Game Controller

Project

Participants

- دينا ابوزيد فرغلي cs1
- علياء احمد علي cs2
- الشيماء محمد فولي cs3
- عبدالرحمن ماهر مخلوف cs1
- محمد احمد حسن شحاتة cs2

General idea

A simple game controller that controls the motion of a player in a unity based game .

Components
Hardware

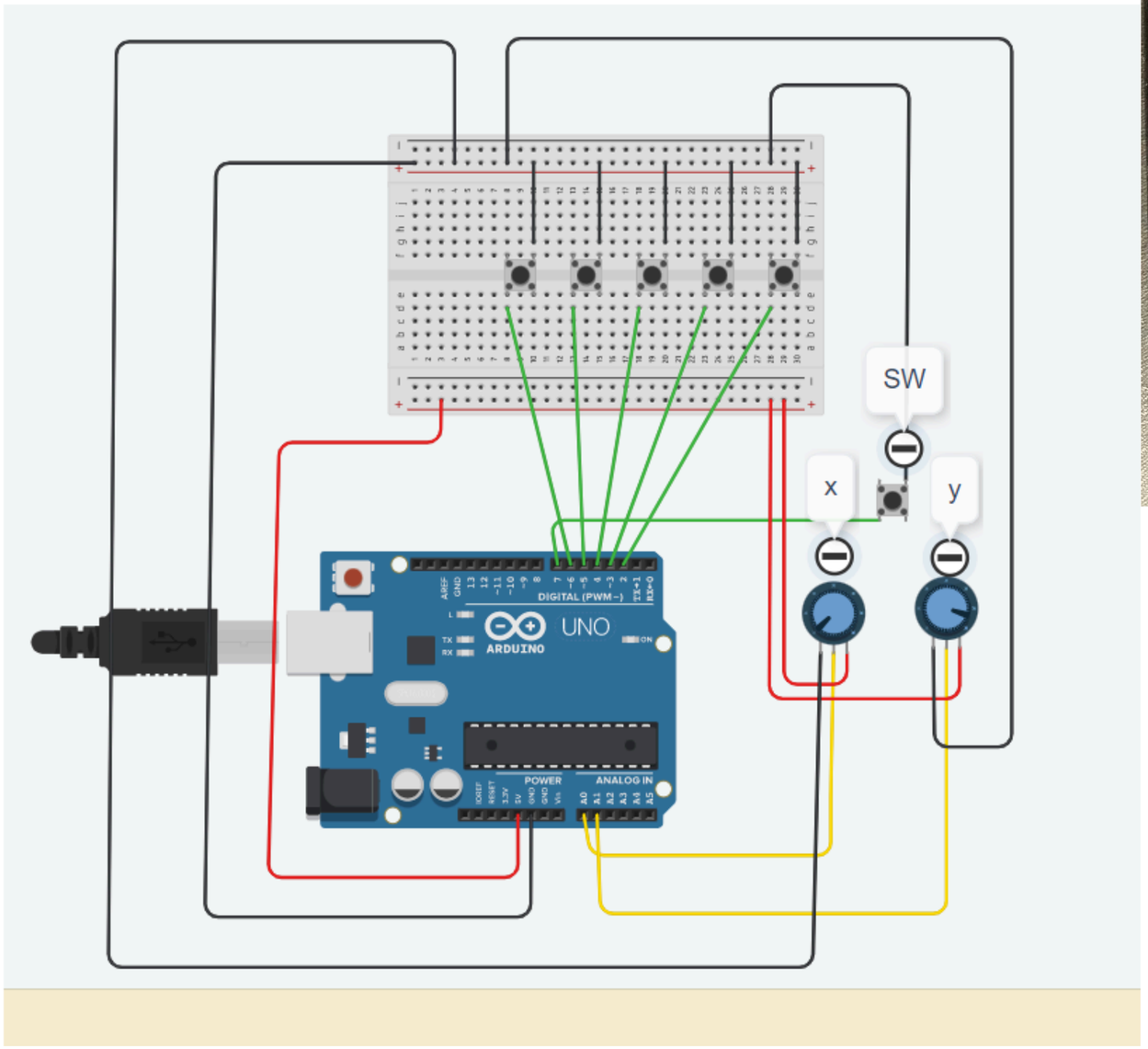
- 5 push buttons
- 1 joy stick
- 1 PCP Dot (Fr2)
- 1 Arduino UNO
- Cables

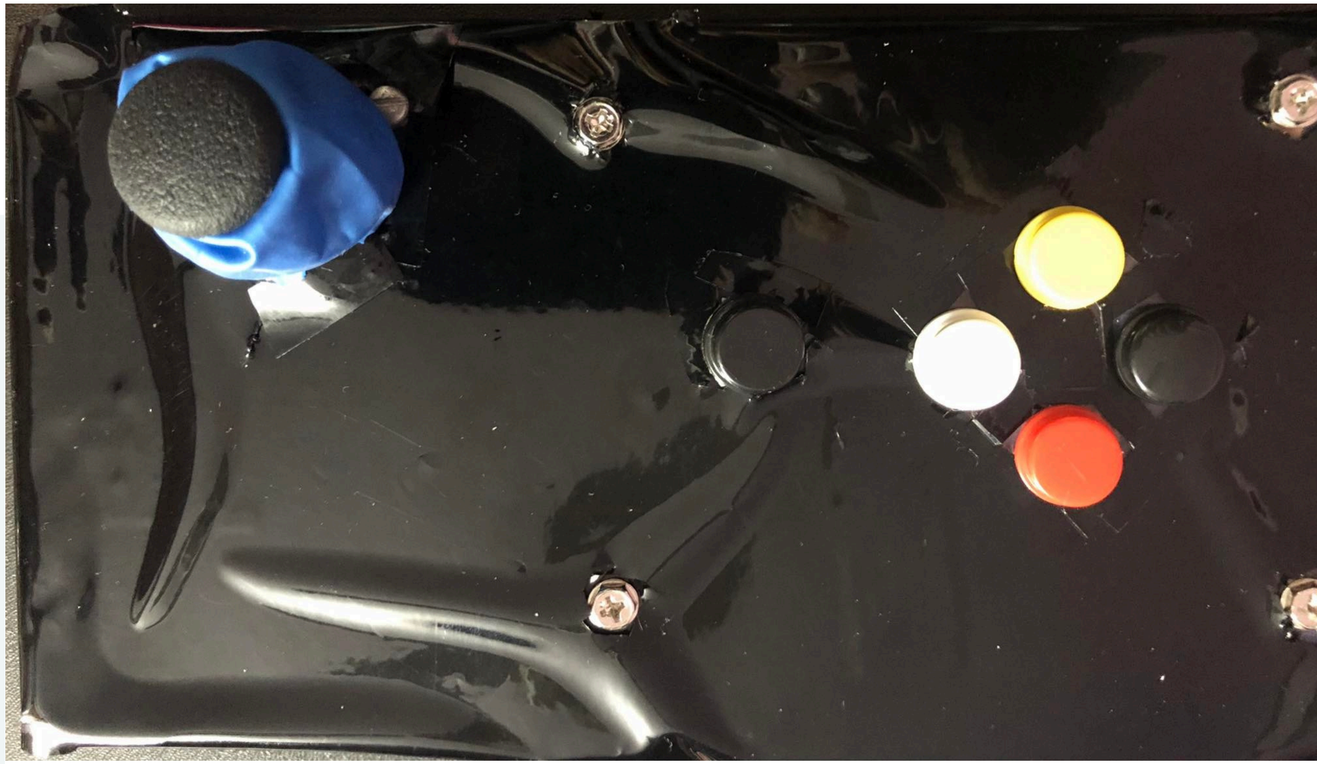
Components
Software

- Arduino
- Unity
- Tinkercad

How would the project work

The project would work by using the serial communication between aruduino and unity . When the user hits a button or moves the joystick , a serial message would be sent to unity . When unity receives the message , player movement can then occur.


SW
x
y
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN



Arduino Code

Declarations

- At the begging , We will declare variables we would use later on .
- SW_pin , x_pin and y_pin are the pin numbers for the joystick
- buttonState1~5 are the states of buttons (clicked or not)
- sw_pinRead , x_pinRead and y_pinRead are the states for the joystick
- del is the amount of delay used

```
const int SW_pin = 7;
const int x_pin = 0;
const int y_pin = 1;

int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
int buttonState5 = 0;
int SW_pinRead = 0;
int x_pinRead = 0;
int  y_pinRead = 0;
int del = 100;
```

Arduino Code

Setup

- Then we would initialize pin Mode
- Pins 2~6 are for buttons as input with pull-up
- SW_pin is for joystick as input
- Begin the serial transmission with a baud rate of 9600

```
void setup() {
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(4, INPUT_PULLUP);
  pinMode(5, INPUT_PULLUP);
  pinMode(6, INPUT_PULLUP);
  pinMode(SW_pin,INPUT);
  Serial.begin(9600);
}
```

Arduino Code

Loop

- We begin the loop by reading button states from their pins numbered 2~6

```
void loop() {

  buttonState1 = digitalRead(2);
  buttonState2 = digitalRead(3);
  buttonState3 = digitalRead(4);
  buttonState4 = digitalRead(5);
  buttonState5 = digitalRead(6);
```

Arduino Code

Loop -continue

- Then we will check for button states , if it was low then a button had been pressed .
- For each button press we send a number that implies the direction the player would move to
- If no button is pressed then we send 0

```
if(buttonState1==LOW)
 {
   Serial.println("1");
 }
 else if(buttonState2==LOW)
 {
   Serial.println("2");
 }
 else if(buttonState3==LOW)
 {
   Serial.println("3");
 }
 else if(buttonState4==LOW)
 {
   Serial.println("4");
 }
 else if(buttonState5==LOW)
 {
   Serial.println("5");
 }
 else{
   Serial.println("0");
 }
```

Arduino Code

Loop -continue

- Then we will read the states of the joystick as a button and two values x and y that represents the joystick direction

```
SW_pinRead = digitalRead(SW_pin);
x_pinRead = analogRead(x_pin);
y_pinRead = analogRead(y_pin);
```

Arduino Code

Loop -continue

- First , we check the joystick button state if it was 0 then it had been pressed
- Then we will check the values of x,y direction of joystick and then we would send a number that implies the direction the player would move to
- Then a general delay for debouncing

```
if (SW_pinRead == 0){
  Serial.println(5);
}

if (x_pinRead > 600){
  x_pinRead = 1;
}
else if (x_pinRead < 400)
  x_pinRead = -1;
else
  x_pinRead = 0;
if (y_pinRead > 600){
  y_pinRead = 1;
}
else if (y_pinRead < 400)
  y_pinRead = -1;
else
  y_pinRead = 0;
if (x_pinRead == 1)
  Serial.println(3);
else if (x_pinRead == -1)
  Serial.println(1);
if (y_pinRead == 1)
  Serial.println(2);
else if (y_pinRead == -1)
  Serial.println(4);
delay(del);
```