3D Drink Collecting Game Using Position Tracking

You Guan  
you17@stanford.edu

Qixiang Zhang  
quixiang@stanford.edu

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drink 1</td>
<td>+2 points</td>
</tr>
<tr>
<td>Drink 2</td>
<td>+4 points</td>
</tr>
<tr>
<td>Drink 3</td>
<td>+8 points</td>
</tr>
<tr>
<td>Stone</td>
<td>-4 points</td>
</tr>
<tr>
<td>Boost Drink</td>
<td>Double item scores for 20 seconds</td>
</tr>
</tbody>
</table>

Table 1. Item Types

1. Introduction

The drink collecting game is very popular worldwide. In this project, we implement its 3D version. We used VR-duino to track player’s hand position. The position data are loaded into Unity through serial pipelining. The position is then attached to a basket to collect different drinks falling down. The game image is displayed on a HMD through stereo rendering.

2. Game Details

The game is implemented using Unity and Arduino. Fig.1 shows a screenshot of the game’s GUI. It consists of following components:

2.1. Item Spawner

Items are spawned to be collected by a basket (controlled by player’s hand position). Different item types lead to different score addition/deduction. Table 1 above shows all item types and their corresponding scores/functions.

Items are spawned randomly and uniformly using a spawner script. A new item is spawned from with fixed y (vertical) position but random x and z position every 0.05 - 0.1 seconds, and destroys itself once it collides with the basket or falls below certain height.

2.2. Basket Control

The basket is a game object with position loaded from Vrduino. When the basket collides with any falling item, its OnCollisionEnter method is triggered to conduct scoring calculation. Updated game score will then be displayed on the canvas.

2.3. Music and Sound Effects

Background music is attached to an audio source and is played repeatedly on game start. When an item collides with a basket, a sound effect of scoring is played accordingly. This is implemented in the OnCollisionEnter method of basket.

2.4. Position Tracking

The position tracking part is implemented using Arduino and C++ language. Here we used the code directly from Unity starter project the course staff provided. The Unity app loads player’s hand position (x, y, z coordinates) from Vrduino board.

2.5. Challenges and Future Improvement

We also tried adding a second Arduino to implement orientation tracking with Quaternion and use it to track player’s head orientation at the same time, thereby changing...
viewing perspective accordingly. But due to hardware limitation the trackings became badly delayed when running with two Arduinos. For future work we could try some more optimizations in both software and hardware to make it work with both hand position and head orientation tracked at the same time.

2.6. Reference

Part of this project refer to the Unity starter code FruitNinja provided by course staff.