Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4316: Senior Design II
Spring 2020

oVRworked
Cookums

Kasi Cross
Geetesh Kalakoti
John Livesay
Quinton Tompkins
Kevin Tung

Revision History

Revision	Date	Author(s)	Description
0.1	05.08.2020	KC, GK, JL, QT, KT	initial draft

CONTENTS

List of Figures

List of Tables

1 Introduction

Cookums is a virtual reality food truck simulation game built with Unity. The game will consist of 3 main features. The ability to cook and interact with food, the ability to receive orders from customers, and the ability to deliver the orders to customers and receive a score based on performance and correctness.

Unity games are built with a component based structure. Rather than outline how Unity organizes classes, this document will outline the overall gameplay system's major components and interactions with the engine, along with OpenVR and the VR kit's role in the overlying system architecture.

Gameplay will have an emphasis on using your hands to gather ingredients, prepare them into complete foods, and then serving them in a fast paced, exciting environment.

2 System Overview

The system is comprised of three distinct layers: the hardware layer, the API layer, and the engine layer.

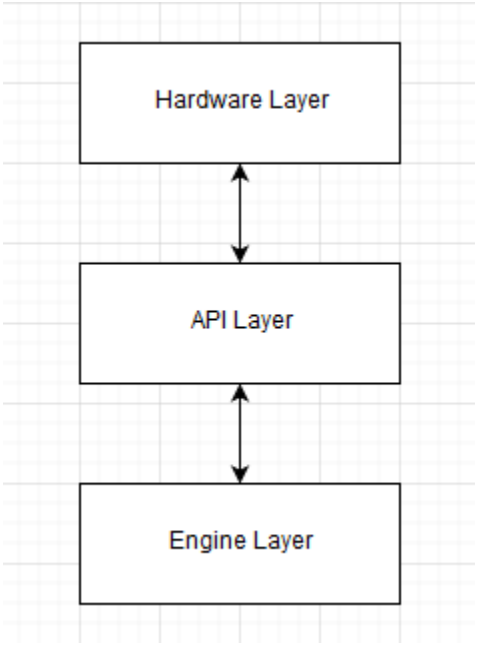

Figure 1: A simple architectural layer diagram

2.1 Hardware Layer

The hardware layer includes the virtual reality kit that serves as the one and only source of input and output for the user, making it critical to the project. At minimum, the kit is composed of a left and right hand controller and a head mounted display (HMD). The kit will supply the rest of the system with information about the user's keypresses along with hand and head transforms relative to the play area. In return, the hands may receive haptic feedback and the HMD will have continuous imagery provided by the system. No business logic is present in this layer; it is entirely an I/O peripheral.

2.2 API Layer

The API layer consists of the OpenVR system's API and runtime. It is responsible for serving as an intermediary of data between the hardware layer and the engine layer, bridging the gap between hard-

ware and software. This layer provides support to a multitude of virtual reality kits and abstracts each individual kit's I/O into a unified interface that works for all.

2.3 Engine Layer

The engine layer includes the entirety of gameplay elements, and handles several critical components. Unity handles lighting, rendering, physics and collision detection. The remaining subsystems handle other gameplay elements unique to Cookums. Unity coordinates interaction between the player controller, entities, and gameplay direction through ordering, difficulty, and customer AI agents. The entities are grouped collectively and labelled as the entity director subsystem, and the SteamVR Interaction System works to create a seamless connection for the player controller to work with other systems.

3 Subsystem Definitions & Data Flow

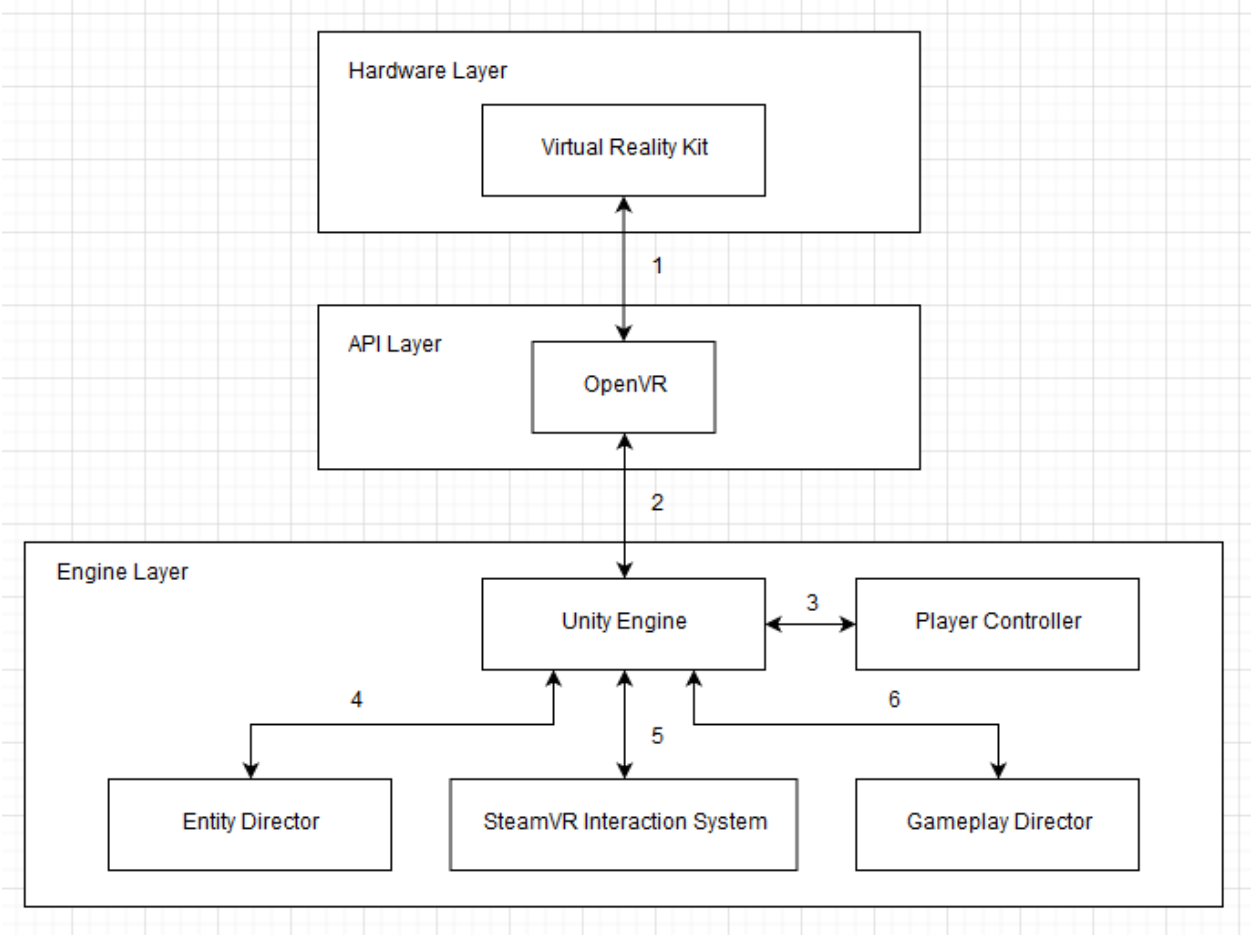

Figure 2: A simple data flow diagram

4 Hardware Layer Subsystems

The Hardware layer contains the Virtual Reality Kit. This layer will manage the player's interactions and observations of the system, communicating to the API layer about the player's actions and receiving input for the player to see and feel.

4.1 Layer Hardware

Component	Recommended system requirements	Minimum system requirements
Processor	Intel Core i5-4590/AMD FX 8350 equivalent or better	Intel Core i5-4590/AMD FX 8350 equivalent or better
GPU	NVIDIA GeForce GTX 1060, AMD Radeon RX 480 equivalent or better	NVIDIA GeForce GTX 970, AMD Radeon R9 290 equivalent or better
Memory	4 GB RAM or more	4 GB RAM or more
Video output	HDMI 1.4, DisplayPort 1.2 or newer	HDMI 1.4, DisplayPort 1.2 or newer
USB port	1x USB 2.0 or newer	1x USB 2.0 or newer
Operating system	Windows 7 SP1, Windows 8.1 or later, Windows 10	Windows 7 SP1, Windows 8.1 or later, Windows 10

Figure 3: System spec requirements.

4.2 Layer Operating System

Windows 7 SP1, Windows 8.1 or later, Windows 10

4.3 Layer Software Dependencies

OpenVR

4.4 Virtual Reality Kit

The Virtual Reality Kit communicates with the player via visual and physical stimulus. It will send and recieve data to the player. It will send data about the player's movement that is collected via the headset and the controllers. It will receive data for the player to see through the headset and feel through vibrations in the controllers.

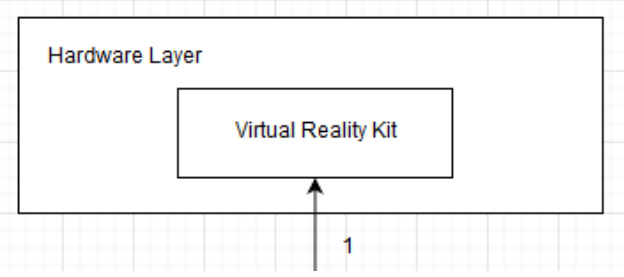

Figure 4: The Virtual Reality Kit Subsystem.

4.4.1 Subsystem Hardware

VIVE VR headset, wireless adapter (for headset), two wireless controllers, two base stations, and one link box.

4.4.2 Subsystem Programming Languages

The programming language used by the subsystem is C#.

4.4.3 Subsystem Data Structures

This portion of the system is a blackbox portion of the VR system. Therefore there are no data structures to reference.

4.4.4 Subsystem Data Processing

This portion of the system is a blackbox portion of the VR system. Therefore there is no data processing to reference.

5 API Layer Subsystems

The API layer consists of the OpenVR subsystem.

5.1 Layer Hardware

No hardware is used in this layer.

5.2 Layer Operating System

No Operating system used for this layer

5.3 Layer Software Dependencies

This subsystem relies on the OpenVR library to communicate between the other two layers.

5.4 OpenVR

The OpenVR subsytem communicates with the virtual reality kit subsystem and the Unity Engine subsystem, bridging the gap between hardware and software in a unified interface for all virtual reality kits. This was chosen in order to reduce the amount of work necessary for the project.

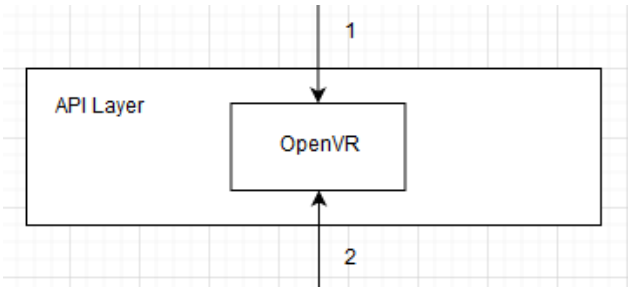

Figure 5: Example subsystem description diagram

5.4.1 Subsystem Hardware

No hardware used in this subsystem.

5.4.2 Subsystem Operating System

No operating system used in this subsystem.

5.4.3 Subsystem Software Dependencies

This subsystem relies on the OpenVR library to communicate between the other two layers.

5.4.4 Subsystem Programming Languages

The programming language used by the subsystem is C#.

5.4.5 Subsystem Data Structures

This portion of the system is a blackbox portion of the VR system. Therefore there are no data structures to reference.

5.4.6 Subsystem Data Processing

This portion of the system is a blackbox portion of the VR system. Therefore there are no data structures to reference.

6 Engine Layer Subsystems

The Engine layer consists of 5 subsystems. The 5 subsystems are the Unity Engine, Player Controller, Entity Director, SteamVR Interaction System, and the Gameplay Director.

6.1 Layer Hardware

Any high end PC or laptop with Windows 7 or newer installed.

6.2 Layer Operating System

Windows 7 SP1, Windows 8.1 or later, Windows 10.

6.3 Layer Software Dependencies

Unity Engine, Unity library, and SteamVR.

6.4 Unity Engine

The Unity Engine subsystem facillitates communication and coordinates the other components of the game. It also handles physics, objects, lighting, rendering, collision, and other basic modern game engine capabilities.

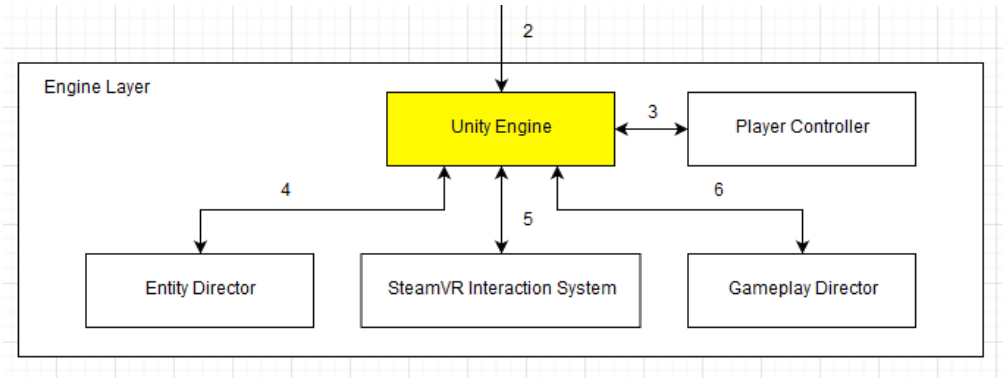

Figure 6: The Unity Engine Subsystem.

6.4.1 Subsystem Operating System

Windows 7 SP1, Windows 8.1 or later, Windows 10.

6.4.2 Subsystem Software Dependencies

Unity Library.

6.4.3 Subsystem Programming Languages

The programming language used by the subsystem is C#.

6.4.4 Subsystem Data Structures

6.4.5 Subsystem Data Processing

6.5 Player Controller

The player controller subsystem is responsible for the player's position and orientation in the world space, as well as movement.

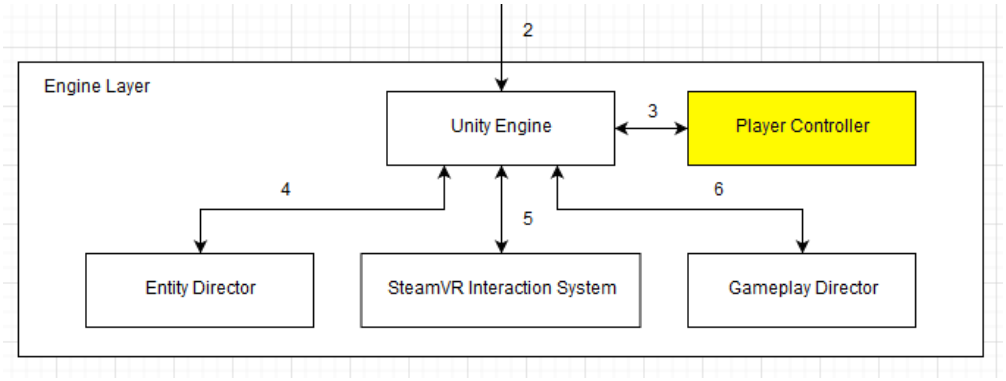

Figure 7: The Player Controller Subsystem.

6.5.1 Subsystem Hardware

Hardware used in this subsystem is included within the VR kit. This includes the headset, and the handheld controllers.

6.5.2 Subsystem Operating System

This subsystem doesn't include any subsystem ther than the one preloaded into the headset.

6.5.3 Subsystem Software Dependencies

This subsystem is able to connect to Unity through SteamVR.

6.5.4 Subsystem Programming Languages

The programming language used by the subsystem is C#.

6.5.5 Subsystem Data Structures

Packets are structured within SteamVR

6.5.6 Subsystem Data Processing

All algorithms are within the Unity library.

6.6 Entity Director

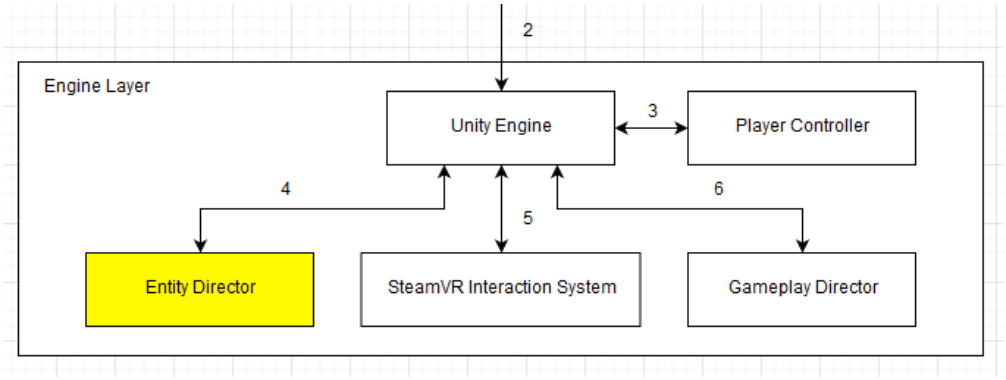

Figure 8: The Entity Director Subsystem.

6.6.1 Subsystem Hardware

6.6.2 Subsystem Operating System

6.6.3 Subsystem Software Dependencies

6.6.4 Subsystem Programming Languages

The programming language used by the subsystem is C#.

6.6.5 Subsystem Data Structures

6.6.6 Subsystem Data Processing

6.7 SteamVR Interaction System

The SteamVR Interaction System (SRVIS) handles much of the virtual reality based interactions. Without it, the game would still work in a 3D physical sense, but this system allows for our hands to become manipulators of the VR world.

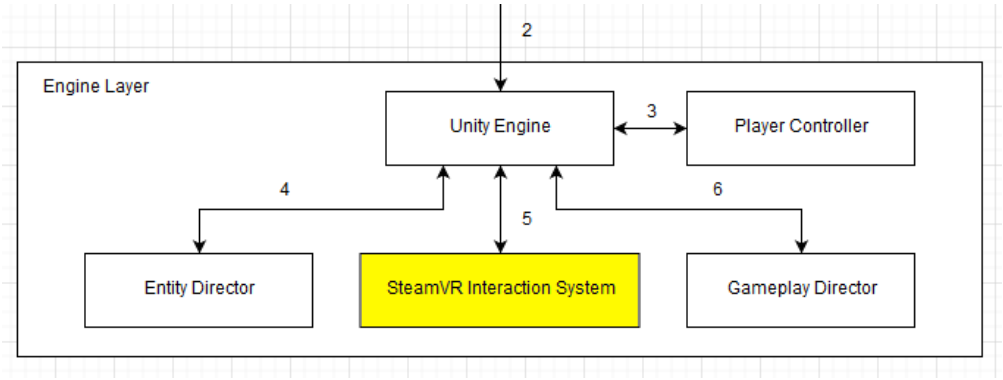

Figure 9: The SteamVR Interaction Subsystem.

6.7.1 Subsystem Hardware

No hardware needed.

6.7.2 Subsystem Operating System

No operating system needed.

6.7.3 Subsystem Software Dependencies

Steam is needed as is making sure settings within Unity were set up correctly.

6.7.4 Subsystem Programming Languages

The programming language used by the subsystem is C#.

6.7.5 Subsystem Data Structures

Packets were assembled automatically within SteamVR

6.7.6 Subsystem Data Processing

No algorithms are needed to be explained.

6.8 Gameplay Director

The gameplay director handles actual gameplay objective elements, not necessarily the entire element of gameplay.

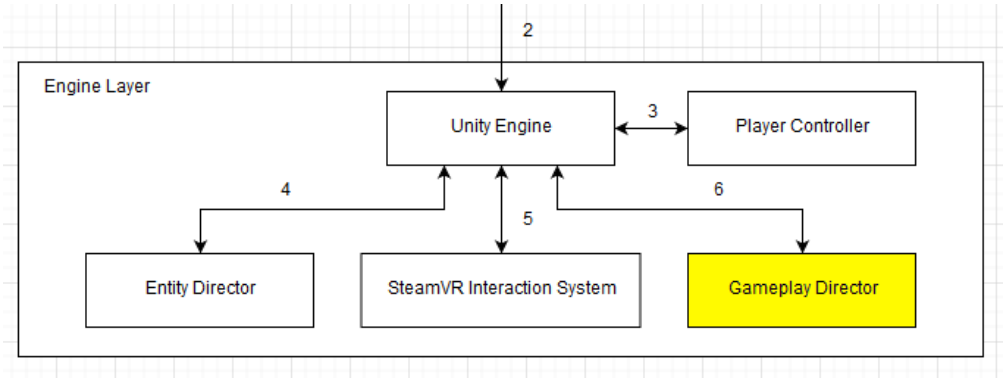

Figure 10: The Gameplay Director Subsystem.

6.8.1 Subsystem Hardware

No hardware needed.

6.8.2 Subsystem Operating System

No operating system needed.

6.8.3 Subsystem Software Dependencies

Player input required.

6.8.4 Subsystem Programming Languages

The programming language used by the subsystem is C#.

6.8.5 Subsystem Data Structures

Player actions and movement are recorded by the headset and controllers and then transferred to the computer via SteamVR.

6.8.6 Subsystem Data Processing

Algorithms processed and provided within Unity libaries

7 Appendix A

References