Object oriented programming

Luis L. marques
Enzo Carré

2019/2020

Contents

The game 2
- Used tools 2
- The basic version 2
- The advanced version 2
- Known bugs 3

How we did the game 3
- The common work 3
- The personal work 3

Personal Experience 3
- Luis L. Marques 3
- Enzo Carré 3

The game

Used tools

Because the course that ordered this project is based on the Java language, we had to use it along with JavaFX for the graphics.

As for the IDEs, we had a mix of Eclipse and IntelliJ users.

The basic version

The client ordered a basic and an advanced version of the game.

For the basic one, we had the following rules:

- Only one type of troop. We chose to keep the knight.
- Only one level for castles.
- The troops don't evade castles or obstacles when moving.
- The troops don't leave the castle by the door.
- Per turn, castles produce troops instead of money.
- The bots don't do anything.

This was in order to have a first functional prototype without all the features, but that could be played by someone.

The advanced version

For the advanced version, we took out the rules we specified in the previous subsection, and we added the following ones:

- Many types of troops. We chose 4: Knight, Onager, Pikeman and Camel.
- Many levels for castles. We chose to have an infinity of levels by using mathematical formulas.
- The troops evade castles and obstacles when moving.
- The troops leave the castle by the door.
- Per turn, castles produce money, and money can be used to create troops and level up the castle.
- The bots do actions.

As bonus features, not included in contract, we added:

- Walls around the castles.
- A troop type that can move money from one castle to another.

Known bugs

Basic version:

- Random crashes when launching the application.
- Sometimes, troops follow the path but are a bit off them when moving to another castle.

Advanced version:

- Random crashes when launching the application.

How we did the game

We were two on the project. In the first part, we just started coding on the same parts to find what we could do using JavaFX.

The common work

The common work was the design of the interfaces and classes we would use as well as the beginning of the code, as it was shared between both versions.

The personal work

By the time we had separated our project into the basic and the advanced version, we also deletaged work.

I took on the basic version along with the javadoc and the documentation for both versions while Enzo took on the advanced version.

Personal Experience

Luis L. Marques

Personnally, I found this project a bit boring at the beginning, as it is not my type of game, and I wasn't familiar with Java. By using C++, I could have coded much faster as I already knew the language and its features, as well as some graphic libraries.

Afterwards, I familiarised myself with JavaFX. Althought it doesn't resemble any other library I already used, it was quite easy to use.

Contract side, I feel the contract had some unprecisions about the players and the attack system. I also felt we lacked a bit of time, one or two more weeks would have been perfect for our team.

Code side, it was a great opportunity to discover and master a new language, and as for every other project, it was a great opportunity to develop team-play.

Enzo Carré