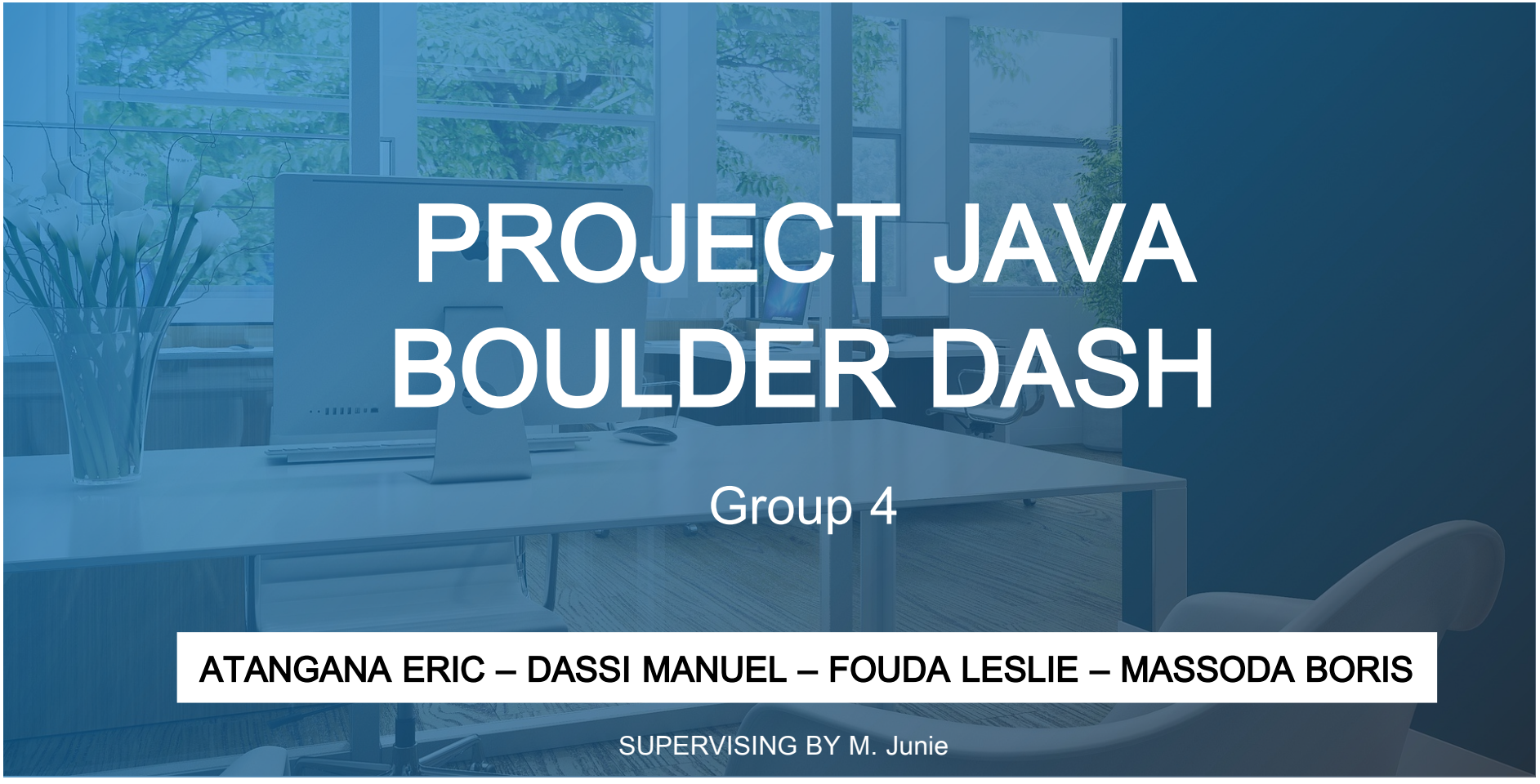
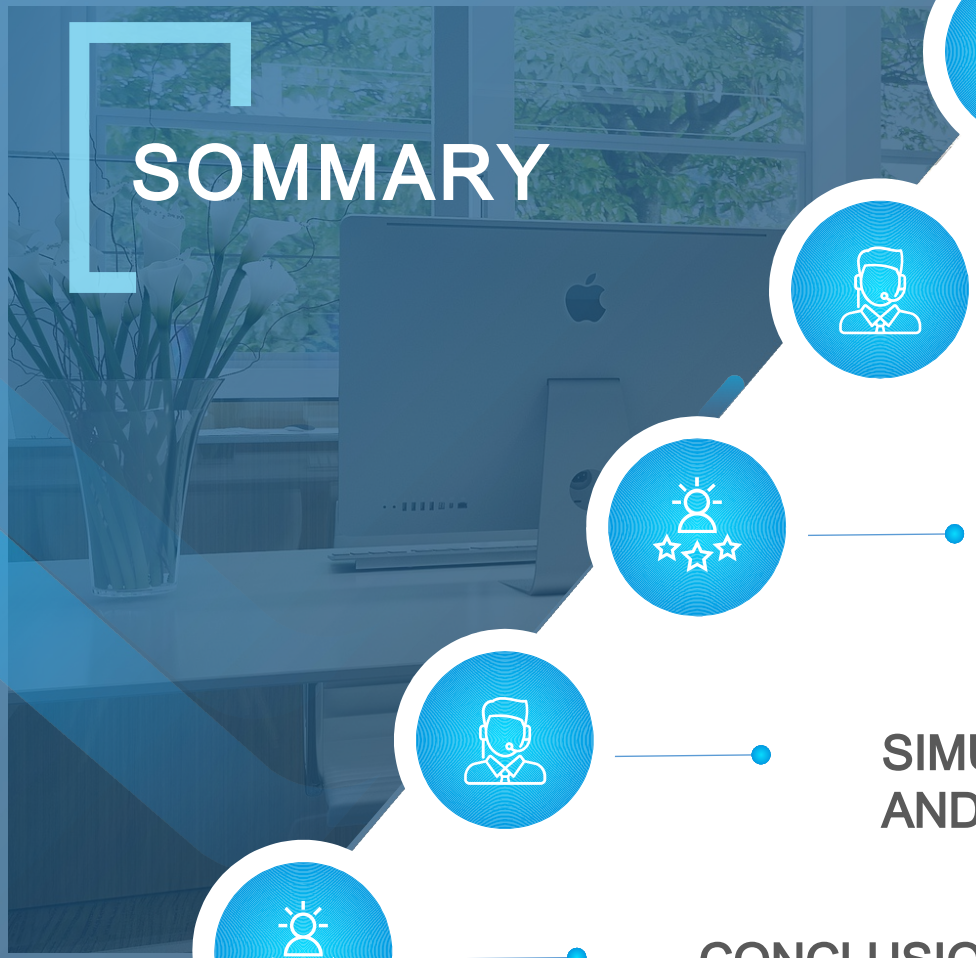


PROJECT JAVA BOULDER DASH

Group 4

ATANGANA ERIC – DASSI MANUEL – FOUDA LESLIE – MASSODA BORIS

SUPERVISING BY M. Junie



SOMMARY

INTRODUCTION

PROJECT PRESENTATION

STRUCTURING AND PRODUCTION

SIMULATIONS AND TESTS

CONCLUSION

PROJECT PRESENTATION

01 Context

02 Problematique

03 Goals

04 Resources

05 Planning

Context

I What is it about ?

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

II What does it have ?

He is promoting the game and increase the capacity of students that we are

III To whom he gives a service ?

Students of more peoples

IV What is it effective on ?

It acts on students of university

Problematique

How to convey and set up
The boulder dash Game ?

Goals

- Model your program using diagrams such as class diagrams, components
- We also need to use Git Hub which is an online versioning system
- Build automation tool used primarily for Java projects and a database to stock our different levels.
- Code it in JAVA and use Maven the game

Software Resources

JAVA EE 1.8

Visual Paradigm

Eclipse IDE

Wampp server and Mysql

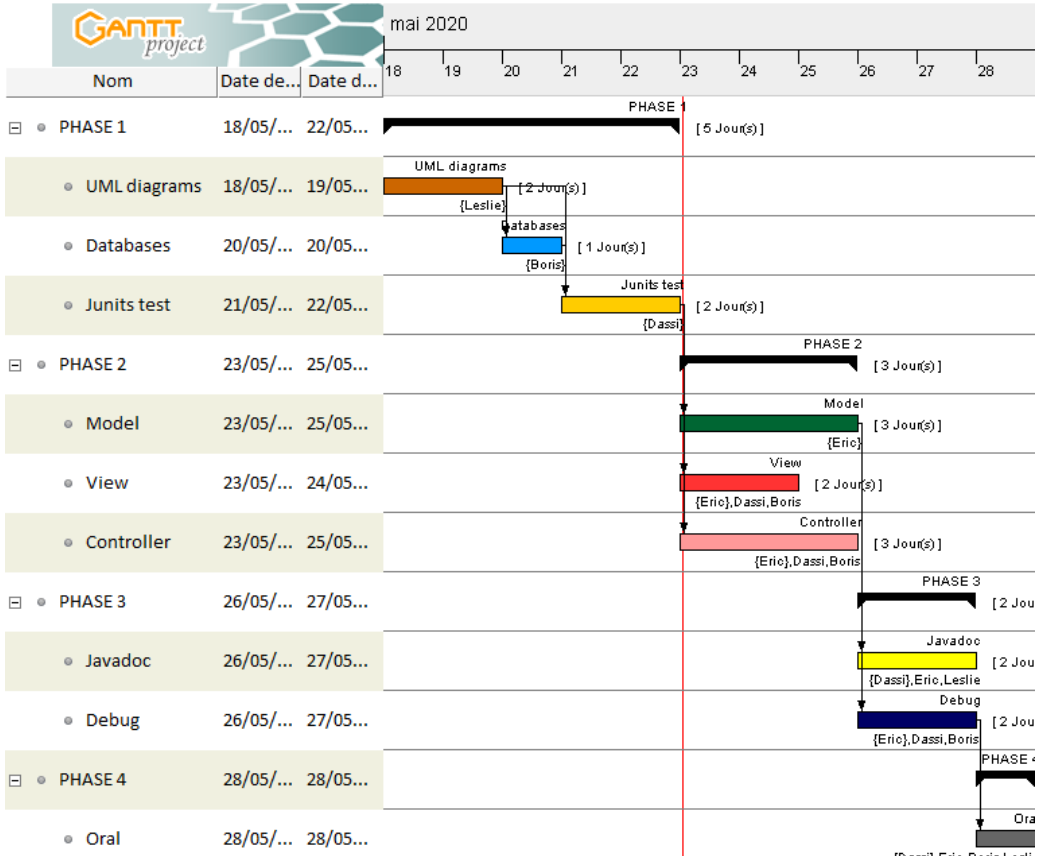
Git

Gantt Project

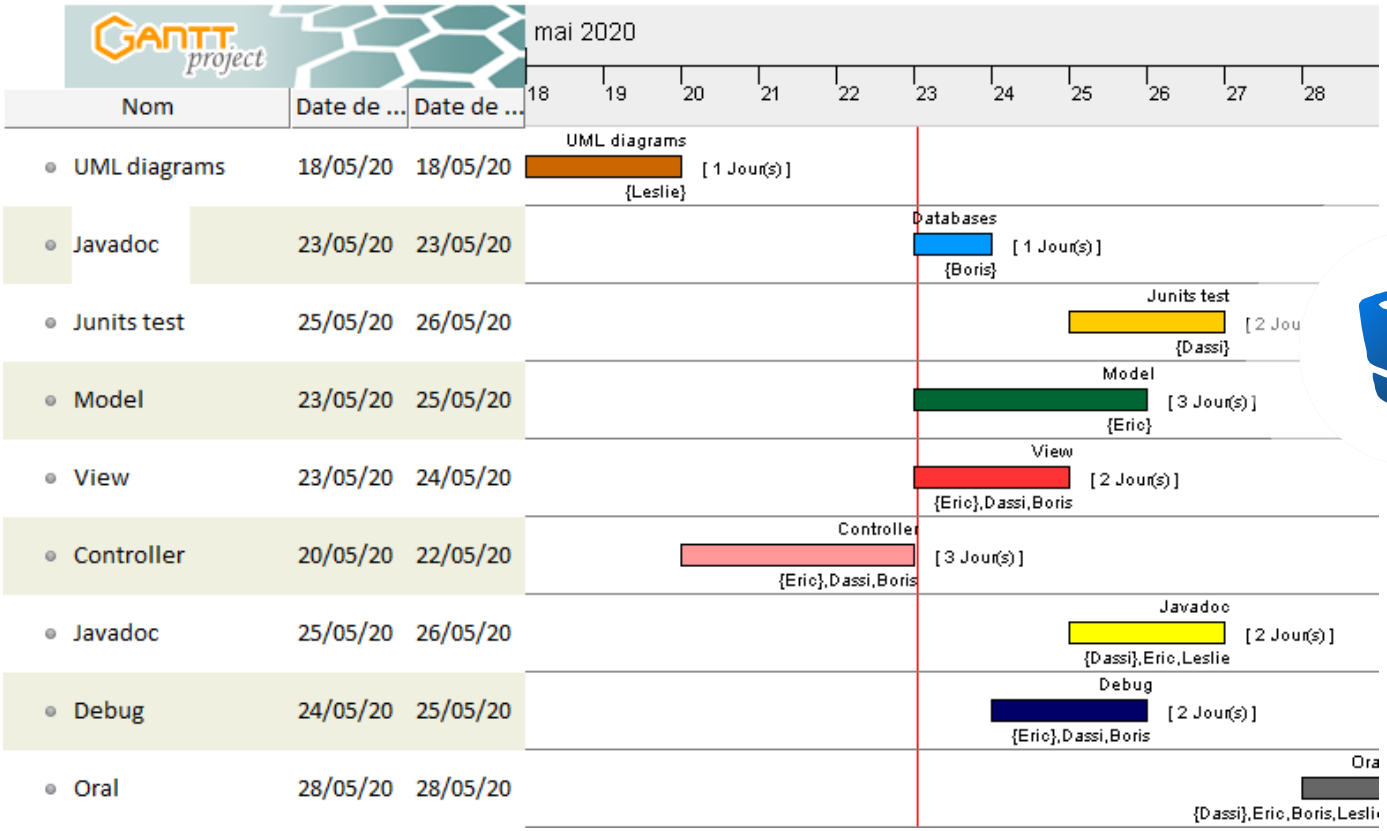
Hardware Resources

Laptop

Provisional schedule

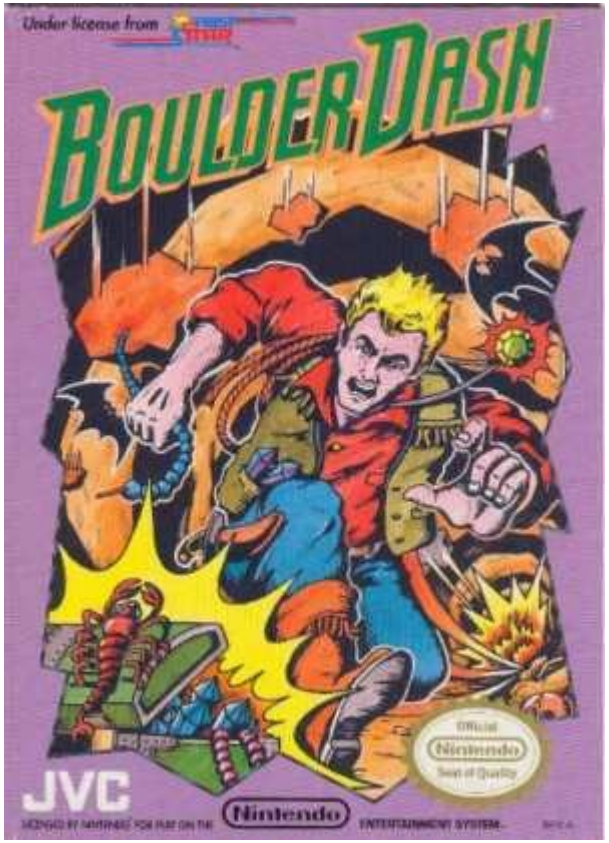


Real schedule



BOULDER DASH

The original Boulder Dash® is a pioneering interactive game dating back to the early times of the home computer and video game era being first launched in 1984. Boulder Dash® and its many sequels have delighted and challenged casual and hard-core players of all ages and sexes for over three decades.



STRUCTURING AND PRODUCTION

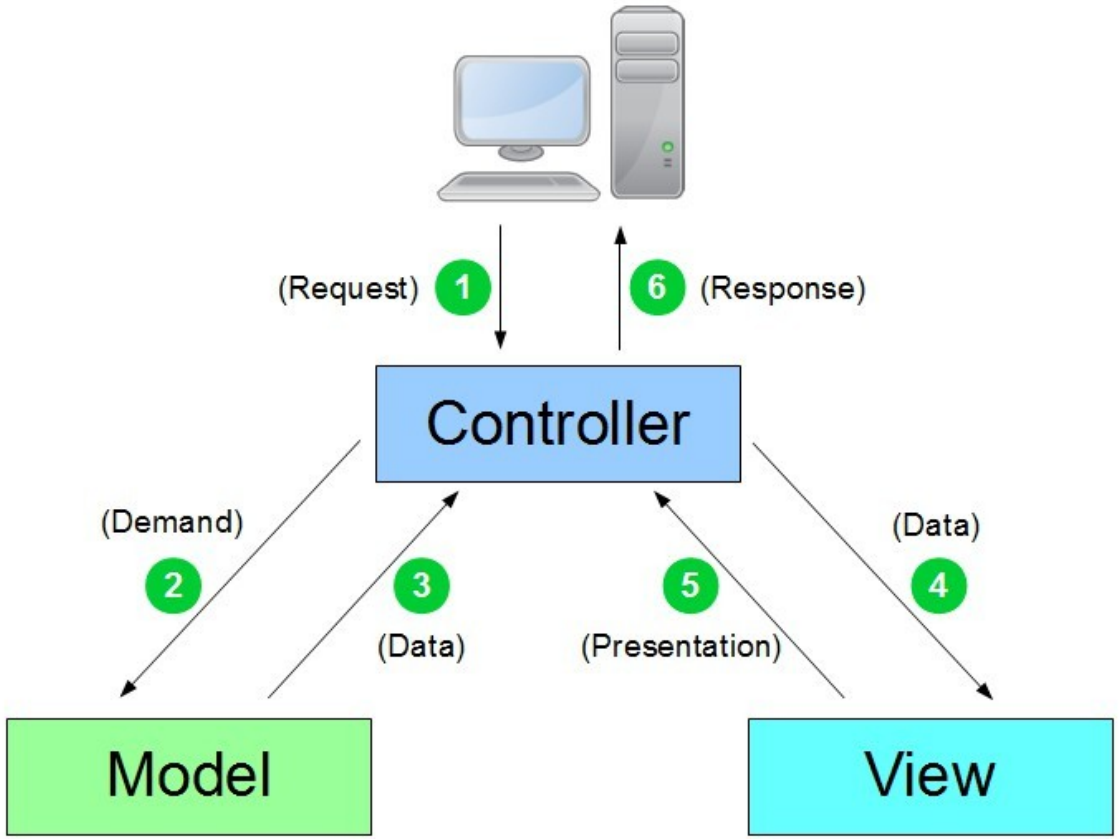
JAVA MODELLING

DATABASE MODELLING

PRODUCTION

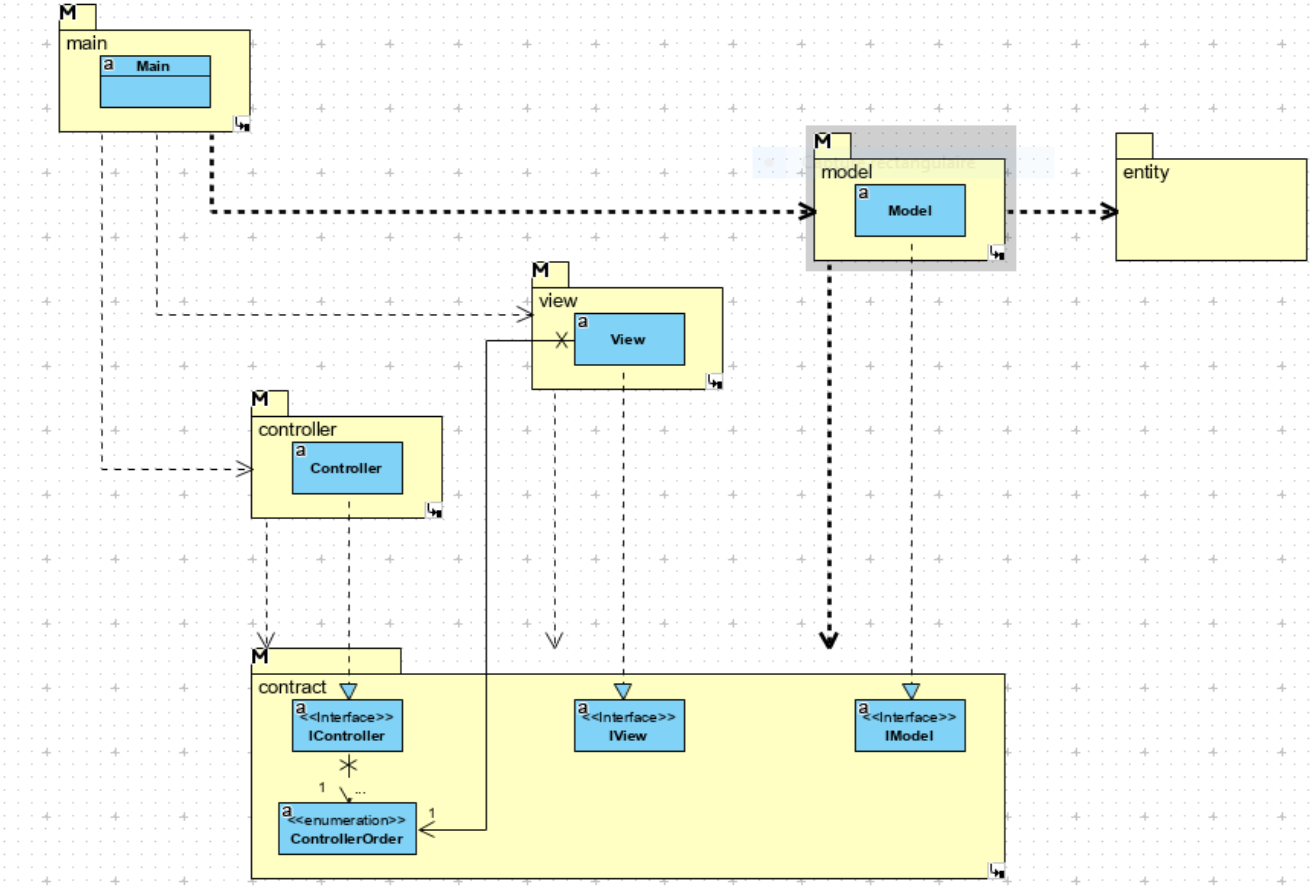
MVC

MVC STRUCTURE



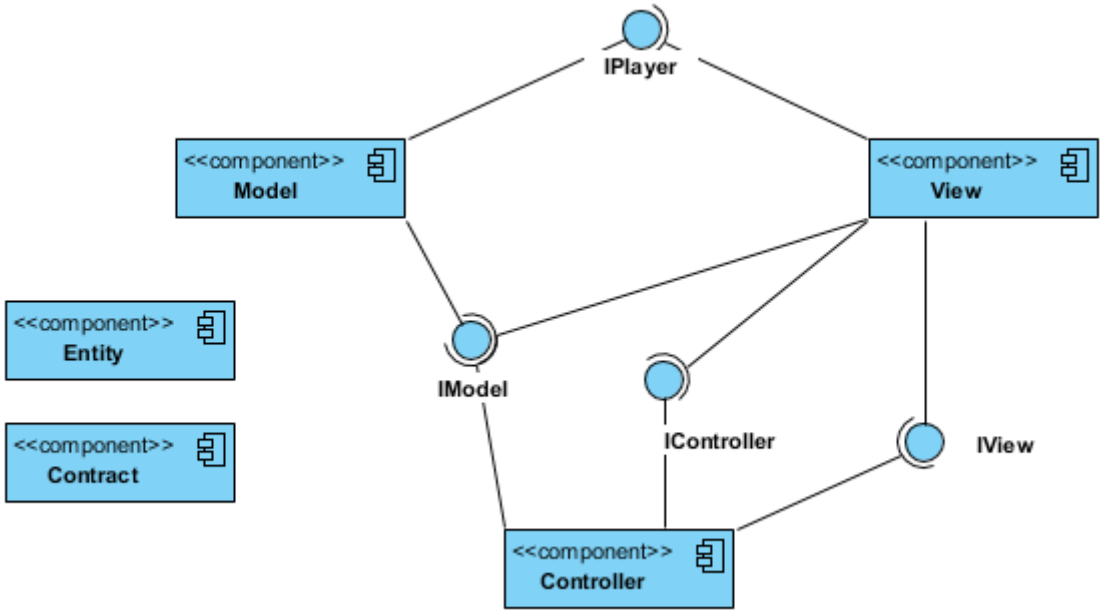
JAVA MODELLING

Diagram of Package



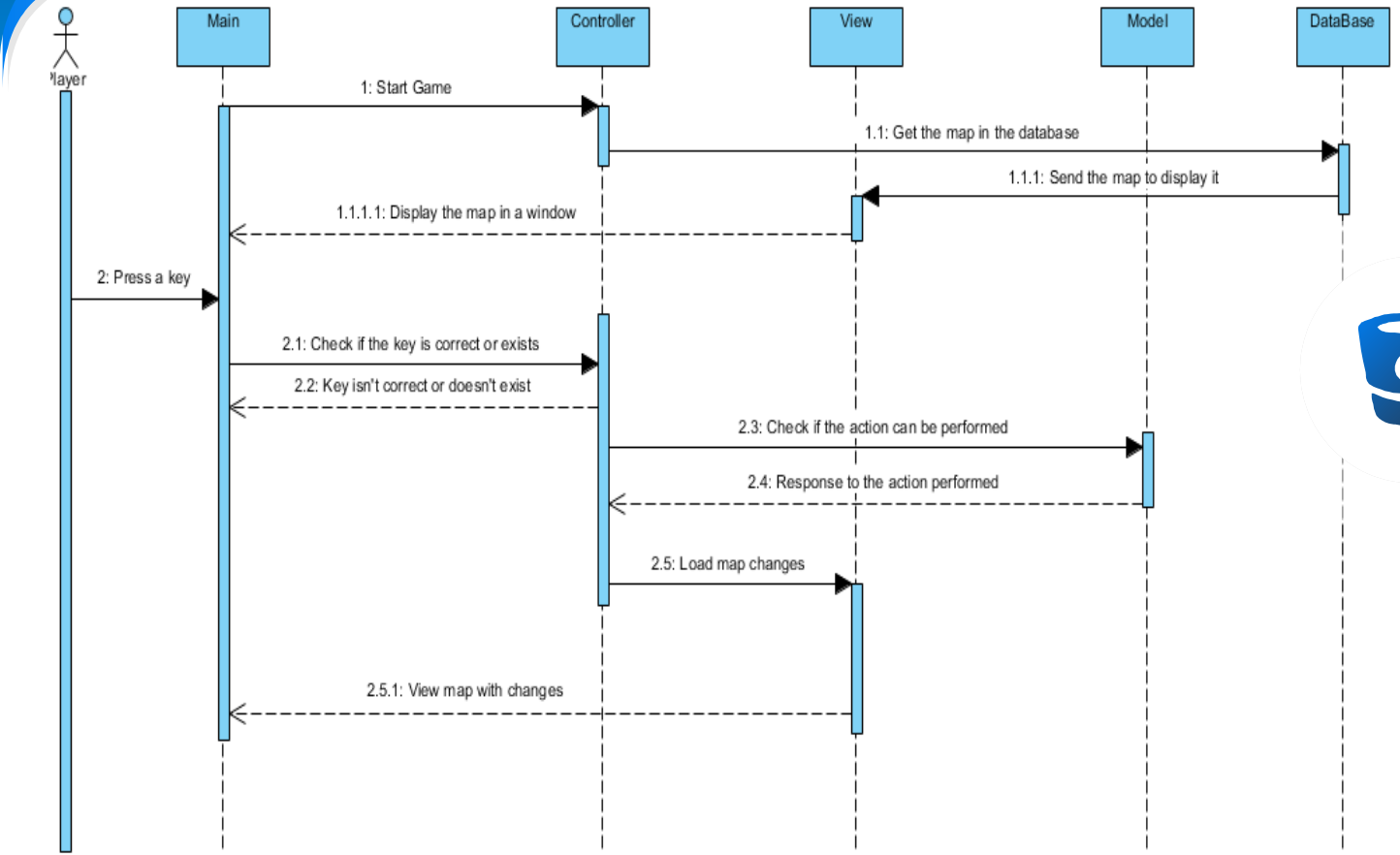
JAVA MODELLING

Component diagram



JAVA MODELLING

Sequence diagram



DATABASE MODELLING

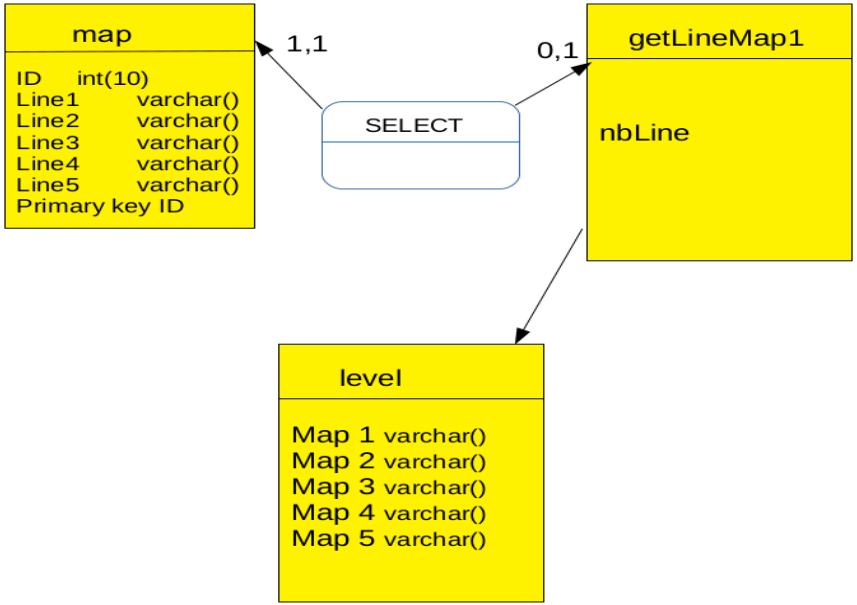
```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données :  `map`
--

DELIMITER $$
--
-- Procédures
--
DROP PROCEDURE IF EXISTS `boucle`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `boucle` ()  NO SQL
BEGIN
DECLARE param VARCHAR(50);
DECLARE colonne VARCHAR(50);
DECLARE nom VARCHAR(50);
DECLARE cmd VARCHAR(500);
SET @param = 1;
SET @colonne = "line ";
boucle_loop: LOOP
SET @nom = CONCAT('`',@colonne, @param, '`');
SET @cmd = CONCAT('INSERT INTO `map1.1` (line1) SELECT ',@nom,' FROM `level
1`');
PREPARE stmt1 FROM @cmd;
EXECUTE stmt1;
SET @param = @param + 1;
```

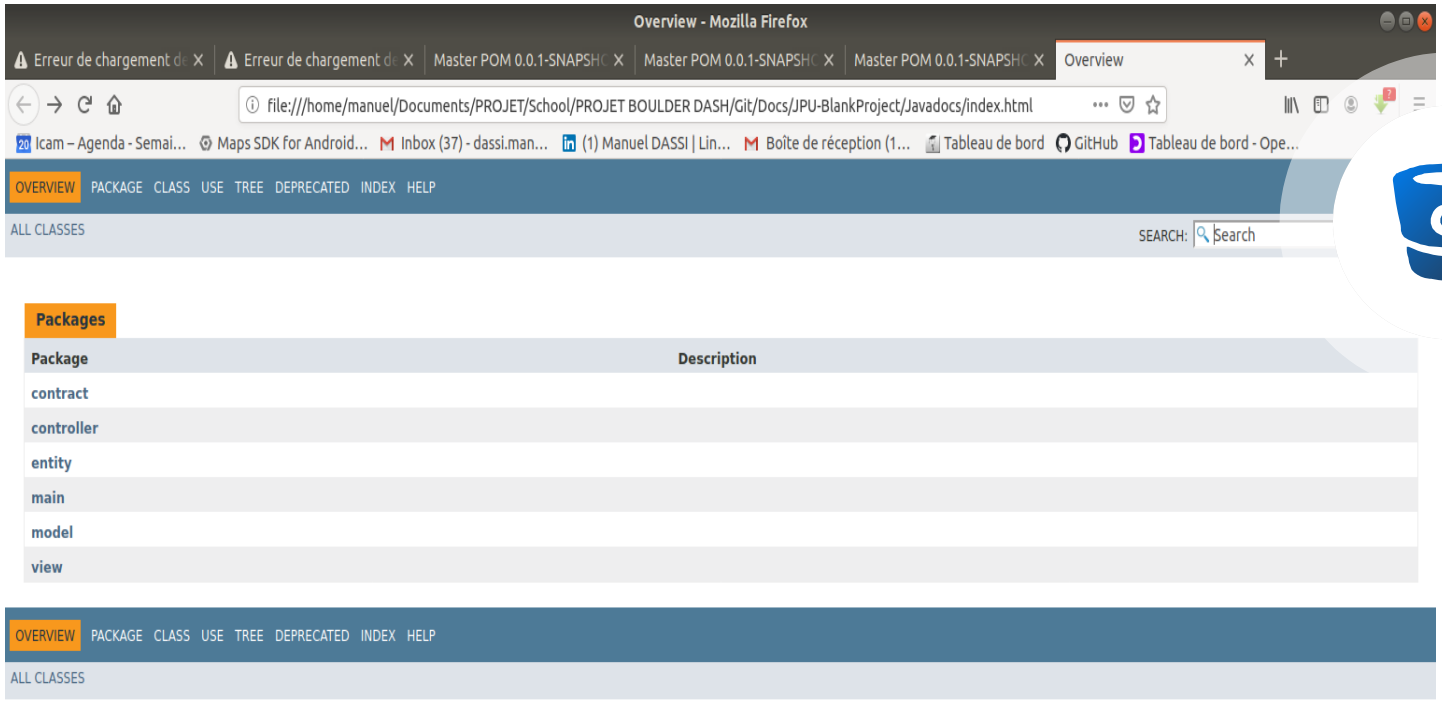
DATABASE MODELLING



JAVADOC + RAPPORT SURE FIRE / XREF

JAVA DOCS COMPLETE

Generate a Java docs of our project



JAVADOC + RAPPORT
SURE FIRE / XREF

RAPPORT
XREF

All Classes

Packages

contract
controller
entity
main
model
view

view

Classes

View
ViewFrame
ViewPanel

```
83
84      /**
85       * Gets the controller.
86       *
87       * @return the controller
88       */
89      private IController getController() {
90              return this.controller;
91      }
92
93      /**
94       * Sets the controller.
95       *
96       * @param controller
97       *          the new controller
98       */
99      protected void setController(final IController controller) {
100             this.controller = controller;
101     }
102
103     /**
104      * Gets the model.
105      *
106      * @return the model
107      */
108     protected IModel getModel() {
109             return this.model;
110     }
111
112     /**
113      * Sets the model.
114      *
115      * @param model
116      *          the new model
117      */
118     private void setModel(final IModel model) {
119             this.model = model;
120     }
121
122     /**
123      * Builds the view frame.
124      *
125      * @param model
126      *          the model
127      */
128     private void buildViewFrame(final IModel model) {
129             this.setModel(model);
130             this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
131             this.setResizable(false);
132             this.addKeyListener(this);
```

JAVADOC + RAPPORT SURE FIRE / XREF

RAPPORT SURE FIRE

Model

Last Published: 2019-06-03 | Version: 0.0.1-SNAPSHOT

Model

Parent Project
- Master POM

Project Documentation
- Project Information
  - **Dependencies**
  - Dependency Information
  - Distribution Management
  - About
  - Plugin Management
  - Plugins
  - Summary

Built by: maven

Project Dependencies

compile

The following is a list of compile dependencies for this project. These dependencies are required to compile and run the application:

GroupId	ArtifactId	Version	Type	Licenses
jpublankproject	contract	0.0.1-SNAPSHOT	jar	-
mysql	mysql-connector-java	5.1.39	jar	The GNU General Public License, Version 2

test

The following is a list of test dependencies for this project. These dependencies are only required to compile and run unit tests for the application:

GroupId	ArtifactId	Version	Type	Licenses
junit	junit	4.12	jar	Eclipse Public License 1.0

Project Transitive Dependencies

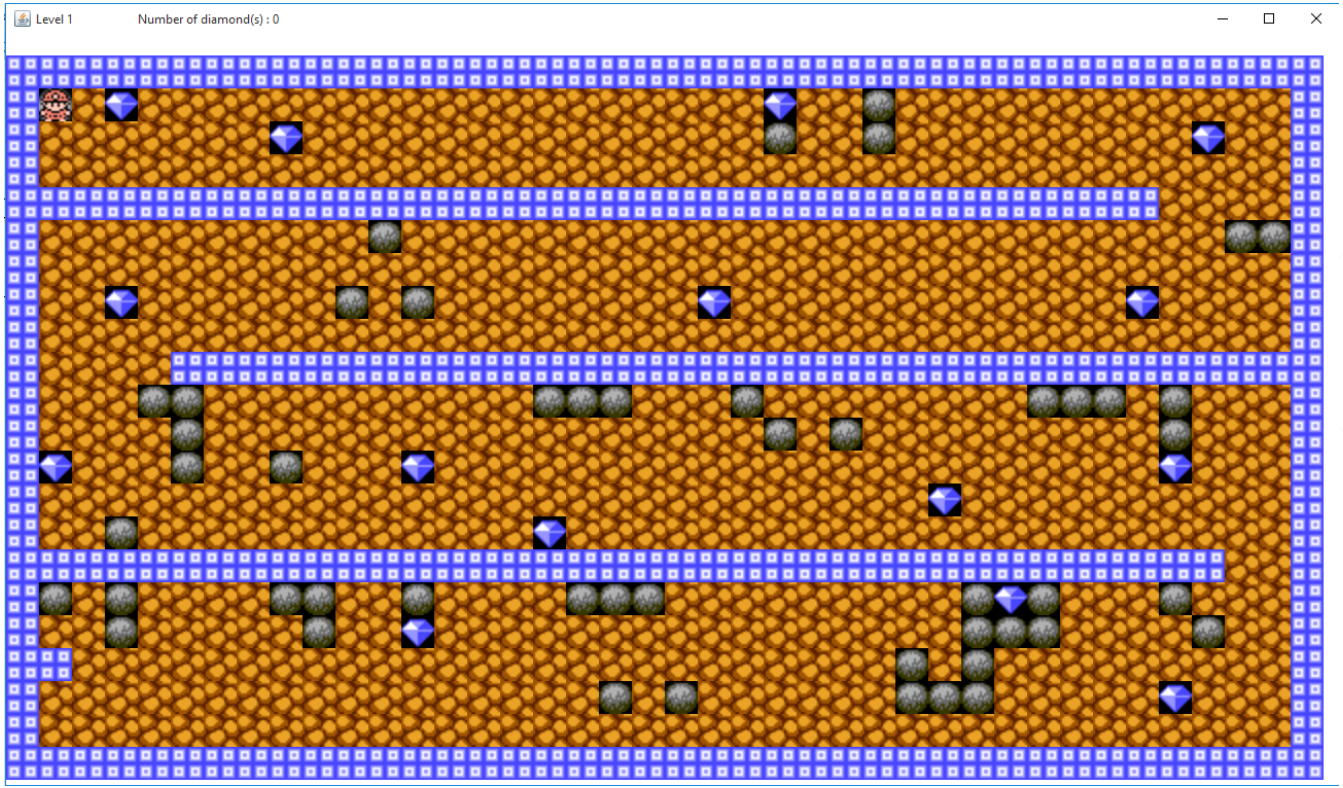
The following is a list of transitive dependencies for this project. Transitive dependencies are the dependencies of the project dependencies.

compile

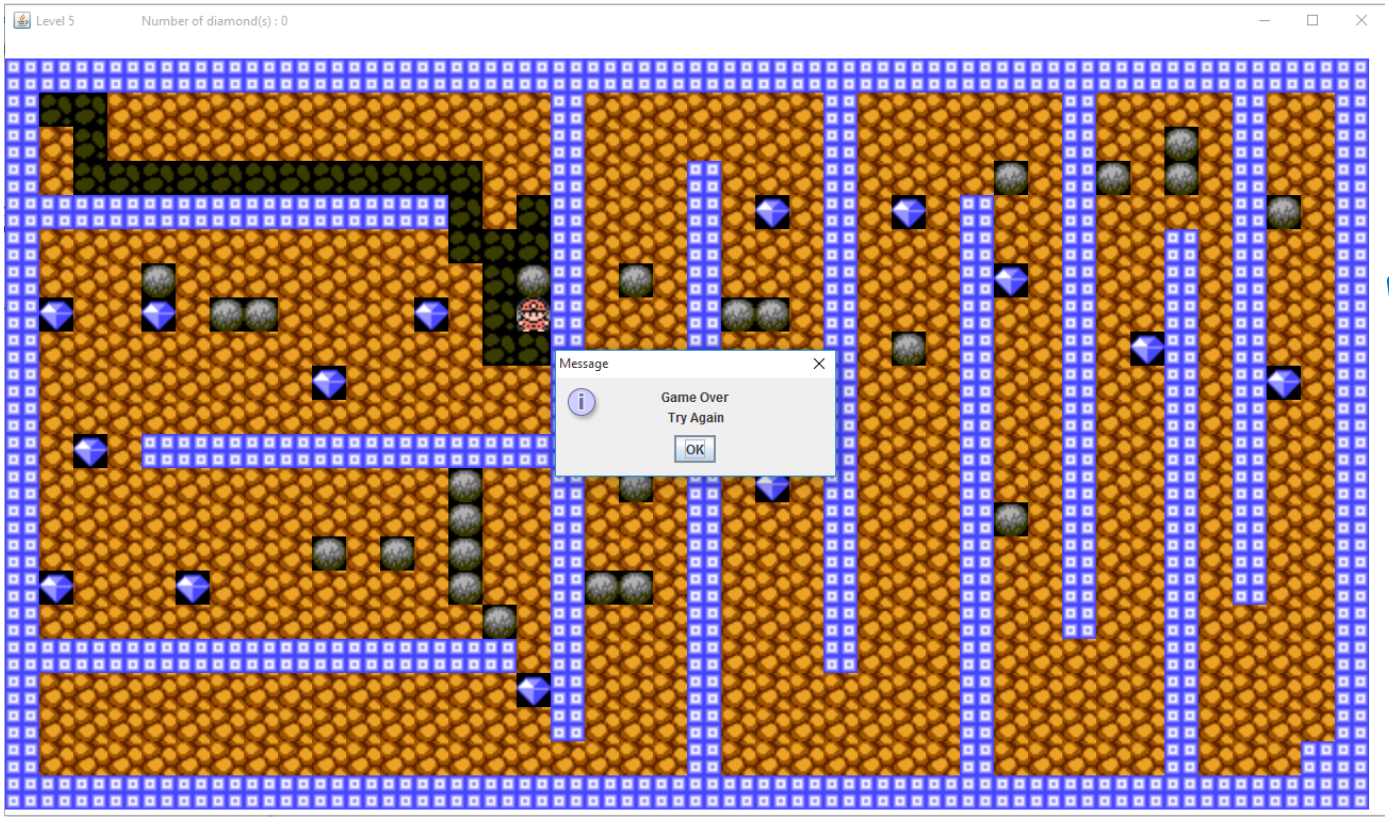
The following is a list of compile dependencies for this project. These dependencies are required to compile and run the application:

GroupId	ArtifactId	Version	Type	Licenses
jpublankproject	entity	0.0.1-SNAPSHOT	jar	-

PRODUCTION



PRODUCTION



CONCLUSION

We had a lot of difficulties during this project and we couldn’t implement all the features we wanted and we couldn’t fix all the bugs. There are a lot of ways to improve !

THANKS

AVIS DES JURY