COMP140 Report

Christopher Robertson

March 2020

//Half a page

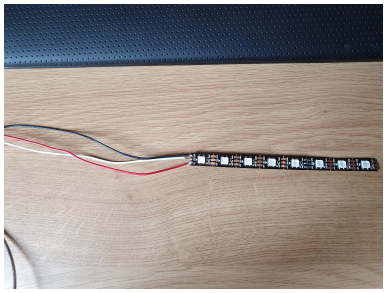
1 Project Proposal

The game design for the controller is a game where you control two sets of colours which are you able to change the hue of using a set of dials - or for debugging two keys on the keyboard. Each colour you control has a lane of queued up colours which the player must match their colour to the next colour in the lane. When the colours are matched they are able to press the dial - or a third keyboard button for debugging - to submit. The game will then either add points, or deduct lives if successful or not. As time goes on the speed of the game will increase much like any re-playable arcade game like this.

2 Hardware and Components

There are two main components that I am using for this controller which is the Rotary Encoder (with an LED Light) which acts as the input and output. I also am using an RGB LED strip which only acts as the output for the player and is the primary visuals for the game experience. The other components I'll be using include the Arduino Uno board which will be processing everything and holding the sketch itself. I will also be using a breadboard too as the Arduino Uno doesn't have enough live and ground slots for me to use.

2.1 Rotary Encoder and RGB LEDs



(b) LED Strip

(a) Rotary Encoder

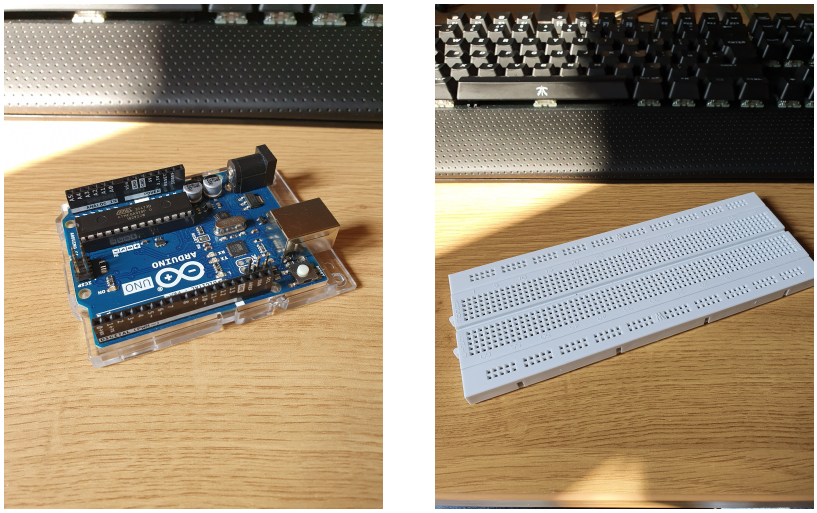
Figure 1: External Components

The rotary encoder is used to control the colour currently selected by rotating the dial which then cycles through an array of available colours. The player is informed of the colour selected by the RGB LED on the rotary encoder itself. Once the player thinks the selected colour is the same as the next colour they

press the button to submit. The rotary encoder I have has 8 connections in total; 3 for the RGB, 2 for live and ground, and then 3 for the rotaryA, rotaryB and the button.

The LED RGB Strip is used to feedback the current state of the game to the player. It represents the queue of colours that are next in order. The player will then use the rotary encoder to match the next colour. This piece of hardware only has three connections which are the live, ground and information wire. I am then using the Adafruit Neopixel library to individually address each LED with a different colour.

2.2 Arduino and Breadboard



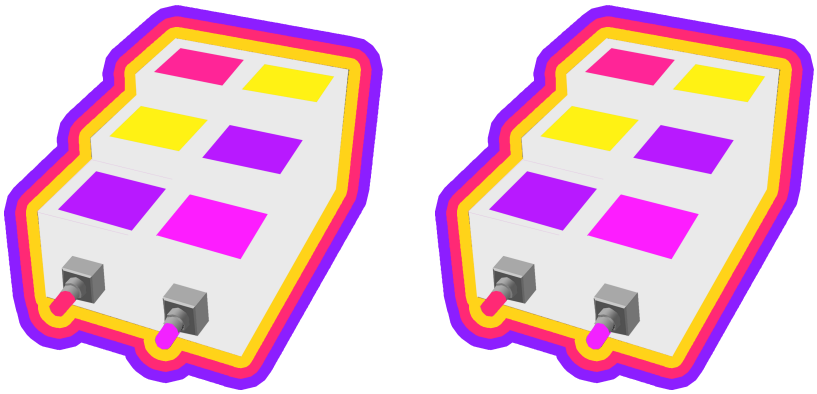
(a) Arduino Uno (b) Breadboard

Figure 2: Main Hardware

The Arduino will be the main piece of hardware in this project since it will be storing the entire game and the logic behind it. It will take in the input from the rotary encoder and then produce an output to both the rotary encoder LED and the LED strip. This will make the controller entirely in-built and will not require a PC, or more specifically Unity, to run, but will just have Unity as an added debug screen. I will also be using a breadboard as I wouldn't have needed one if the Arduino had 3 extra 5v slots and 1 extra ground slot.

3 Design of the Controller - Wiring, Casing and Attachments

3.1 Design



(a) Concept (b) Final

Figure 3: Controller Design

The design of my controller is to have 3 layers of boxes at 3 separate heights because I feel it is just a bit more aesthetically appealing than just being flat. The top of each of these boxes has a clear layer of plastic so that the player may see the LED colour inside. On the front are the two rotary encoders which are used to control each respective lane. I went for a simple design approach for this and tried to make it look like a sort of toy since that is the kind of game I was making with the in-built controller idea.

3.2 Wiring and Casing

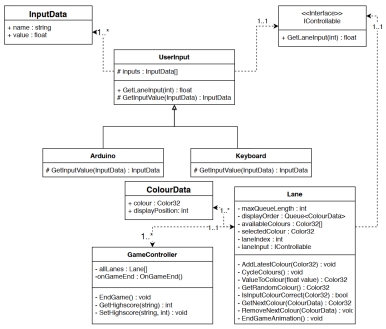
The casing of my controller is made out of thick cardboard - although plastic had been planned - where I made 3 sets of rectangular boxes with no top by cutting a sheet of cardboard and using PVA and craft glue to stick it together. Once it had dried I cut around the rough edges and then made each rectangular box its own height in descending order. I then cut the LED strip and wiring holes as well as the slots for the rotary encoder. From there I glued the 3 boxes together to create my final design. I then added some plastic to the top of each box as a lid. The wiring is ran through the controller round to the back where the breadboard and Arduino Uno is stored.

The wiring itself is fairly straight forward. The power and live of the rotary encoders and LED strips are all connected together on the breadboard which is then connected to the Arduino. For the LED strips I then connect the informa-

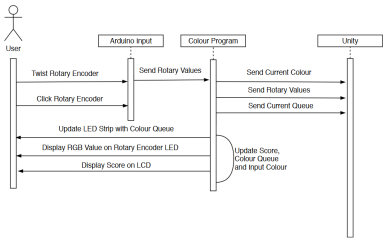
tion wire to the breadboard through a resistor and then to its designated digital read slot. This is to prevent it from melting the LEDs. The rotary encoder has all of the remaining wires - RGB and input - wired straight to the Arduino. These can be seen on the diagram.

4 Game Experience/Concept

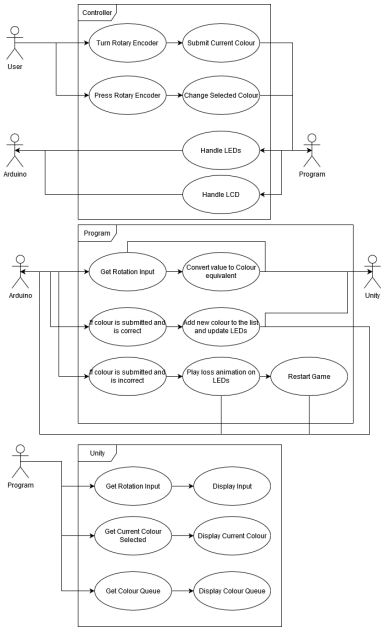
5 Software Design with UML Diagrams



(a) Class Diagram



(b) Sequence Diagram



(c) Case Diagram

Figure 4: UML Diagrams

//One Page

6 Reflection on the Project