

In the Shade - Programmers Documentation

Pascal Ziegler

March 16, 2016

Abstract

In the virtual reality game "In the Shade" the user can switch between two roles: A God with powerful hands and a small, pretty helpless slug. As God she is placed in space, looking at a planet which can be zoomed, rotated and manipulated. Objects on the planet can be replaced and rotated. As slug the user is placed on this planet. She can move around on it and collect items, but needs to stay out of the light as it can hurt her. Sometimes she will need godly help to achieve this; for example by turning the planet out of the sunlight, or blocking light with objects. Here is where the God mode kicks in again.

One of the key features of the game is that is playable in many different spatial contexts.

1 Introduction

The virtual reality project "In the Shade" was developed during the ws 2015/16 in a group of two: [Markus Frei \(MAD\)](#) and myself as programmer. The goal was to create a VR game experience that is both simple and innovative.

2 The game

2.1 Input and Display

The game is targeted towards consumer VR headsets like the Oculus Rift or the HTC Vive. To simulate the (not yet available) input controllers of those head mounted displays, Sixense's Razer Hydra was used.

2.2 Spatial Context

One of the key features of the game is that is playable in many different spatial contexts. The user may sit still and just move her head, on a swivel chair or even move or walk around a room (with the Vive for example). This makes the concept of the experience adaptive to all current and coming VR devices, especially mobile ones. Still the quality of the experience will increase with every degree of freedom.



Figure 1: In the Shade wallpaper

2.3 Concept

In the game "In the Shade" the user can switch between two roles: A God with powerful hands and a small, pretty helpless slug. As God she is placed in space, looking at a planet which can be zoomed, rotated and manipulated. Objects on the planet can be replaced and rotated. As slug the user is placed on this planet. She can move around on it and collect items, but needs to stay out of the light as it can hurt her. Sometimes she will need godly help to achieve this; for example by turning the planet out of the sunlight, or blocking light with objects. Here is where the God mode kicks in again.

2.3.1 Controls and movement

The inner control buttons of the Razer Hydra are used to switch between the two game modes.

When in God mode the user can not move further than her hmd will allow (which is not far in space) but look around and see her hands. By pressing both upper triggers of the Razer Hydra she can zoom in and out of a planet by performing a zooming gesture like on touch inputs. By pressing the other trigger she can pick up objects to rotate or replace them. While an object is placed "inside" a hand of the user this hand will not be visible. This decision was based on Alchemy Labs work on "[Job Simulator](#)". They state that the user will prefer an invisible hand over a bad animation which may even need to move the hand mesh away from its actual position. None of the users that tested the game did even notice that their hands "disappeared" while holding an object. The user can not just pick up objects but also rotate the planet by grabbing it. The concept of this rotation was inspired by the "[Trackball](#)" [3D visualisation concept](#). This concept can be used if a 3D object on a 2D screen needs to be rotated by using just two degrees of freedom (horizontal and vertical movement

of the mouse for example).

When in slug mode the user can set a target point by just looking around. If she looks at a position where she can possibly move to a glowing ball will mark this position. By pressing a trigger the slug (user) will slowly move towards this point. This VR movement concept was inspired by the game [Lands End](#). It strongly reduces the risk of motion sickness, as it completely eliminates rotation, which is the main cause of it. The difference here is that the target points are not fixed, but can be set by the user intuitively by just looking at the position where she wants to move. The few users that tested this did not experience any motion sickness at all.

2.4 Implementation

The project was realized with Unreal Engine 4.10. To synchronize the work, GitHub was used. All code was written in C++.

2.4.1 Plugins and Externals

As the gravity of the game could not be realized with the build in physics simulation on the Unreal Engine a [custom gravity plugin](#) was used. It made it possible to use point gravity at the center of the planet. For the Razer Hydra [Getnamos plugin](#) was used.

2.4.2 Custom Rotations

Besides the countless bugs and incomplete documentation of the Unreal Engine one of the trickier parts of the project were the custom rotations. To make the slugs "up vector" always point out of the planet was possible by simple vector math. The rotation of the planet was a different task:

Given the center of the planet, the position of the user's hand and the position of the user's hand the frame before a new rotation needs to be computed. To do so the concept of [quaternions](#) turn out very helpful. Quaternions can describe a rotation of a certain angle around an axis which is exactly what we need here. Let's call the vector from the center of the planet towards the position of the hand v_2 and the vector from the center of the planet towards the old position of the hand v_1 . Then the cross product of v_1 and v_2 will be the rotation axis and the dot product of the two the cosine of the angle.

2.4.3 Blueprints

The Blueprints Visual Scripting system in Unreal Engine is a complete gameplay scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor. This system is extremely flexible and powerful as it provides the ability for designers to use virtually the full range of concepts and tools generally only available to programmers (<https://docs.unrealengine.com/latest/INT/Engine/Blueprints/index.html>).

This made it possible for the designer to easily rearrange and test simple game logic implemented by the programmer. An example in this project is the "Earth Walker" (see figure 2) or "God Character" (see figure 3), both Blueprints which described the player during the two modes.

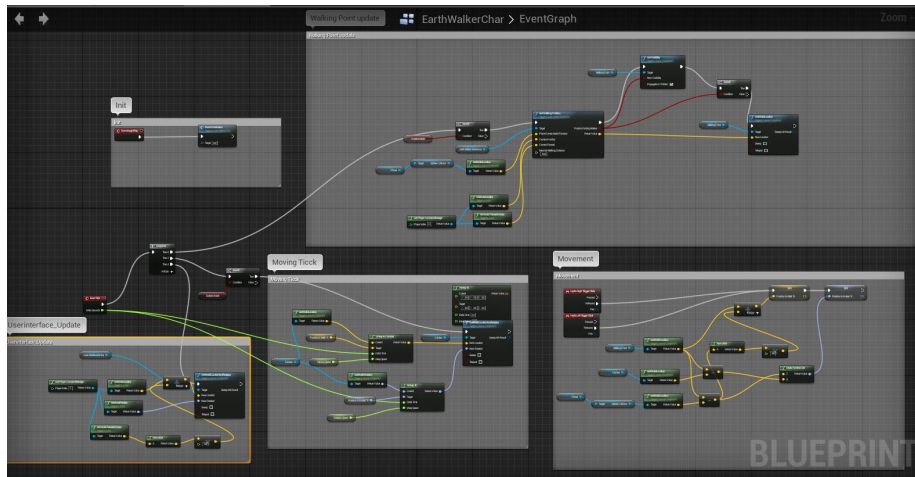


Figure 2: The Earth Walker Blueprint

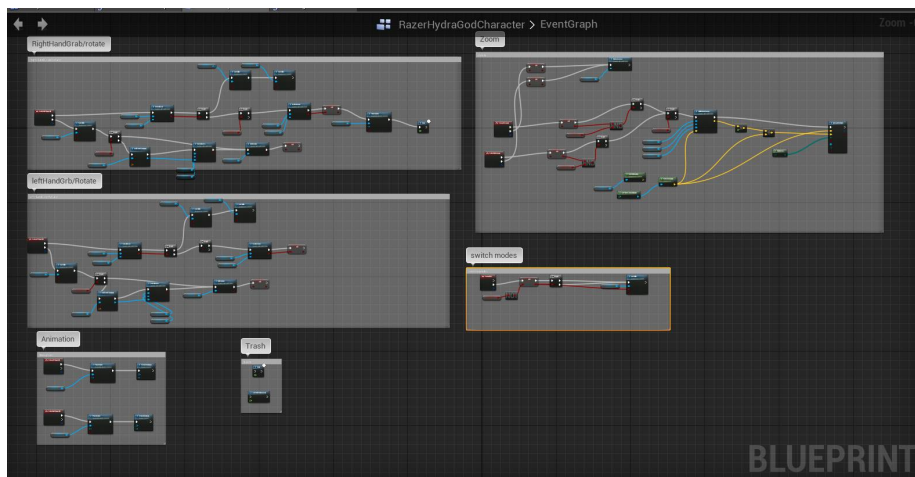


Figure 3: The God Character Blueprint

3 Art and Design

All content like assets, textures, etc. were done by [Markus Frei](#) at HBKsaar.

4 Future Work

The project was presented as "proof of concept" at the Bachelors presentation of Markus Frei. From now on this prototype needs to be tested by as many different people as possible to improve it. The user interface needs to be finalized and the game content itself needs to be multiplied. The next step could be a Kickstarter campaign.