18-5-2023

Technical report

Battle of races

Marc Argüelles, Yuhen Zhou, Alex Chica
INS ESTEVE TERRADAS I ILLA

Index

1. Introduction

This program is a basic fights game between fantastic races developed in Java using Java Swing as a graphic interface. This project has also a local SQL database and a local website that shows the battle history.

2. Applied technology

2.1. Programming

To develop this project, we used Java. The backend of this project consists of multiple packages of classes divided into the game logic, the player classes and the graphic interfaces.

The logic package consists of three classes, one for operating with the points distributed inside the game, other to run the logic of the fights and the last one to store the stats of the entities.

In the player classes package we find the database connector used to download and upload certain parameters, the player class to store all the methods related to the character fighting and the weapon class where we find the methods to use the weapons.

Finally, we have the windows package where we created a class for every window we had to develop using Java Swing.

At last, we have a directory with all the images used in the game. We use the database to store the path of every image and to use them we query the mentioned path to the database.

2.2. Database

The database has been developed using SQL.

It's composed of 6 tables. The battle history table that stores everything related to the result of the battles, the players table that stores the player ID and its name, the races table that stores all the base stats of every race, the warriors table that stores its ID, the name, the image path and its race, the weapons table where it's stored the weapon ID, name, image path, power and speed and finally a table that stores the weapon ID and the warrior ID that can use the weapon.

This database is used to run the program through the JDBC driver connector.

This database is used to show the match history on a local website.

Also note that this database is automatically generated in Linux based SO. If used on a Windows SO it's necessary to run the SQL scripts manually to create the database.

2.3. Website

The website shows a battle history using the "battle" table of the database.

We have used python convert the "battle" table to XML, from there we used the XSL programming language to style the XML file and to convert it to HTML, to do that we have also used python. At last, we have used a CSS stylesheet to show a fancy version of the battle history table originally stored on the database.

To run this properly it's necessary first to have the database on and to play some games in order to have some data in the database. After that you just must run first the dataToXml.py then the xmlToHtml.py. Once done that just open the server to see the website.

2.4. Operating system

The operating system used to run the game does not matter as soon as the database is correctly created and the data is correctly inserted.

If you use Linux, the program itself creates the database automatically, otherwise if you use Windows you must create it manually using the scripts located in the player classes package.

3. Technical issues

3.1. The database creation

The first issue has been the database creation, in Linux it is possible to access the MySQL shell through the Command Prompt, using that we can create the database automatically. But Windows, in order to access the MySQL shell, first it must access its directory and from there is possible to access the MySQL shell. That is why in Windows it is necessary to create the database manually.

3.2. Automatic website generation

Due to lack of knowledge related to the python-xml-html connection we have spent plenty of time figuring out how to extract the data from the database to XML and from there, using XSL, to HTML. The solution was some python files we had from previous M04 projects, we could convert those files to the desired ones without changing them too much.

3.3. The images used

We have focused more on functionality rather than beauty, that's why the images used in the project look ugly and off. The fact that the images had to be 300x300 pixels limited the resources we had access to.

3 Improvement proposals

If we had more time and freedom, there are things we could have done way better.

First of all, the aesthetics. We didn't know much about pixel per pixel drawing but doing that, we could have created a retro-like game that paints every pixel as we wanted to. Instead, we just used the classic JavaSwing style.

Other thing we could have done better with more freedom is the fact that the dwarfs are kind of overpowered, lowing their stats a little bit would result in a more entertaining experience.

The last thing I think we could have done better is the fact that there are no objectives or bosses. Would be interesting to have milestones, bosses and an objective inside the game.

4 Personal thoughts

This project has been interesting because after several months of learning how to code, finally, we visually see the potential of the things we have been learning. Also I think that it's really simple to divide a project based on Java due to its object oriented structure.

Overall a growing and enjoyable experience.