Daydream

Project Part 4

Team 13
JC Orolfo
Matt Fredrickson
Sarah Koh

CS425
University of Nevada, Reno
Department of Computer Science and Engineering
Instructor: Sergiu Dascalu
Advisor: Eelke Folmer
December 11th, 2014

Introduction

Daydream has largely remained true to the design proposed and has made large leaps in development. Currently, our project has a working, 3D menu system with game options and terrain. We are also about to get small demos working for the Cardboard and the Mindwave that showcased destroyable objects when you look at them and reach high values for attention and meditation brain readings. The Mindwave implementation consisted of checking connections with the Mindwave, grabbing raw data values, and translating them into values that are usable for our scripts. Cardboard implementation consisted of importing the dual camera prefab and adjusting render settings to optimal values, and modifying prefab to work with our scripts. The problem we encountered was the integration of the Mindwave with the Cardboard. We are working on getting these components to work together.

The only major change to occur was is the scale of the game. The main focus of the original design was to create an extensive, large game. We have since started to err towards the creation of several small, simply mini-games that really showcase the Mindwave and Cardboard’s symbiotic relationship. Future development will focus on delivering rapid content that emphasizes a specific type of interaction between the player, the Mindwave, and the Cardboard. For example, one such mini-game could be placing objects in a virtual reality environment by focusing on a point in the terrain.

Daydream is written in Unity. The source code attached is only for the scripts we wrote ourselves. The scenes containing the objects, terrain, cameras, etc. is not included with the source code, but the objects are outlined below in the “Overview” section along with what scripts they use.

Contributions

JC was mainly in charge of scripting and raycast implementation

Researching Unity Functionality - 10 hours
Attempted Mindwave Implementation - 5 hours
Attempted Cardboard Implementation - 2 hours
Raycasting Research - 5 hours
Raycasting Implementation - 4 hours
Raycast Script Implementation and Refactoring - 2 hours
Unity Testing Scene Creation and Design- 1 hour
Unity Particle Creation - 1 hour

Sarah was in charge of Mindwave integration and scripting for the Mindwave

Researching Unity Functionality - 10 hours
Mindwave Implementation - 4 hours
Mindwave Research and Testing - 3 hours
Mindwave Script Implementation - 3 hours
Unity Testing Scene Creation and Design - 1 hour

Matt was in charge of Google Cardboard integration into the Unity project, as well as designing the terrain and main menu scene.

Research for Unity engine proficiency: 10 hours
Cardboard implementation: 4 hours
Terrain design: 2 hours
Main menu implementation: 3 hours
Integration testing: 3 hours

Overview

Scene

The scene consists of all objects that create our world.

Object	**Scripts**	**Description**
Main Camera	Mouse Orbit Raycasting Raycast Destroy Display Data	The main camera serves as the main view of the player. It also handles the raycasting onto objects and the HUD that shows brainwave readings.
Destroyables	Highlight Spin	Objects that are able to be destroyed if brainwave readings exceed a certain threshold. Particle effects come off of these when destruction is attempted.
Player	None	The player is your character, viewed in first person. Movement is viewed from the player's perspective.
Terrain	None	The terrain is the the ground on which the player stands along with any other objects part of the scenery. In our case, we had a grassy field and mountains.
Skybox	None	A panoramic textures that encompassed the entire scene. This ensures that there is always something(a sky) in the distance.
Particles	None	Two such particle systems exist for our project that emit small objects. Particles are used to indicate the position of the cursor as well as indicating whether a

Scripts

All scripts overviewed below were written by our team and used with the objects in the scene.

Name	**Description**
Raycasting	Used to indicate a cursor at the center of the screen with particle effects and detects any intersections with other objects.
Raycast Destroy	A modified version of the raycasting script seen in the Cardboard, but used for the Mindwave. Used to indicate a cursor at the center of the screen with particle effects. The modifications include object destruction and brain wave readings
Mouse Orbit	A script to have the mouse orient around a player object. This gives the appearance of a first-person view. Looking around is done with the mouse.
Display Data	Display Data is used to grab information from the Mindwave brain readings and
Highlight	Highlights objects depending on whether the cursor is over them or not.
Spin	Spins an object at a constant rate. This was used on destroyable objects to make them easier to identify as destroyable.