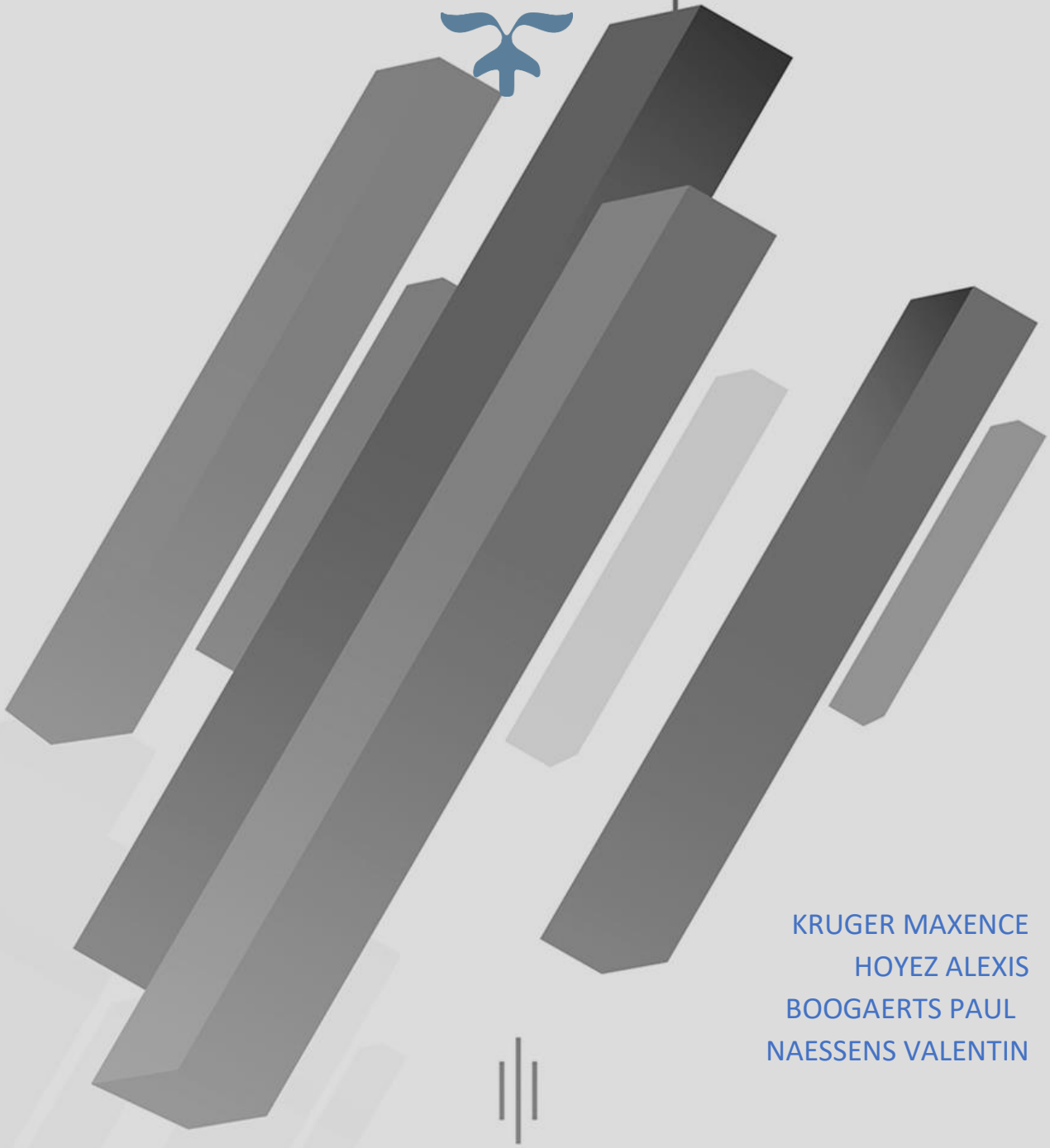
JAVA PROJECT



KRUGER MAXENCE
HOYEZ ALEXIS
BOOGAERTS PAUL
NAESSENS VALENTIN

Summary :

https://github.com/Paulbgt/Projet-Java

I/ Context

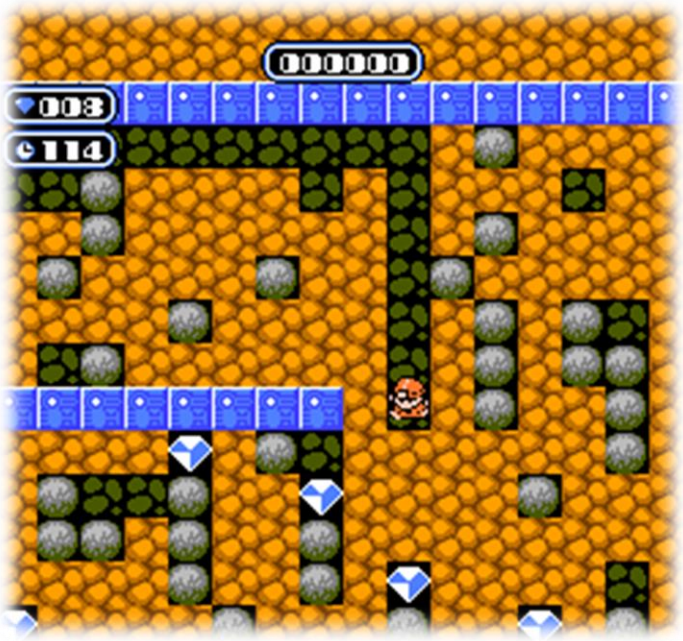
In a first part, we have to acquaint with the subject, we have to create a video game : « Boulder Dash »

The hero « Rockford » have to dig in order to find many diamond to spend a level.

The goal is to work in groupe to realize the game on two weeks.

This game have many constraints :

- Monsters : They can move, kill the hero
- Dirt : Who can be destroyed by the hero
- Wall : This is on the floor but can't interact with something
- Stone : The stone can move, kill the hero and the monsters but when the hero do nothing, it stay immobile.
- Diamond : It can be taken by the hero, he can kill the hero and monsters too, and can appear when a monster has been killed.
- Floor/Background : The floor appear when something is broken or when someone has been killed, it's the thing behind the others (Monsters, Dirt, Wall, Stone, Diamond, Hero). We can see it too when something or someone is moving.
- In fact, the hero can move and dig.



In our opinion, the situation might be shown with a kind of dependency matrix, in order to show all of the links between the differents elements :

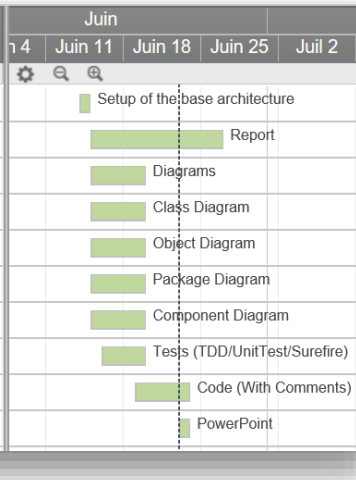
So, we can divide our diagram on 5 big parts:

	Hero	Monster	Dirt	Wall	Stone	Diamond	Floor
Hero		X			X	X	
Monster	X				X	X	
Dirt	X						
Wall							
Stone	X						
Diamond	X						
Floor	X	X			X	X	

- First, the Main part where we'll start the project.
- Secondly, the View part where we put the Jpanel and Jframe, it's the display part:
  - Jframe corresponds to the window in general.
  - Jpanel, it displays the score, the game so he "fills" the Jframe.
- Thirdly, the Controller part where we'll show the orders requested to be executed:
  - For example, the things linked by the Keylistener who'll allow the interaction between keyboard and the video game.
- Fourthly, the contract which allows to put all the interfaces who is an abstract class where methods are abstracts and where attributes are constants.
- At last, the Model, this is the most important part where we can put :
  - The differents Objects (Rockford : The Hero, The Monsters, Dirt, Wall, etc…)
  - The Map who'll be load in the Model, Link between database and java.
  - The Movements (Monsters, Hero).

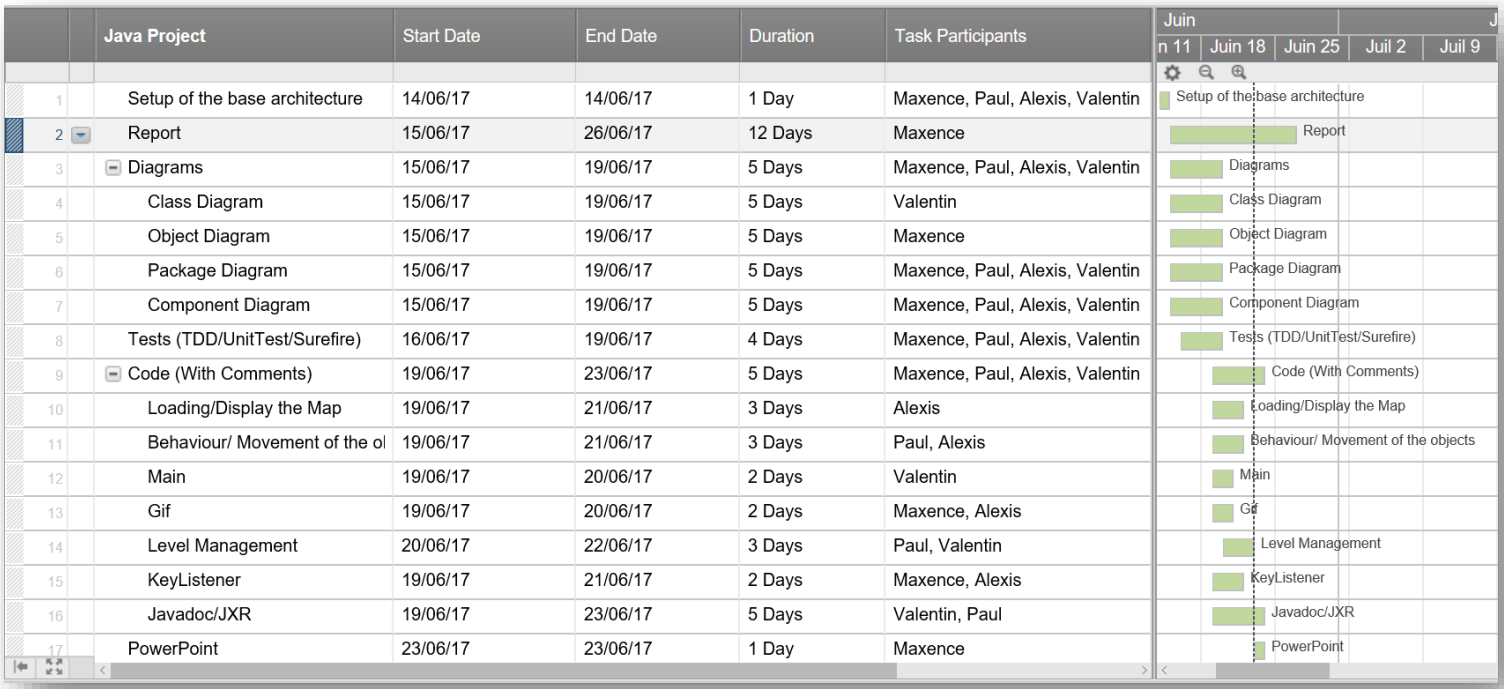
Our provisional schedule for this project :

	Java Project	Start Date	End Date	Duration	Task Participants
1	Setup of the base architecture	14/06/17	14/06/17	1 Day	Maxence, Paul, Alexis, Valentin
2	Report	15/06/17	26/06/17	12 Days	Maxence
3	Diagrams	15/06/17	19/06/17	5 Days	Maxence, Paul, Alexis, Valentin
4	Class Diagram	15/06/17	19/06/17	5 Days	Maxence, Paul, Alexis, Valentin
5	Object Diagram	15/06/17	19/06/17	5 Days	Maxence
6	Package Diagram	15/06/17	19/06/17	5 Days	Maxence, Paul, Alexis, Valentin
7	Component Diagram	15/06/17	19/06/17	5 Days	Maxence, Paul, Alexis, Valentin
8	Tests (TDD/UnitTest/Surefire)	16/06/17	19/06/17	4 Days	Maxence, Paul, Alexis, Valentin
9	Code (With Comments)	19/06/17	23/06/17	5 Days	Maxence, Paul, Alexis, Valentin
17	PowerPoint	23/06/17	23/06/17	1 Day	Maxence



Finally we opted for a more detailed diagram :

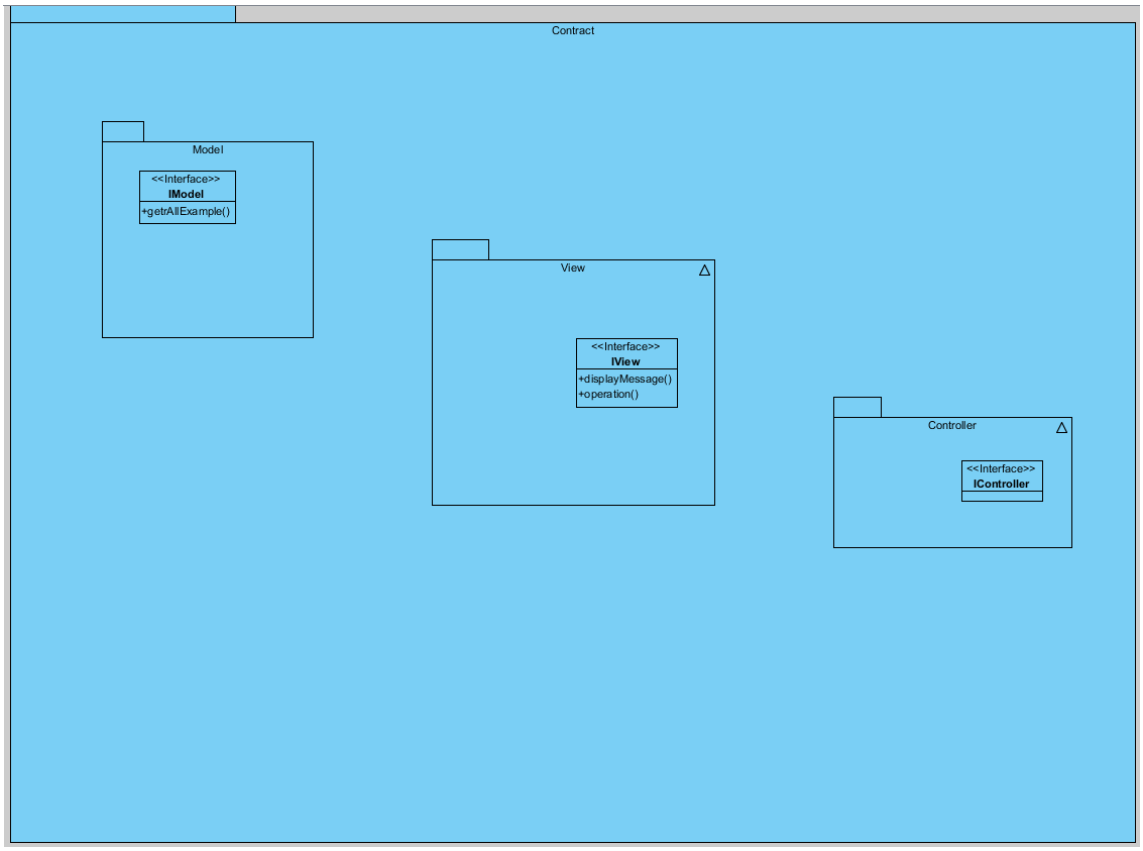
	Java Project	Start Date	End Date	Duration	Task Participants
1	Setup of the base architecture	14/06/17	14/06/17	1 Day	Maxence, Paul, Alexis, Valentin
2	Report	15/06/17	26/06/17	12 Days	Maxence
3	Diagrams	15/06/17	19/06/17	5 Days	Maxence, Paul, Alexis, Valentin
4	Class Diagram	15/06/17	19/06/17	5 Days	Valentin
5	Object Diagram	15/06/17	19/06/17	5 Days	Maxence
6	Package Diagram	15/06/17	19/06/17	5 Days	Maxence, Paul, Alexis, Valentin
7	Component Diagram	15/06/17	19/06/17	5 Days	Maxence, Paul, Alexis, Valentin
8	Tests (TDD/UnitTest/Surefire)	16/06/17	19/06/17	4 Days	Maxence, Paul, Alexis, Valentin
9	Code (With Comments)	19/06/17	23/06/17	5 Days	Maxence, Paul, Alexis, Valentin
10	Loading/Display the Map	19/06/17	21/06/17	3 Days	Alexis
11	Behaviour/ Movement of the ol	19/06/17	21/06/17	3 Days	Paul, Alexis
12	Main	19/06/17	20/06/17	2 Days	Valentin
13	Gif	19/06/17	20/06/17	2 Days	Maxence, Alexis
14	Level Management	20/06/17	22/06/17	3 Days	Paul, Valentin
15	KeyListener	19/06/17	21/06/17	2 Days	Maxence, Alexis
16	Javadoc/JXR	19/06/17	23/06/17	5 Days	Valentin, Paul
17	PowerPoint	23/06/17	23/06/17	1 Day	Maxence



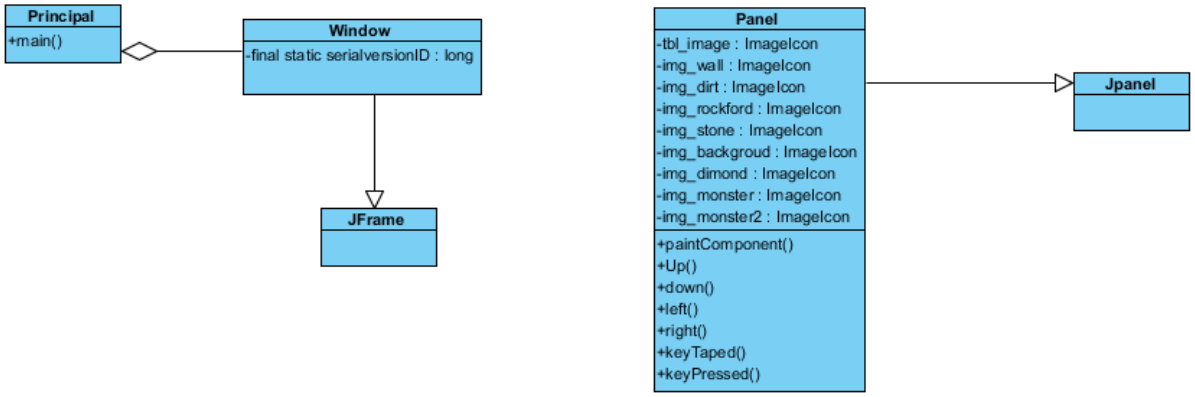
II/ The Project

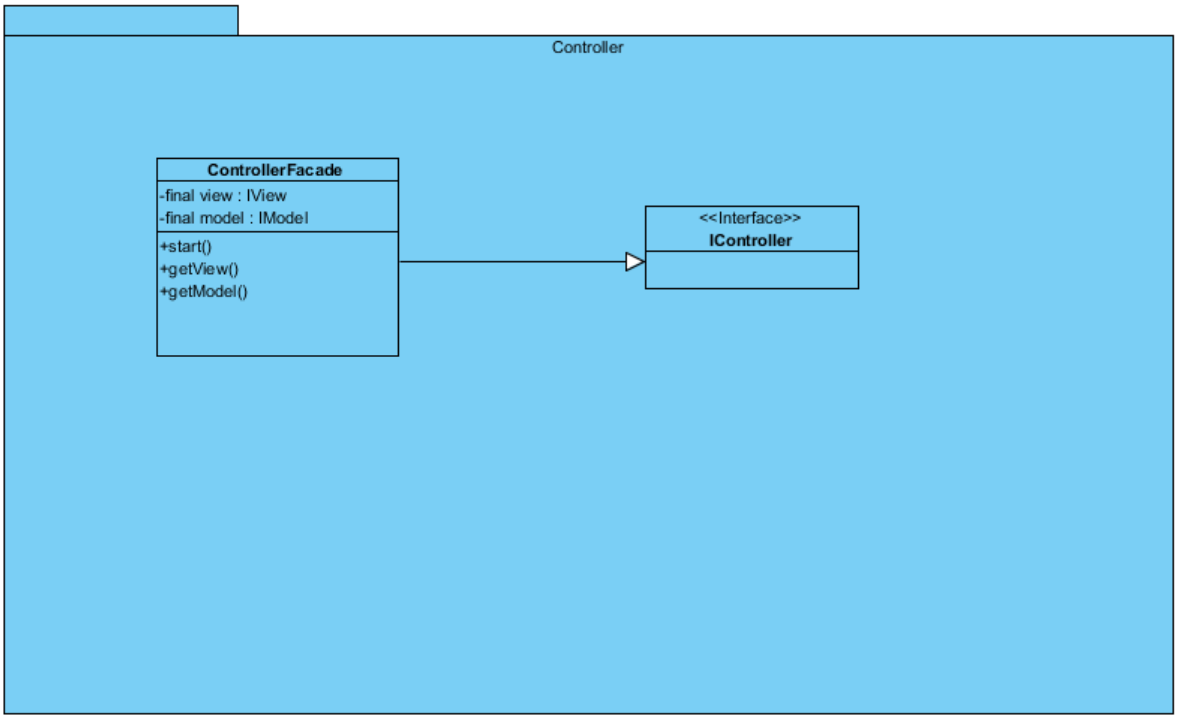
A-Diagrams

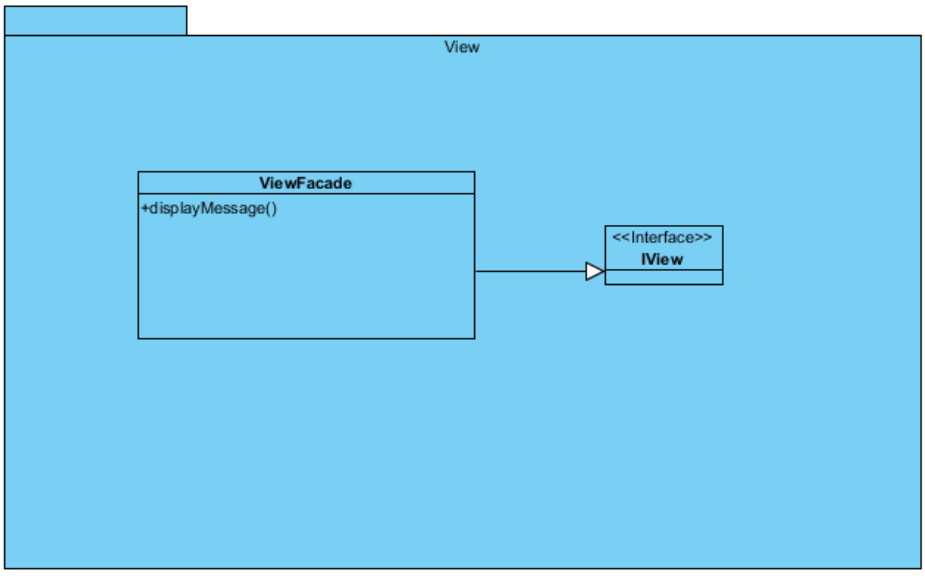
1. Class Diagram

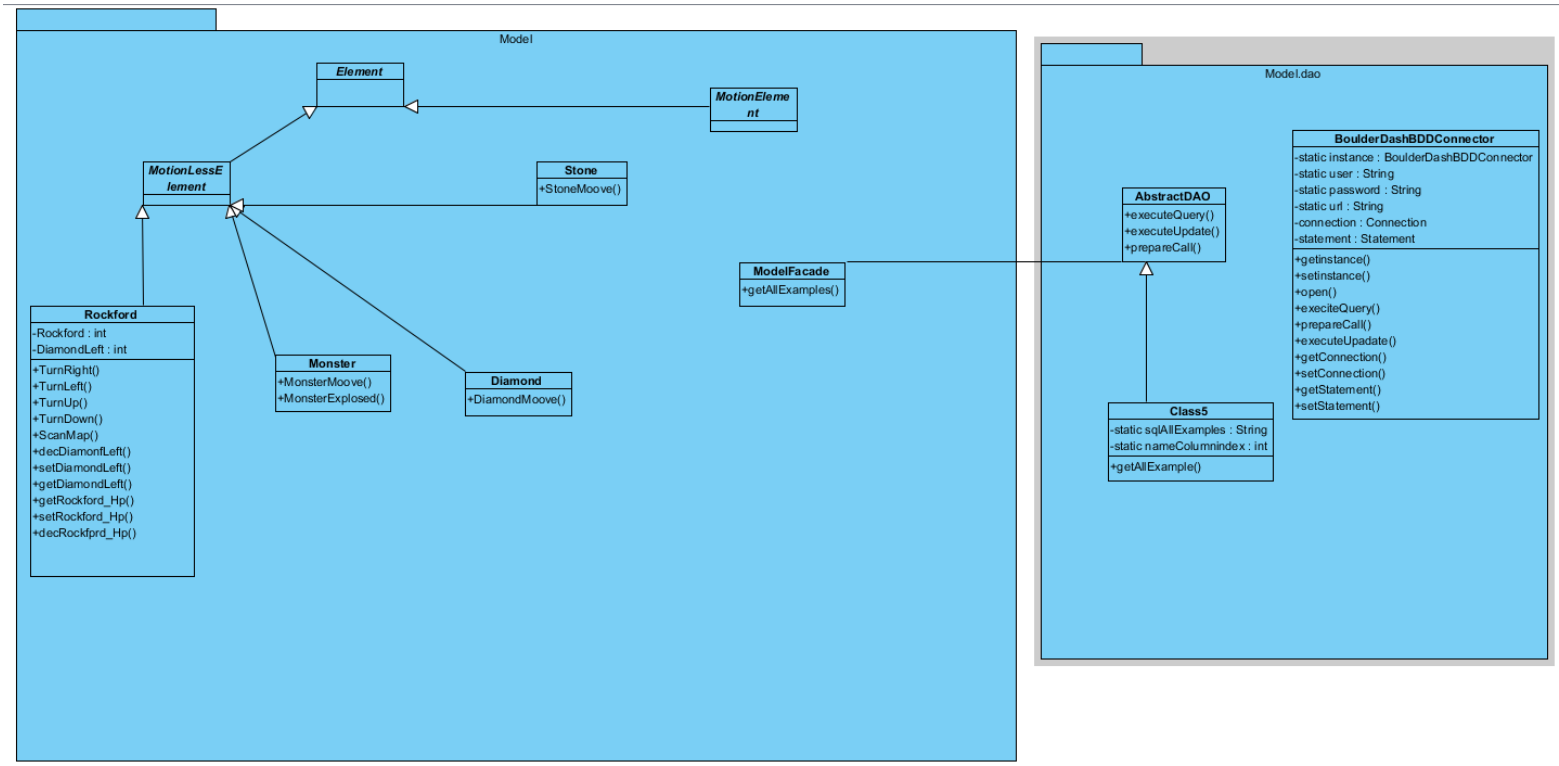


Main

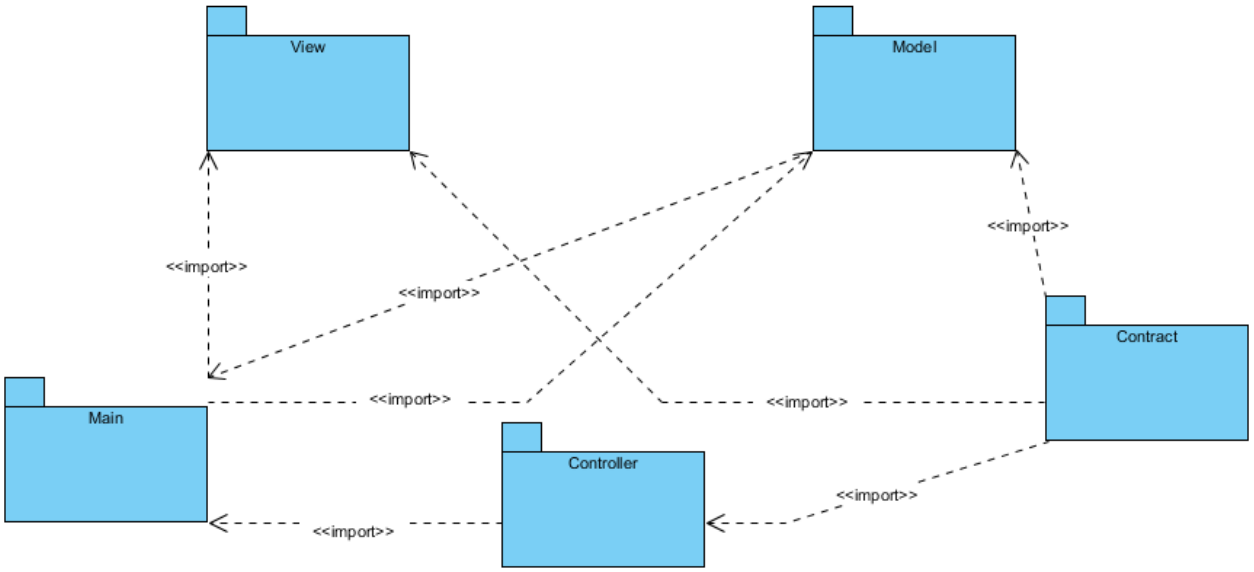

Principal
+main()
Window
-final static serialversionID : long
JFrame
Panel
-tbl_image : ImageIcon
-img_wall : ImageIcon
-img_dirt : ImageIcon
-img_rockford : ImageIcon
-img_stone : ImageIcon
-img_backgroud : ImageIcon
-img_dimond : ImageIcon
-img_monster : ImageIcon
-img_monster2 : ImageIcon
+paintComponent()
+Up()
+down()
+left()
+right()
+keyTaped()
+keyPressed()
Jpanel


Controller
ControllerFacade
-final view : IView
-final model : IModel
+start()
+getView()
+getModel()
<<Interface>>
IController

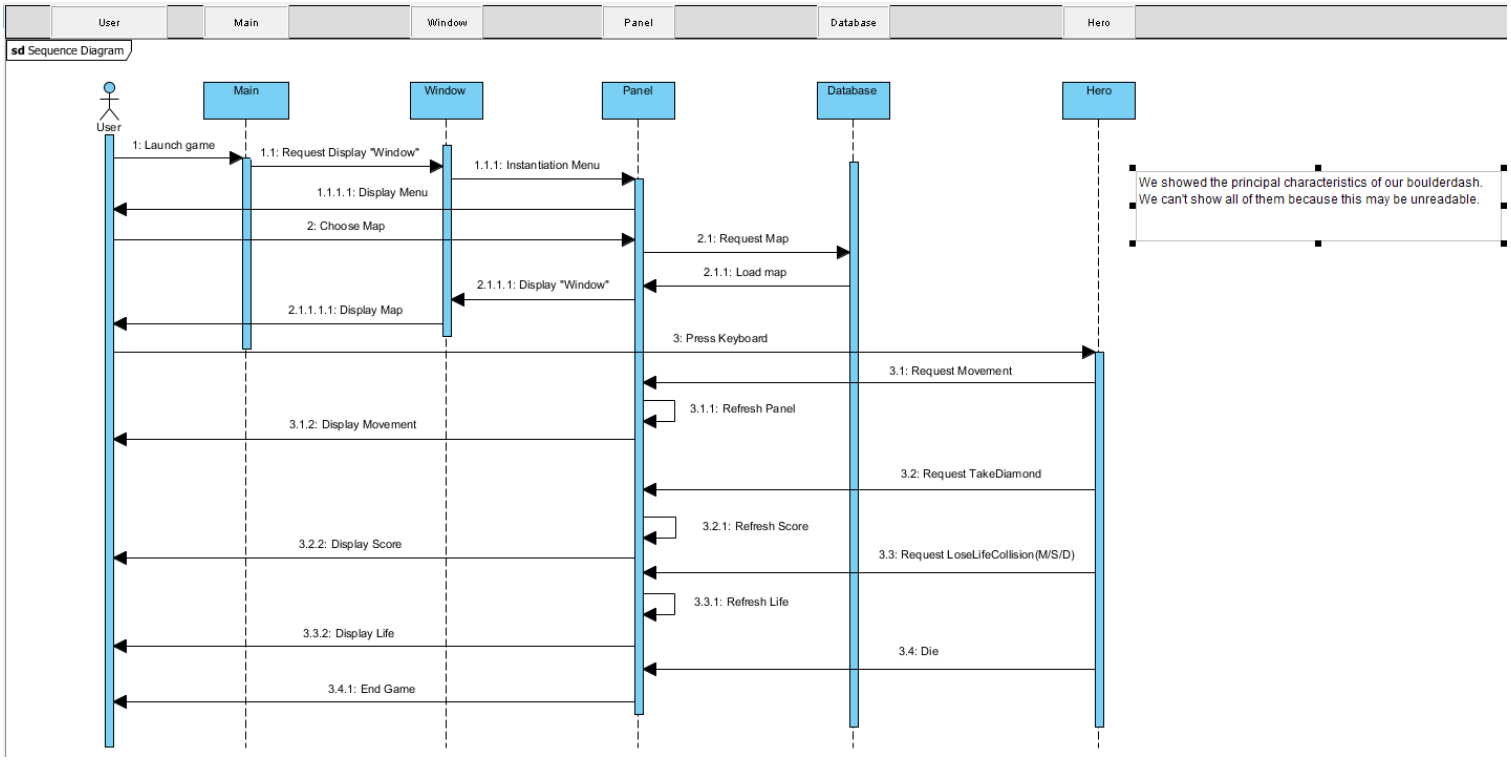

View
ViewFacade
+displayMessage()
<<Interface>>
IView



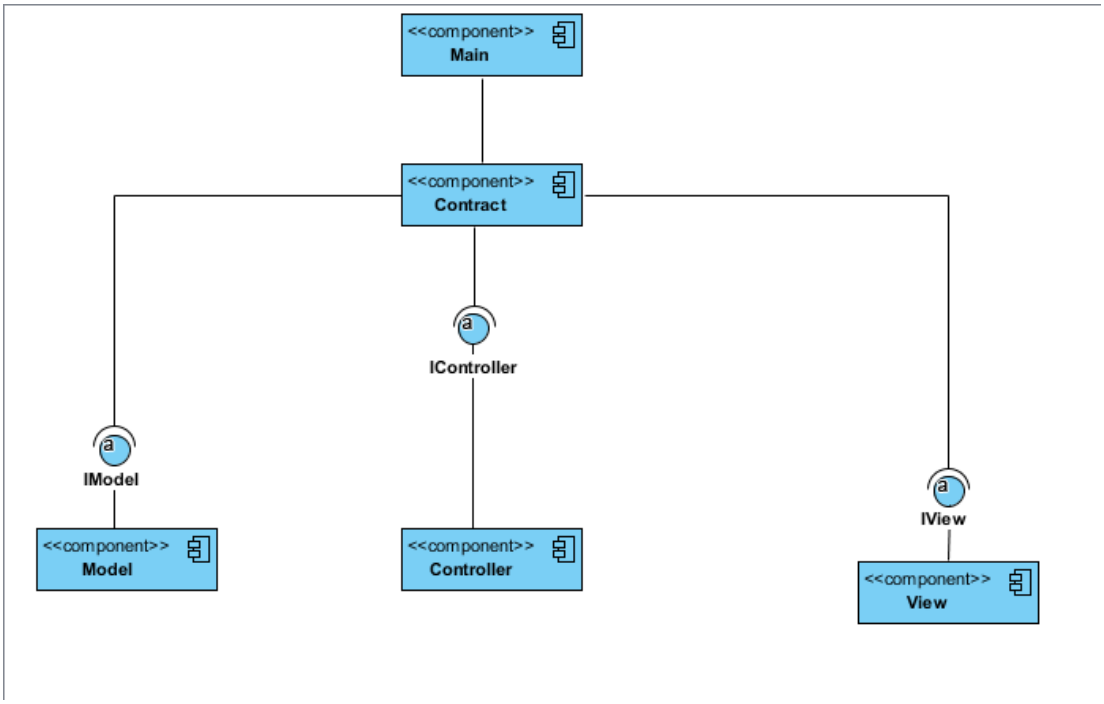
2. Package Diagram



3. Sequence Diagram



4. Component Diagram



B- Code

At the beginning of this part we have to know « where are we going ? ».

We need to do a boulderdash in java and to respect the contraints.

When we have established our path for the game we can therefore begin to code:

First, we should do a map. We did it stored in a database, in characters.

Then we can start the code, we must associate the character with the desired picture (Wall, Monsters, Dirt, Hero, etc.…).

After this, one of the most difficult part is to load the map and to reveal the it in the form of pictures.

When we did it, we must insert the movement knowing that the objects have different behaviors, so it's important to separate them to be able to call them.

The movement and the loading are in the "panel "class.

In our code, we should manage the different behaviors, here is what we have do during this time of project:

- We have several types of monsters, thereby they must act differently.
- The Hero can dig dirt.
- The Hero can die by stone, diamond and monsters (Collision)
- The Hero can't cross stones and walls but can push stones.
- The Hero can pick up the diamonds.
- Stones and Diamonds can fall.
- When Stones are on Diamonds or other Stones, it falls in an empty adjacent case.
- The Monsters can be killed by Stones or Diamonds when it falls on them.

We have other things to do in our code too, and we did somethings more than what it was requested:

- We must be able to change map, we did it via a menu when we launch boulderdash.
- We have put the gif of Stone, of Diamond to have a more beautiful game.
- The game end when the hero dies and when the diamonds are all picked up.

That explain the results of our project.

To explain our code, we have done a javadoc who explain our methods in GitHub.

III/ Conclusion

A- General overview

Good group overall, each member was very involved in the project, everyone was productive and very good results despite the expectations of the designer. The project was interesting, and we had a beautiful group cohesion in contrary to what one might have thought at the outset. The project was concrete and pleasant to realize.

Difficulties : It was difficult for us to use Maven, and some problems with GitHub.

B- Personnal review

Paul : I found the project interesting despite that I found the part of software engineering a little too big for our first approach to the P.O.O. I have developed my knowledge in Java and software engineering.

Maxence : I found the project interesting, i learned a lot of things during this project and I appreciated it in contrary I don't really like developement. I liked the group cohesion, it was a pleasure to work with them. I can add that the results was great and I enjoyed program in java.

Alexis : The project is really interesting, it allowed me to apply and improve the knowledge learned during prosits, but especially during workshops and exercise baskets. The game we need to create requires a lot of knowledge and do several searches on the internet, which I really liked. Moreover, we were rather free in the design of the games, we were free in the choice of sprites, maps, etc ... The group atmosphere was very pleasant which allowed us to move fast and well from the first day.

Valentin : The java Project was an experience with positive and negative point. Indeed, I liked this project because the objective, create a game was fun and concrete project. Furthermore, the ambience in the group was better than I think at the beginning. Moreover, I think use so much genius software was a bad idea, we practice java since a month ago so we are not very strong and it was too hard to mix the different logical with our level. I think it will better to practice some less in this project to be very comfortable with and see the other in an other years.

C- Prospects for improvement

Improve the MVC and the Unit Test.