LINGI1131 Programming language concepts
Project report

Bronchain Olivier - 47651200
Staelens Jean-Sébastien - 86831200

May 4, 2015

1 Introduction

This report contains the architecture of the project and the implementation choices that were made during the conception of the game.

2 Component diagram

This diagram contains the different PortObjects, their characteristics, and the functions that can be called on them.

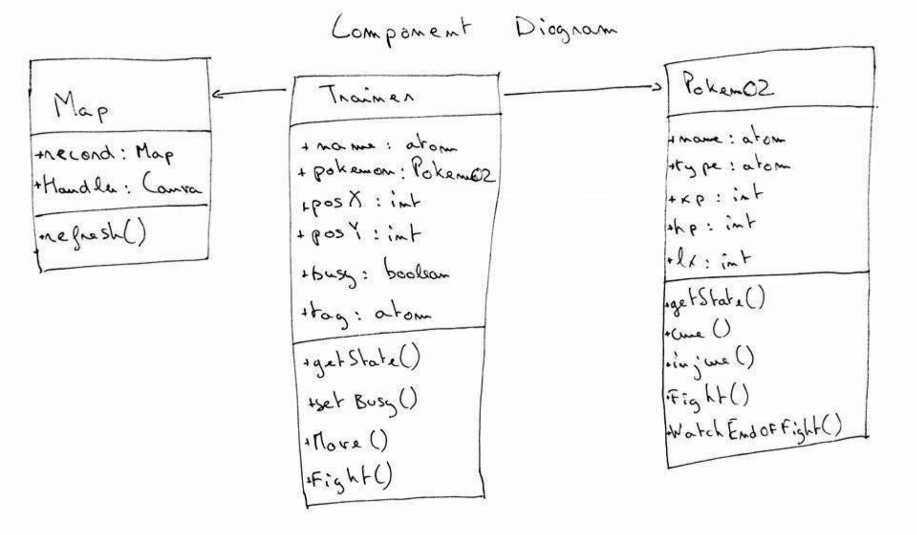

Figure 1: Component Diagram

3 State Diagram

This diagram contains the different states that you can reach during the execution of the game, and what has to be done to go from one state to another.

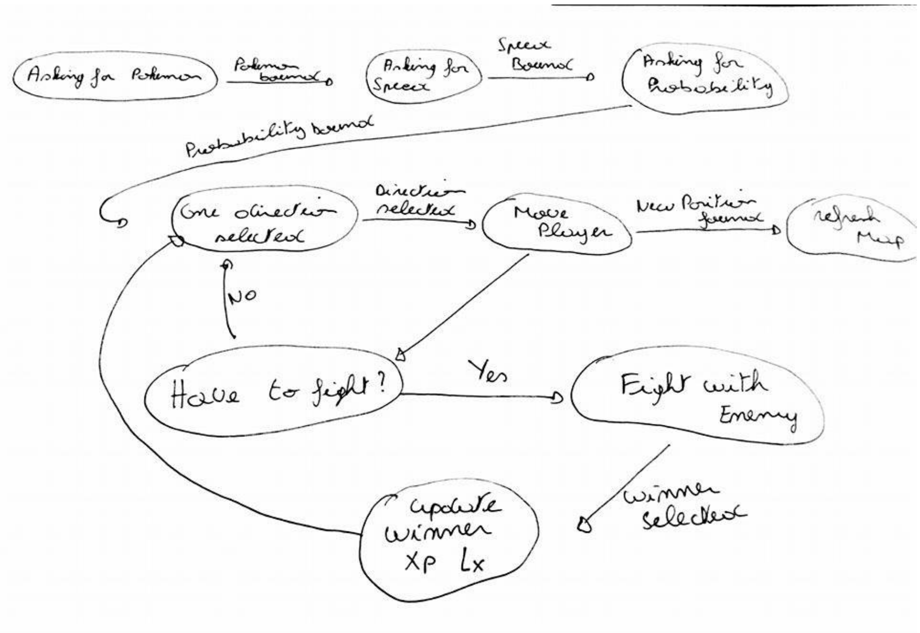

Figure 2: State Diagram

A UML sequence diagram is provided in the appendix.

4 Data structures

Trainer A record containing the name of the trainer, the PortObject Pokemoz that he carries, his position, a tag and his state (busy or not).

Pokemoz A record containing its name, its type, its Hp/Level/Xp.

Map A record containing the record of the map that was given and the handler of the canvas used to display the map.

Those data structures are implemented by PortObjects containing the records, allowing their state to change during the game. Behaviour functions are used to apply the changes on the records, and to pass data between the PortObjects.

5 Implementation choices

5.1 Infirmary

We decided to make a spot on the down left corner of the map were the player and the enemy trainers can cure their pokemoz. The trainers only have to walk on the spot, and their pokemoz is cured. It is shown by the image of an infirmary.

5.2 AutoFight - Enemy trainers

For the Autofight implementation, we decided the player to have a simple behaviour : he tries to reach to top right corner of the map with an alive pokemoz. When his pokemoz is dead, he goes to the infirmary to cure it and then tries to go back to the top right corner.
The IA of the enemy trainers is even simpler. They move randomly on the map and fight the player when they're next to him. They don't fight wild pokemoz.

5.3 Dead pokemoz

We decided that when the pokemoz of the player is dead, no fight can be declared, either by an enemy trainer or by a wild pokemoz. The player can then move freely on the map and go to the infirmary. When the pokemoz of an enemy trainer is dead, he continues to move randomly on the map, but he can't start a fight with the player. He doesn't try to reach the infirmary, but when he reaches it, his pokemoz is cured and he can try to fight the player again.

5.4 Pickle

After many tries, we were unable to make Pickle work on any of our computers. We decided to allow the player to design his own map at the beginning of the game. He can also decide to use the default map.

6 Appendix

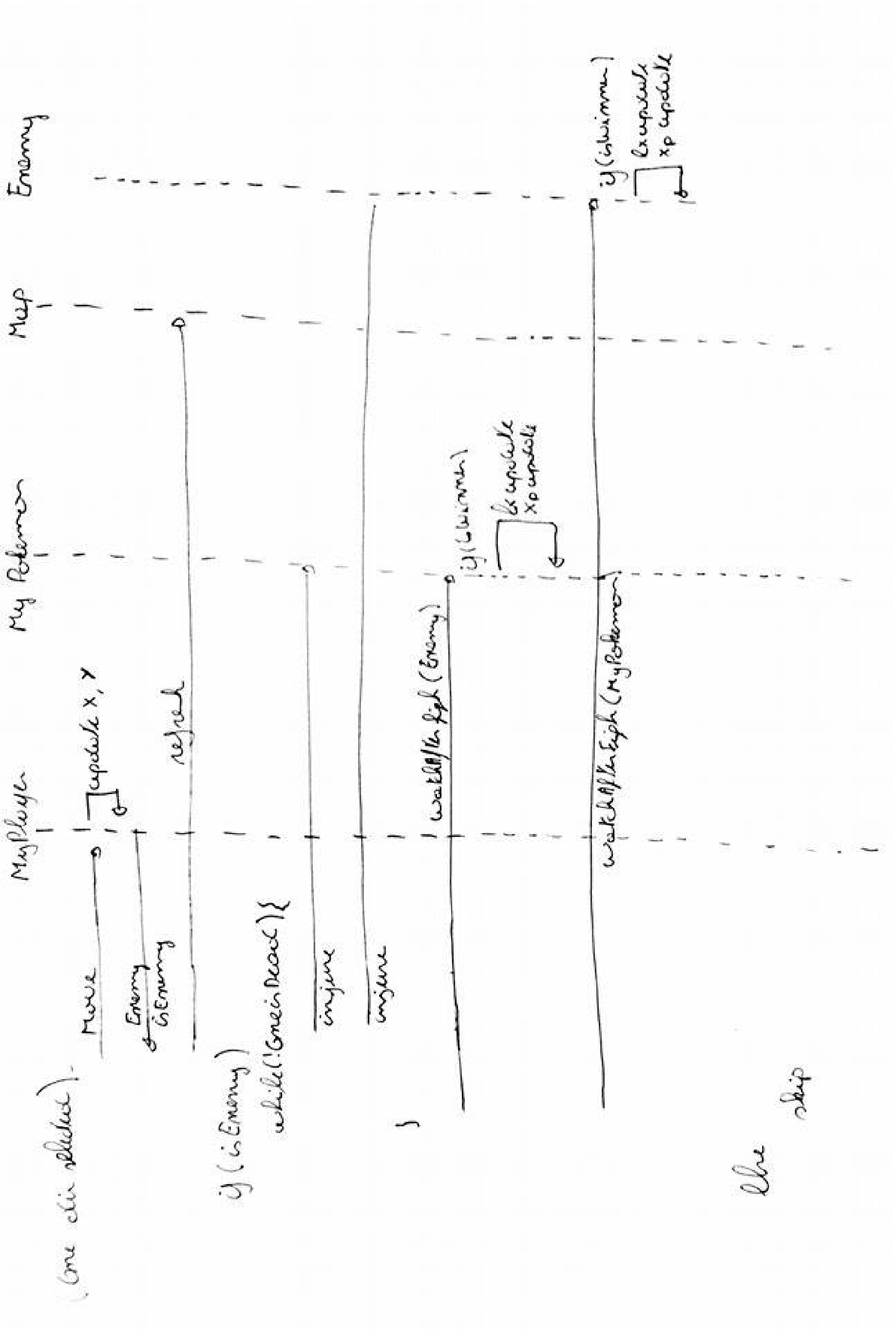

Figure 3: Sequence UML diagram