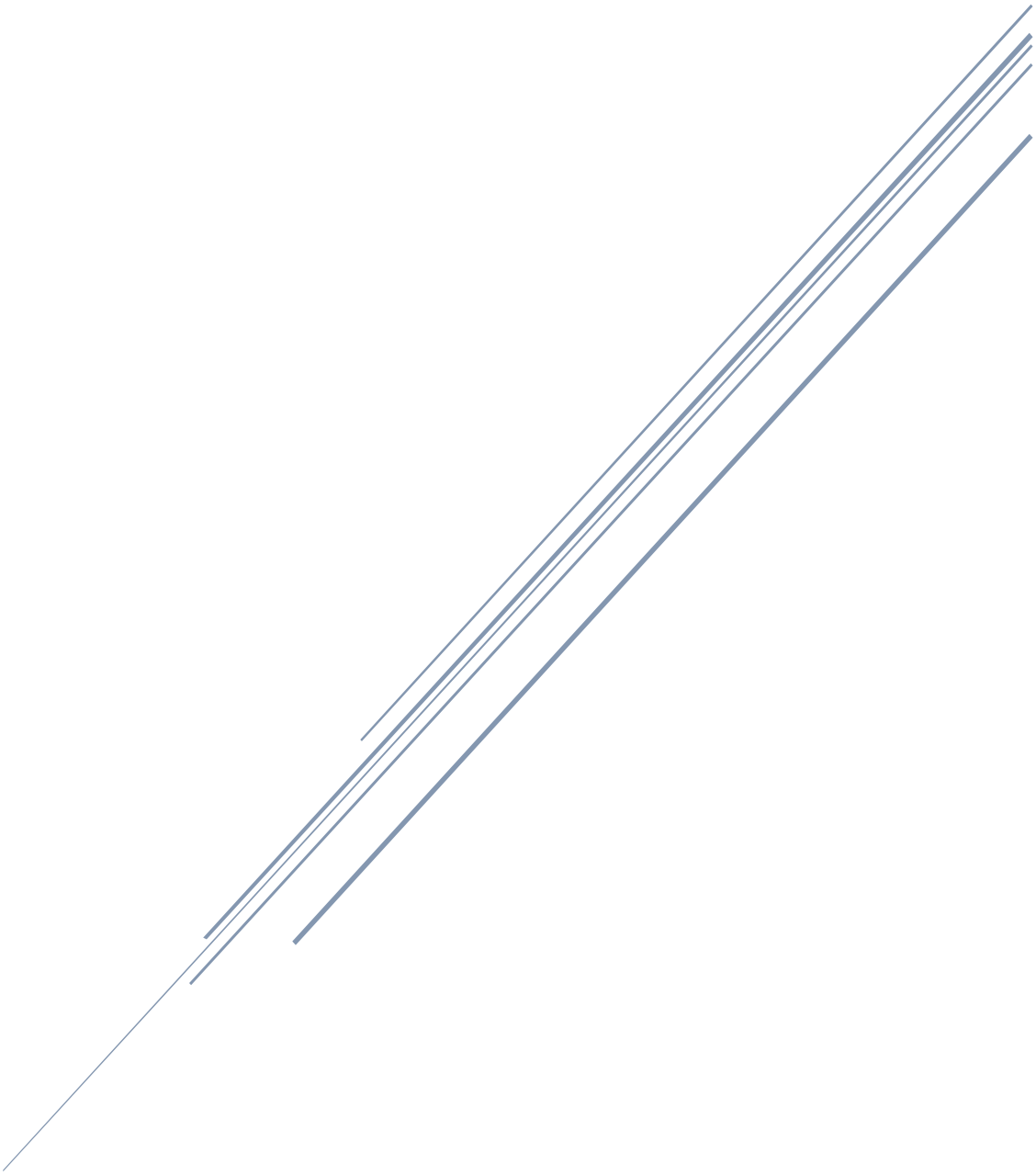
PROJECT REPORT

JAVA/POO/UML



Cesi Exia
A1-2018

❖ Group members:

MAITRE Maxime
AZZOUZI CLAUSEL Zacharia
BOULESTEIX Tristan
KARDOUS Jean-Pierre

Summary

I. Context/Goals

The main goal of our project is to recreate the Lorann video game in Java language with five different levels. The levels are stored in a database.

	Lorann, the main character
	The multicolored spell
	Energy sphere
	The exit door of the level
	The impassable elements of scenery
	The four demons
	Treasures to earn points

Figure 1- Different sprites used

The figure above shows the different sprites that are given to us and that must be used for our program. To kill demons, Lorann can launch a spell. He can only use his spell once, if he wants to resume his spell, he must touch it.

II. Real schedule

Days / Name	Friday 05/25/18	Saturday 05/26/18	Sunday 05/27/18	Monday 05/28/18	Tuesday 05/29/18	Wednesday 05/30/18	Thursday 05/31/18	Friday 06/01/18	Saturday 06/02/18	Sunday 06/03/18	Monday 06/04/18	Tuesday 06/05/18
AZZOUZI Zacharia	Discovering the project subject and group organization			Reflection about the project structure and making the first UML		Making the plugged code about the DAO, creating the database and maps	Finish the DAO and popup code to select the level	Starting Junit tests	Starting the Javadoc, continue Junit tests and modify README.md		Trying to repair the Junit test, and upgrade the Javadoc	Finish deliverables and compilation of the program
BOULESTEIX Tristan						Making the plugged code of model	Finish model code and add part in the model code	Update Controller code and making contract code	Starting the Javadoc, finishing the last part of the code and debugging		Continue to debug the code (add new functionality) and try to repair Junit tests	
MAITRE Maxime						Making the plugged code of view	Finish code of the frame and update class diagram/making the component diagram	Making the Sequence diagram and update the class diagram	Starting the Javadoc, update UML		Create the second sequence diagram and update UML	
KARDOUS Jean-Pierre						Making the plugged code of controller	Making the package diagram and start the project report	Making the sequence diagram and create the score popup	Starting the Javadoc, and update the report		Create the second sequence diagram and update the project report	

III. Implementation of the UML

A. Diagrams

1. Diagram of packages

To create the diagram of packages, we used the class diagram. In fact, we took the name of the different packages and we linked them together with dependency links.

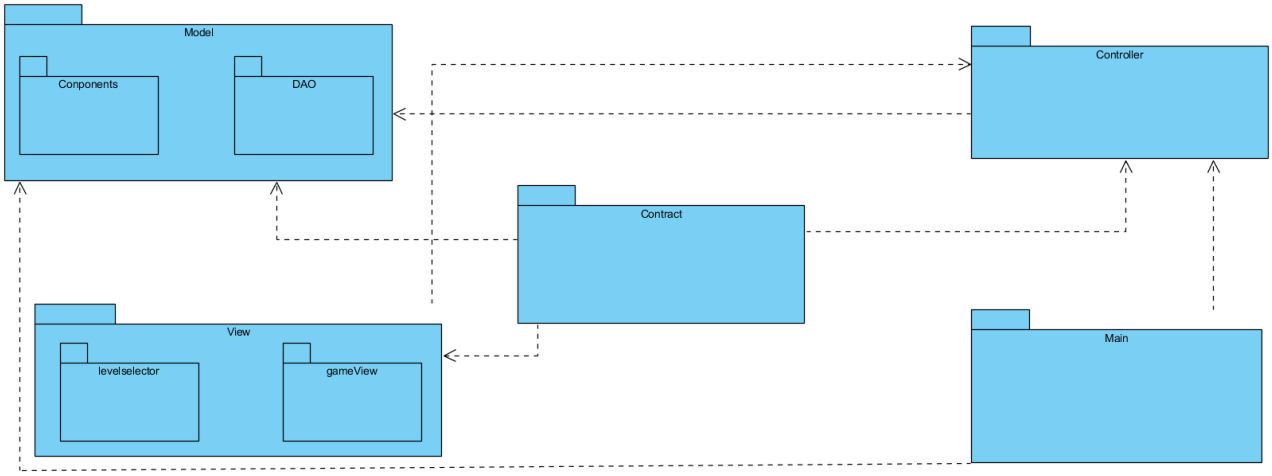


Figure 2- Diagram of Packages

B. Codes

1. Model

In the package Model, there are every component which will be used in the game. Furthermore, there are also classes which can read external data like the database or sprites of the components

2. View

In the package View, there is the code of the frame and class to update the display. It has also a list about the selection of different level.

3. Controller

In the package Controller, there are all the element that make the link between the model and the view. It contains also alternate thread which run the game, and which move the components.

4. Contract

In Contract, there are all interfaces which make link between all other packages.

IV. Organization of the database

The database is used to store the different levels which are related with maps.

Figure 3- Database Table

Furthermore, we used stored procedure to find a map by it ID. We select the map we want to display. This map is stored in the table “level”.

A. Entity relationship diagram

We have stored procedure which make link with all level which have an ID. So, we must create an entity relationship diagram which have two characteristics, an ID which have a unique number and the map which use a string.

Figure 4- Entity relationship diagram of project

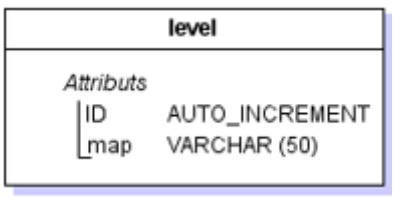


Figure 5- Logical model of data

B. SQL

To have the possibility to find map by ID we have created a stored procedure which select the map from level with parameters the ID of the map.

```
1  CREATE PROCEDURE findMapById(Int id)
2
3  BEGIN
4      SELECT map
5      FROM level
6      WHERE level.id = id
7
8      END;
```

Figure 6- The SQL code of the project

V. Conclusion

A. Problems encountered

During this project, even if it was easier than expected, we have meet some issues. The biggest problem was about Maven. The dependencies caused a lot of problems. For example, we could not generate a SureFire report with Maven despite the command “run as JUNIT Test” in Eclipse worked perfectly.

Another issue was with Git. One of our teammate was unable to sync the project with git desktop. He succeed by using git on Eclipse but he has to reconfigure the URL of the project each time because git seem to forgot the settings when he pull.

We also have some strange error like a null pointer exception which seem to appear randomly when we launch the spell but doesn’t append when we debug the program.

B. Group report

Globally, we liked work in this project because we were very organized. Everyone listened to each other and was useful thanks to our various skills. We learned a lot of things for example, to work with the team on a program or to organize a Java code.

C. Personal report

Tristan BOULESTEIX:

I found this project really interesting and I really enjoyed working with this group. I already known Java before the teaching unity (UE) but I still learned a lot of things like the utility of the interfaces and the part Contract (thanks to Maven dependencies) and how to generate jxr, surefire report. I think this project doesn’t helped me to be a good programmer in Java, but it makes me learned to respect the rules, the Design Pattern…

Jean-Pierre KARDOUS:

According to me, the project was very interesting because it shows us how to create a video games and how to use and abandonware. At the beginning, I found this project difficult, because I’m not very good at Java code. I still learned things, for example I learned how to use Javadoc, and how to structure well an UML diagram.

Maxime MAITRE:

We work well as a group, everybody was involved and motivated. We have effectively separated tasks to optimize our time of work. As in my group my classmates are good in code, I make all the diagrams and some littles parts of the code. This week was productive, and we finish our project on time.

Zacharia AZZOUZI:

Result production

The project has been completed in its entirety. I focused on the DAO part at the beginning of the project. As a result, I also helped Tristan in the realization of the corked code and its realization. At the beginning of the project we also produced a UML team.

OPERATION OF THE GROUP

Work steps and procedures Methodological aspects

Our group to work properly after we get along on our tasks.
I really liked doing UML, the programming part was harder, but we managed to finish it in the given time.

Developed competence

I learned a lot about Maven configuration but also about the Singleton and Factory design patterns. As a result, I also reviewed a lot of basic java.