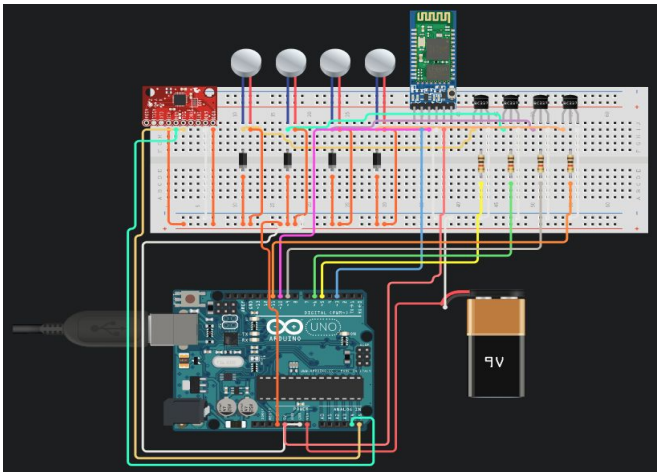
Comp140 Assignment 2 Report – Project Proposal

Thomas O'Leary - to231922@falmouth.ac.uk – 1903716

The main design for the controller will be a simple box. When the box is tilted (forwards, left, right or backwards), it will tilt the maze within my game (depending which direction it is tilted). The aim for the player is to tilt the maze to allow a ball to get from its starting position to the finish. If the ball collides with a wall within the maze, a short haptic vibration will occur within the controller - notifying the player that the ball has collided. This haptic feedback will occur where the ball collided (front side, left, right and back).

Components Required:
Arduino Uno (68.6mm length × 53.4mm width)
MPU-6050 Accelerometer (21.2mm length × 16.4mm width × 3.3mm height)
Vibrating Disk Motor × 4 (3.4mm length × 10mm diameter)
HC-05 Bluetooth Module (26.9mm length × 13mm width × 2.2mm height)
9V Battery Connector
9V Battery (27mm length × 17mm width × 48mm height)
Solderless Breadboard (170mm length × 65mm width × 10mm height)
TIP120 Transistor × 4
2.2k Ohm Resistors × 4
1N4001 Diode × 4

- As a user, I can control the pitch and roll of the maze by changing the pitch and roll on the controller.
- As a user, I want to have some sort of feedback when the ball in the maze has collided with an object.
- As a user, I want the controller to be Bluetooth so that it can be wireless.



Hardware of the Controller

The main hardware component required for this controller to work is the Arduino Uno itself. Without this, it would make it very hard for myself to make sure that all other components integrate with each other and overall much harder to interact with Unity. The Arduino Uno, supplied by the Games Academy, allows me to control all my other components, also allowing me to utilise them to what I deem necessary for the project.

The MPU6050 component is a simple 3-Axis Gyroscope, Accelerometer and Temperature module that will allow me to calculate Pitch & Roll for real life movement in my game. This small component is a very important module for my project as it will control the movement for my maze in the Unity Project.

The HC-05 Bluetooth module is a small Bluetooth module that is able to transmit and receive Serial data. Because of this, I will be able to use the module to communicate over serial communication without a USB cable – allowing it to be wireless and handheld.

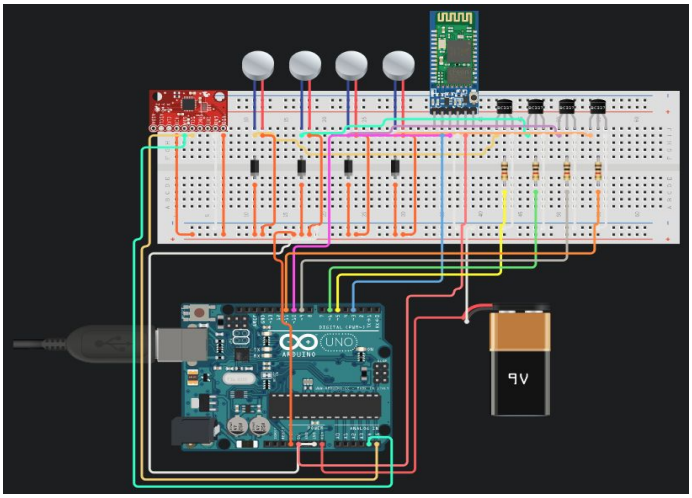
To control the haptic feedback for the controller, I am using 4 separate 3.3v Vibrating Motors. The starting current of these motors are ≈85mA, which exceeds what the Arduino Uno can supply. To resolve this issue, I will use TIP120 transistors to allow me to supply an external power (9V Battery) to my motors.

Other hardware in my controller consists of:

- Breadboard
    - To allow me to connect all the components together with jumper wires
- 2.2k Ohm Resistors
    - This will control the flow that goes through the Arduino Uno from the TIP120 transistor
- 1N001 Diodes
    - This will make the ground (negative) of the motor’s electrical current flow in one direction

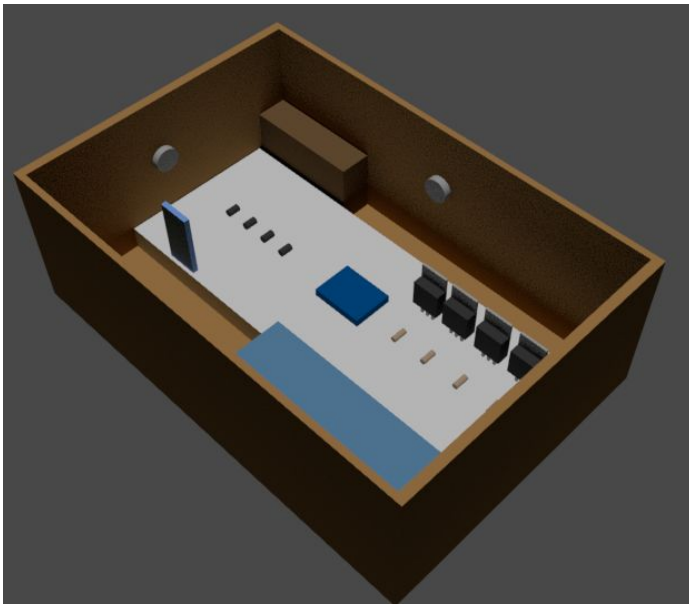
Design of the Controller

The controller itself has a very simple design as the overall casing is just a simple enclosed box that all the components are placed into.
One possible issue with the design itself will be the jumper wires as there would be so many required for each component and it will be easy to be confused on which jumper wire is for what.



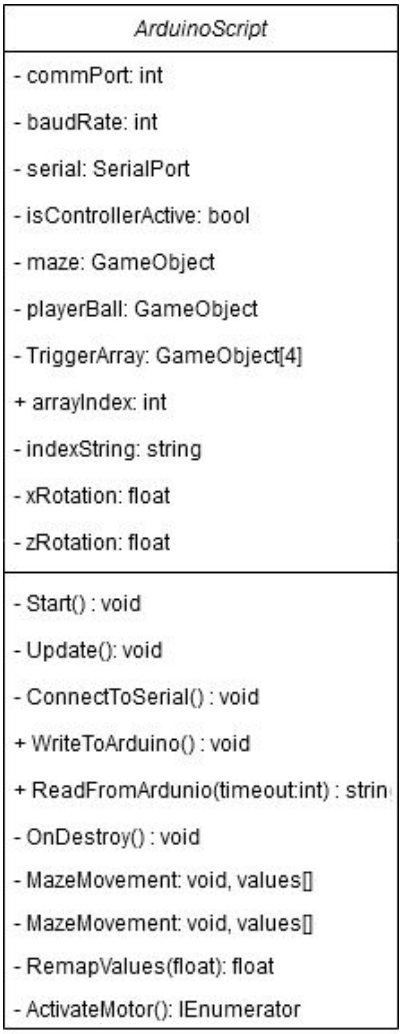
The wiring will not be overly complicated in the controller. The most confusing part that I will encounter with is having to use the TIP120 transistors as I have never experienced in having to use these with my limited experience overall in electronics.

In the casing of the controller itself, the 4 separate 3.3v motors will be stuck to the sides of the box (this can be done with the motor itself as it has an adhesive back). This will allow the player to know directly on the controller where the haptic feedback is coming from.



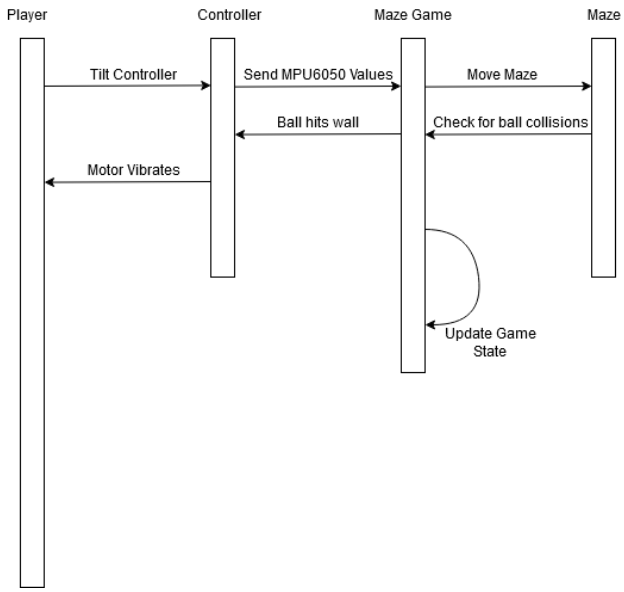
Software Architecture

The original idea for the software architecture was to have 4 main scripts to control the game. One that contains all related functions in connecting, reading and writing to the Arduino (Arduino.cs), a script that controlled the movement of the maze by utilising the reading function from Arduino.cs (this script was called MoveMaze.cs) and the last two were related to controlling the ball. These 2 scripts worked together by detecting collisions on the ball and writing specific values to the Arduino – these scripts would have controlled the motors for the game (BallScript.cs & BallTrigger.cs).



After managing to write all the scripts, many errors then occurred – the main being that the COM port was busy and could not be used. This was due to both MoveMaze.cs and BallScript.cs both trying to communicate with the Arduino at the exact same time. After many attempts at fixing the error (which caused countless more such as Null References), I decided to merge the functionality of Arduino.cs, MoveMaze.cs and BallScript.cs all into one.
I wrote this script with the anticipation of failure, mainly due to the fact that my scripts could work separately but not together. After a couple of minutes of merging all the scripts together I had created ArduinoScript.cs.

When running the game for the first time just by using ArduinoScript.cs, everything worked the way I wanted to. This script allowed me to read and write to the Arduino finally allowing me to play test the mechanics all together.



Reflection

Link to poster

When preparing for my project proposal, I did not have a single creative idea in my head that I thought would go great with Comp140 Assignment 1. After many hours of trying to come up with ideas, I stumbled across the MPU-6050 Accelerometer on YouTube. After seeing what this module was able to do, I instantly came up with the idea of Maze Roller. A simple maze game where the player has to move a maze to roll a ball through it! One main inspiration for this project is a small party bag toy that consisted of a small maze in which kids have to roll a ball to its centre!

When writing the project proposal I also included that I would use the HC-05 Bluetooth module – this, in theory, would allow me to use Serial Communication over Bluetooth and make the controller wireless (rather than having to have a Type B USB cable coming out of one the sides). My idea for using the HC-05 was that it would be a gateway for me to communicate from the Arduino straight to the computer (without having to need a USB cable for Serial communication), so this would mean that the HC-05 would also need to be used for printing within the COM port (so that Unity would be able to read the MPU-6050 values). But many hours later after struggling to get it to work, wondering why it's printing random ASCII values – I discovered that the HC-05 could not be used in the way I intended it to. So sadly, I decided to remove the HC-05 from the controller.

As I had only recently been introduced to both the Arduino and electronics overall (in January), I struggled a lot in trying to implement all the electronics together. One main aspect I struggled with was trying to power my 3.3v Vibrating Motors – this is because just one of these motors drew so much current that it could blow the Arduino. Andy Smith (GA technician) showed me a wiring diagram that used a TIP120 transistor with an Arduino – this specific diagram explained to me how I would be able to use an external power source to power all of my motors. But following this diagram deemed impossible for me as well. Many hours I tried to recreate this diagram but it would just not work. I finally resolved the issue by discovering that the breadboard that was supplied by the GA (along with our Arduino) had a split down the middle that stopped all current going from one end to the other.

When I arrived back home in London, after leaving Falmouth due to Covid-19 at the time, I came up with the idea of creating a 3D printed case for my controller. The idea of this was much better than my original case for the controller as it was only a spare

Amazon box I had lying around in my room. After talking with a family member of mine (who has a 3D printer) my idea for a printed case was well on its way. Maybe an hour or so of research later, I discovered that I was able to use the 3D modelling software that I had learnt for 2 years prior at College. After 2 hours of drawing mock-up designs and then modelling the pieces themselves – I had an accurate 3D print ready model for my case. Sadly however, the 3D print was not finished before the Assignment 1 Deadline for the 1st of April – but will be ready for the Viva on the 1st of May.

Overall I believe that this assignment has gone well. If I was able to make the scripts work separately in Unity and the HC-05 to work how I intended it to, I believe the project would have gone very well for me. If I were to ever do it again, I would spend much more time into research regarding physical components (what their main pros and cons are and what their main functionality is) and also spend more time into looking a Serial Communication regarding C#. This is because by researching more into how C# handles communication with COM Ports, I believe I would've been able to make my software architecture within Unity much better.