System design document for Kangaroo Project

Table of Contents

Version: 0.3

Date: 15-5-12

Author: Henrik Alburg, Simon Almgren, Arvid Karlsson, Sean Pavlov

This version overrides all previous versions.

1 Introduction

1.1 Design goals

The model shall be not be dependent by the view. It shall be possible to change tiles.

1.2 Definitions, acronyms and abbreviations

•GUI, graphical user interface.
• Java, platform independent programming language.
• JRE, the Java Run time Environment. Additional software needed to run an Java application.
•MVC, a way to partition an application with a GUI into distinct parts avoiding a mixture of GUI-code, application code and data spread all over.

2 System design

2.1 Overview

The application will use the MVC model. The controller will take input and modify the model and the view will read the model and render.

2.1.1 Unique identifiers, global look-ups

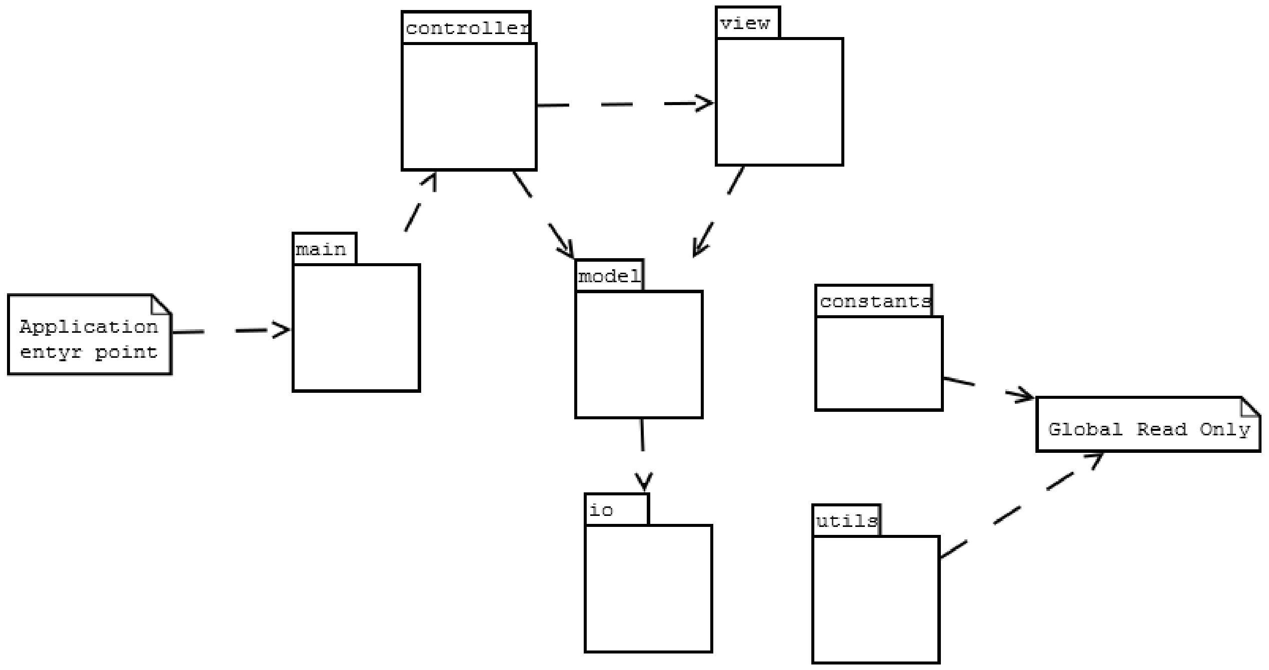
All the global constants are saved in Constants.java. They are used to know what the type is of a blocks ID. And also the resolution of buttons and the game. There are also some for the number of levels and number of high-scores.

In this section we explain the overall design choices.

2.2 Software decomposition

2.2.1 General
The application is decomposed into the following modules, see Figure X.

- view, is the top level package for all GUI related classes including the main window
- main, is the application entry class
- controller, taking input and changing the model
- model, the OO-model
- io, file handling
- util, tools and general utilities
- constants, application global constants



Package diagram. For each package an UML class diagram in appendix

2.2.2 Layering
Higher layers are at the top of the figure, lower layers at the bottom. The layering is shown in the figure above.

2.2.3 Dependency analysis
As shown in figure. There is no circular dependency between the packages.

2.3 Concurrency issues
NA. We have a thread that runs the main program. This is complimented by some small threads that handles things like sound and special moves by some enemies. There’s always just one thread that changes one object.

2.4 Persistent data management

We will store our controls and hi score in two different files.

2.5 Boundary conditions

NA. Application launched and exited as normal desktop application.

3 References

1. MVC. see
   http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

APPENDIX