Hacettepe University

BBM104 – 2021 Spring

ASSIGNMENT 3

May 6, 2021

Melike Nur DULKADİR

21992919

1. Problem

In this assignment, we were expected to implement an application that is a simulator of a simple board game with using polymorphism. In this game, there are two sides and each side has 3 different characters each and the characters have different characteristics from each other. The aim of this game is to kill all the characters of the enemy side and remove them from the board.

2. Solution

I think the most important part of this task was to correctly apply the attributes given in addition to the characters besides the general characteristics. Because even though they were on the same side, the characteristics of each character were different. And with this situation in mind often, I set up a design by overriding the methods in the required parts and checking the conditions.

Firstly, I created classes in my project and I implemented an inheritance relationship between them. I first created the super class, the Warrior class, then the Zorde and Calliance classes that extend the Warrior class, and then the Ork, Troll, Goblin classes that extend the Zorde class and the Human, Dwarf, Elf classes that extend the Calliance class. Then I created the methods required for each class. Accordingly, I created the "fight method " in the Warrior class and added the event that will take place jointly, and I added different parts in the Zorde and Calliance classes by overriding this method for different events that will take place on both sides. In addition, I created the "checkNeighbors" method which using for the normal attack and using for the heal property of Ork characters, "toString" method by overriding, and the "defineZordeAttacker" method to determine the type of the attacker character. And I created the method of "move", which will perform the move steps using these methods. For the heal property of the Orc character, which is different from other Zorde characters, I wrote the move method by overriding it in the Orc class. Likewise, in the Calliance class, I overrided the "toString" method, I created the "checkNeighbors" method for the range attack of the elf characterand the normal attack, and the "defineZordeAttacker" method to determine the type of the Calliance character, and I created the "move" method. For determining the character type in my "defineZordeAttacker" methods, I used the downcasting to create an attacker of that type.

In the Main class by reading the given initials.txt file, I created the Warrior[][] board according to board size and placed each warrior in its initial place. I created arraylist zordes and calliances with the super class references for keep the wariors. Then I created the Command class for reading commands.txt and performed the move commands in accordance with the rules using "move" methods. I created the MoveCountException and BoundaryException classes to handle exceptions that do not comply with the rules.

3. UML Diagrams