Glovesy

BY

Alan Devine - 17412402

Sean Moloney - 17477122

A functional specification document

As a requirement for CA400

Dublin City University (DCU)

Plagiarism Declaration

I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work. I/We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I/We engage in plagiarism, collusion, or copying. I/We have read and understood the Assignment Regulations. I/We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment refernces. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.

I/We have read and understood the referencing guidelines found at
`https://www.dcu.ie/info/regulations/plagiarism.shtml`,
`https://www4.dcu.ie/students/az/plagiarism`,
and/or recommended in the assignment guidelines.

Project Title	Glovesy
By	Alan Devine - 17412402
	Sean Moloney - 17477122
Field of Study	Computer science
Project Advisor	David Sinclair
Academic Years	2020/2021

ABSTRACT

Glovesy is a wearable computer interfacing device in the form of glove which will allow the user to interface with their computer by using custom macros, or use the device for hand-tracking in VR or AR applications.

Keywords: : Wearables, human-computer interfacing, VR, AR, Arduino

TABLE OF CONTENTS

Introduction

Glovesy is a wearable device which will allow the user to inteface with their computer, either by using user-defined macros, which will be set up using our program which will allow a number of gestures do be defined to certain actions within the pc, or by allowing the user accurate hand and finger tracking for use in Virtual and Augmented Reality.

General Description

1 Product / System Functions

Glovesy is a wearable human/computer interacing device, which will allow users to interact with their pc in a number of different ways, for different scopes. The primary focus of the device, will be to allow the user to set up macros, or certain movements or gestures, which the computer will recognise as a specific command, thereby allowing ease of use. Another function of the device will be to track user hand and finger movements for increased accuracy and control in VR applications, since the device is so low profile, as opposed to current VR controllers which tend to be bulky, handheld devices.

2 User Characteristics and Objectives

Glovesy is designed for users who engage in VR/AR applications. This includes, but is not limited to, Gaming, Product prototyping across remote teams, Creating virtual art, etc. In addition to VR/AR users, given that Glovesy is designed to be a standard input device, we also envision it being used in environments such as, a presenter navigating a stage, users who for whatever reason may not feel comfortable/can use traditional input devices, etc. We hope to make Glovesy as accessible as possible to the vast majority of users, that being said, here is our idea of the "Ideal User".

- Experience with gesture-based input devices
  - e.g. Nintendo Wii, or an Air mouse.

- Experience with defining Macros
  - Keyboard and Mouse Combinations that are present with devices with extra additional keys/ buttons.

3 Operational Scenarios

1. **Gaming in a VR setting:**
   - The User would see a virtual representation of their hand (this is entirely dependent on the Game itself). The user would be able to interact with their surroundings using Gloves rather than a traditional controller.
2. **Using an AR Device:**
   - The User would be able to control their AR Device using hand gestures.
3. **Mouse Emulation:**
   - Within a normal desktop, a User would be able to control their mouse cursor by moving their hand in a way similar to the WiiMote. The user would be able to left click by tensing their index finger and right click using their middle finger.
4. **Gesture Control:**
   - The User would define some sequence of hand movement combined with finger position, and create a mapping to certain actions on their computer, e.g. raising and lowering system volume, advancing slides, etc.

4 Constraints

There are a number of constraints that we foresee will have some impact on the development process of this project.

- **BlueTooth:** We can imagine that there may be some problems with connectivity over bluetooth.
- **Distinguishing Gestures:** It may be challenging to distinguish hand movements between general movement and purposeful gestures.
- **Application Support:** It could be difficult to set up programs to use the device, as, particularly in games, there may be different controls that are pre-defined.
- **Cross-platform:** a different driver will need to be made for different operating systems, which could be challenging.

Functional Requirements

- **Hand Tracking:** The device must be able to track the overall hand's orientation and position in 3D space.
  - **Criticality:** This aspect is quite important to the overall accuracy of the device, especially for tracking the hand for VR and for Gesture Recognition.
  - **Technical Issues:** A significant issue with this is that the accuracy is almost entirely dependent on the hardware, meaning if the IMU device we use is innacruate, we wont be able to do anything to improve the tracking.
- **Finger Tracking:** The device must be able to track the fingers' individual joints accurately.
  - **Criticality:** This aspect is also quite important, as it is one of the main functions of the device.
  - **Technical Issues:** One of the main issues with this is that the flex sensors that we are using are made by us, so there will need to be some calibration when first using the device.
- **Data Transfer:** The device must be able to transfer data to the pc reliably.
  - **Criticality:** This Aspect is very important, since, even if the rest of the data is accurately measured, if the data isn't transferred reliably, the device is useless.

  - **Technical Issues:** One of the issues with this will be getting the bluetooth module on the board we are using to be reliable, otherwise we can use the device with

- **Gesture Recognition:** The system must be able to distinguish between random movements and intentional gestures.

  - **Criticality:** This requirement is important for a specific use-case scenario which is not focused on gaming or vr, but rather on workflow.
  - **Technical Issues:** Due to the nature of the device, we only have a certain number of inputs, and thereby limited on the amount of information we get from the device, meaning that we will have to focus on some way to distinguish between gestures and non-gestures through pattern recogntition.

System Architecture

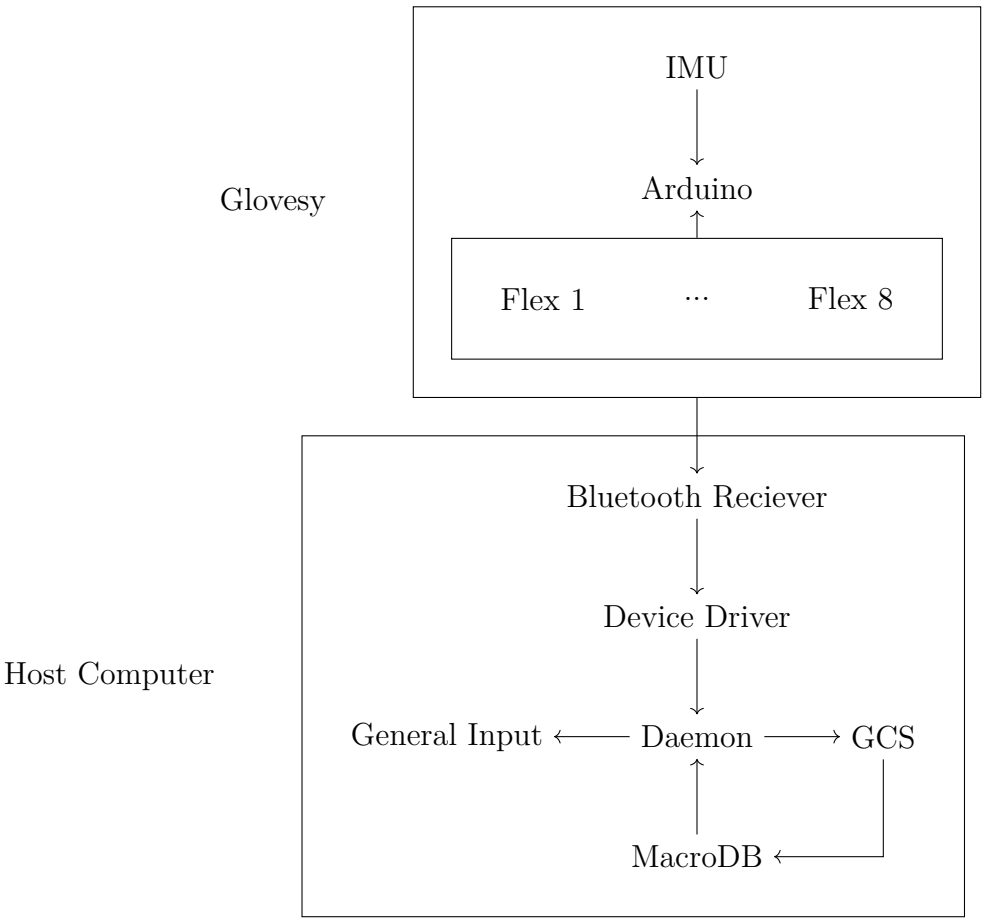


Figure 1: High-Level System Architecture Diagram

5 Arduino

The arduino in use is an *Adafruit Feather M0 Bluefruit*. This has an integreated Bluetooth controler which allows us to transfer data to and from the host com-

puter. This Arduino has 10 analog pins which we can use to measure resistance on our flex sensors.

6 Flex sensor Array

A flex sensor is a vairable resistor that changes resistence based on how much the sensor is bent. For this project we will be making use of 8 flex sensors which will measure the position of each knuckle as well as each finger joint. With the exception of the Thumb and pinky fingers, which will only have one flex sensor each. To help mitigate the costs associated with this project, we will be constructing our own flex sensors using several layers of velostat and conductive thread which will act as our positive and negative terminals.

7 IMU

The IMU, or *Inertial measurement unit* is the device which we will use to measure the device in 3D space. Our device in question, measures Acceleration and Gyroscopic data on the X, Y and Z Axises in meters per second and radians respectivly. Our IMU also measures temperature, however that will not be used for this project.

8 Macro Database

User defined macros will be stored in a database implemented in MongoDB. We chose MongoDB as conceptually, it makes more sense to group these actions as an object rather than in a relational database.

High-Level Design

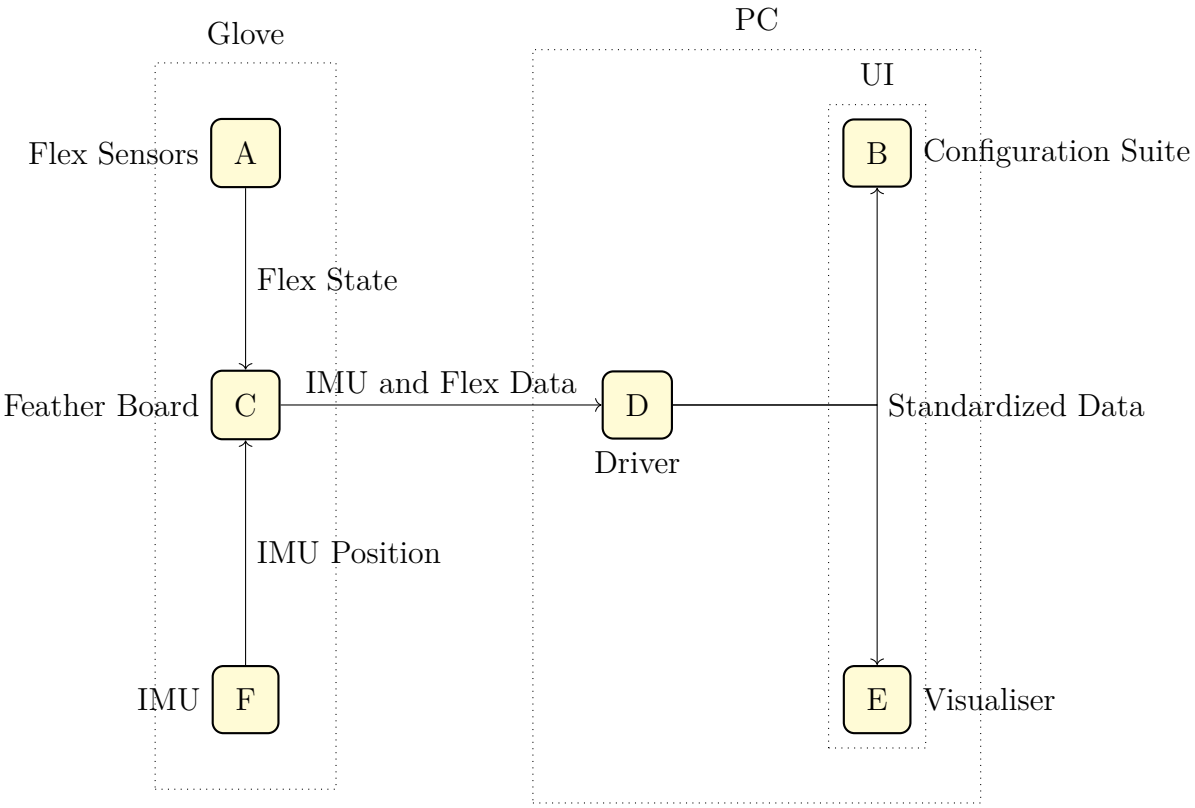


Figure 2: Data Flow Diagram

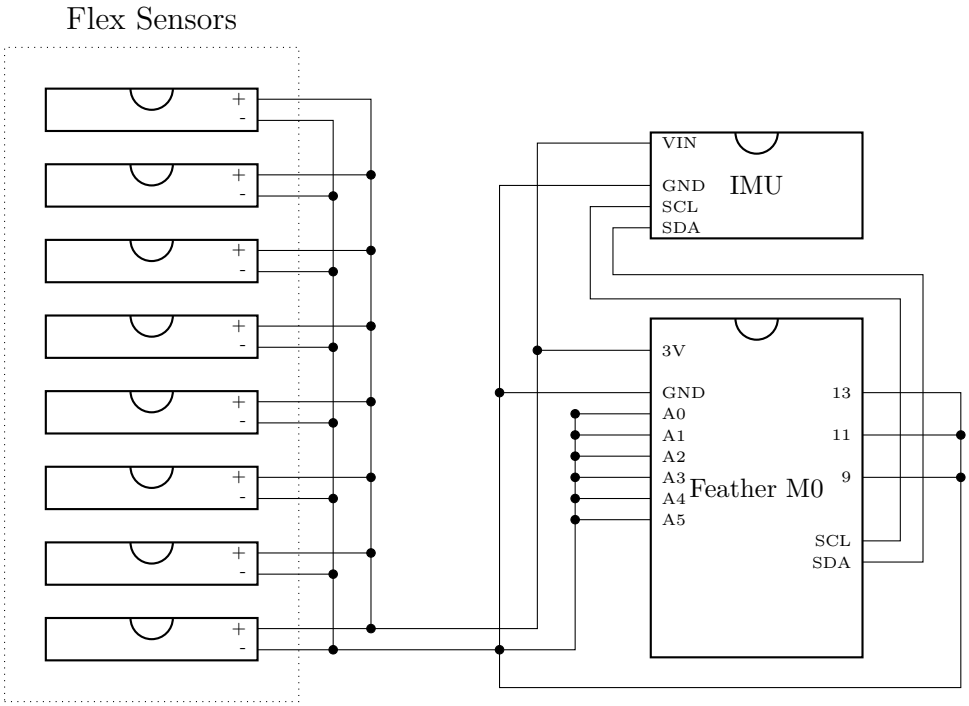


Figure 3: Circuit Diagram

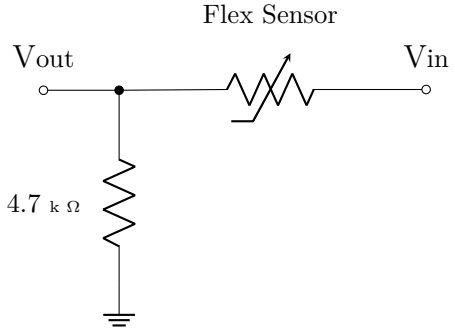


Figure 4: Flex Sensor Circuit

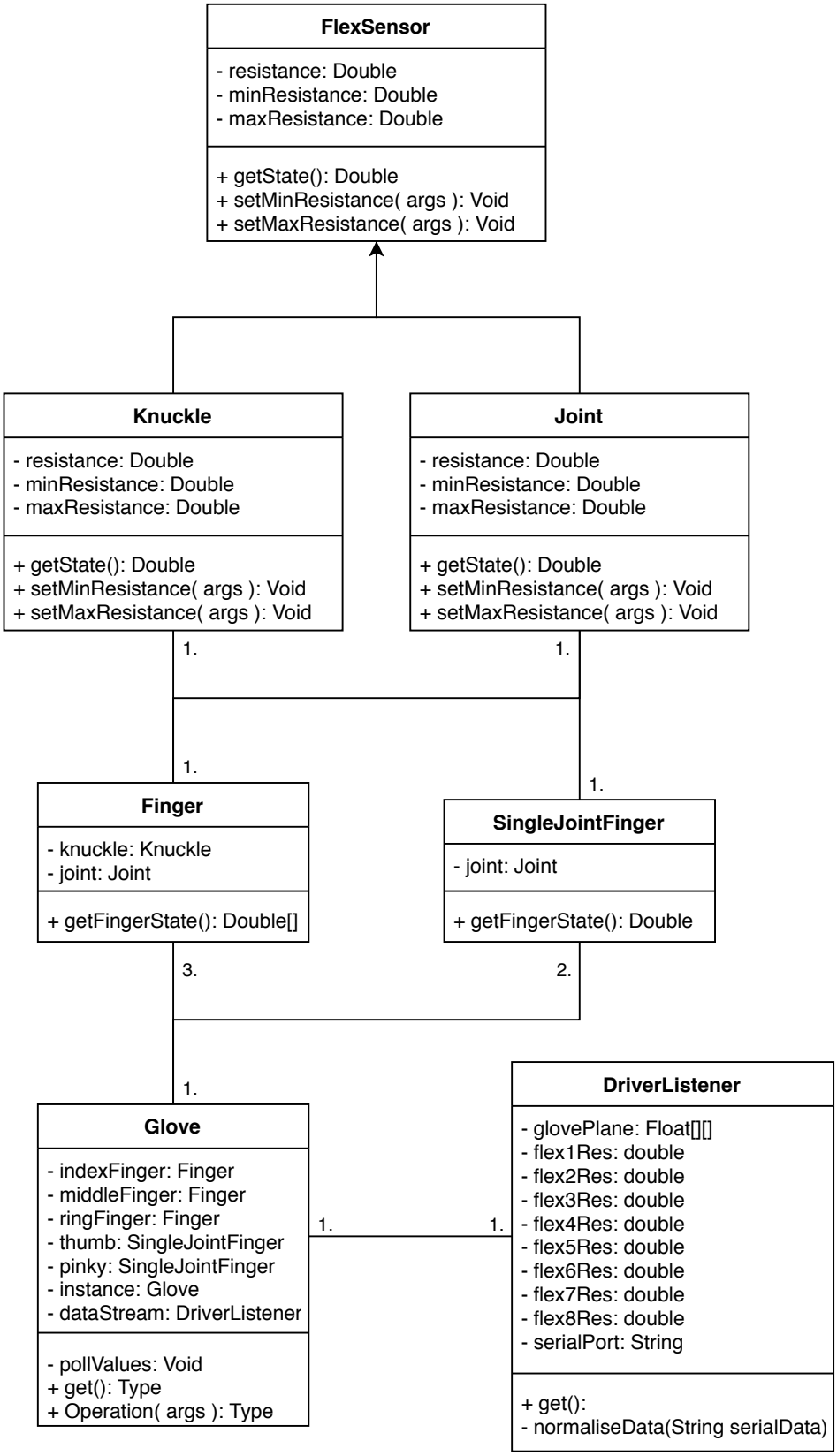


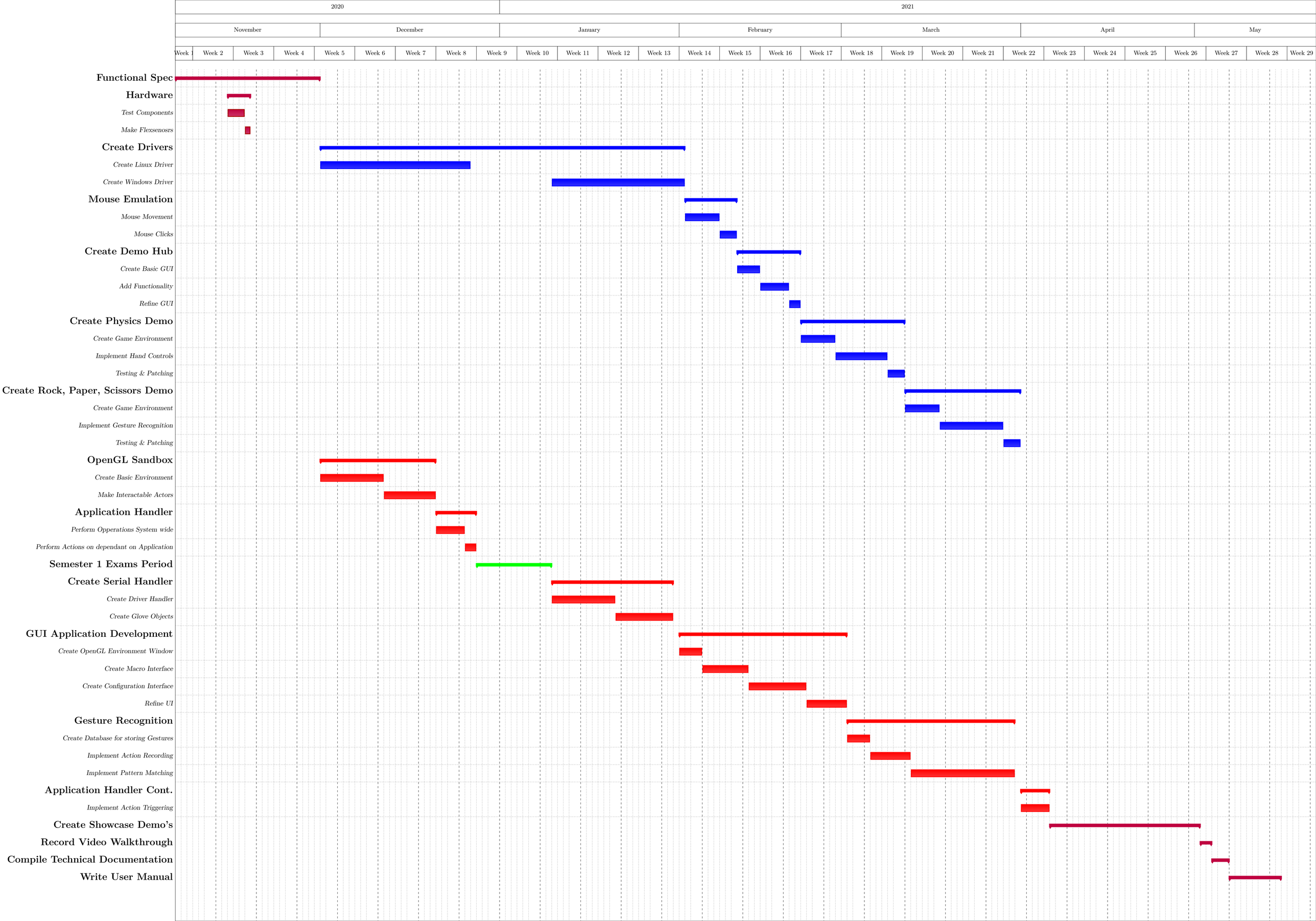
Figure 5: UML Class Diagram

Preliminary Schedule

In this section we will outline our provisional Gantt chart depicting how work will be completed through the duration of this project. The document is layed out as follows:

- Items which are Blue will be completed by Sean Moloney.
- Items which are Red will be completed by Alan Devine.
- Items which are purple will be completed by both of us.

Note: Time to write unit tests is factored in with each development task presented below.


2020
2021
November
December
January
February
March
April
May
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14
Week 15
Week 16
Week 17
Week 18
Week 19
Week 20
Week 21
Week 22
Week 23
Week 24
Week 25
Week 26
Week 27
Week 28
Week 29
Functional Spec
Hardware
Test Components
Make Flexsenosrs
Create Drivers
Create Linux Driver
Create Windows Driver
Mouse Emulation
Mouse Movement
Mouse Clicks
Create Demo Hub
Create Basic GUI
Add Functionality
Refine GUI
Create Physics Demo
Create Game Environment
Implement Hand Controls
Testing & Patching
Create Rock, Paper, Scissors Demo
Create Game Environment
Implement Gesture Recognition
Testing & Patching
OpenGL Sandbox
Create Basic Environment
Make Interactable Actors
Application Handler
Perform Opperations System wide
Perform Actions on dependant on Application
Semester 1 Exams Period
Create Serial Handler
Create Driver Handler
Create Glove Objects
GUI Application Development
Create OpenGL Environment Window
Create Macro Interface
Create Configuration Interface
Refine UI
Gesture Recognition
Create Database for storing Gestures
Implement Action Recording
Implement Pattern Matching
Application Handler Cont.
Implement Action Triggering
Create Showcase Demo's
Record Video Walkthrough
Compile Technical Documentation
Write User Manual

Appendices