Context Project Group 8

Rob van Bekkum(rvanbekkum)	4210816
Thijs Brands(tlmbrands)	4247132
Soheil Jahanshahi(sjahanshahi)	4127617
Aidan Mauricio(amauricio)	4195175
Joost van Oorschot(jjevanoorschot)	4220471

Thursday 15th May, 2014

Draft Design

Contents

1 Introduction

This document has been written with the goal of providing insight in the architecture design of our software solution. The architecture design will be subjected to change frequently due to the need for additional features or problems encountered with the current design. This document will be updated accordingly when these changes are made.

1.1 Design goals

We have divided our classes mostly over two categories: model and view. All classes that represent the game world or objects of the game world are part of the model. All classes that construct a way to view a part of the game world are part of the view category. One of our design goals was to keep coupling between these two categories as loose as possible. This way we can easily make changes to the game world our implement a new view without having to adjust a lot of classes.

2 Software Architecture Views

2.1 Subsystem Decomposition

DECOMP IMAGE HERE -will update tomorow - thijs.

The architecture of the system is divided into subsystems. These subsystems are the Game Model, the Driver Interface, the Navigator Interface and the Network Interface. All subsytems will be explained in this section.

The Game Model
The Game Model consists of the game world and all data related to the game. The actual game takes place in this subsystem. All other subsystems are interfaces that are used to alter the game model.

The Driver Interface
This is one of the two interfaces that will be directly used by the players to interact with the Game Model. The Driver Interface enables the user to see the car and its direct surroundings. It also gives the user the means to controll the car (steering and acceleration).

The Navigator Interface
The second way for players to interact with the Game Model is the Navigator Interface. This interface gives the user an overview of the game world in the form of a map. This enables the user of this interface to (verbally) guide the user using the Driver Interface through the game world. The Navigator Interface can be used to interact with the Game Model through the activation of powerups, which will alter specific parts of the game world.

The Network Interface
This interface is used to connect players with eachother. It is responsible for the connection of the two players within each team as well as the connection between all the teams. The Network Interface is also responsible for the concurrency between all Game Models.

2.2 Hardware/Software Mapping

In the Software aspect of the game, so called Libgdx Game engine is deployed in order to ensure that implementing the Physics and Collision of the game.

(mapping of sub-systems to processes and computers, communication between computers)

2.3 Persistent data management

(file/ database, database design)

3 Concurrency

(processes, shared resources, communication between processes, deadlocks prevention)