UMassAmherst

Step

A Step Forward in Virtual Reality

Department of Electrical and Computer Engineering

Team Step

Ryan Daly
Electrical Engineer

Joseph Roberts
Electrical Engineer

Jared Ricci
Electrical Engineer

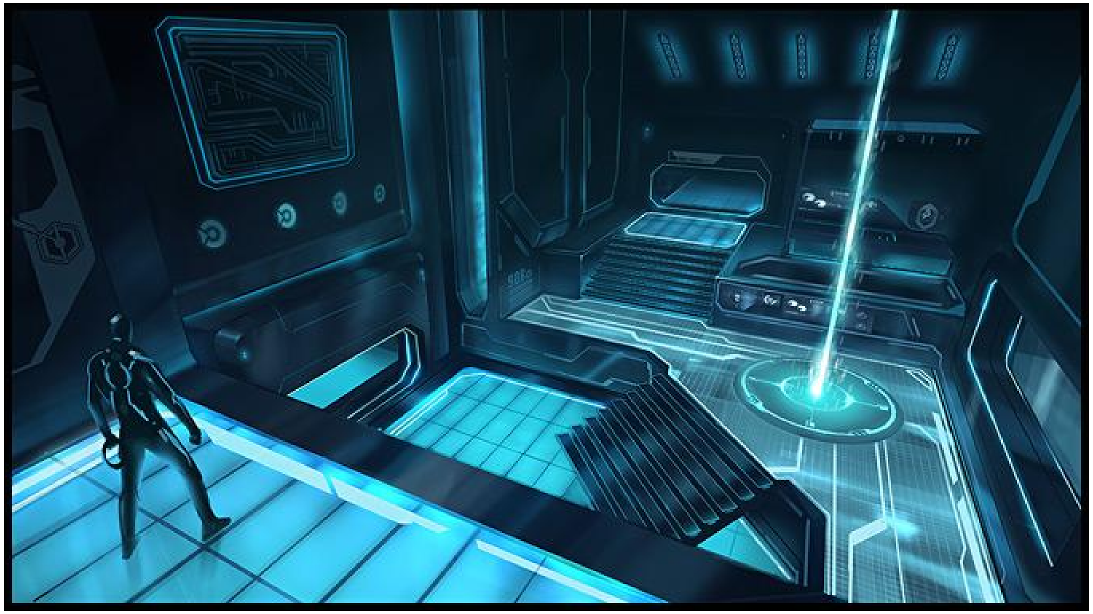
Steven So
Electrical Engineer

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

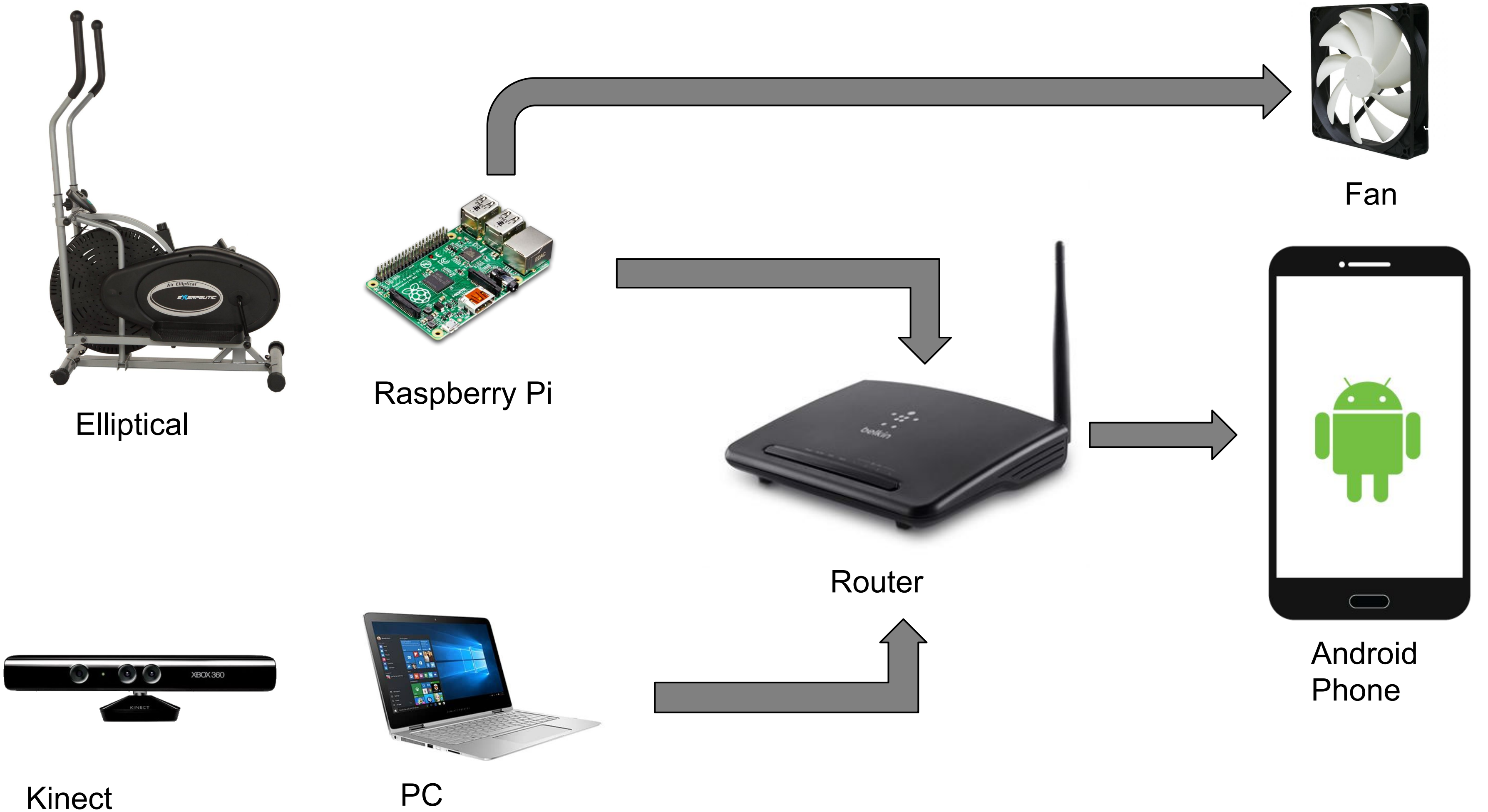
- Create a 360 degree, 3 dimensional virtual environment on the android
- Create a system that will detect the user's movement like walking, running, and arm motions
- Create a structure in which the virtual world can be mimicked

Overall Requirements

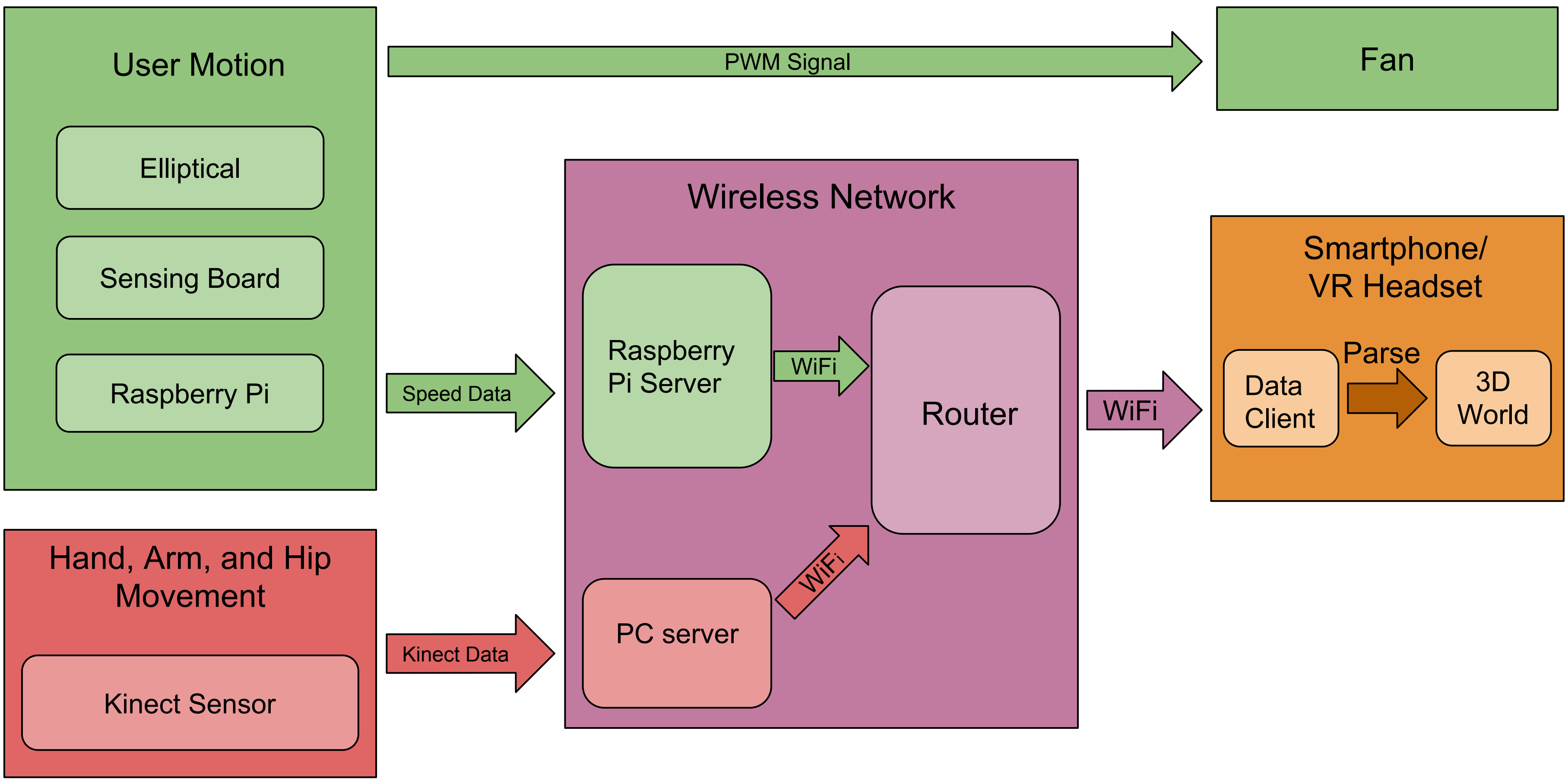
- User is able to freely move in virtual environment and control movement speed
  - Speed Accuracy within .5 MPH
- Hand and arm motion is translated to in-game action
  - Depth Accuracy, standard deviation within 1 inch
- Control latency less than 200 ms[1]
- User does not have to wear any sensing equipment beyond VR headset
- Maintain framerate at 60 FPS

[1] Leadbetter, Richard. "Console Gaming: The Lag Factor." Eurogamer.net. Eurogamer, 09 May 2009. Web. 01 Dec. 2016. http://www.eurogamer.net/articles/digitalfoundry-lag-factor-article.

Design


Fan
Raspberry Pi
Elliptical
Router
Android Phone
Kinect
PC

CDR Block Diagram



CDR Deliverables

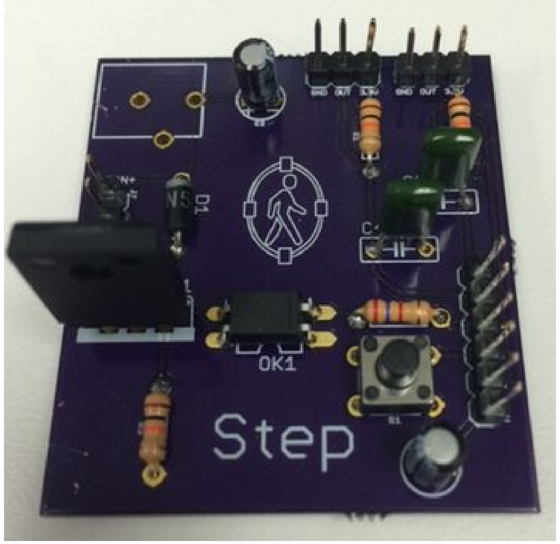
1. PCB design / Fan control
2. Refined Game (Reset)
3. Structure / Safety Harness
4. Turn sensing

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

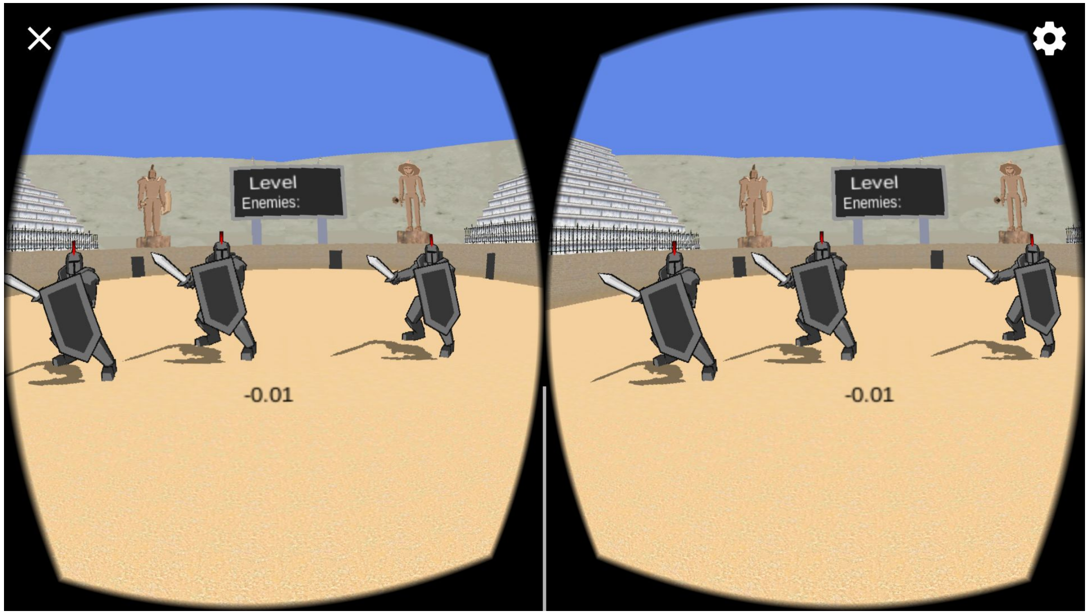
CDR Demonstration

- 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



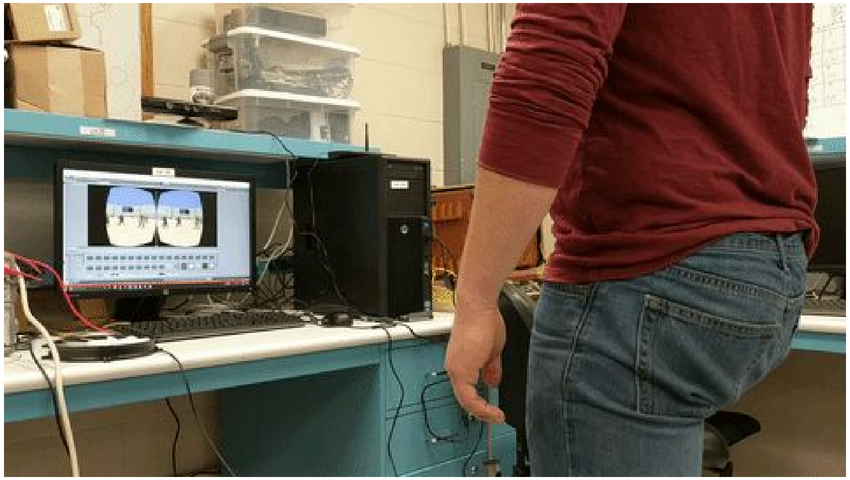
CDR Demonstration

- 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

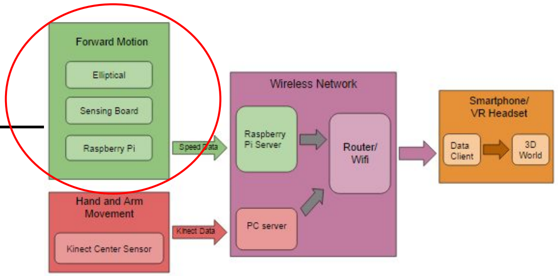
- Aesthetics on the structure
  - Better wood for mounting
  - Paint the structure
  - Backdrop for structure
  - Kinect and fan mount
- Completed PCB and mount
- Tutorial menu environment

Demo

CDR Demonstration of End-to-End Functionality

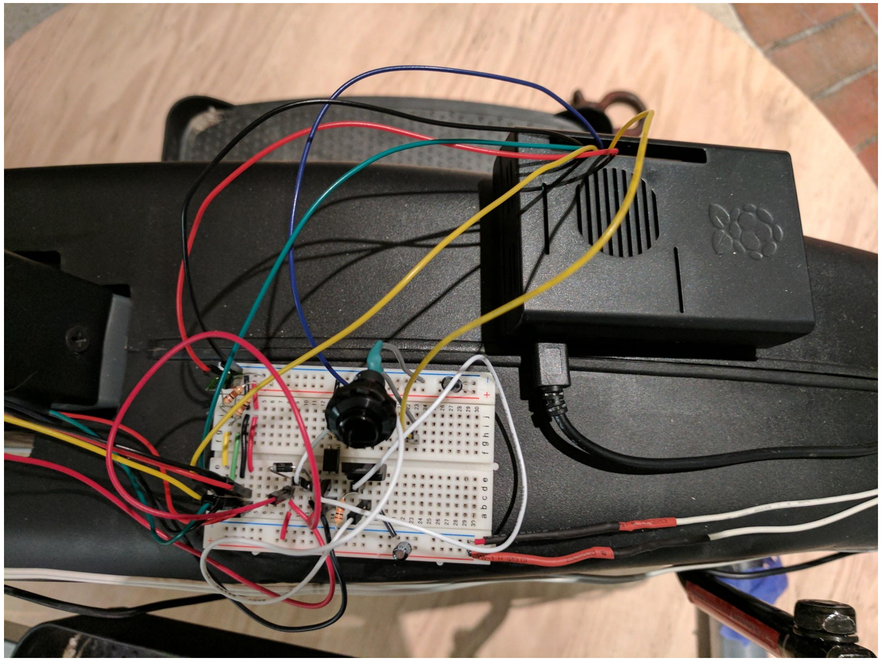
- Forwards and backwards movement
- Fan control
- PCB design
- Game refinement
- Reset button
- Safety structure and harness
- Turn sensing and virtual turning

User Motion



Implementation

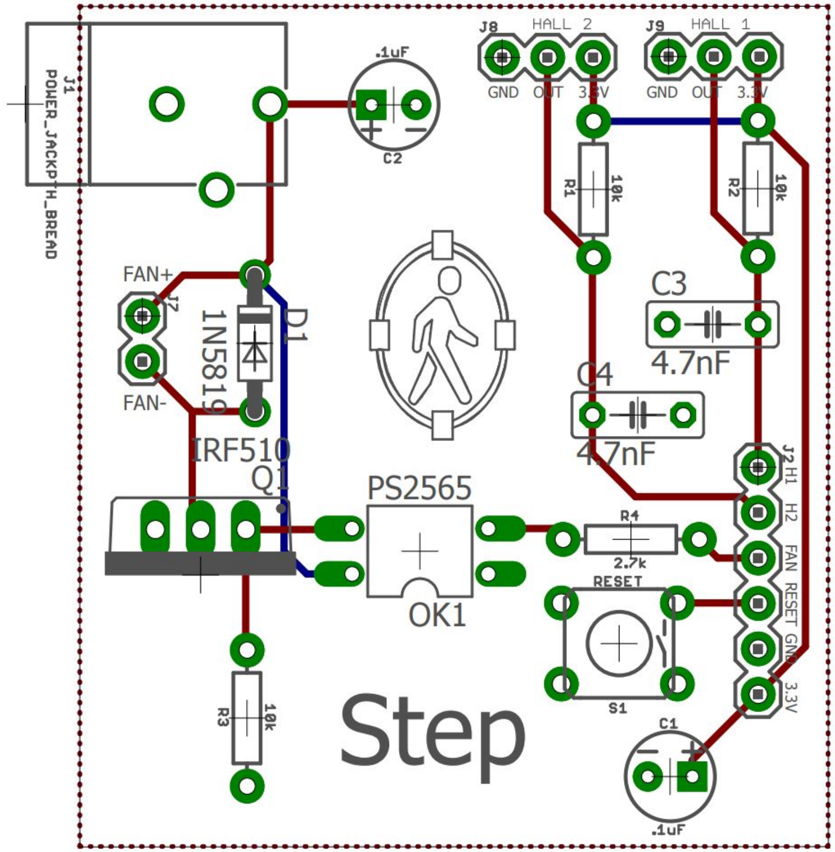
- Raspberry Pi with attached sensor board
- Rotational speed measured using magnetic sensor
- Data transmitted wirelessly to smartphone



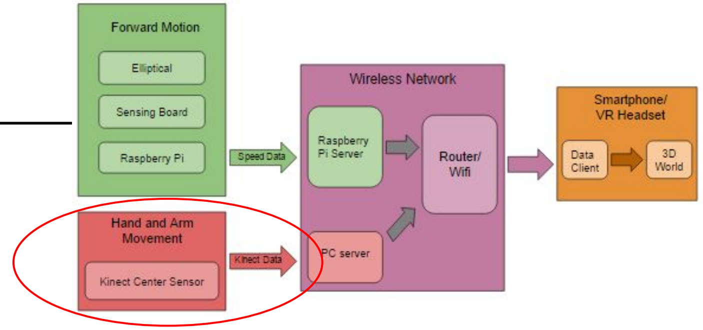
Additions since MDR

- Reverse motion
- Fan Control based on user speed using PWM
- System Reset

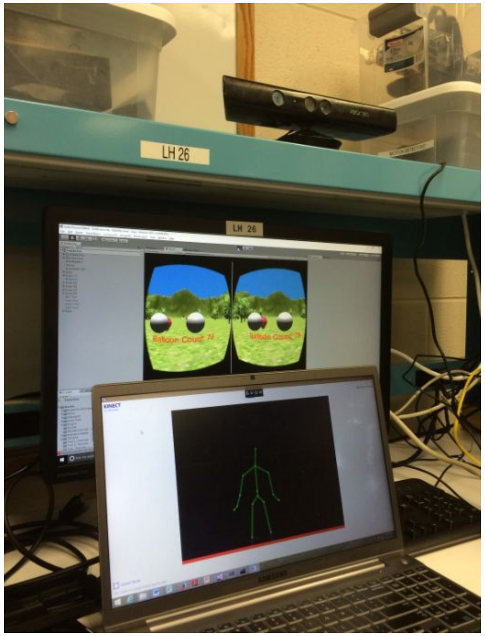
PCB Design


J1
POWER_JACKPTH_BREAD
.1uF
C2
J8
HALL 2
GND
OUT
3.3V
J9
HALL 1
GND
OUT
3.3V
R1
10k
R2
10k
C3
4.7nF
C4
4.7nF
FAN+
FAN-
D1
1N5819
IRF510
Q1
PS2565
OK1
R4
2.7k
RESET
S1
J2
H1
H2
FAN
RESET
GND
3.3V
R3
10k
Step
C1
.1uF

Hand tracking - Kinect

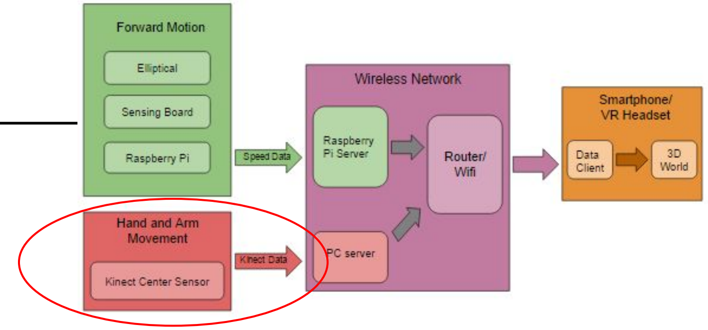


- **Requirements**
  - Depth within 1” standard deviation for arm movement
  - Real-time processing and transmitting

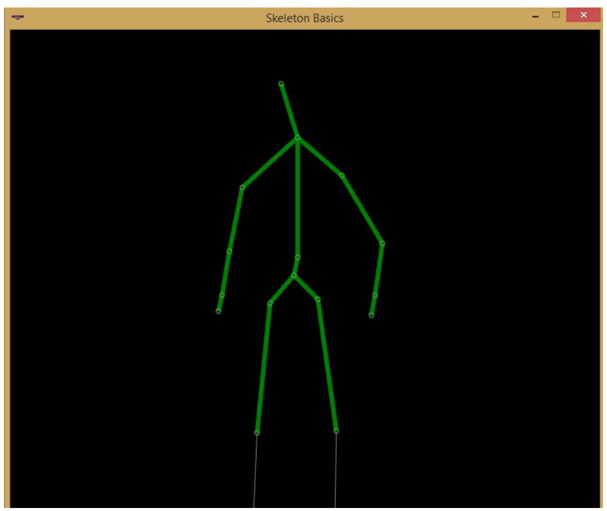


- **Steps**
  - Create server
  - Extract Position
  - Wait for request
  - Write position as string to server
  - Handle reset

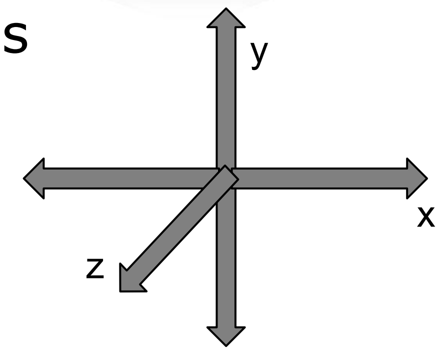
Kinect Joint tracking



- 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

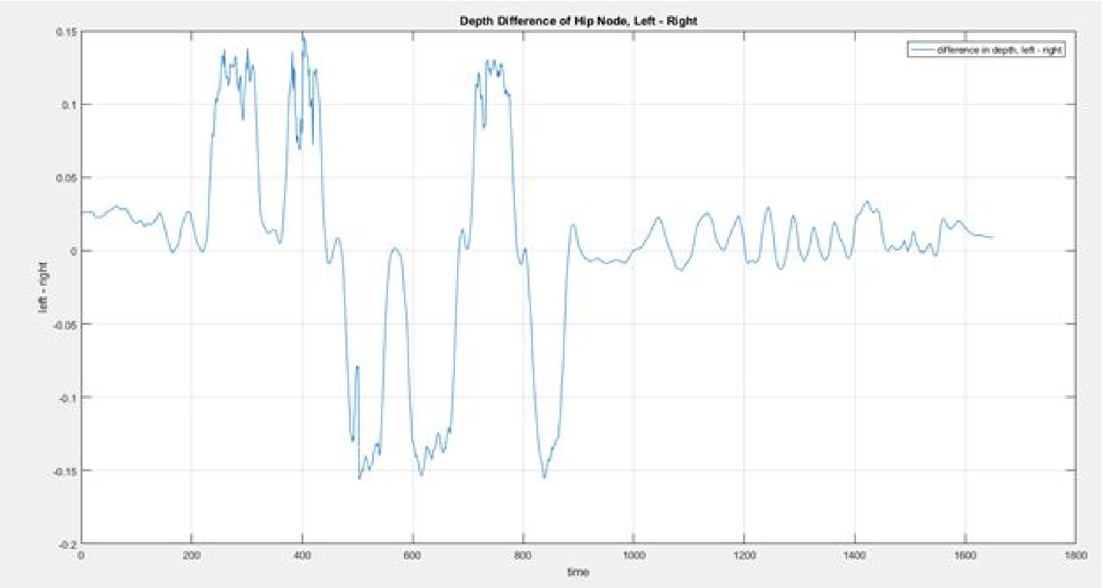
- Depth was thought to be an issue
  - Collected data and measured depth performance

Tape Measured	Calculated Kinect mean (n=100)	Standard Deviation (m)	Standard Deviation (in)	Performance <inch (=0.0254 m)	Performance <4cm
1.0 m	1.0021 m	0.0214	0.844	80%	93%
2.25 m	2.2452 m	0.0218	0.859	77%	96%

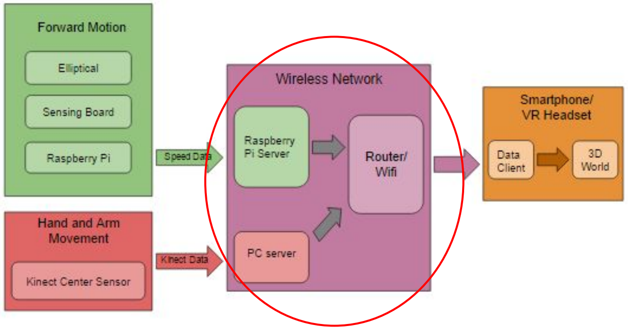
- Experiment setup
  - Measured points from Kinect with tape measure
  - Compare with what Kinect returned

Turn Sensing

- 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

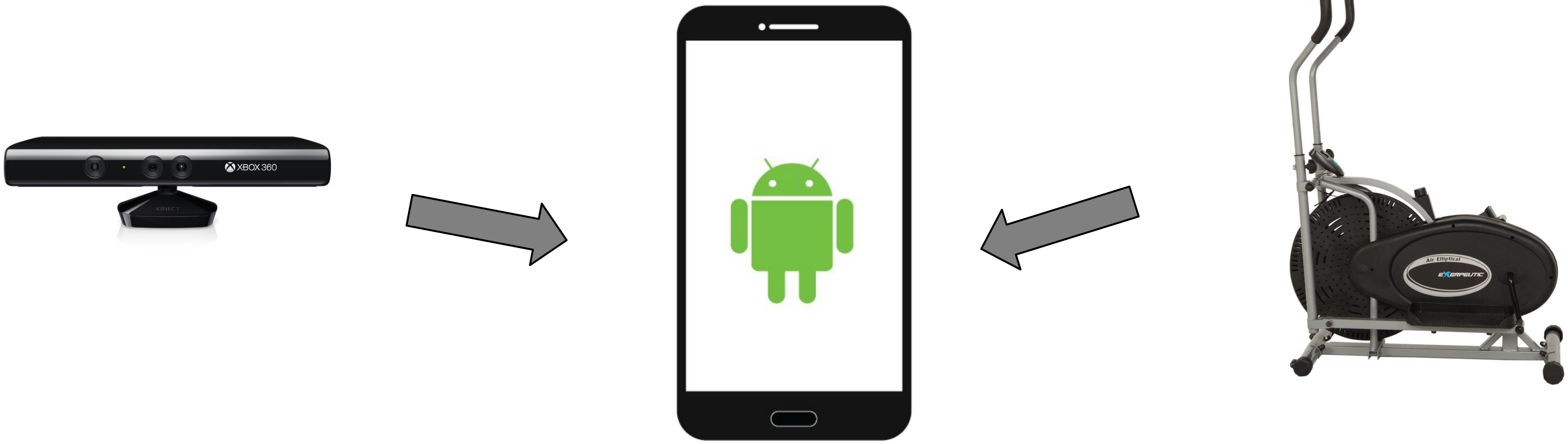


Wireless Network



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?

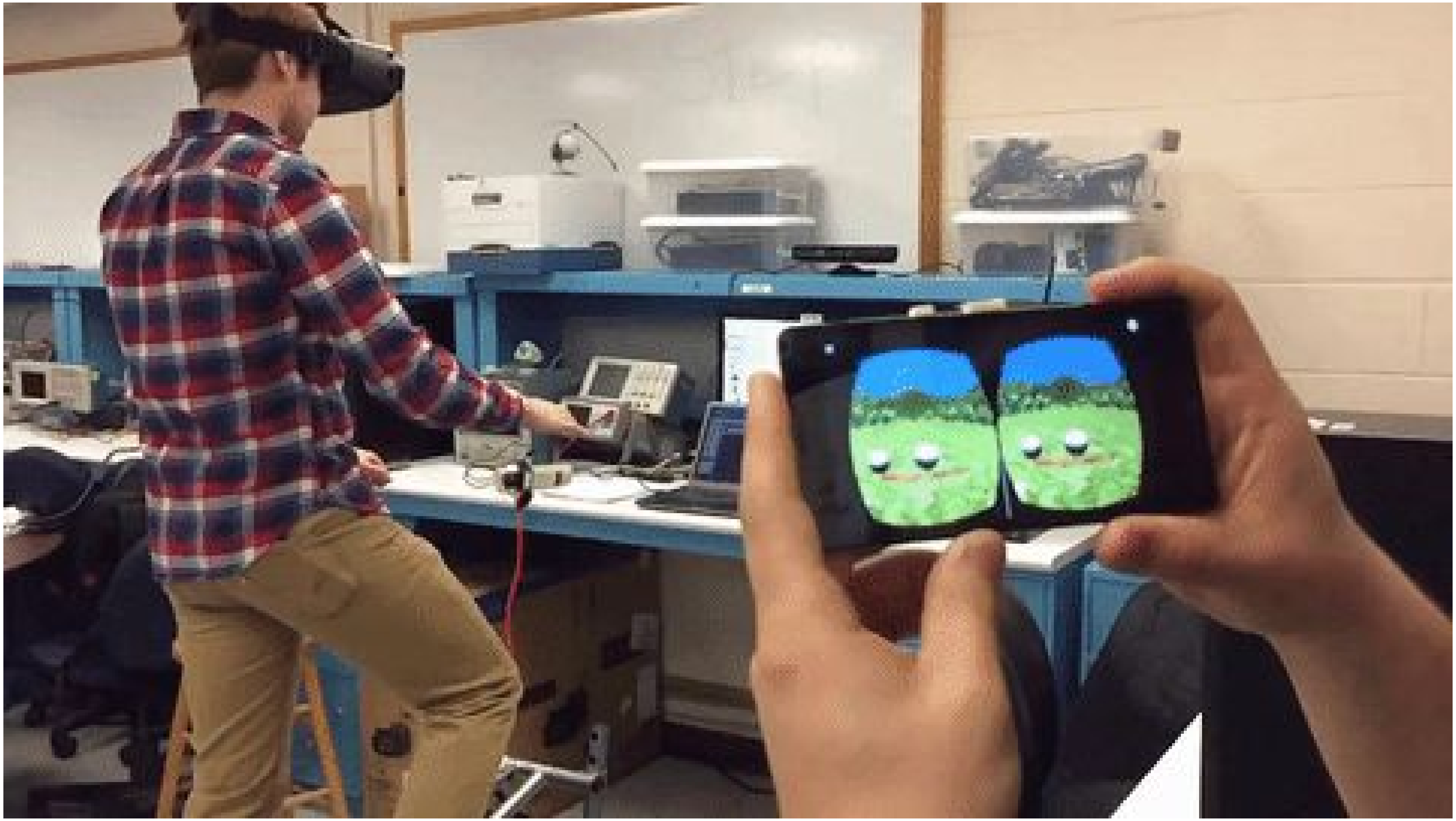
Very clearly the Kinect and the Elliptical can communicate with the android phone through the router since the phone can see

The latency

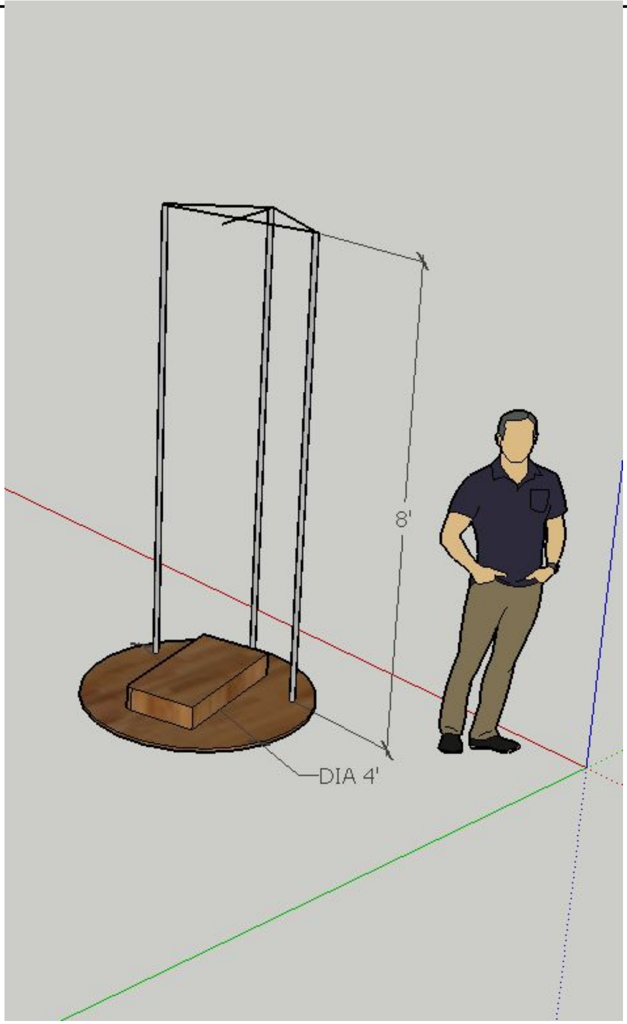
Used slow motion camera and matlab to calculate

End-to-End Latency: ~163ms

Latency : end-to-end

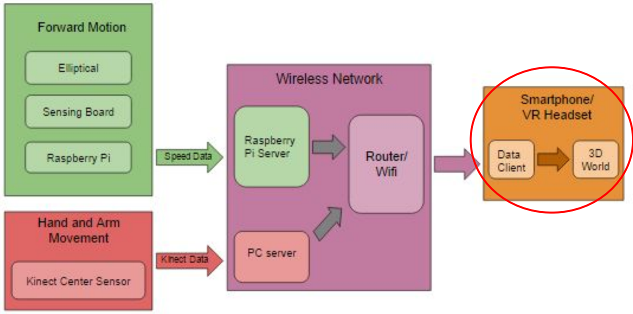


Safety Structure



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

Smartphone Application

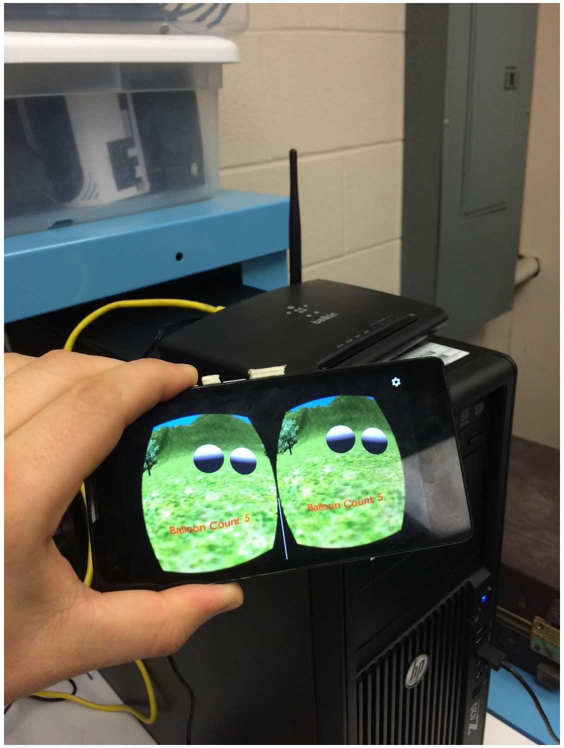


Requirements

- 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



Unity


Unity Personal (64bit) - FirstScene.unity - Balloon VR - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component GoogleVR Mobile Input Tools Window Help
Hierarchy
FirstScene
GvrViewerMain
Main Camera
EventSystem
GvrFPSCanvas
Terrain
Directional light
Cube
Cube (1)
Cube (2)
Cube (3)
Cube (4)
Cube (5)
Win Text
Lose Text
Reset Text
Start Text
Body
Inspector
TCPclient Import Settings
Imported Object
TCPclient
using UnityEngine;
using System.Collections;
using System;
using System.IO;
using System.Net.Sockets;
using UnityEngine.UI;
public class TCPclient : MonoBehaviour {
//variables for client 1
private Text displayText;
private TcpClient client;
private NetworkStream stream;
private Byte[] data = new Byte[2048];
private String responseData = "1";
private Int32 bytes;
//right hand variables
private String rightHandResponseData = "1";
private Int32 bytesR;
private Byte[] dataR = new Byte[2048];
//variables for writing
private String please = "please send data";
private String please2 = "1";
private Byte[] send = new Byte[2048];
private Byte[] send2 = new Byte[2048];
//variables for client 2
private Text displayText2;
private TcpClient client2;
private NetworkStream stream2;
private Byte[] data2 = new Byte[2048];
private String responseData2 = "0";
private Int32 bytes2;
private Text displayText3;
private Text displayText4;
private int frameCount;
//left hand
private String[] handCoords = new String[3];
public static double xHandCoord;
public static double yHandCoord;
public static double zHandCoord;
private int i;
//right hand
private String[] righthandCoords = new String[3];
public static double xHandCoordR;
public static double yHandCoordR;
public static double zHandCoordR;
//elliptical velocity
Asset Labels
Project
Console
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Editor
GoogleVR
Party Pack
Plugins
Scripts
SOUND ATELIER
Standard Assets
Terrains
Tree 1_Textures
Tree 2_Textures
Tree_Textures
Assets ▸ Scripts
BalloonCou...
Begin
ButterflyAn...
Destroy
LeftHandM...
LoseObject
Move
MoveCame...
MovePlaye...
Reset
RightHand...
TCPclient
WinObject
TCPclient.cs

CDR Deliverables

1. PCB design / Fan control
2. Refined Game (Reset)
3. Structure / Safety Harness
4. Turn sensing

Questions?