

Faculdade de Engenharia da Universidade do Porto



Final Project Report

PlainJump

T02G02

João Vieira up201603190

Susana Lima up201603634

UML

Located on */Part2 - Final Project/uml/uml.png*

Design/Architectural Patterns

The developed application implements the following patterns:

1. Model-View-Controller: used to separate the user interaction portion of the program from the game's development, resorting to a mediator.
 - *Model*: responsible for the game's development
 - *View*: responsible for the application graphical portion
 - *Controller*: serves as mediator between Model and View allowing a complete independence between them
2. Singleton: used in classes that are accessed in various parts of the program assuring that there is only one instance of that class. Classes where is implemented:
 - *GameController*: controls all game states, from menus to gameplay
 - *MapController*: serves as a junction for all other entities that are involved somehow with level creation.
 - *BallController*: used to control ball movements and status, represents the player.
 - *ConfigsModel*: contains all the information regarding the configuration menu colors possibilities.
 - *EntryModel*: contains all the information transmitted in the main menu
 - *MapModel*: contains all the information regarding the game's high score count
 - *GameModel*: responsible for loading and saving the ball, plains and background colours as well as the high score
 - *MapView*: used to load the entire level making it visible for the player to see it.
 - *GameView*: responsible for the entire game's camera management

Design Decisions

As previously referred, the code was structured by the Model-View-Controller architectural pattern.

Regarding the graphical aspects of the application, the game itself was implemented using the 3d api offered by libgdx but the menus were implemented in 2d.

Difficulties

- Organizing the code
- Learning how to work with libgdx

Time spent developing

In order to be able to achieve the final result of the project, several hours of work and research were dedicated individually and jointly by the students. Estimated way around 160 hours.

Work distribution amongst team members

The implementation of the application in question was possible due to the joint effort of both students so that there was not great disparity of contribution. Each element focused more on the module with which it was more comfortable, but overall the work was divided evenly between the two.