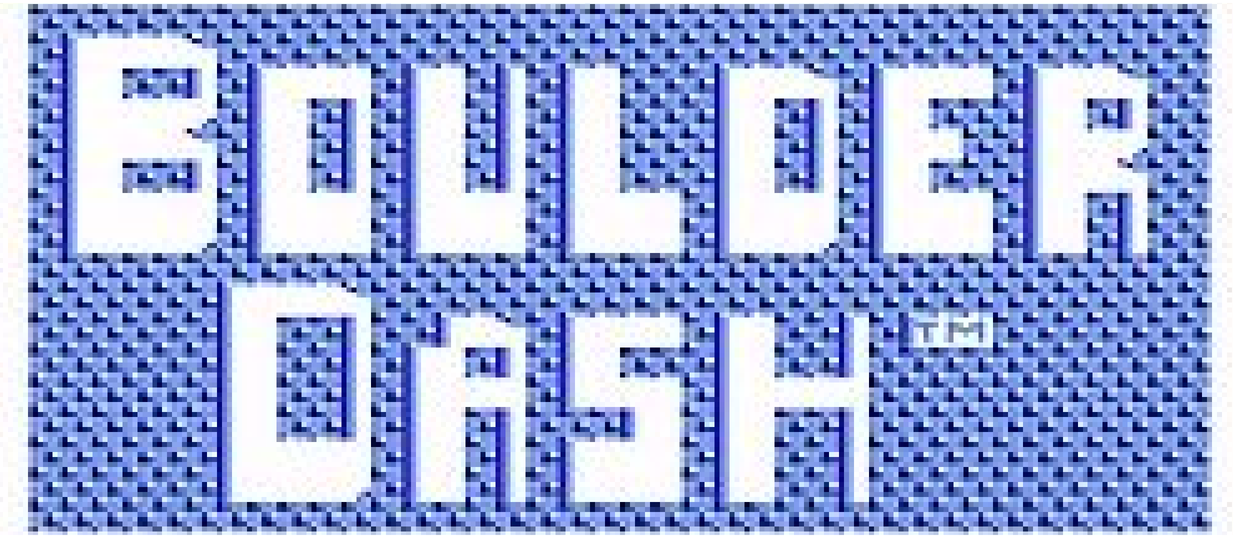


Boulder Dash Remake

By Wild Box Studio

Wierez Loïc
Duleu Lucas
Brocherieux Etiennes
Terrier Antoine

04/06/2019

Introduction

The goal of the project were to remaster the game Boulder Dash. This game was edited in 1984 and created by Peter Liepa and Chris Gray.

Moreover, we wanted a reliable and reusable work. The project had to be updatable, even by another team easily. That’s why we made some choices and applied some design patterns. We will explain these choices here, and talk about our group experience in this project.

Technicals choices

As explained in the introduction, we made choices and we imagined the development to make the project reworkable.

The MVC design pattern

First, we choose the Model View Controller design pattern. It means our code is split into three packages :

- The Model, which manipulate the game data ;
- The View, which control the graphics ;
- The Controller, which manage the two other packages, make the exchanges between the model and the view, trigger actions… ;

The advantage of this design pattern is that each package can be modified and reworked without impact on the whole code.

Also, user interactions and the database access is made easier.

Maven

To improve the efficiency of this method, we used Maven, a development tool that manage the project to avoid inter-dependencies. So the modification of a package is very easy.

Maven and the MVC design pattern are two elements that ensure the project can be modified and updated.

Unit tests

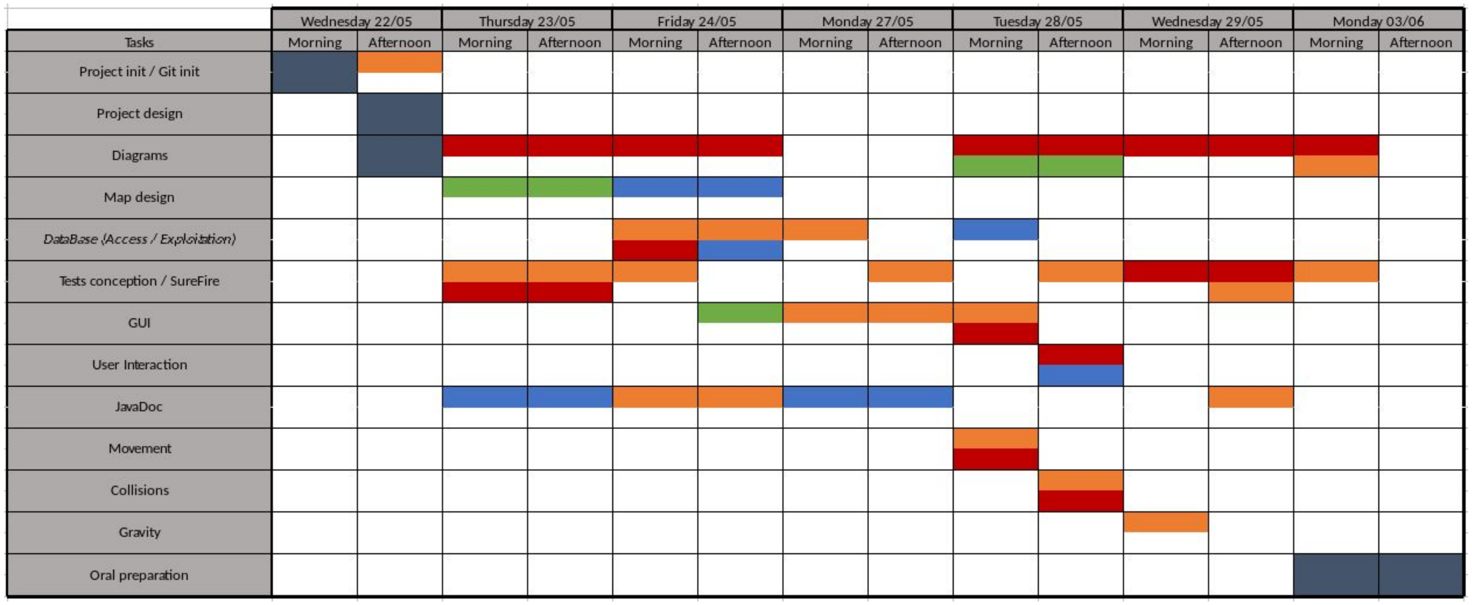
The stability of the code was also a big concern. To make sure each function of the game works perfectly. Also, each test is referenced in a SureFire report on an HTML page.

JavaDoc

Our HTML reports also contain a full java documentation, so every method of each class is explained.

The group experience and organization

The organization



Tasks	Wednesday 22/05		Thursday 23/05		Friday 24/05		Monday 27/05		Tuesday 28/05		Wednesday 29/05		Monday 03/06	
	Morning	Afternoon	Morning	Afternoon	Morning	Afternoon	Morning	Afternoon	Morning	Afternoon	Morning	Afternoon	Morning	Afternoon
Project init / Git init	⬛	🟧												
Project design		⬛												
Diagrams		⬛	🟥	🟥	🟥	🟥			🟥 🟩	🟥 🟩	🟥	🟥	🟥 🟧	
Map design			🟩	🟩	🟦	🟦								
DataBase (Access / Exploitation)					🟧 🟥	🟧 🟦	🟧		🟦					
Tests conception / SureFire			🟧 🟥	🟧 🟥	🟧			🟧		🟧	🟥	🟥 🟧	🟧	
GUI						🟩	🟧	🟧	🟧 🟥					
User Interaction										🟥 🟦				
JavaDoc			🟦	🟦	🟧	🟧	🟦	🟦				🟧		
Movement									🟧 🟥					
Collisions										🟧 🟥				
Gravity											🟧			
Oral preparation													⬛	⬛

This is the gantt diagram recording the work of each member of the team.

To organize the group, we made daily meetings to check the done and the future work.

The issues encountered

We had few issues during this project :

- First, Maven plugins. The plugins were outdated in our project, and we had to download and configure them again in the main pom.xml file.
- Then, the sprite draw, Maven have a special way to manage files, and a configuration was needed to make the sprite loading work. Again, we had to modify the pom.xml file.
- The model package is bigger than others and contain way more classes, and the organization of these classes was changed a few times during the development.