EE267 Final Project Report

Yun Zhang, Qiwen Fu

June 9, 2018

1 Introduction

VR Games provide users with an immersive game experience, which have superiority beyond traditional games. For the final project, we built an interactive VR game **Interstellar**, with Unity, View Master VR Glasses and a single VRduino. The main character is a spaceship traveling in the universe with various planets and hearts flying towards the character. The goal of this game is to survive as long as possible by avoiding collision with planets and trying to collide with the hearts. The spaceship has a fixed forward speed and the player can use a VRduino to control the orientation of the spaceship.

2 Game Design

Our spaceship travels in a starry universe, just like those in the film Interstellar directed by Christopher Nolan. The user wearing the VR glasses is in a first-person view with the spaceship. The spaceship travels forward with a fixed speed, whose orientation is controlled by the user with a VRduino. The IMU in the VRduino tracks the orientation (yaw, pitch, roll) of the head, so that the orientation of the spaceship is synchronous with that of the user's head. Traveling in the universe is full of danger, as various mysterious planets are flying by unpredictably. Our spaceship has 10 lives by default. Collision with the planets will lead to explosion and reduction on the lives. There are also lovely hearts flying by randomly. Collision with the hearts will add one additional life. The final score is the surviving time in the game before the lives are exhausted.

3 Game Development

The game is developed in Unity 2018.1.2 and Arduino IDE. We built three scenes, including a start scene, a main playing scene and an end scene. The start scene enables the user to hit "space" to start the game. The main scene displays the current score and lives and through which the user can see the movement of the spaceship. The end scene displays the final score and enables the user to hit "space" to return to the start scene. We use stereo rendering to render two images for the left eye and the right eye respectively.

The orientation measurements are obtained from the IMU in VRduino, which communicates with the PC through USB port. In each time frame, the orientation of the spaceship is updated accordingly. In order for the camera to follow the spaceship, the spaceship, directional light, scores displayed and rendered cameras are all mounted on a single first person controller, which travels with a certain speed.

The planets and hearts are spawn in a random position within a box ahead of the spaceship. To be detailed, the initial positions of the planets and hearts are always in the front relatively to the spaceship. The direction of them are towards the spaceship, with some randomness, and the velocity and angular velocity are also random.

We use Unity's collider component for detecting collision between the game objects. When collision between the spaceship and the planets is detected, a fire object is instantiated to display explosion effect, a sound effect is played and the number of lives is reduced by 1. When collision between the

spaceship and the hearts are detected, a different sound effected is play and the number of lives is increased by 1.

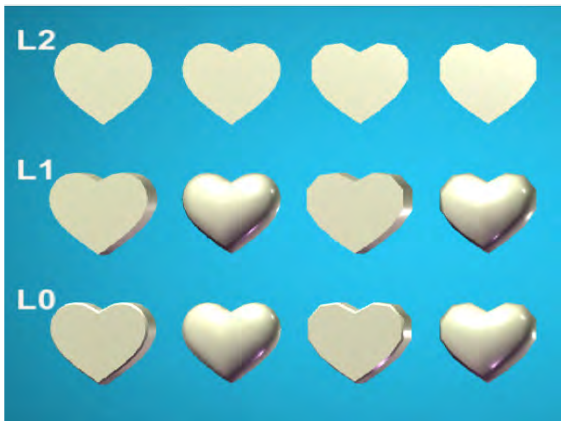
4 Screenshot samples

This section shows the components we purchased from Unity asset store and screenshots of each scene.

Skybox

Spaceship

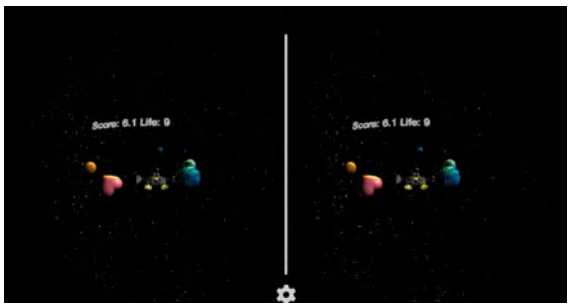
Planets



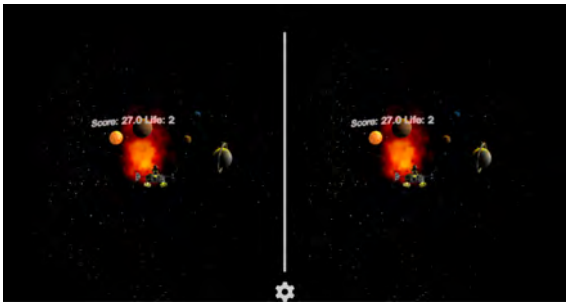
Hearts

Figure 1: Components Purchased

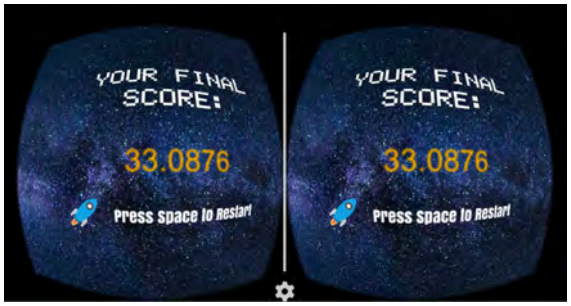
Start Scene



Main Scene



Main Scene-Explosion



End Scene

Figure 2: Sample Screenshots

5 Reference

Part of our project, i.e. scripts to read data from the VRduino and to perform stereo rendering for the HMD, is based on the starter project provided by the EE267 staff.