Object oriented programming

Luis L. marques
Enzo Carré

2019/2020

Contents

How to play

The realm is in chaos after the king passed away. You are a duke and have to fight all the other aspiring dukes to get the throne. There are also neutral dukes who don't specially want the throne.

Your castle is the one with the red flag, while aspiring dukes's castles are the colored ones. The neutral dukes all have grey castles.

When clicking a castle, you have information displayed about it, even if it's an enemy castle. Then you will have many buttons in the top center of your screen.

The list in order is:

- Recruit.

- Move (Move if ally target, Attack if enemy target).

- Level up castle.

- Transfer money between castles.

- Build wall around castle.

- Level up the castle's barrack.

With these commands, you should be able to play instinctively and have fun. The game is over once you have no more castles (lose) or when you conquered all the other active dukes (win).

Note: When transfering money, you can write the amount in the spinner. However, to transfer that amount, you first have to press Enter so the application gets the value you typed.

Making 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 3: Knight, Onager, Pikeman.

- Many levels for castles. We chose to have a max of 10.

- 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.

- No victory screen.

Known bugs

Basic version:

- Randomly, the application won't start up.

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

- After conquering a castle, sending troops again will make them appear twice, but the game only considers them once as we wanted.

- White bar on the right side of the window.

- Random crashes (problably AI related).

- No victory screen.

Advanced version:

- Randomly, the application won't start up.

- White bar on the right side of the window.

- Multiple recruitement has some bugs.

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.

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 felt 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

This project has been a real opportunity for me to discover the organisation that is required for a project of this extent. But in my opinion this project was, by far, too long for the time we had and because of this i have a taste of unfinished. We had the idea, and the skills to do it but we couldn.I think this is because I am in MATH-INFO give me a real disadvantage compared to CMIs and INFO class, because we dont know how to work in this kind of format. Anyway it was still a very good experience for me and my mate, I discovered a new way or programing (by pair).