Final Project Report

Jonathan Sumner Evans*, Robinson Merillat†, and Sam Sartor‡
Department of Computer Science, Colorado School of Mines
Golden, Colorado
Email: *jonathanevans@mines.edu, †rdmerillat@mines.edu, ‡ssartor@mines.edu

***Abstract*—Virtual Reality (VR) Technology has been developing rapidly over the past decade. Current solutions such as Unity attempt to use old programming languages and paradigms to implement VR environments and thus limit developers' abilities to create new and unique environments. With every cutting edge technology, new paradigms and design patterns must be invented. In this paper, we discuss a deferred immediate mode (DIM) application architecture suitable for implementing large virtual reality applications. We present a library which utilizes this architecture and a few case studies of this library in use.**

I. Introduction

The Virtual Reality (VR) market is growing rapidly. The International Data Corporation (IDC) projects that revenues for the combined Augmented Reality (AR) and Virtual Reality markets will grow from $5.2 billion in 2016 to more than $162 billion in 2020 [1]. This flourishing new industry has created an exciting new field of Software Engineering with great potential for revolutionary new design paradigms and program architectures.

Most current frameworks and libraries attempt to apply old design paradigms and program architectures that are well-suited for 2D user interfaces and rendering 3D environments to a flat screen to virtual reality. Although these endeavors have been successful, they do not fully explore potential new design paradigms and program architectures.

Our goal was to a) explore and find a generally applicable system for developing virtual reality applications and b) explore systems for interacting with a VR environment. We wanted a modern, fast, and practical framework to create virtual reality applications. We developed our approach to meet the following requirements.

A. Performant

Virtual Reality requires a very high frame rate. A virtual reality program must be capable of generating 90 frames per second (fps). Because there are two eyes, and thus two screens to render to, the effective required frame rate is 180fps. Achieving this frame rate is resource intensive and requires highly efficient and optimized code. Additionally, multithreading is imperative so that long-running processes can occur without blocking the user interface. This is unlike a traditional desktop user interface where blocking the UI process for a second does not have a major affect on the usability of the program.

B. Volumetric

C. Programmable

D. Flexible

E. Modular

We chose to take a step back from the predominant design philosophies and explore alternative architectures.

In this paper, we present a new program architecture and user interface design paradigm which builds on and combines concepts from many previous architectures.

II. Evolution of Deferred Immediate Mode

A. Entity Component System

B. React-Like Architecture

C. Event Tree

III. Deferred Immediate Mode

A. Deferred

B. Immediate Mode

IV. Flight

Flight is our implementation of a VR UI library using the DIM architecture. It is written in Rust and is designed to be highly modular. It implements the DIM architecture using Rust `FnOnce` closures and resolves state using a *guru* system.

A. Language

B. Dependencies

C. Modular

D. Deferred

E. Immediate Mode

F. Gurus

V. Case Studies

A. Let's Get Physical and Snowflakes — Physics

B. Workbench — SOMETHING

C. VRsh — Global User Interface

VI. Comparison to Alternative Libraries

A. A-Frame

VII. Conclusion

References

[1] IDC. *Worldwide Revenues for Augmented and Virtual Reality Forecast to Reach $162 Billion in 2020, According to IDC*. Aug. 15, 2016. URL: https://www.idc.com/getdoc.jsp?containerId=prUS41676216.