Architecture Design (Draft)

Computer Games Contextproject 2015-2016
Course TI2806, Delft University of Technology

Group PixelPerfect

April 28, 2016

Supervisor
Dr. ir. Rafael Bidarra

Teaching Assistents
Sander van den Oever
Jurgen van Schagen

Team

David Alderliesten	4368703	*J.W.D.Alderliesten@student.tudelft.nl*
Floris Doolaard	4362748	*F.P.Doolaard@student.tudelft.nl*
Dmitry Malarev	4345274	*D.R.Malarev@student.tudelft.nl*
Jesse Tilro	4368142	*J.Tilro@student.tudelft.nl*
Wouter Zirkzee	4398858	*W.J.Zirkzee@student.tudelft.nl*

Table of Contents

Introduction

Pixelperfect was assembled during the fourth quarter of the academic year of 2015-2016 for the Context Project course of the Computer Science bachelor program at the Delft University of Technology. The team consists of five members and was assigned to design, develop, and release a virtual reality game for the Oculus Rift platform.

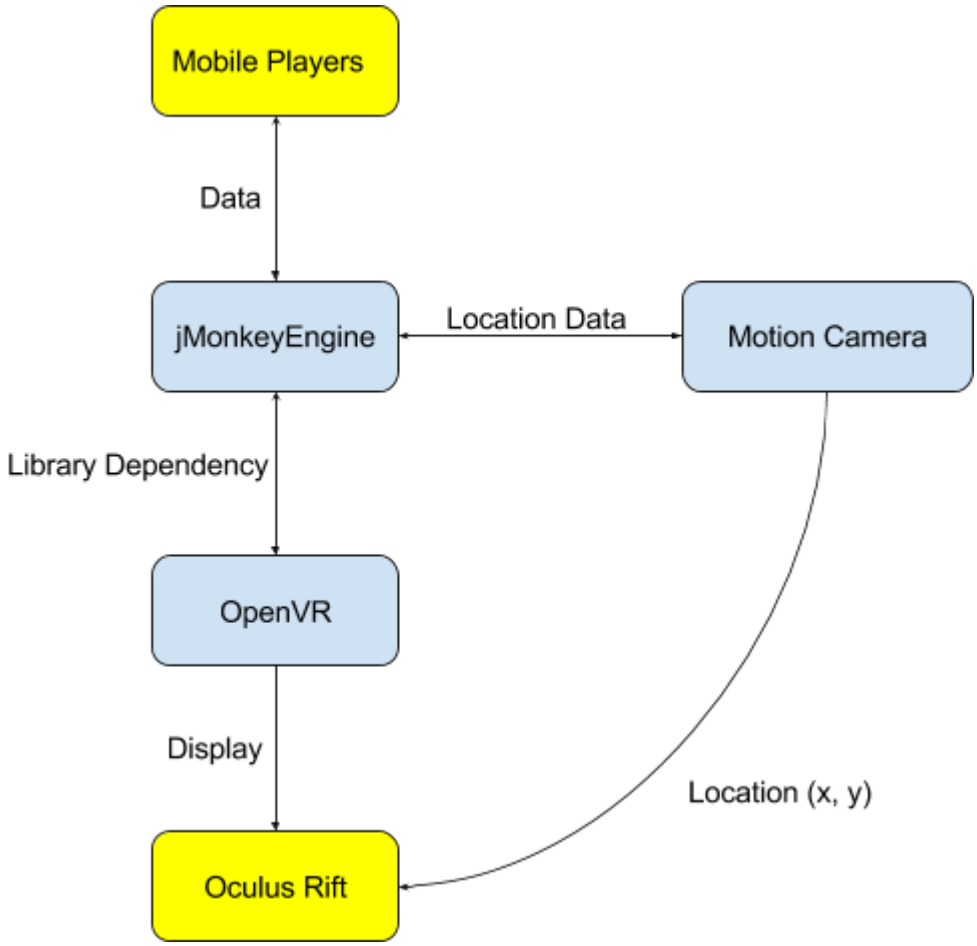
The project requires an array of specialized hardware devices in order to be played, including a virtual reality headset called the “Oculus Rift” which is at the heart of the software product. The connectivity between the other participants of the game on their phones or devices and the dependeability of the game on the motion tracking hardware makes for a tough challenge which will be solved using [TECHNOLOGY]. The continuous interdependability is outlined in this document and was at the core focus of the design for the game.

The focus of the design of the system architecture was to find a balance between performance, reliability, and manageability. The performance of the system was crucial due to the interdependability between the Oculus Rift player and all the mobile players, who must be able to reliably communicate during time-restricted moments of the game. The reliability refers to the reliability of the connections and interactions between the players, which must be stable to allow for a simple implementation and to prevent frustration during the act of connecting of the players. The maintainability is of the essence for the maintainence of the project, which must be advanced enough to offer a rich exeprience to the player, yet maintainable for the programmers to prevent bloat and clutter.

This document outlines the decision made for the software project with regards to the subsystem decomposition, the mapping of hardware and software, the method utilized for the storage of data, and how concurrency in systems is handled. A glossary can be found at the end of the document outlining certain important terms for the project.

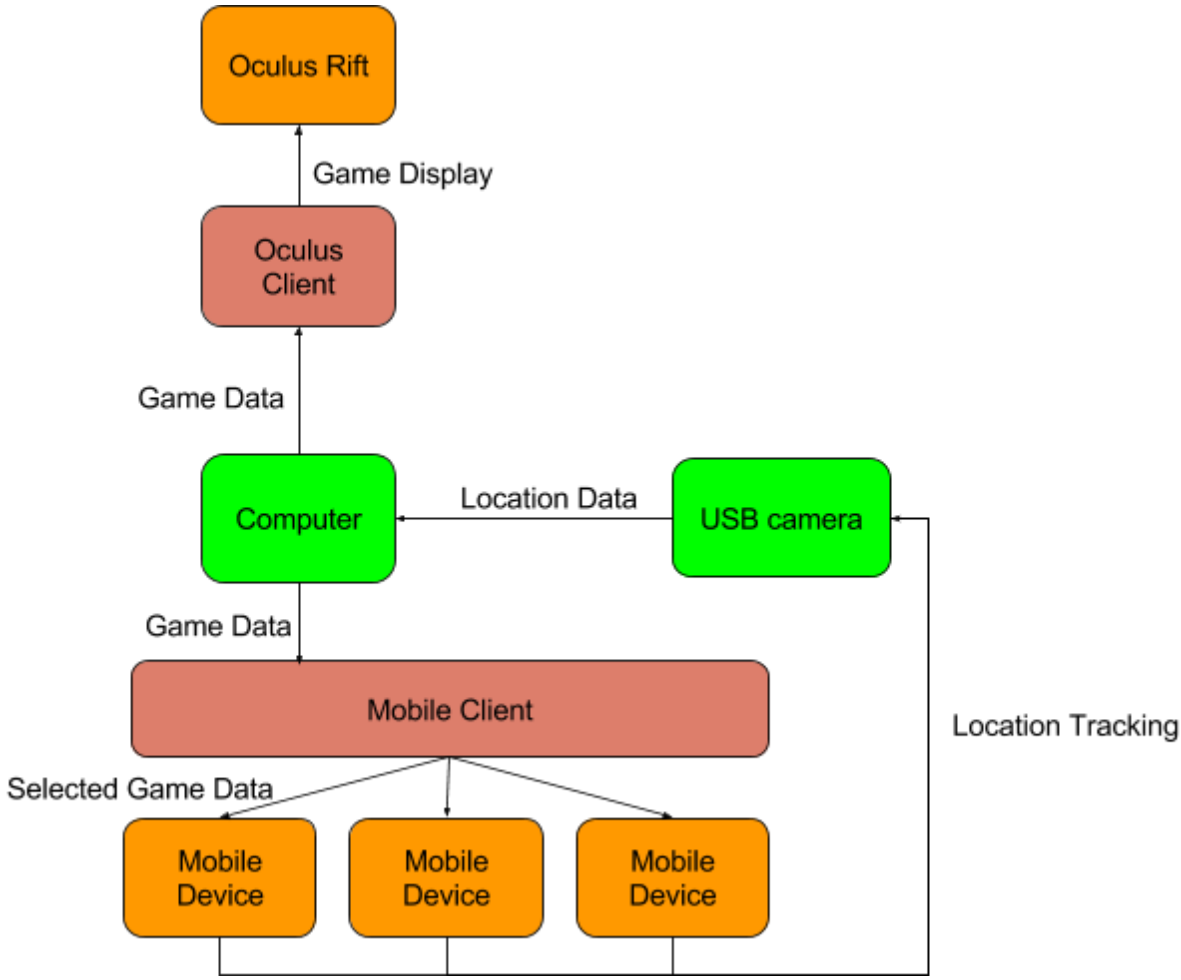
Architectural Views

Subsystem Decomposition



Hardware & Software Mapping

The peripherals required for the project are the Oculus Rift virtual reality system, the mobile devices used by the other players, and a central computer that acts as the processor for the Oculur Rift and the central access point for the other players. A USB camera acts as a peripheral which sends the location data of the mobile players to the computer, which is reflected in the view of the Oculus Rift player. The Oculus client handles the game data and displays the 3D interface on the Oculus Rift. The Mobile Client takes the game data and takes selected game data (player inventory, tools available, current score and disasters if active) and displays it on the mobile devices. The green colors represent peripherals, the red represents clients, and the orange represents player used hardware.



Persistent Data Mangement

The game will include a file system to track local highscores that will be updated and synchronized whenever the game is active. This will be done by using a file that stores a finite amount of scores. the file will be then read everytime the game is launched and updated whenever a game is over with a score higher than any of the ones exisiting already.
If a score has to be written in the file, the program will check at what place does it have to be in the file and write it, shifting the lower scores one spot and removing the score of the last place.

Concurrency

This is a multiplayer game, but most of the mini games will be solved on an induvidual basis. However some mini games will require some team effort to complete. This is where concurrency will come into play as a wrong order in a mini game will be important to complete or fail your mission. More information on concurrency will be added as we progress with the game and more issues become apparent.