# System design document for JumpyDash

Alex Sundbäck
Johannes Mattsson
Marcus Bertilsson
Oscar Hansson

May 2016

# Contents

# 1 Introduction

## 1.1 Design goals

The main design goal is to make the source code module and flexible in order to make it possible to change physics engine, thereby reducing the dependency upon a specific framework. This will be achieved by using interfaces and abstract classes. The current physics engine in use is Box2D.

The second design goal is the strictly follow the MVC design pattern in order to improve code readability and maintainability

## 1.2 Definitions, acronyms and abbreviations

- **Box2D**, physics engine
- **libGDX**, framework for 2D game development
- **MVC**, software design pattern

# 2 System design

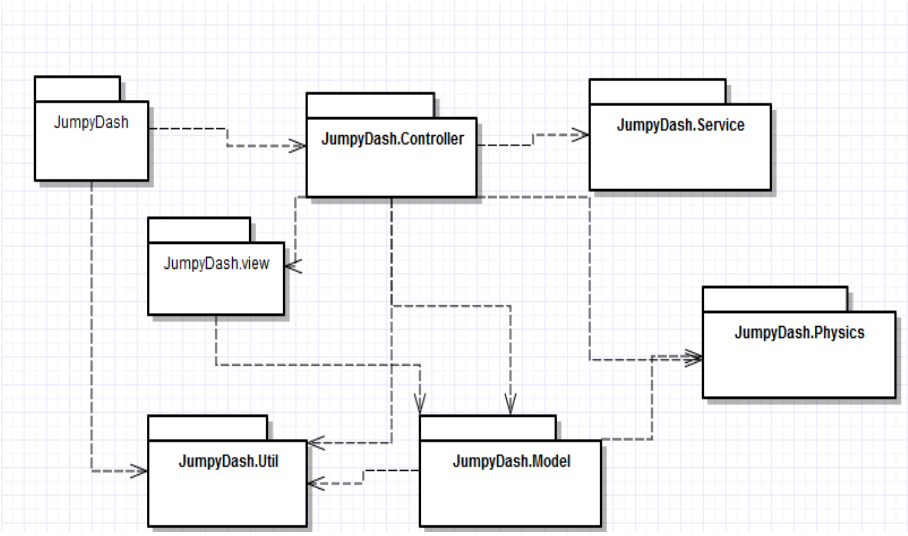
## 2.1 Overview

The application will be based on the MVC design pattern. Note that every object on the screen will have its own model, controller and view class. libGDX will be used to handle drawing of graphics, input control and audio.

## 2.2 Software decomposition

### 2.2.1 General

The application is decompsed into the following packages. The dotted lines indicate a dependency.




- **controller**, constains all the controller classes
- **view**, contains all the view classes
- **util**, contains all the util classes
- **service**, contains all the service classes
- **model**, contains all the model classes
- **physics**, contains all the Box2D classes with abstraction

### 2.2.2 Decomposition into subsystems

The application will have two separate subsystems, the first is responsible for handling all the physics and the second holds the functionality of reading a text file into an array which later will be used to create objects on the screen and set their initial positions.

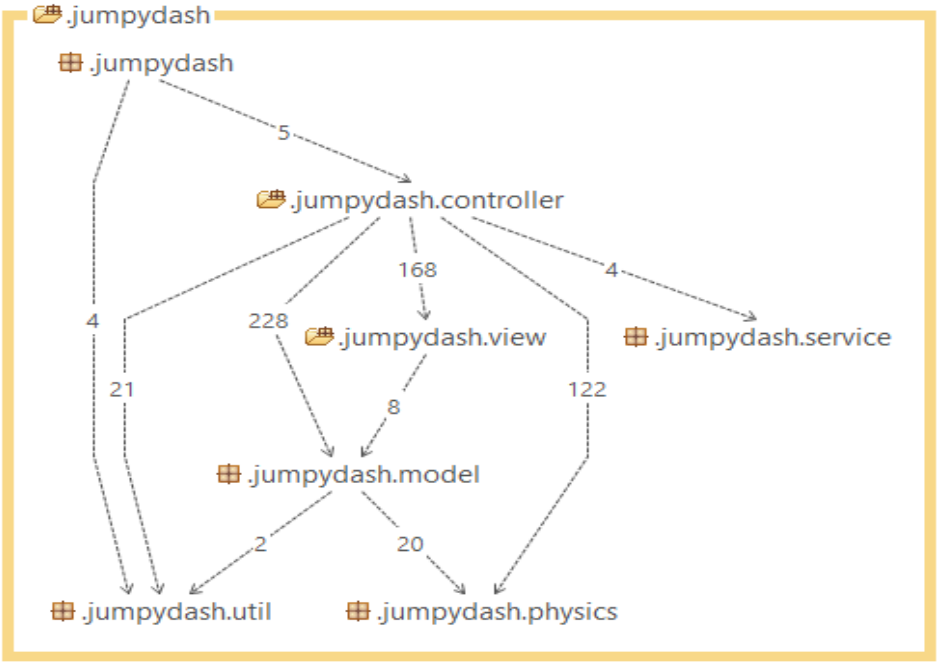
### 2.2.3 Layering

The system will be split up into the following layers:

- **controller**
  1. **screen**
  2. **collision**
- **view**
  1. **screen**
- **util**
- **service**
- **model**
- **physics**

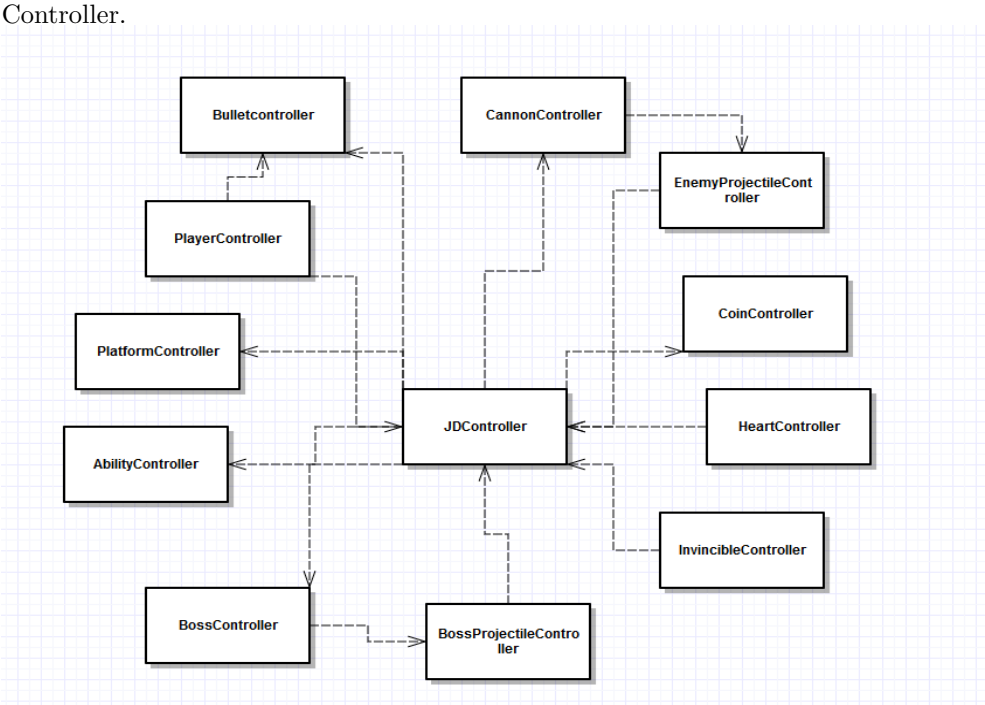
### 2.2.4 Dependency analysis

Dependencies between packages are as shown

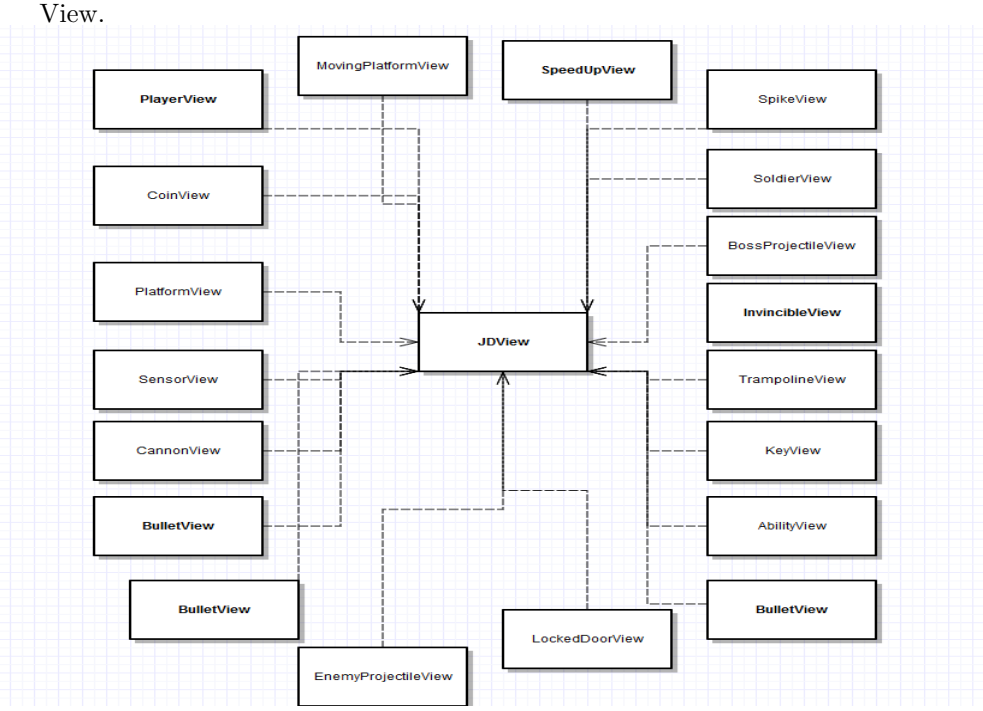


# Appendix

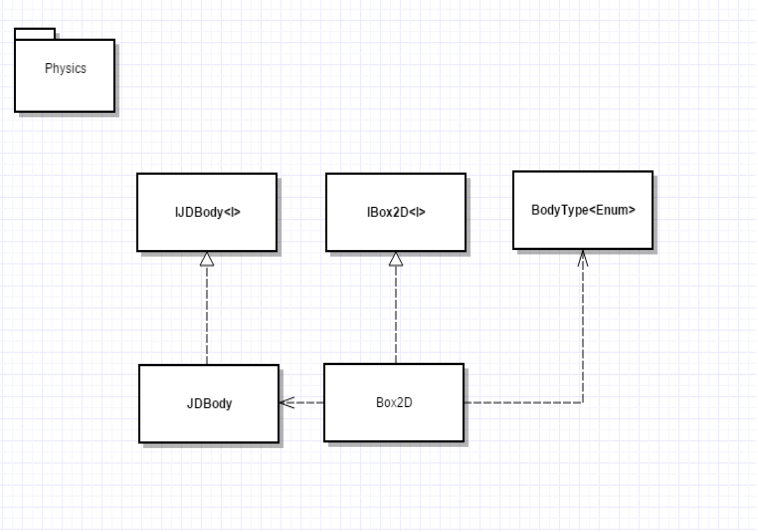
Controller.



View.




Physics.



Service.