan to have the argument of execution being the name of the profile you want to load, and maybe add some sort of password just for the jist of it (files are locally available and not admin protected).

3 Main classes, files and folders:

Game The class that is used to launch the game.

Fight The class that handles the fighting loop and all rewards gainned from fights.

Player Class that holds all the methods tied to the player.

Item Class from which all different items extends (even gears).

MenuSystem Class that will allow the player to navigate in different Menus. As a start: fighting and accessing and changing the gear and inventory.

PlayerFile Class that is the core of the player data saving functionnality.

Gear/ItemFactory Classes that hold the same use for Gear pieces and Items, they are used with PlayerFile in order to save and load a player profile.

Levels

PlayerFiles

4 Tests Available

4.1 TestInventory

Test uses visual test, not JUnit.

This test exists to ensure that the inventory read in the PlayerFiles is correctly read and that the display of the Inventory is clear.

4.2 TestFile

Test uses visual test, not JUnit.

This test is used to verify that the Player Profile is correctly saved in the PlayerFiles folder.

4.3 TestGame

Test uses visual test, not JUnit.

This test allows to verify that the fighting loop is indeed a loop, that the rewards in case of a victory do work and that the display is coherent.

4.4 TestPlayer

Test uses JUnit.

This test verifies that a player profile is correclty loaded into the player when using the PlayerFile class.