A Step Forward in Virtual Reality
Team Step

Ryan Daly
Electrical Engineer

Joseph Roberts
Electrical Engineer

Jared Ricci
Electrical Engineer

Steven So
Electrical Engineer
Motivation

▪ Current Virtual Reality has lacked the REALITY aspect

▪ The market is pushing smartphone VR; peripherals can make the experience better

▪ Users do not feel immersed in the environment

▪ VR hasn’t reached its full potential
Introduction

- Step is a new virtual reality environment that will immerse the user with no added hardware

- The user will be able to interact, move, and feel the environment
Our Solution

▪ Create a 360 degree, 3 dimensional virtual environment on the android

▪ Create a system that will detect the user’s movement like walking, running, and arm motions

▪ Create a structure in which the virtual world can be mimicked
Overall Requirements

- User is able to freely move in virtual environment and control movement speed
  - Speed Accuracy within .5 MPH

- Hand and arm motion is translated to in-game action
  - Depth Accuracy, standard deviation within 1 inch

- Control latency less than 200 ms\[^{1}\]

- User does not have to wear any sensing equipment beyond VR headset

- Maintain framerate at 60 FPS

Design

Elliptical

Raspberry Pi

Router

Fan

Android Phone

Kinect

PC

Department of Electrical and Computer Engineering
CDR Block Diagram

User Motion
- Elliptical
- Sensing Board
- Raspberry Pi

Hand, Arm, and Hip Movement
- Kinect Sensor

Wireless Network
- Raspberry Pi Server
- PC server
- Router

Fan

Smartphone/VR Headset
- Data Client
- Parse
- 3D World

PWM Signal
- Speed Data
- Kinect Data
- WiFi
- WiFi
CDR Deliverables

1. PCB design / Fan control
2. Refined Game (Reset)
3. Structure / Safety Harness
4. Turn sensing
CDR Demonstration of End-to-End Functionality

- Forwards and backwards movement
- Fan control
- PCB design
- Game refinement
- Reset button
- Safety structure and harness
- Turn sensing and virtual turning
CDR Demonstration

- Forwards and backwards movement
  - User can now move backwards
- Fan control
  - User’s speed correlates to fan speed
- PCB design
  - PCB is designed and manufactured
CDR Demonstration

- Game refinement
  - New and improved game (more fun than balloon popper!)
- Reset button
  - Button automatically resets subsystems
CDR Demonstration

- Safety structure and harness
  - Structure welded, supports weight, stability improvement

- Turn sensing and virtual turning
  - Tracked hips allow user to virtually turn
FPR Goals

- Aesthetics on the structure
  - Better wood for mounting
  - Paint the structure
  - Backdrop for structure
  - Kinect and fan mount
- Completed PCB and mount
- Tutorial menu environment
Demo
CDR Demonstration of End-to-End Functionality

- Forwards and backwards movement
- Fan control
- PCB design
- Game refinement
- Reset button
- Safety structure and harness
- Turn sensing and virtual turning
User Motion

Implementation

▪ Raspberry Pi with attached sensor board
▪ Rotational speed measured using magnetic sensor
▪ Data transmitted wirelessly to smartphone

Additions since MDR

▪ Reverse motion
▪ Fan Control based on user speed using PWM
▪ System Reset
Hand tracking - Kinect

- **Requirements**
  - Depth within 1” standard deviation for arm movement
  - Real-time processing and transmitting

- **Steps**
  - Create server
  - Extract Position
  - Wait for request
  - Write position as string to server
  - Handle reset
Kinect Joint tracking

- **Skeleton**
  - Each node is a “Joint type” object
  - Using left and right hands
    - Shown as balls in balloon popper

- **Position**
  - \((x, y, z)\) coordinates
  - Unit: meters
Kinect Depth Results

- Depth was thought to be an issue
  - Collected data and measured depth performance

<table>
<thead>
<tr>
<th>Tape Measured</th>
<th>Calculated Kinect mean (n=100)</th>
<th>Standard Deviation (m)</th>
<th>Standard Deviation (in)</th>
<th>Performance &lt;inch (=0.0254 m)</th>
<th>Performance &lt;4cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 m</td>
<td>1.0021 m</td>
<td>0.0214</td>
<td>0.844</td>
<td>80%</td>
<td>93%</td>
</tr>
<tr>
<td>2.25 m</td>
<td>2.2452 m</td>
<td>0.0218</td>
<td>0.859</td>
<td>77%</td>
<td>96%</td>
</tr>
</tbody>
</table>

- Experiment setup
  - Measured points from Kinect with tape measure
  - Compare with what Kinect returned
Depth of hip nodes recorded in c# script, analyzed with MatLab

- Graph indicates (depth of left hip) - (depth of right hip)
- Positive values indicates a left turn
Wireless Network

What it needs?

- Should allow the data from inputs to communicate with android
- Minimal latency in order to have accurate movements in the game
Wireless Network Blocks

Servers
- The raspberry pi
- PC reading Kinect data

Client
- Android

Router
- Wirelessly connects the Servers to the Client
Does this part work?

Very clearly the Kinect and the Elliptical can communicate with the android phone through the router since the phone can see

The latency

Used slow motion camera and matlab to calculate

End-to-End Latency: ~163ms
Latency: end-to-end
Safety Structure

- Should prevent the person from losing their balance off of the elliptical
- Tall and wide enough for person to freely move
Smartphone Application

Requirements

- Render a 3D virtual world
- Receive and translate data sent through network
- Framerate = 60 FPS (Limited by VSYNC)

Implementation

- Virtual 3D environment developed with Unity
  - C# scripting
- TCP client requesting data through router
  - Data parsed to be usable
- Ensure TCP servers are not capping the framerate through slow data availability
- Handle resets from Pi
1. PCB design / Fan control ✓
2. Refined Game (Reset) ✓
3. Structure / Safety Harness ✓
4. Turn sensing ✓
Questions?