VR

*INTERACTIVITY PACK*

*Project S.D.F.*

*Andrea Carver*

*Loyola Marymount University*

*CMSI 402 – Individual Senior Thesis*

*2.0 – Preliminary Project Proposal ––––––*

*Project Proposal*

The arrival of consumer virtual reality (VR) is accelerating the desire for enterprise solutions that utilize VR. Yet, many game engines used to create VR experiences lag behind the progress in hardware. Critical functionalities are still missing, forcing content creators to start from scratch with each project. Content quality and development speed suffer as a consequence.

Unreal Engine 4, a leader among these game engines, needs a set of modular, reusable components that can expedite the creation of VR content. Specifically, Unreal Engine 4 needs a pre-built system for grabbing objects, locomotion, and managing inventory in virtual reality. Nearly every VR experience will use these mechanics; hence, designing them well once will be extremely profitable.

In my individual senior project, I will finish my work from summer 2017 on fulfilling these needs. I will be programming primarily in C++ with occasional support through Unreal Engine 4’s visual scripting language. If possible, I will package my work as a downloadable plugin for public use. In summary, I will create reusable mechanics that will make VR experience creation in Unreal Engine 4 more accessible to non-programmers.

*3.0 – Project Proposal ––––––––––––––––––*

*3.1 Project Description*

The VR Interactivity Pack is a plugin for Unreal Engine 4. It aims to close the gap in available software for virtual reality developers who may have a more artistic than technical background. Such developers may not be able to program fundamental virtual reality mechanic themselves, forcing them to abandon their creative visions for virtual reality experiences.

Given the state of virtual reality hardware development, the most important, non-trivial ability in virtual reality is the ability to pick up objects. Disappointingly, major game engines like Unreal Engine 4 do not offer templated starter solutions for picking up and interacting with objects. The VR Interactivity Pack, using C++ programming and Unreal Engine scripting macros, will provide these necessary templates.

After installing the plugin, users will use the templated interactions by adding one of the following components to an Actor (an object in Unreal Engine 4): a Simple Grab Component, a Twist Grab Component, a Piston Grab Component, or a Lever Grab Component. Each grab component allows the player to grab and interact with the component as specified by the grab type. For instance, a Twist Grab Component can be grabbed, twisted a certain amount, and released.

Each of the components can be configured more specifically to meet the user’s interaction needs. For instance, a Twist Grab Component could only be twisted a certain number of degrees, a Piston Grab Component could only be pulled out a certain length, and a Lever Grab Component could only be rotated a certain number of degrees.

On their own, the grab components are invisible, only offering the interaction functionality. Each different type of grab component will have one or more attributes where users can input geometry to ‘skin’ the grab components. This way, the grab components can have any customized appearance.

Project maintenance will be performed by myself, Andrea Carver. Project updates are likely to occur since I am one of the end users for this project. As I use the plugin and move on to more advanced interaction types, I will be expanding the types of grab components. Possible future grab component types include omnidirectional lever grab, elastic grab, and string grab.

*3.2 Justification*

Completing this project will require me to synthesize knowledge gained from both the animation and computer science department. I must have familiarity with the target platform, Unreal Engine 4, as well as the programming language, C++. The largest challenge of the project will likely be the programming itself, as the computer science major has focused on more modern programming languages. I have a high level of familiarity with the game engine, but not with C++. However, I wish to become more proficient in C++ so that I may apply lessons from computer science classes to future virtual reality projects. Hence, this project is a perfect way for me to gain experience.

*5.0 – Software Requirements Specification*

*5.1 Introduction*

The VR Interactivity Pack provides two core components: a specialized motion controller actor and a new component for grabbing functionality. The motion controller actor is intended to be used as a parent class, from which users can create a child class to meet their more specific needs. The grab component is intended to be added as a component to actors that need to be interacted with; it adds the capability to interact with the player. As seen in the diagram below, the motion controller actor and grab component do not exist only on their own, but are either directly connected to an actor/pawn as a component or pointed to with a pointer.

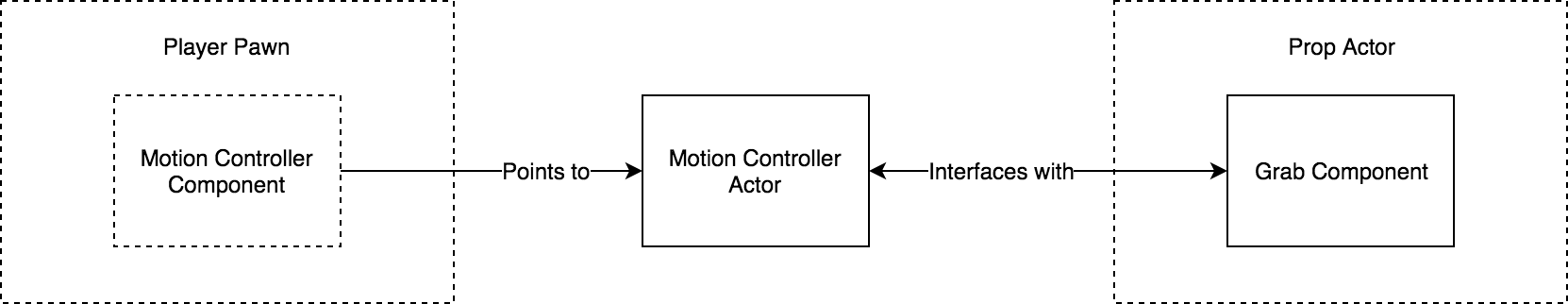


Fig 5A: System components diagram. Components created by the plugin are rendered with solid lines and components created by the user are rendered with dotted lines.

The remainder of section 5 is structured as follows. Section 5.2 contains functional requirements, section 5.3 contains performance requirements, and section 5.4 contains environment requirements. Functional requirements specify what the plugin should be able to do while performance requirements specify how the plugin should meet those functional requirements. Environment requirements specify what platforms, hardware, and software are required to run or develop the plugin.

*5.2 Functional Requirements*

[ Functional Requirements intro ]

5.2.1 Subsection 1

Description of subsection 1. This one wraps around the page and wraps around the page and wraps around the page.

5.2.1.1 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

5.2.1.2 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

5.2.1.3 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

5.2.2 Subsection 2

Description of subsection 2. This one wraps around the page and wraps around the page and wraps around the page.

5.2.2.1 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

5.2.2.2 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

5.2.2.3 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

*5.3 Performance Requirements*

[ Performance Requirements intro ]

5.3.1 Subsection 1

Description of subsection 1. This one wraps around the page and wraps around the page and wraps around the page.

5.3.1.1 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

5.3.1.2 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

5.3.1.3 The subsection item 1 shall do this thing. This wraps around the page and wraps around the page.

5.3.2 Subsection 2

Description of subsection 2. This one wraps around the page and wraps around the page and wraps around the page.

5.3.2.1 The subsection item 2 shall do this thing. This wraps around the page and wraps around the page.

5.3.2.2 The subsection item 2 shall do this thing. This wraps around the page and wraps around the page.

5.3.2.3 The subsection item 2 shall do this thing. This wraps around the page and wraps around the page.

*5.4 Environment Requirements*

[ Environment Requirements intro ]

5.4.1 Development Environment Requirements

5.4.2 Execution Environment Requirements