JAVA PROJECT : BOULDERDASH GAME CONCEPTION

Project actors

OULDHAMOU Chakib **Project manager:**

HOCINE Walid

MRABENT meriem

AOUDJHANE Wassim

Première Année - Cycle préparatoire

Ecole Supérieure d'Informatique Appliqué (EXIA)

Juin 2017

Table of Contents

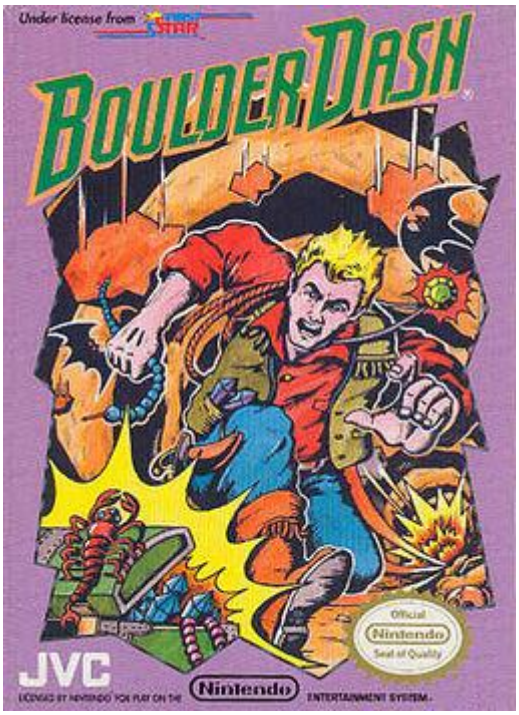
Introduction

Project presentation

For this object-oriented programing project, our client asks us to rewrite a famous game from 1984, “Boulder Dash”, a game where you must collect diamonds to move forward through the game while avoiding enemies and boulders.

we have to code it in JAVA and use Maven, which is a build automation tool used primarily for Java projects and a database to stock our different levels. In order to facilitate teamwork, we also need to use Git Hub which is an online versioning system.

Finally, before we start coding, we need to model our program using diagrams such as class diagrams, components



Study of the need

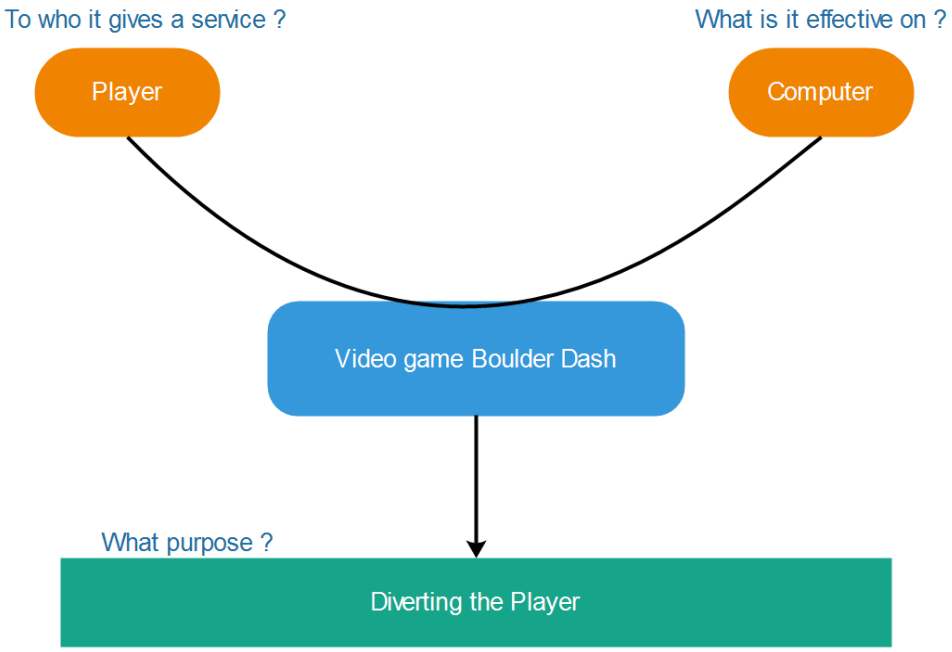


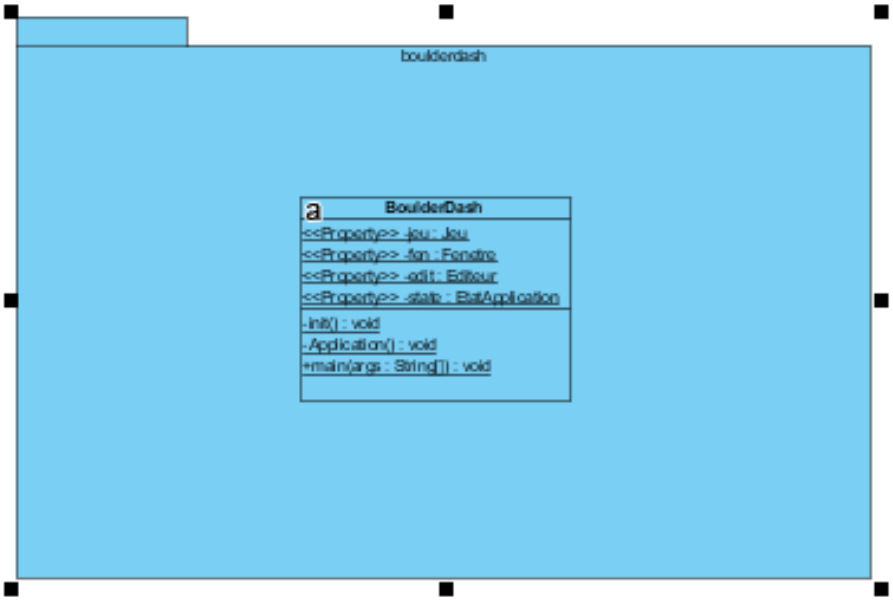
Figure 1Service Diagram

How the game work ?

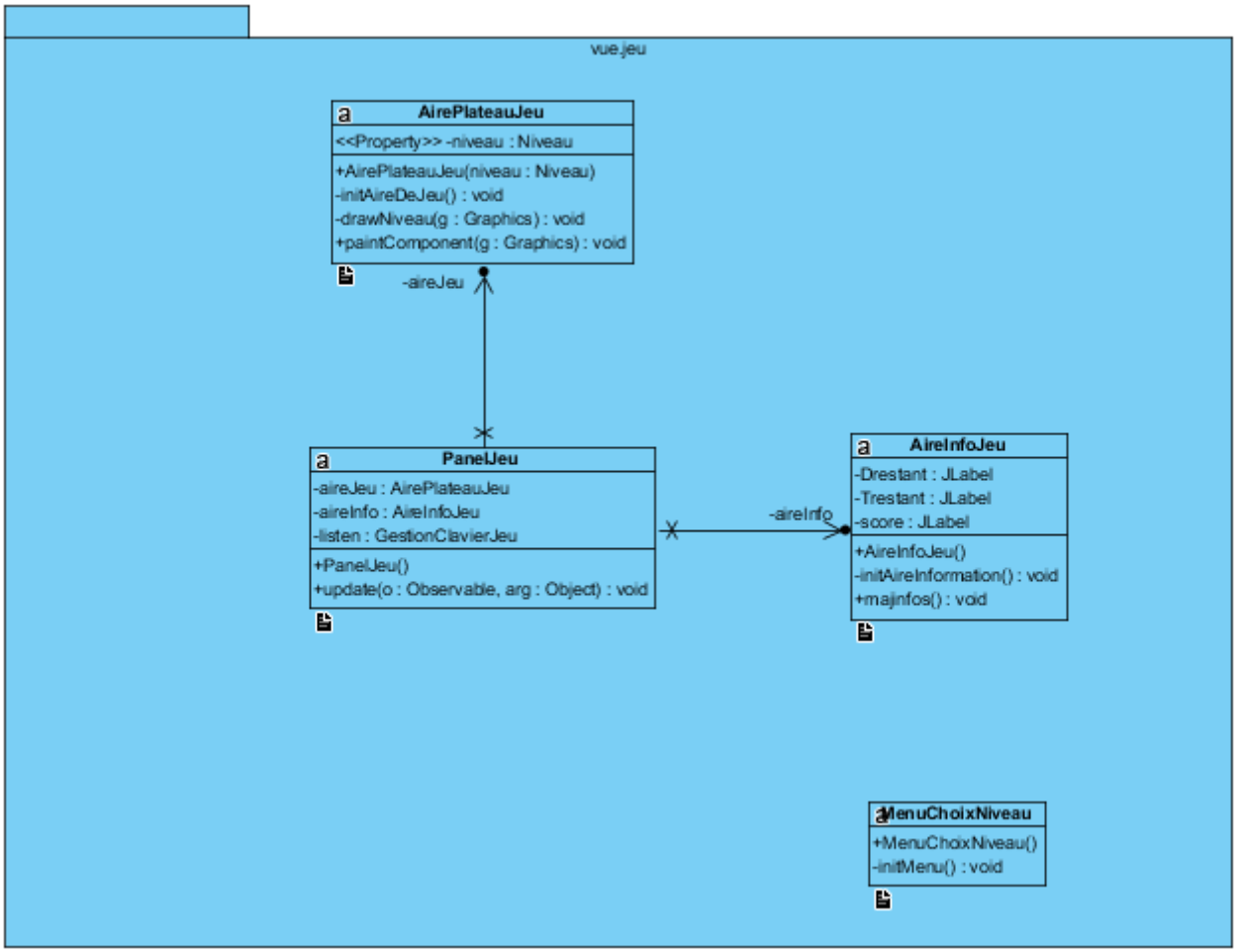
Here are some diagramms created using Visual Paradigm showing the structure of the code and also the architecture and mecanisme of the video game

1. **Class Diagram per Package**

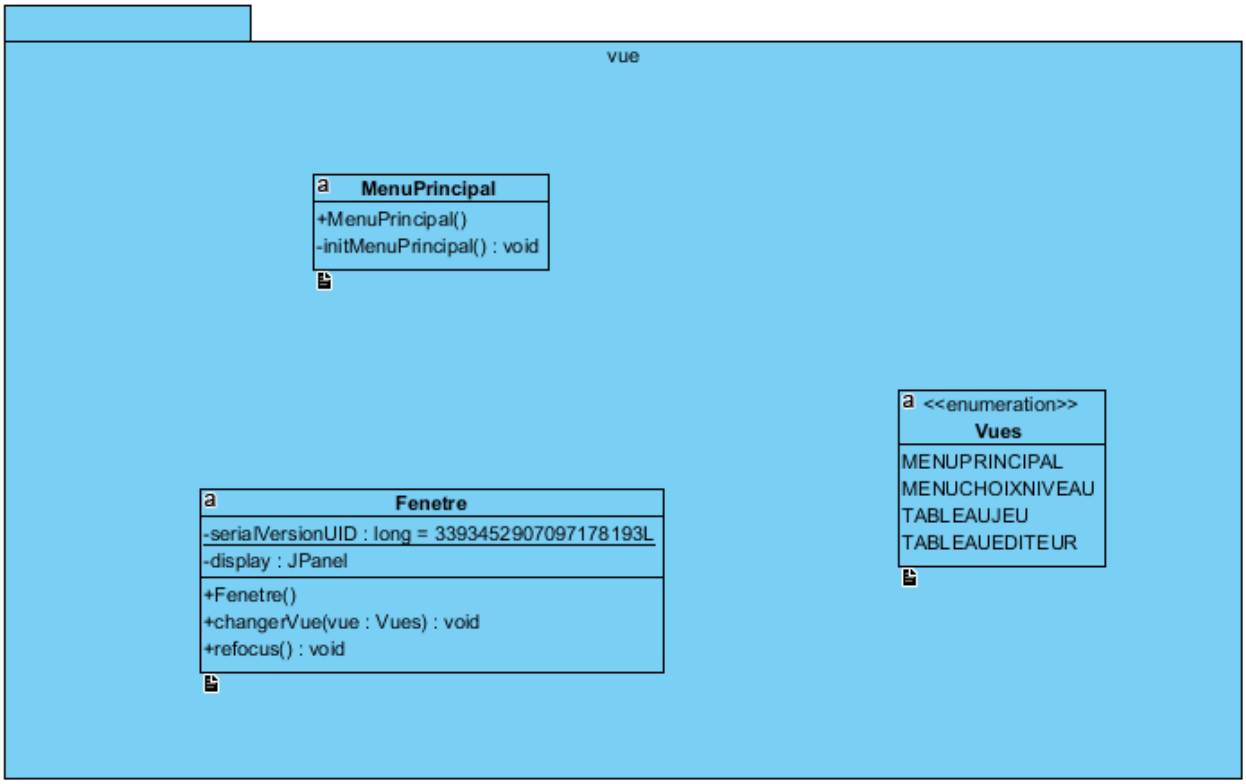
- **Package boulder dash:**



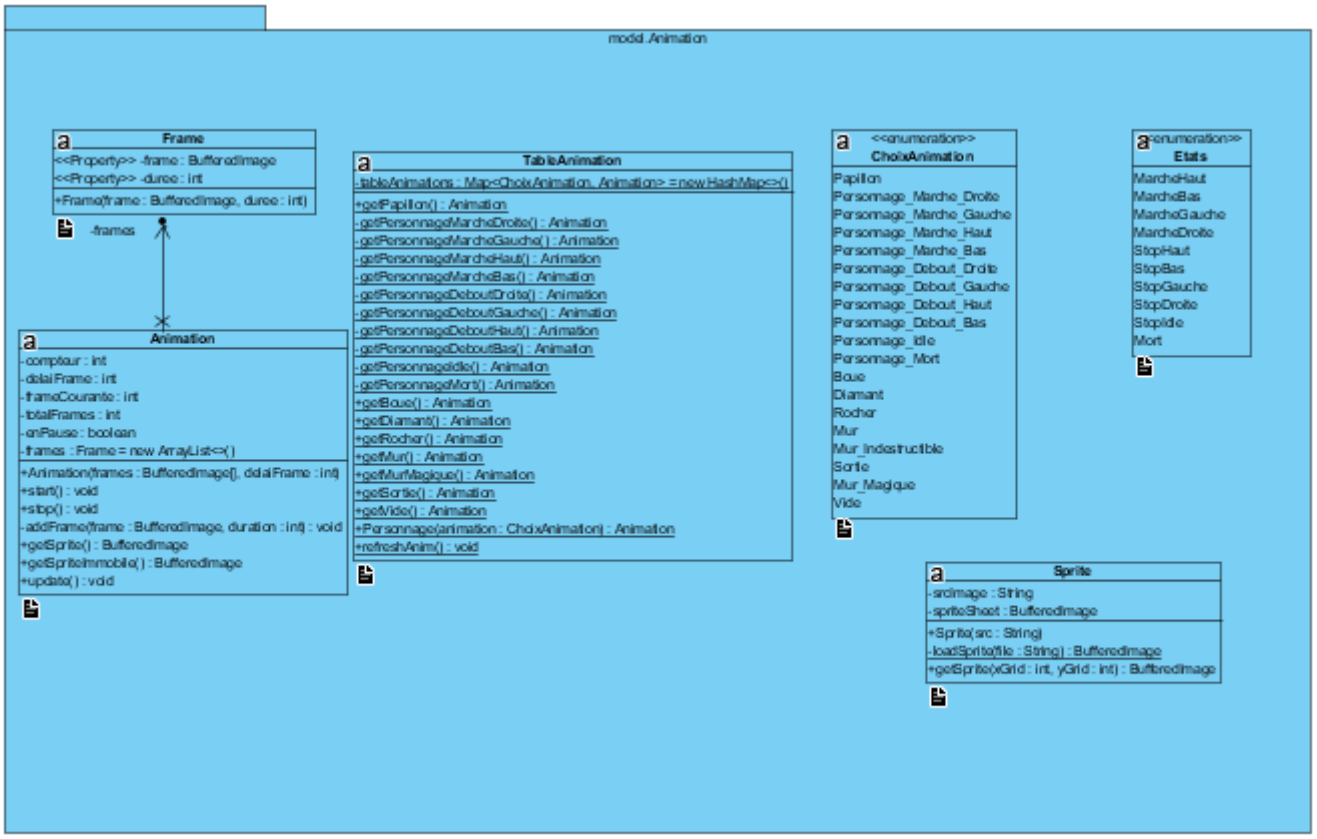
- **Package vue.jeu:**


vue.jeu
AirePlateauJeu
<<Property>> -niveau : Niveau
+AirePlateauJeu(niveau : Niveau)
-initAireDeJeu() : void
-drawNiveau(g : Graphics) : void
+paintComponent(g : Graphics) : void
-aireJeu
PanelJeu
-aireJeu : AirePlateauJeu
-aireInfo : AireInfoJeu
-listen : GestionClavierJeu
+PanelJeu()
+update(o : Observable, arg : Object) : void
-aireInfo
AireInfoJeu
-Drestant : JLabel
-Trestant : JLabel
-score : JLabel
+AireInfoJeu()
-initAireInformation() : void
+majInfos() : void
MenuChoixNiveau
+MenuChoixNiveau()
-initMenu() : void

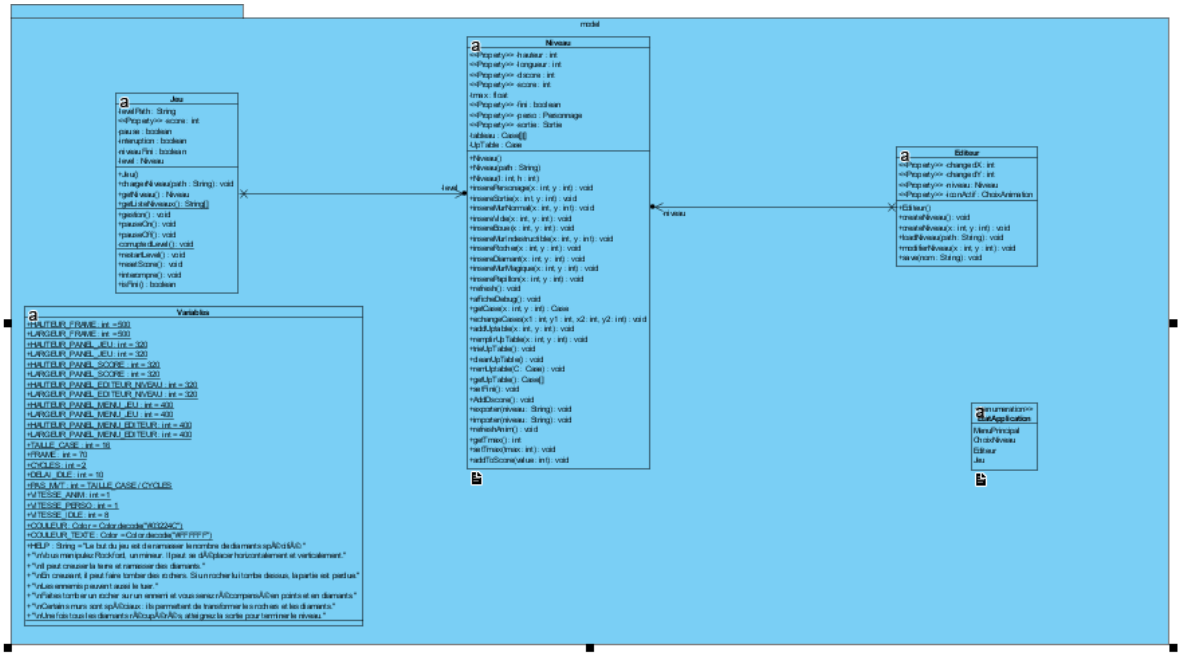
- **Package vue:**



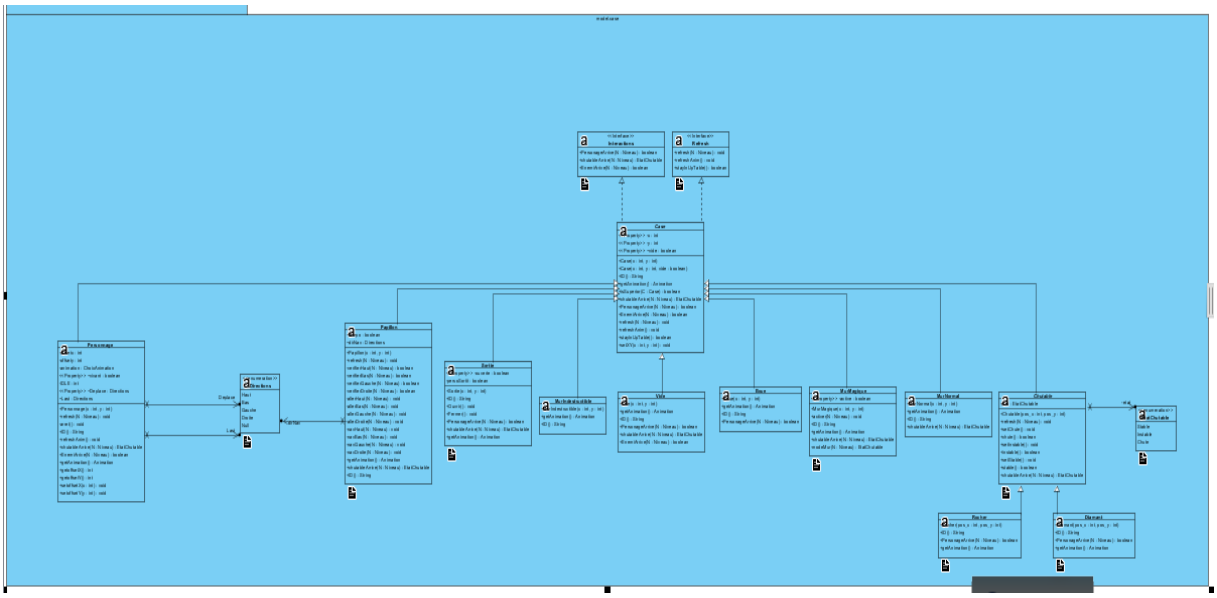
- **Package ModelAnimation:**



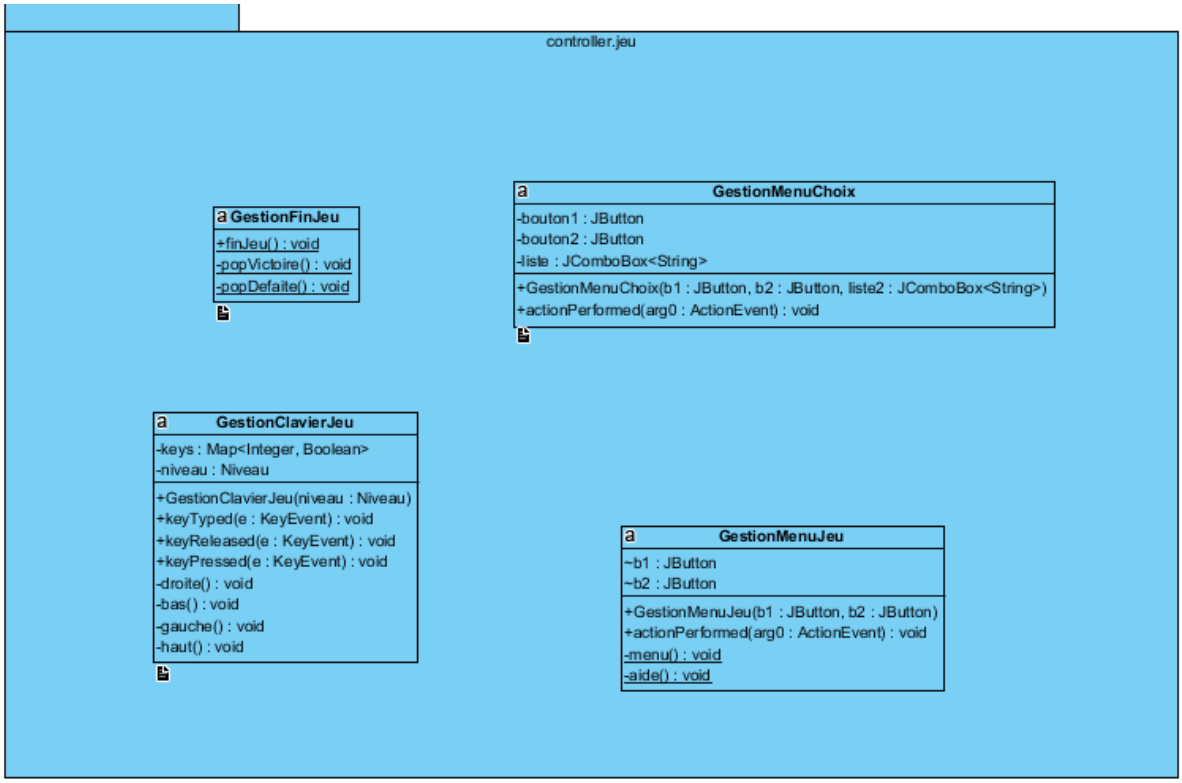
- **Package model:**



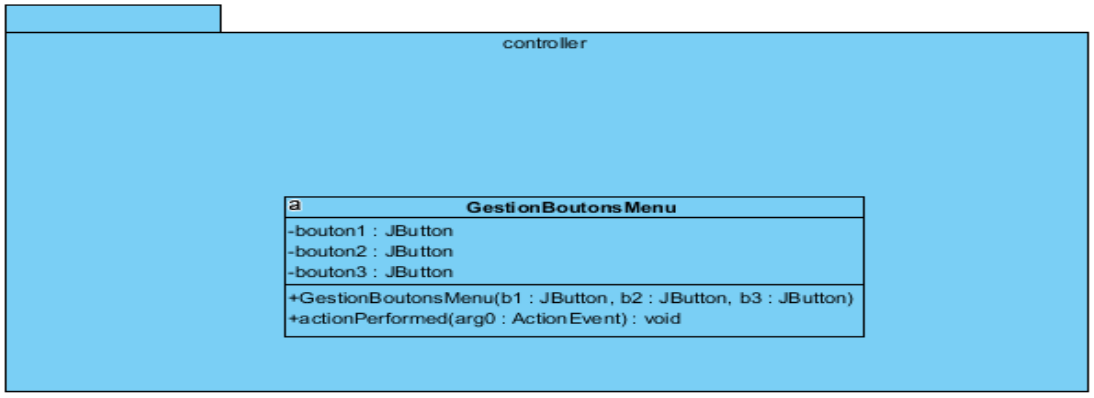
- **Package model.case:**



- **packagecontroller.jeu:**

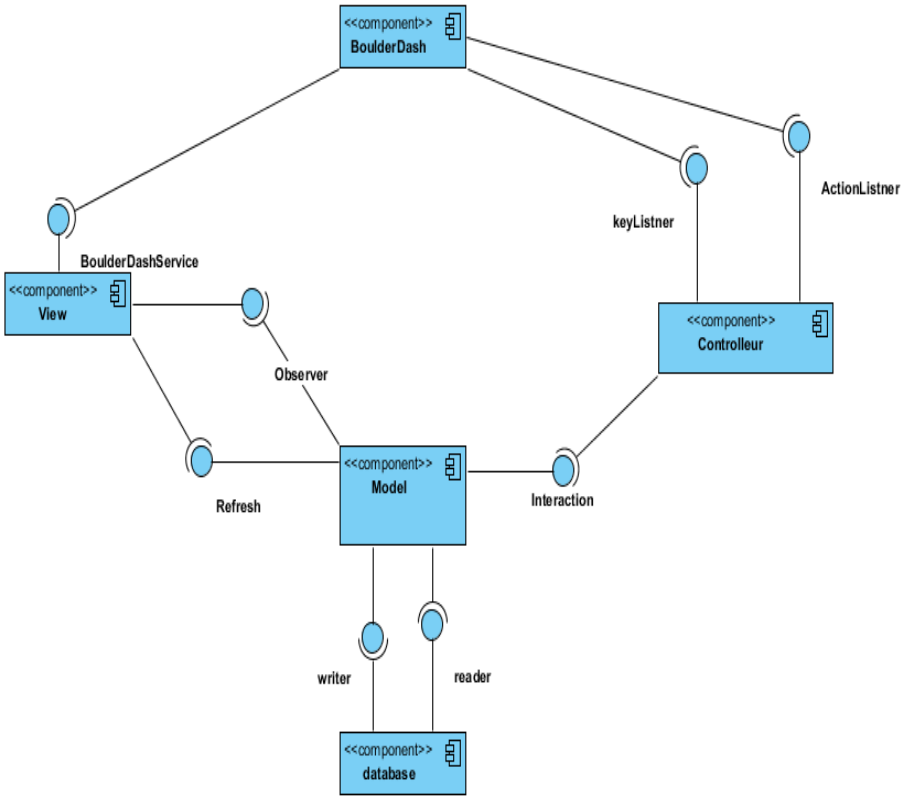


- **Package controller:**



Component diagram:

Component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

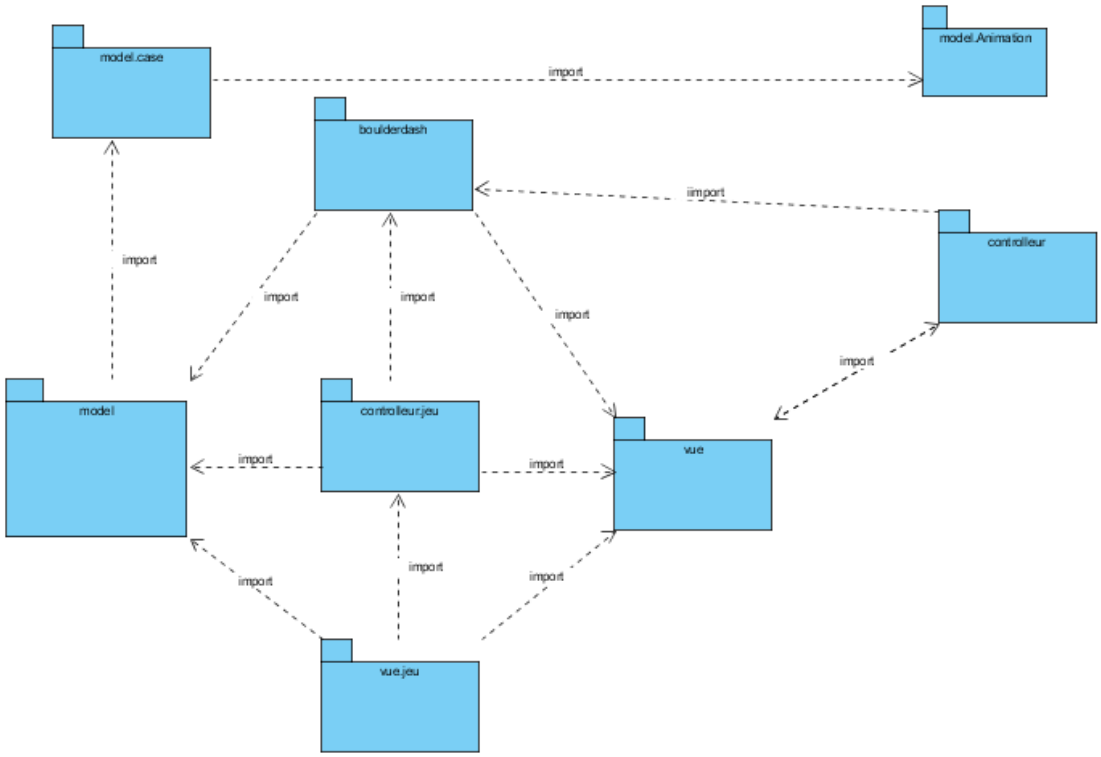


Package diagram:

A package diagram in the Unified Modeling Language depicts the dependencies between the packages that make up a model.

In addition to the standard UML Dependency relationship, there are two special types of dependencies defined between packages:

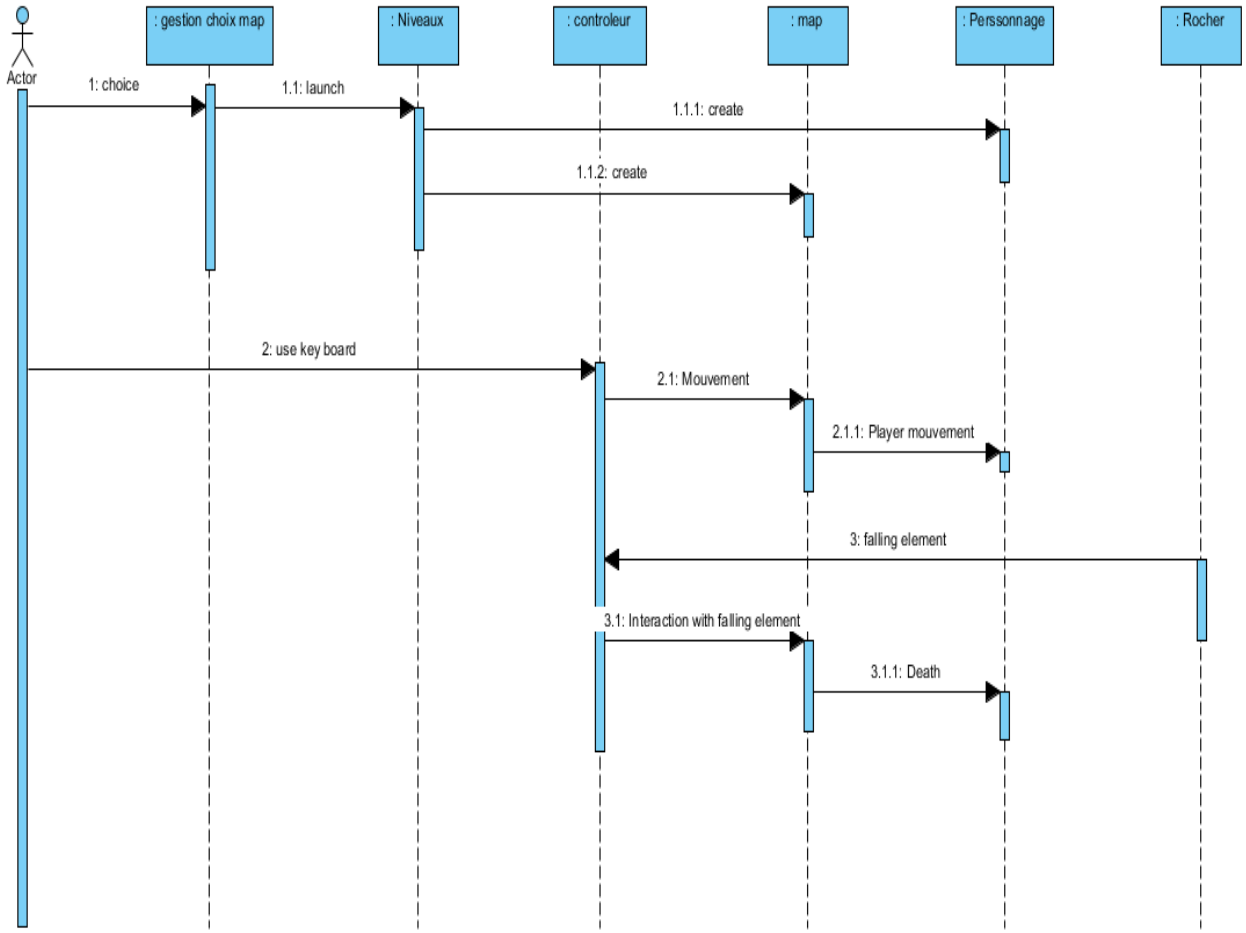
- package import
- package merge



Sequence diagram:

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario



Database modeling:

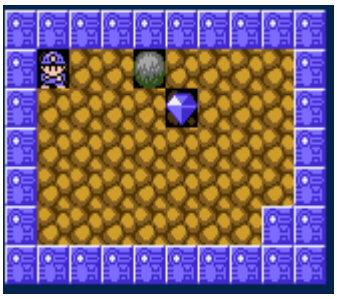
We have used different xls file (exel) as a database to store different levels

The xls file contains characters that define form of the levels map

The level in xls :

M,M,M,M,M,M,M,M,M,M,
M,P,B,B,R,B,B,B,B,M,
M,B,B,B,B,D,B,B,B,M,
M,B,B,B,B,B,B,B,B,M,
M,B,B,B,B,B,B,B,B,M,
M,B,B,B,B,B,B,B,S,M,
M,M,M,M,M,M,M,M,M,M,

This the same level in the game:



M = Mur 

P = Player 

B = Boue 

D = Diamonds 

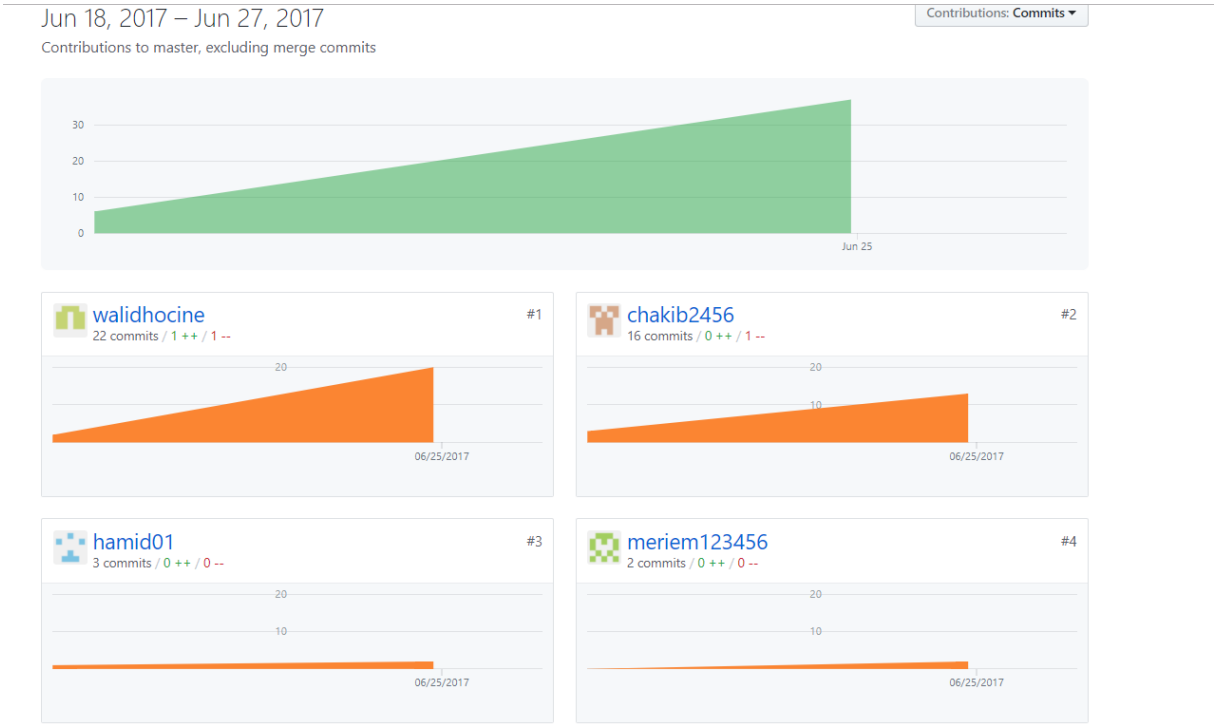
S = Sortie 

Github

For this project we've used Github wich is the most powerfull tool for collaboration between developers is also a great platform where we can find code source shared by other developers

We uploaded (commited) during this project many version of our code and other file like diagrams and github was very helpful to take our work safe so we can recover at any time and by any member of the team a previous version of code or any other file sins it stored in repository.

Here are a report the activity of the team member's group



Project log:

Date	Team members	Tasks realized
15/06/17	Ouldhamou chakib	- Subject study
	Mrabentmeriem	- Global class diagram
	Hocine walid	- Git creation
	Aoudjehane wassim	- Installation of the environment
18/06/17		
	Ouldhamou chakib	- Distribution of tasks
	Mrabentmeriem	- Provisional schedule
	Hocine walid	- Global class diagram
	Aoudjehane wassim	- MVC class diagrams
19/06/17		
	Ouldhamou chakib	- Controller class diagram
	Mrabentmeriem	- View class diagram - Functional analysis
	Hocine walid	- Data EXCEL
	Aoudjehane wassim	- Model class diagram
20/06/17		
	Ouldhamou chakib	- Controller class diagram
	Mrabentmeriem	Cutting sprites
	Hocine walid	-View class diagram - Data dictionary
	Aoudjehane wassim	- Model class diagram - Sequence diagram
21/06/17	Ouldhamou chakib	- Start of Junit tests
	Mrabentmeriem	- Class diagram java export
	Hocine walid	- Start of Junit tests
	Aoudjehane wassim	- Data base creation - Map creation
22/06/17	Ouldhamou chakib	- Junit test
	Mrabentmeriem	- View code
	Hocine walid	- Starting coding Elements

		- Junit test
	Aoudjehane wassim	- Data base filling
24/06/17		
	Ouldhamou chakib	- Junit test
	Mrabentmeriem	- View code
	Hocine walid	- JUnit tests - Elements coding
	Aoudjehane wassim	- Component diagram
25/06/17		
	Ouldhamou chakib	- Controller code - Project Report
	Mrabentmeriem	- View code
	Hocine walid	- Implementation of the java methods that will execute - Class diagram update - Component diagram update
	Aoudjehane wassim	- Map code - Project Report

26/06/17		
	Ouldhamou chakib	- Implementation of a console view - Package diagram

		-
	Mrabentmeriem	- Debug of the view - Debug of elements -
	Hocine walid	- Effective schedule - Creation of map/data base test - Project report
	Aoudjehane wassim	- Controller Javadoc
27/06/17	Ouldhamou chakib	- *Production of the slideshow* - *Oral preparation*
	Mrabentmeriem	
	Hocine walid	
	Aoudjehane wassim	
28/06/17	Ouldhamou chakib	- *Last update on the presentation* - **PRESENTATION**
	Mrabentmeriem	
	Hocine walid	
	Aoudjehane wassim	

Conclusion:

Encountered problems:

- Problem with Maven compilation, which was solved with a pom.xml file modification.(adding dependencies)

Personal review:

- **Ouldhamou chakib:**

  This project give me the concretion of what I knew theoretically with the pratic of what I learn in java

  This project allowed me to understand how to use certain tools like maven, Junit and github

- **Mrabent meriem:**

I'm happy to participate to this project.

Despite my difficulties in modelling I think I have made progress and have a group with I had a good agreement help me for work atmosphere.

- **Hocine walid:**

  This project allowed me to get new skills and sharp those I already have.

- **Aoudjehane wassim:**

I found this project very interesting because it allows us to implement everything we had seen during the various prosits. It also allows us to realize a software starting from the thinking and going up to the realization in passing by the modeling.

Our group was very pleasant and hardworking, which allowed us to respect our deadlines and to succeed in making something functional while maintaining a good atmosphere in the group throughout the project.