**Touch to Win**

Game Design Document

**Goal**

Build a JavaScript game around the LeapMotion sensor interacting with WebGL (specifically using Babylon.js this time). This final project will demonstrate what I have learned in this Advanced Game class and in my CS courses. The specific idea behind this game is to give me experience using the LeapMotion to control JavaScript (specifically WebGL content), and more generally to explore how the LeapMotion sensor could more readily be integrated into future student’s game projects.

**Overview**

Use a simple 3D ball generated using Babylon.js to represent my palm position in the x, y, z dimensions. The ball will match my hand movement. The purpose is to build a demo that will show that the LeapMotion can be used as an effective controller for future students building games.

**Game Description:**

Each game starts with a set of canvas window’s (each a unique color). Each level, starts with the colored canvases and flashes different canvases in a certain order. The user needs to use their hand to “touch” the canvases in the correct order and under a certain time limit.

**Game Assets:**

**Hand Ball:** Simple ball model from Babylon.js (<https://doc.babylonjs.com/)> and gets LeapMotion data in JSON format served through a Web socket connection (see documentation for this: <https://developer-archive.leapmotion.com/documentation/v2/javascript/supplements/Leap_JSON.html#connecting-to-the-websocket-server>)

**LeapMotion Controller:** (Find hardware on: https://www.leapmotion.com/ ). This isn’t quite a game asset, but is necessary for the game. You must plug in a generation 1 LeapMotion and turn on the App Home application that goes with it. Then, you can run the game.

**UI:**

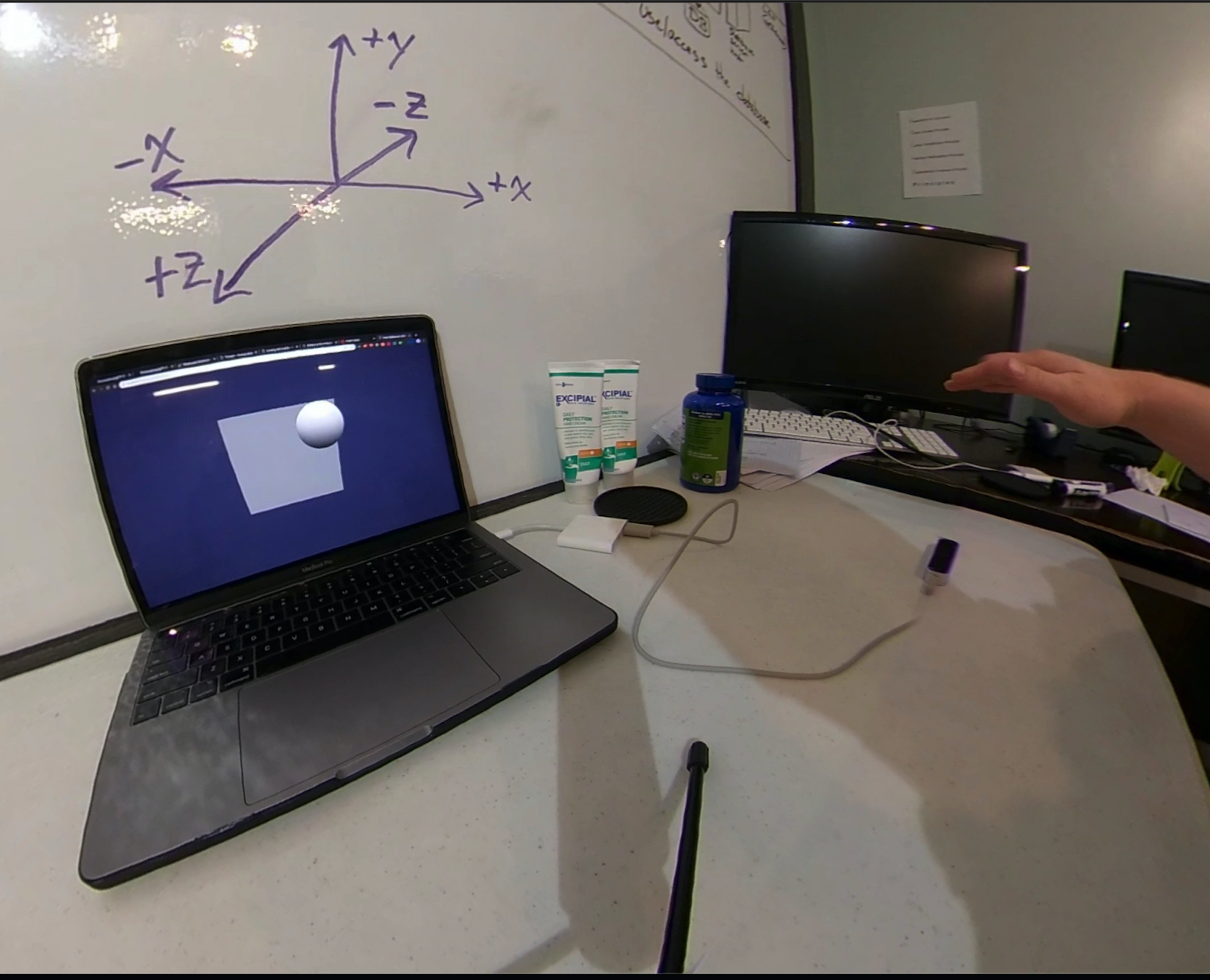
**Game Background:** Demo background from Babylon.js tutorial

**Game Play:** Move your hand over the sensor in the direction you want the ball to move and the ball should map itself to that path.

**Game Object Mechanics:**

**Hand Ball:** A ball that follows the position of your palm. It should move in the negative x direction when you move your hand over the sensor to the left. It should move to the positive z dimension when you move your hand over the sensor to the right. It should move in the positive y-direction when you raise your hand up. It should move the negative y direction when you move your hand down towards the sensor. It should move in the negative z direction when you move your hand away from the sensor towards the top of the sensor (away from yourself). It should move in the positive z direction when you move your hand away from the sensor towards the bottom of the sensor (towards yourself).

**Simple Plane:** Just a static plane with no collisions (It helps show that you are actually moving the ball in the z-dimension).



**States and Transitions:**

First you will need a first generation LeapMotion and need to launch LeapMotion home app on your computer. To launch the game, go to the directory level that the *handBall.html* file is at in command line. Generate a server at port 8000 by running ‘python3 –m http.server’. Then in your browser go to ‘localhost:8000/handBall.html’ and run the game.

**Milestone plan**

**Get data from Leap Motion with web socket connection**

**Control WebGL model that tracks the motion of my hand**

**Build collisions** (wasn’t able to get to this one)

**Tuning:**

I would liked to incorporated collisions. I would like to write a real game or even a plug and play game interface for the LeapMotion to interact with already built JavaScript games.