POO project - Report

dimanche 2 juin 2019 16:20

Report Project POO - UML/Java

Summary :

I. Group & tasks division

A. The group :

- AMARY Clément (Group manager)
- ALBERT Naomie
- MOENCH Baptiste
- MONTEMONT Théophile

B. Departure tasks division :

At the beginning of the project, we saw that we had to do different tasks and use the MVC Design Pattern, so we decided to divide the diagrams and code organization with this MVC Design Pattern, so we had :

- Clément : Controller class diagram & code + Database
- Naomie : Model class diagram & code
- Baptiste : View class diagram & code + Database connection
- Théophile : Work on the different components

And after that, the person in charge of the diagrams and code for a component had to do the tests for this component.

C. Real tasks division :

The first organization seemed good but there were a lot of difficulties in the code implementation and we had to change our organization, so here is the real tasks organization at the end of the project :

- Clément : Controller, View, Model, contract code, database build and connection, UML sequence diagrams, controller and contra ct UML classes diagrams, UML components diagram, level design, git gestion, javadoc, jxr report and model Junit tests and tests repo rts
- Naomie : UML model and view class diagram
- Baptiste : First version of the DAO, first version of the UML components diagram and controller Junit tests.
- Théophile : Code and debug

II. Schedule

A. Original schedule :

	23 May	24 May	25 May	26 May	27 May	28 May	29 May	30 May	31 May	02 June	3 June	4 June	5 June	6 June
UML Diagrams														
Database														
Junit tests														
Model														
View														
Controller														
Javadoc														
Debug														
Oral														

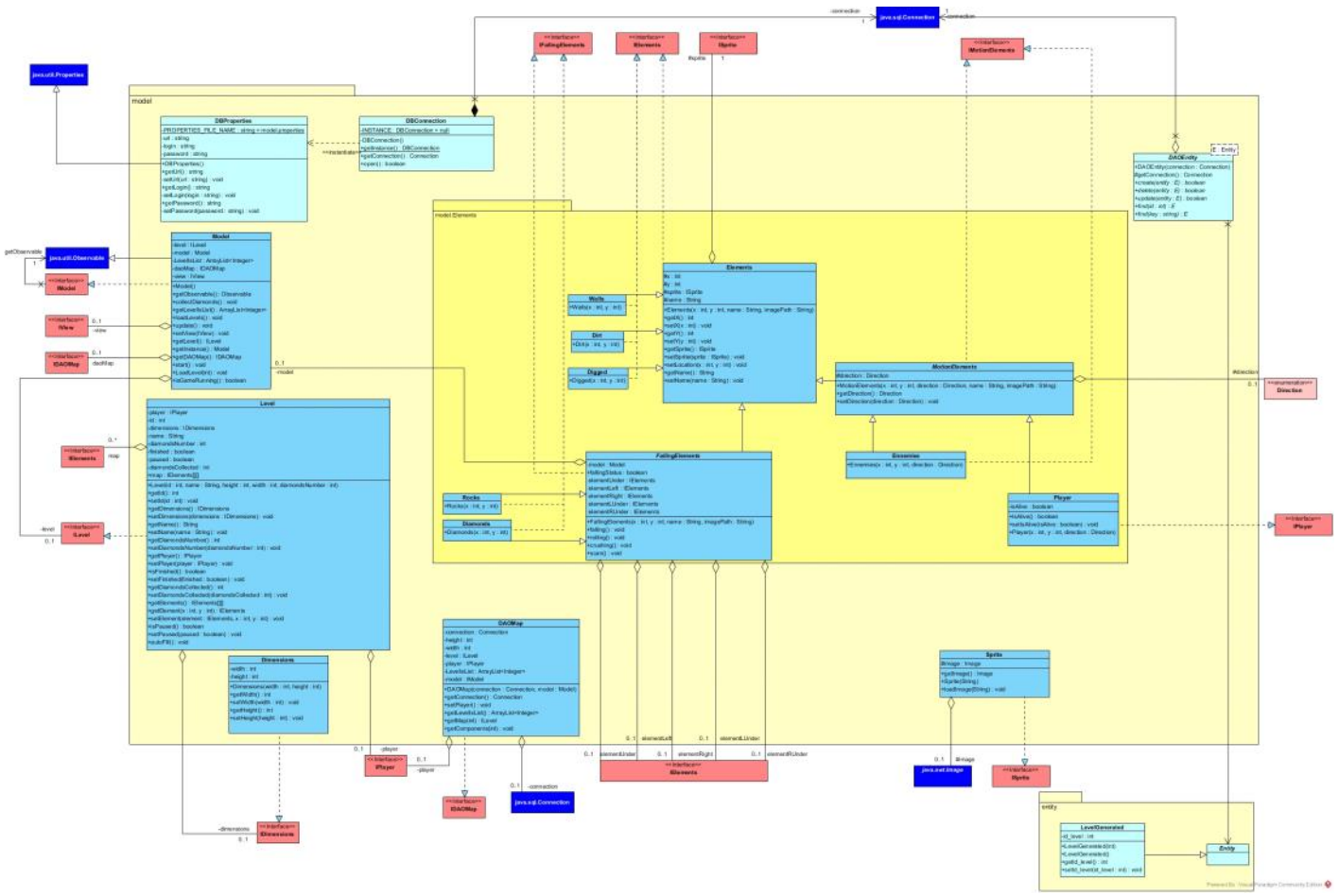
B. Real schedule :

	23 May	24 May	25 May	26 May	27 May	28 May	29 May	30 May	31 May	02 June	3 June	4 June	5 June	6 June
UML Diagrams														
Database														
Junit tests														
Model														
View														
Controller														
Javadoc														
Debug														
Oral														

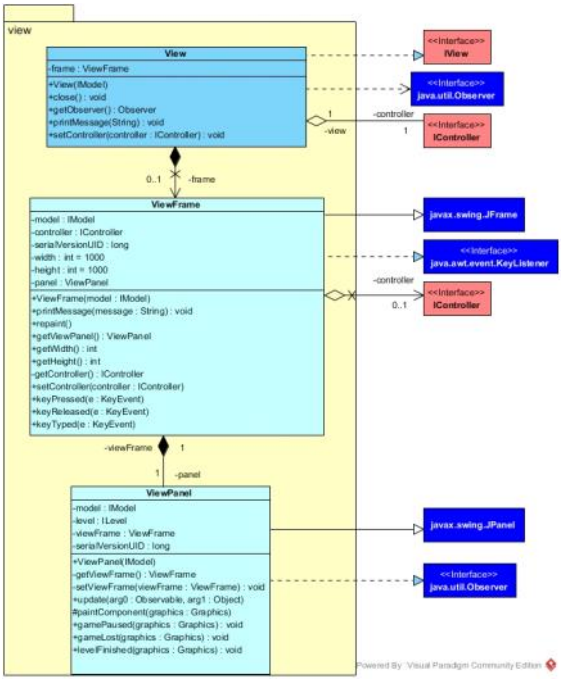
III. UML Diagrams

A. Classes diagrams :

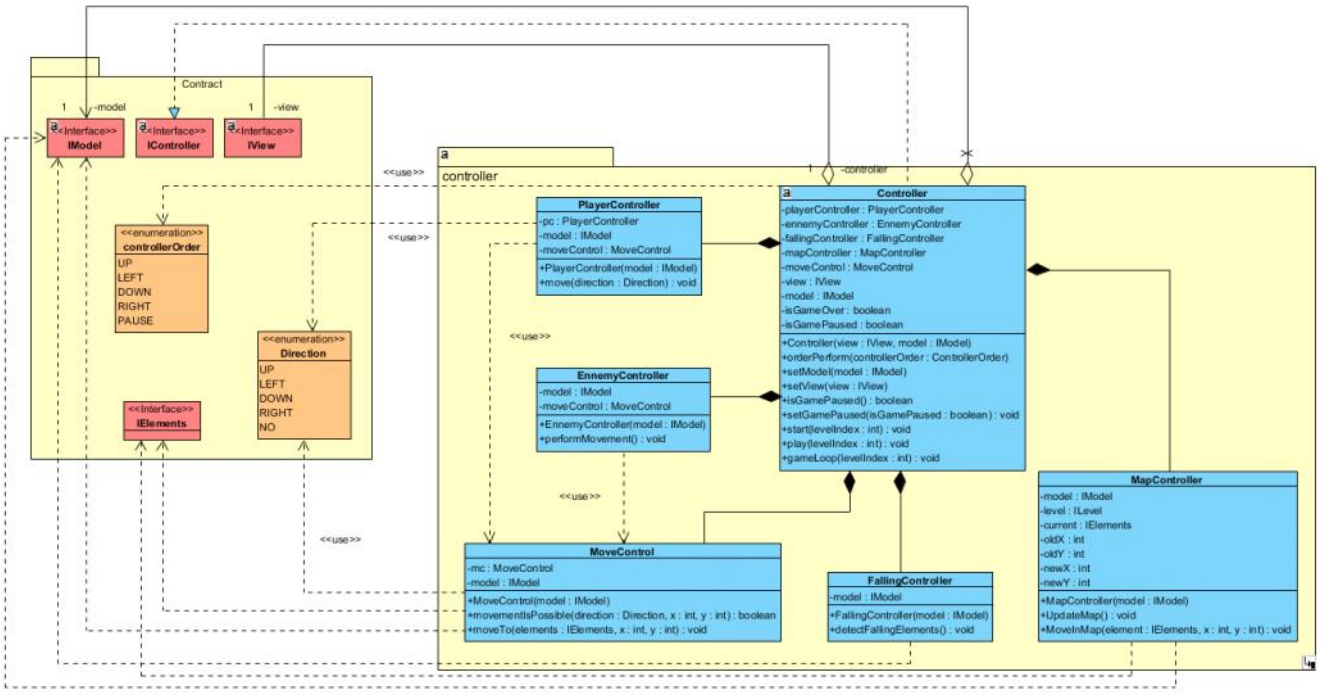
- Model :



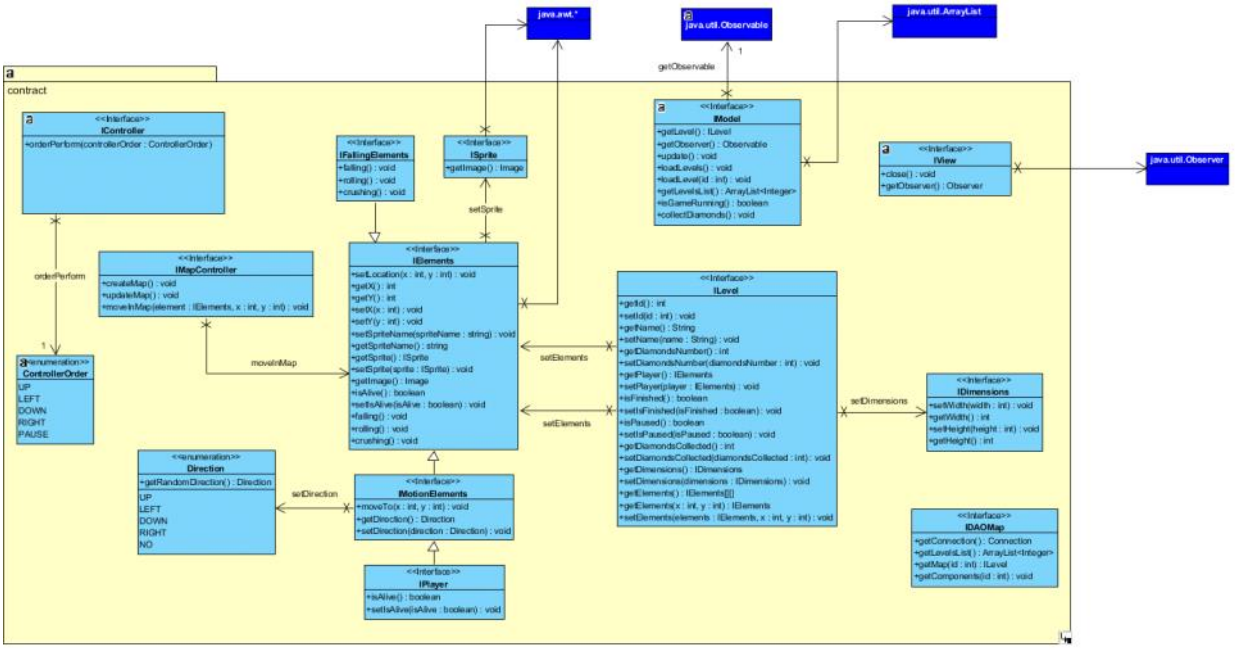
- View :



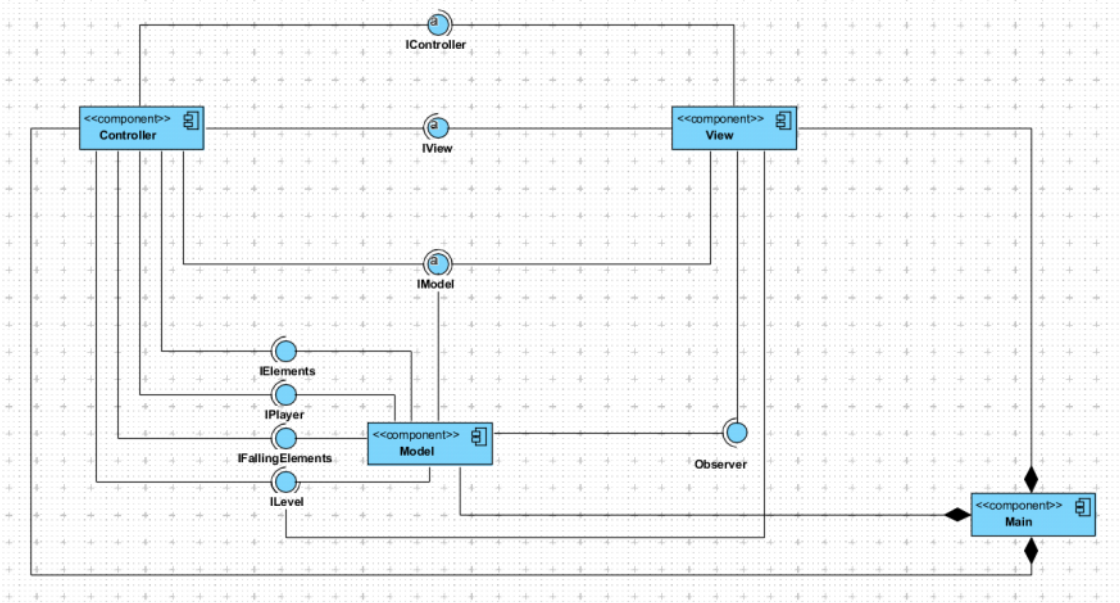
- Controller :



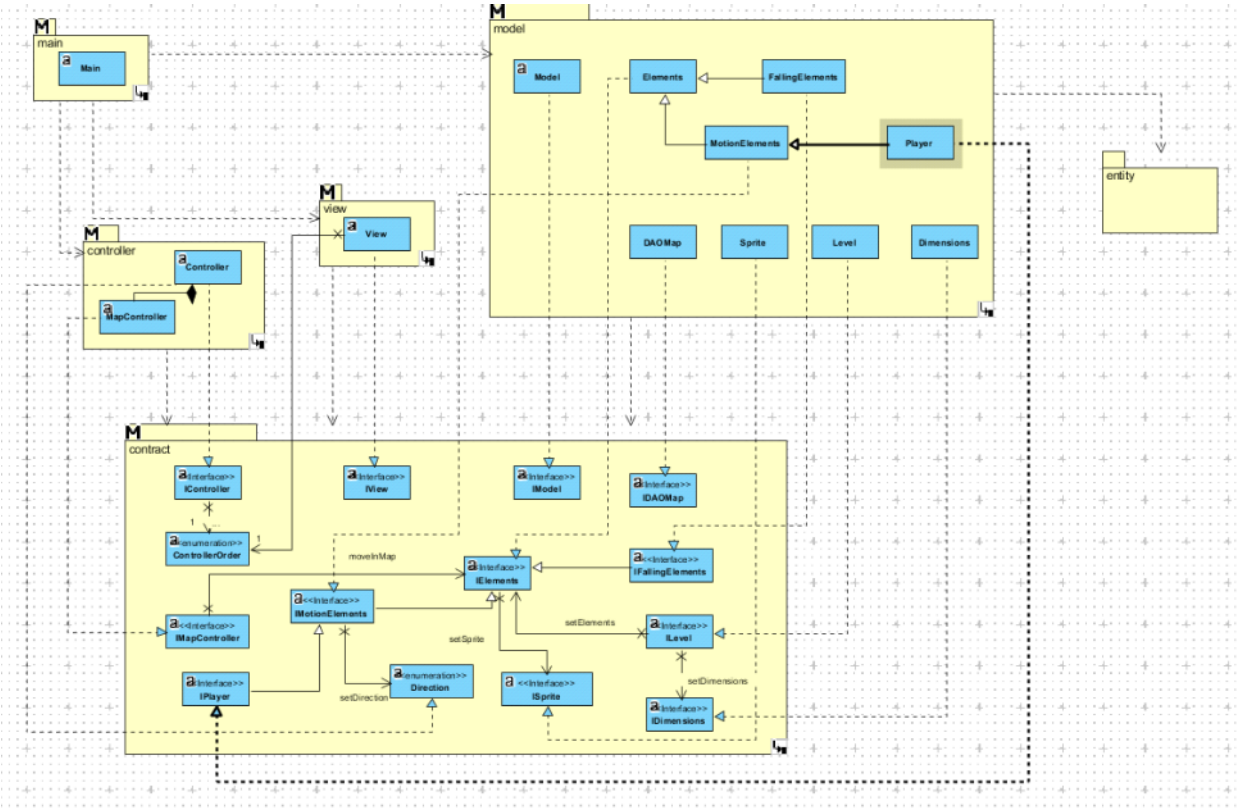
- Contract :



B. Components diagram :



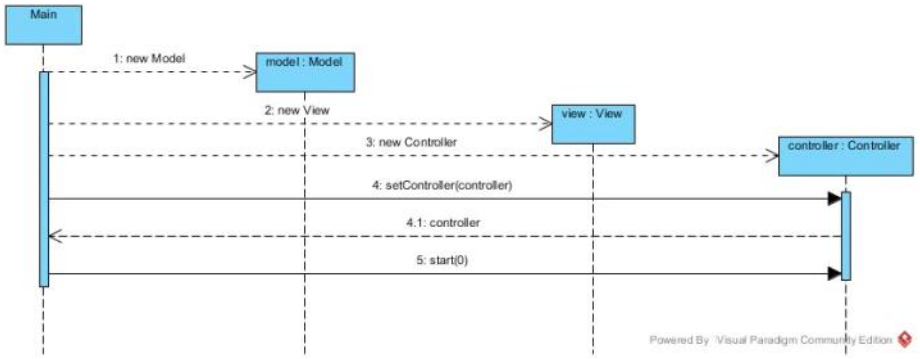
C. Package diagram :



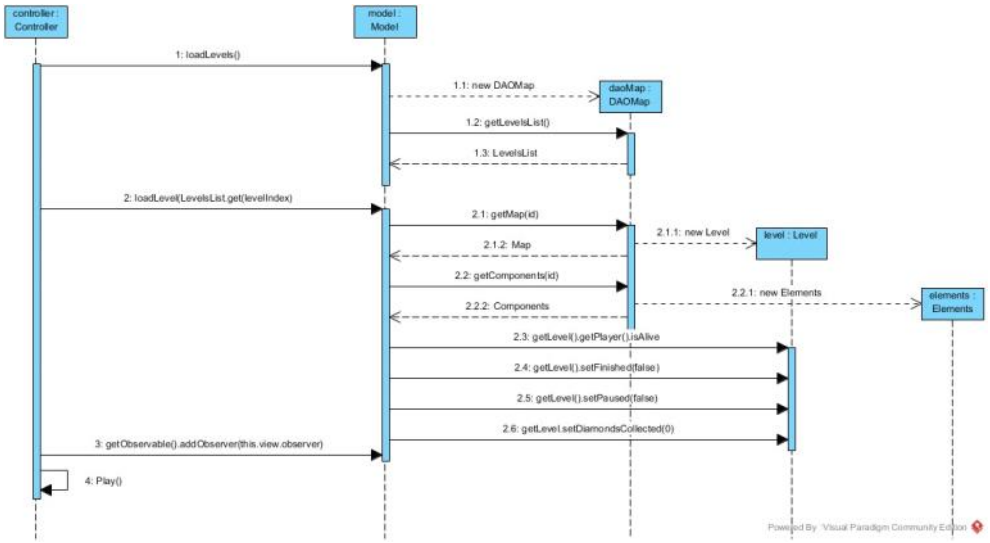
D. Sequence diagram :

We did different sequence diagrams to describe how the system work in our different controllers.

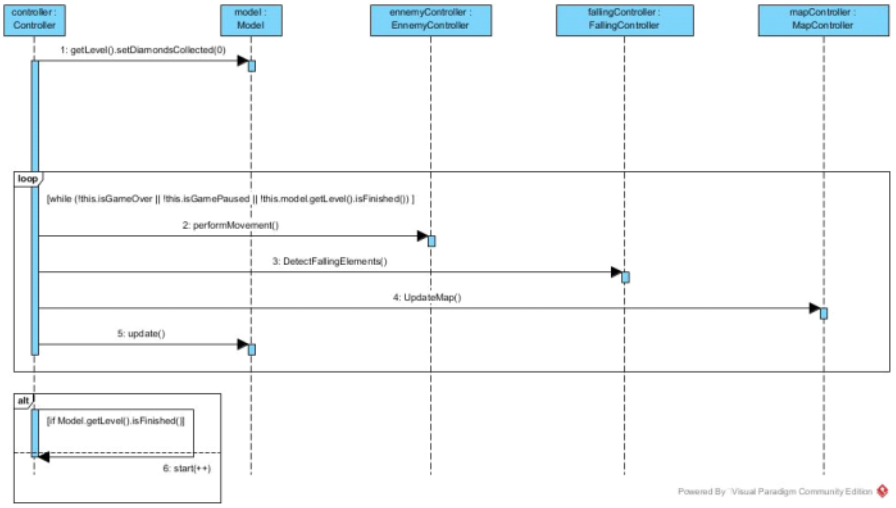
- Main :



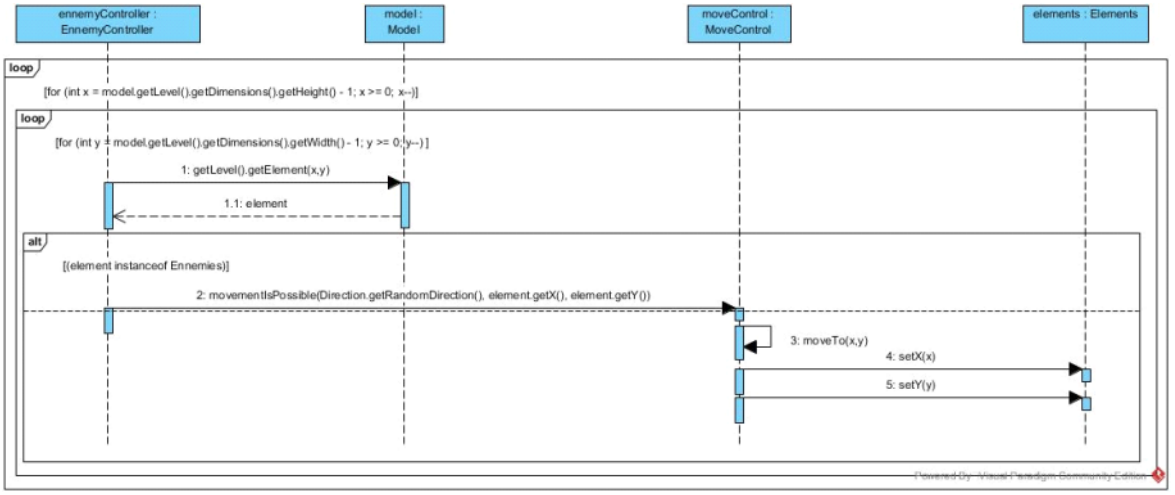
- Controller start :



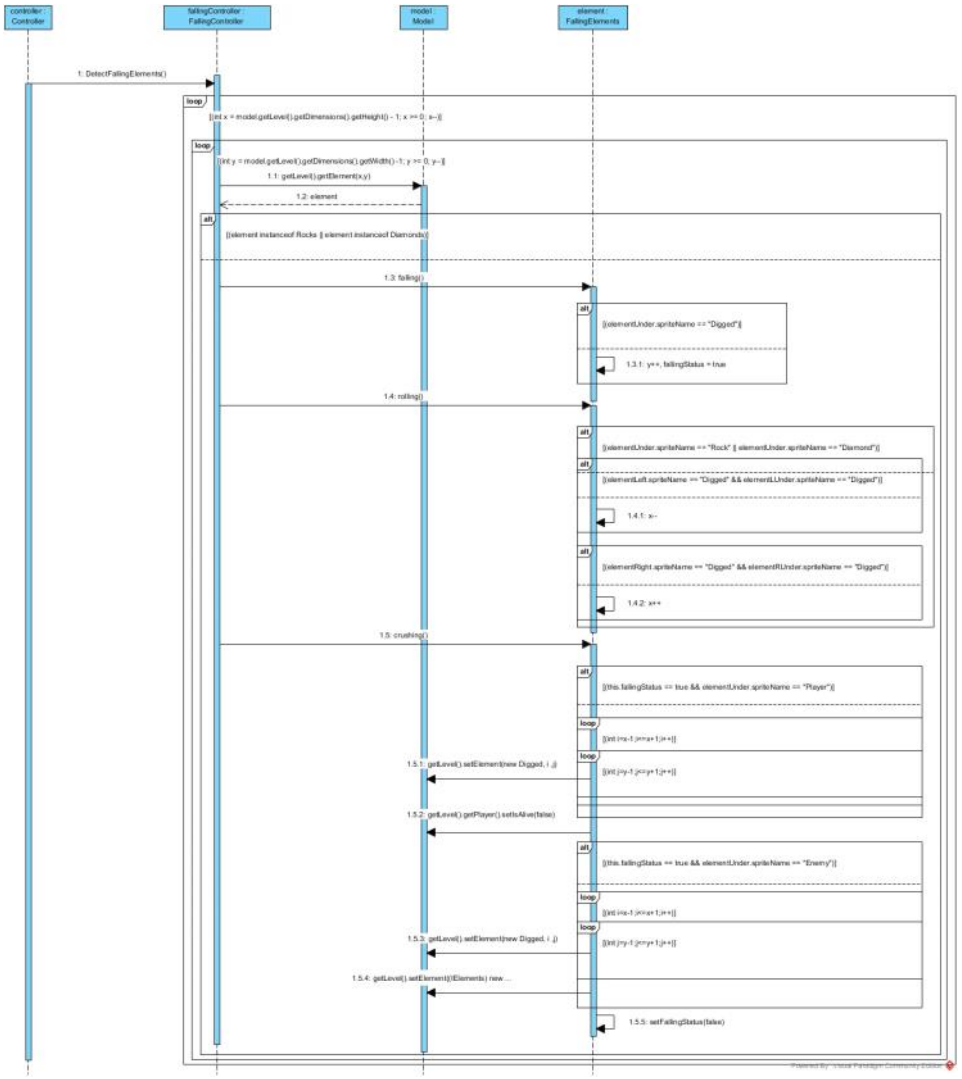
- Controller gameLoop :



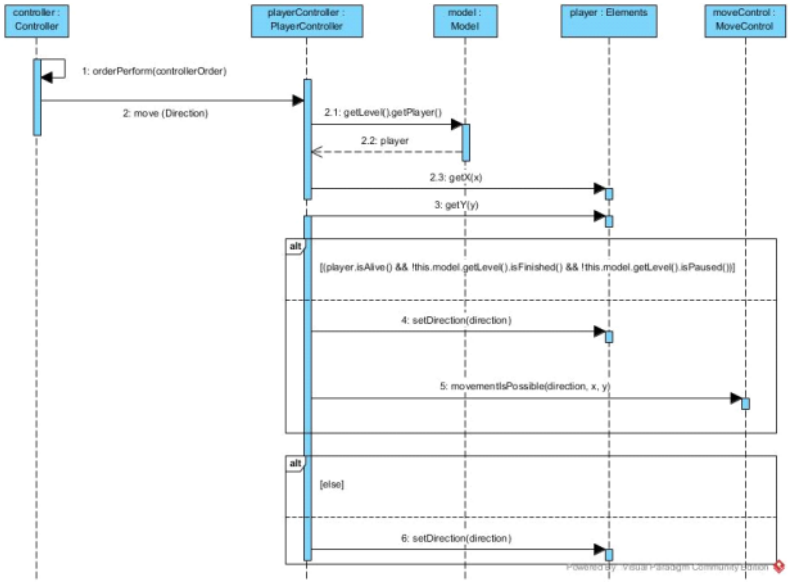
- EnnemyController :



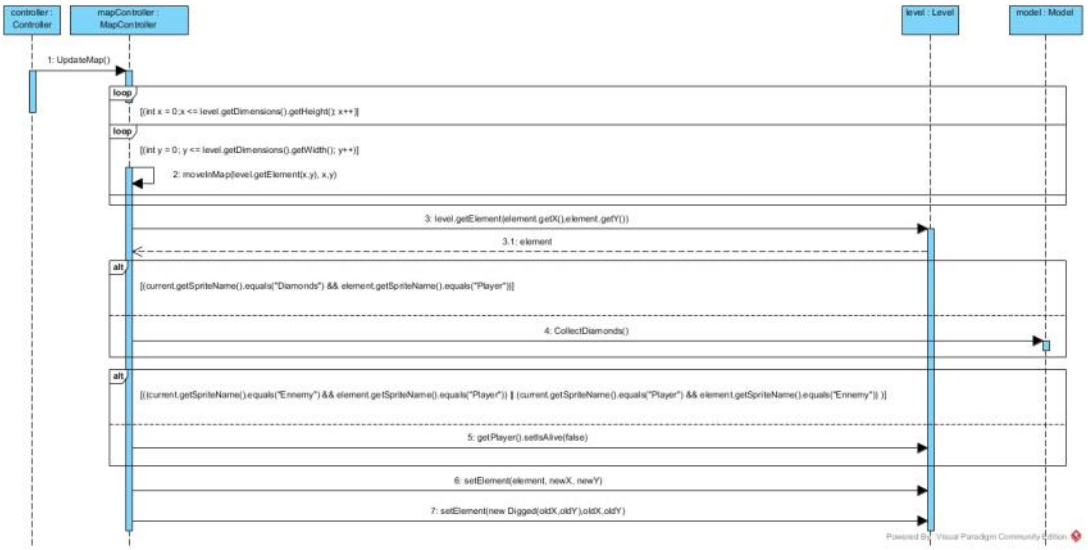
- FallingController :



- PlayerController :

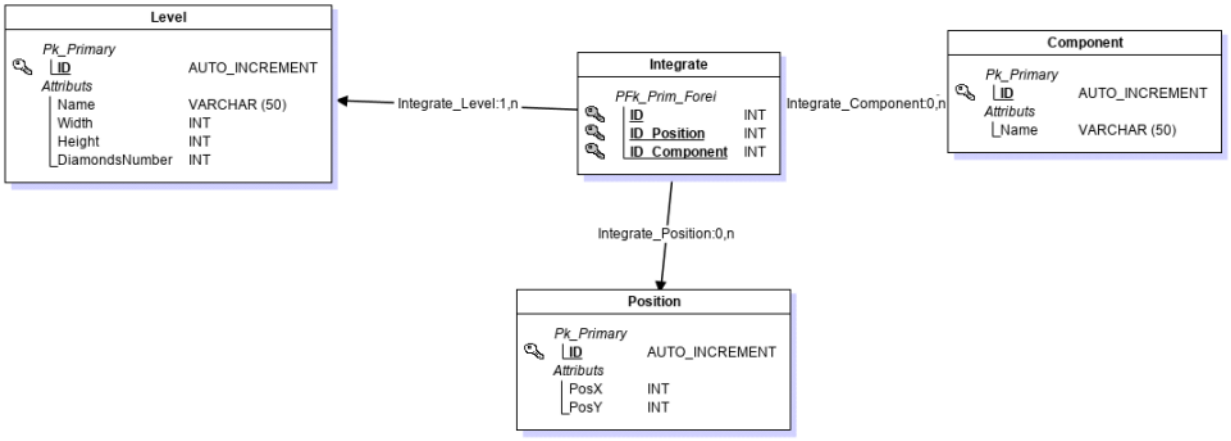


- MapController :



IV. Database

Here is the Data Conceptual Model (MCD) of the database that we have built for the different levels :



And here are the stored procedures used to load the levels and the components on the game :

- Get the list of levels :

```
BEGIN
SELECT DISTINCT level.id
FROM level;
END
```

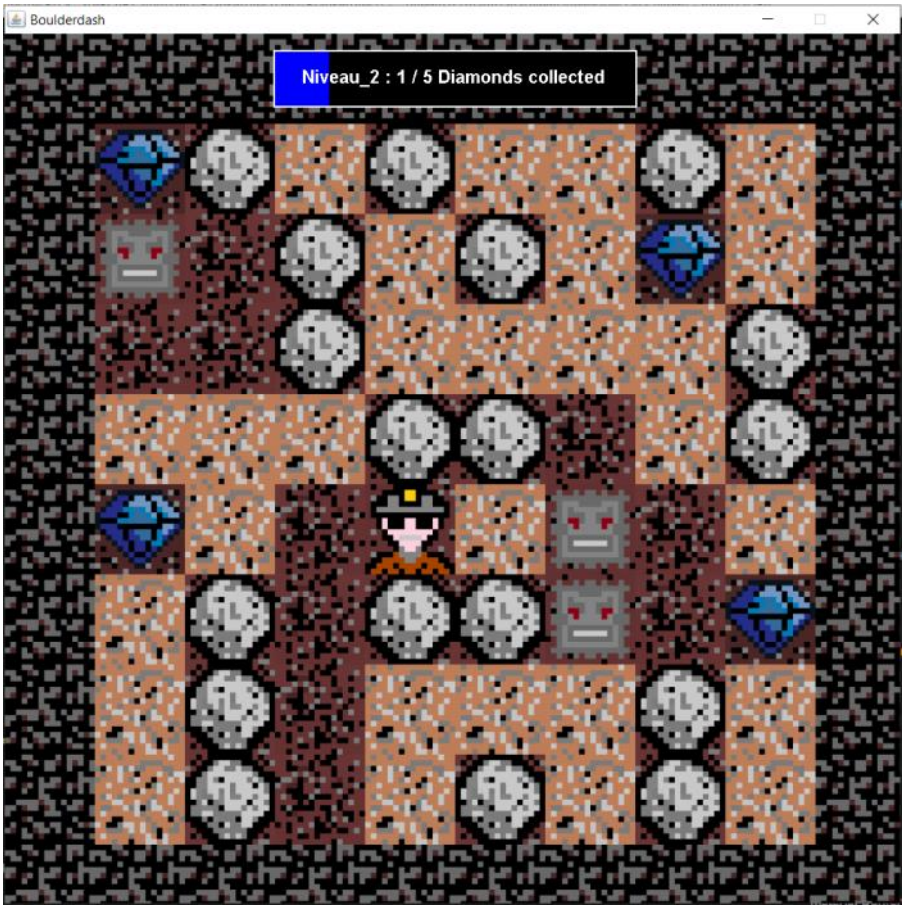
- Choose a level :

```
BEGIN
    SELECT level.Id, level.Name, level.Width,
level.Height, level.DiamondsNumber
    FROM level
    WHERE level.Id = p_Level_id;
END
```

- Get the components of the level :

```
BEGIN
SELECT components.Name, positions.PosX, positions.PosY
FROM components, positions, integrate, level
WHERE components.Id = integrate.Component
AND positions.Id = integrate.Position
AND level.Id = integrate.Level
AND level.Id = p_Level_id;
END
```

V. Final result



VI. Improvement

We had a lot of difficulties in the implementation of our code. So, we have a lot of improvement possible.

First, our code structure isn't the most efficient so our game is pretty slow because we need to check every element of the map to move the rocks/diamonds and ennemies.

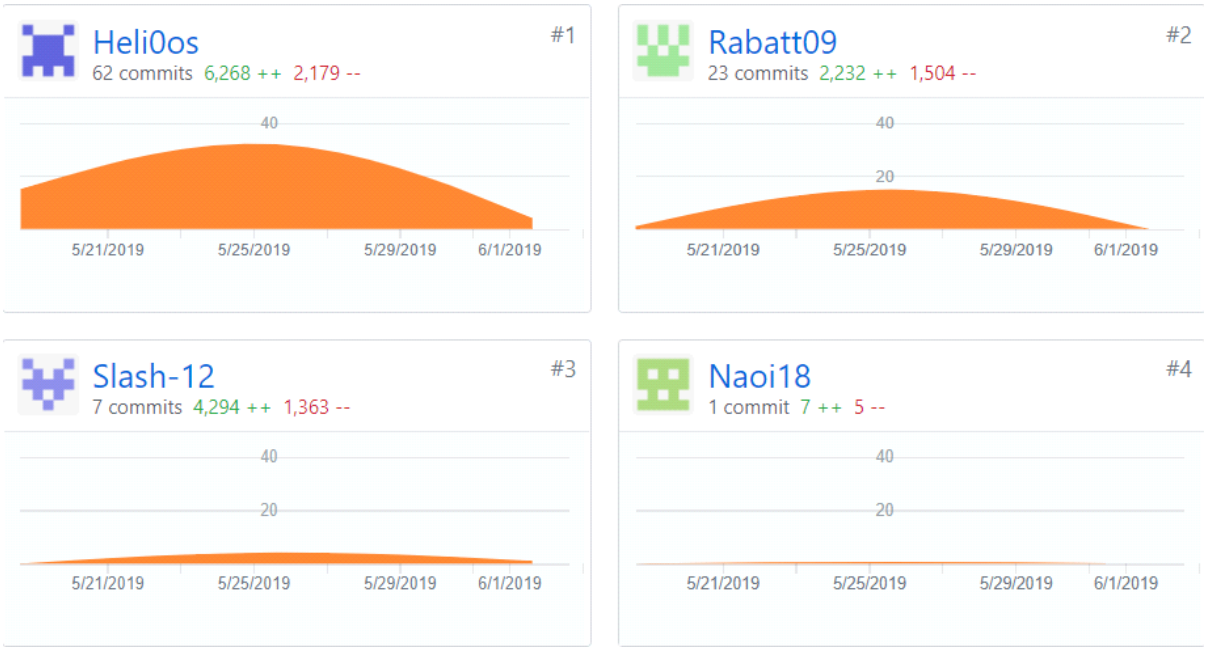
Also, we have different bugs like the ennemy which can dig and collect diamonds. And the rocks fall, crush but don't roll.
We tried to fix those bugs but didn't succeed.

VII. Github summary

Here is a summary of the different commits made by the members of the group on our github :

First, there are the amount of commits for each members of the group on the master branch

Heli0os : AMARY Clément
Rabatt09 : MONTEMONT Théophile
Slash-12 : MOENCH Baptiste
Naoi18 : ALBERT Naomie



And here is a graph for all the branches (master and dev) :

AMARY Clément : 104 commits (A lot of commits because I had a problem with my git connection on IntelliJ and I had to commit the code class per class).
MONTEMONT Théophile : 39 commits
MOENCH Baptiste : 10 commits
ALBERT Naomie : 4 commits

VIII. Group impressions

- **Clément :**

I don't totally agree with my mates because I think that the work atmosphere wasn't as good as possible, we were late on a lot of tasks. For the project, I liked it because it was very understanding and formative.

- **Naomie :**

This project was interesting and having the architecture of HelloWorldMvc helped us with the orientation of the code. Despite some errors in the JDK during the first day of the project we managed to quickly get going. I found that the group atmosphere was good, despite a small lack of communication the first days and the fact that I took time to do some diagrams.

- **Baptiste :**

About the group :

Clément and Théophile were more comfortable than us about the project. Naomie and I had more difficulties to code the project . I think that the work atmosphere was good in the group. Clément and Théophile worked hard on the project trying to code without bugs. I was blocked several times on what I had to do during these past weeks. I also struggled a bit with the Git.
About the project :

It was recommended to us to use the JDK version 12. After minutes of testing this version with Maven, we figured out that the version 12 didn't fit the base of the project. We finally used the version 8 which was the right one. We lost an afternoon trying that which wasn't perfect to start the project.
In a general way I think this project was acceptable. Hopefully, we didn't have another main problem like the previous one. I have nothing bad to say about the project.

- **Théophile :**

for the project itself :

A great deal of difficulty in organizing the distribution and placement of tasks, which has created a delay in overall progress and technical compatibility issues : for the members of the group workload poorly distributed according to the capacity of each.