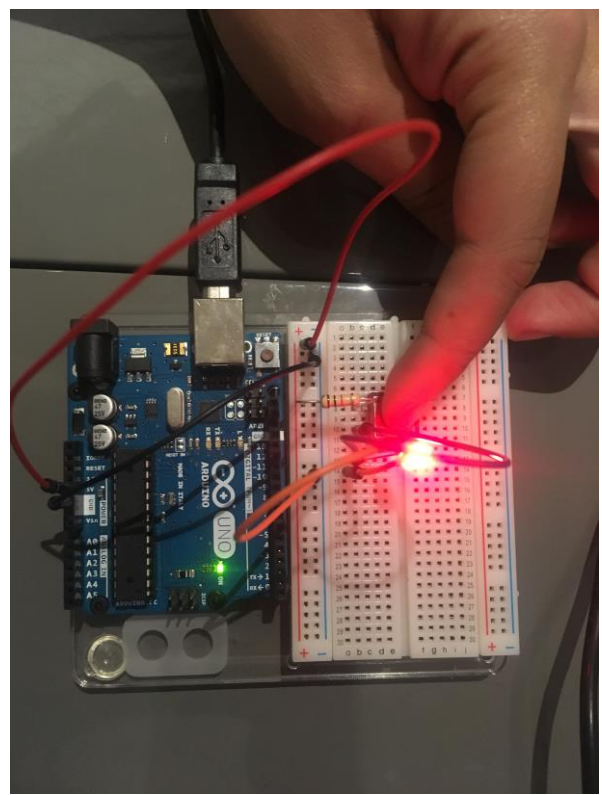
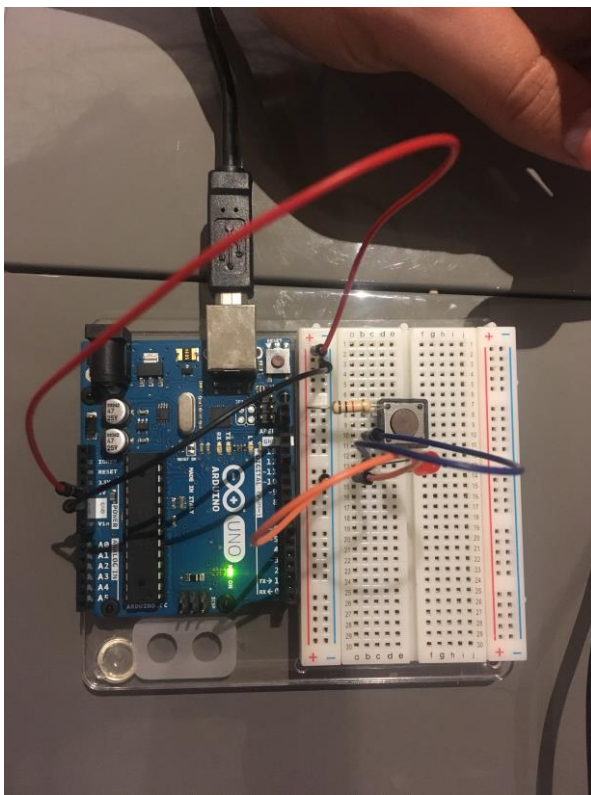


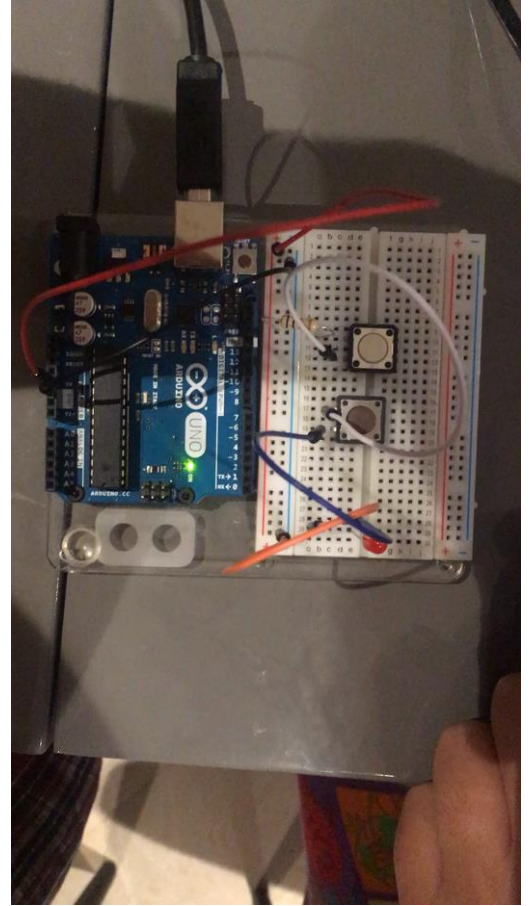
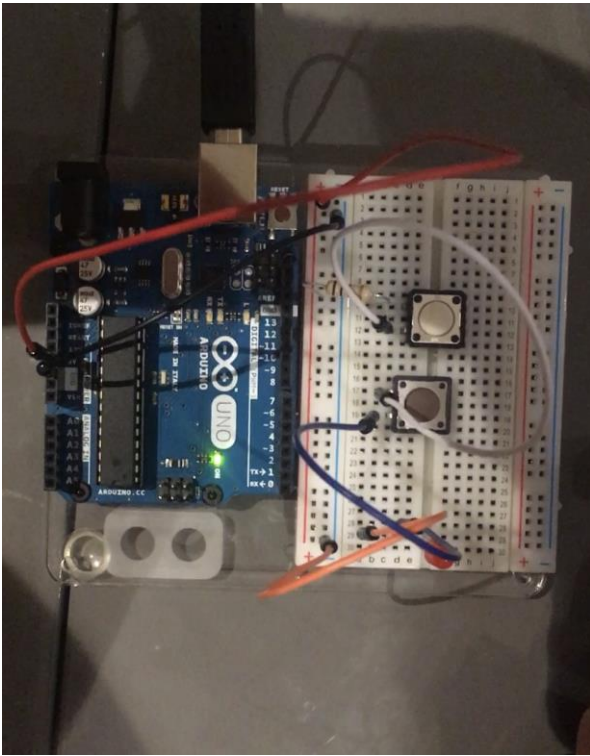
ARDUINO - WEEK 1

Project 1:

After reading the content of the project, I have started to build the first circuit with the LED and the button press. Here are two photos. The first one shows the setup of the circuit and the second one shows it functioning when the button is pressed.

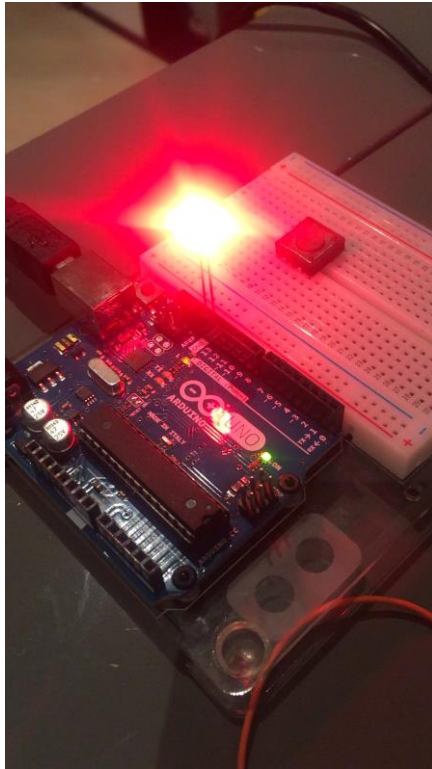


After this, I moved in to the second part of the first project. The two-button circuit. This exercise helped me understand better how succession of button component works. Here is a photo of the setup of the circuit and also a short video where I press both button but with a small delay to show how both buttons need to be pressed down to actually close the circuit completely and allow the current to go through the LED.



Project 2:

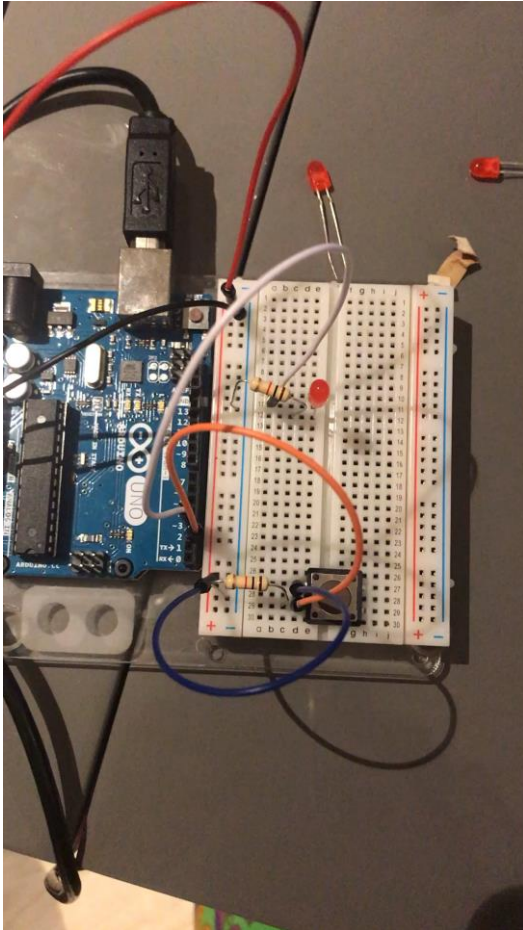
Before starting project 2, I have decided to read all of it to have an overview of what would come next. I then decided to experiment a bit before diving into the second project. I have made my LED flashed by using only the Arduino board by just entering the output through code. Here is a video of it:



The only advantage of using such a method to proceed is that the circuit is actually very clear and easy to “read” as there is no resistor or cable. I also quickly realised that this is also a bad way to proceed because we have no control on the LED itself through button press. However, I had fun exploring à different way too do it.

Now, for the project number 2 I was missing some cables and LEDs. Moreover, I didn't have the right resistor, but I still did my version of the project.

Using the set and the code from project 2, I will first, make the LED blink (with a 1 second delay) as soon as I plug my circuit into the USB port, it's is without button press, here is a video of it and a photo of the code that makes it work:



```
Circuit1

int switchState = 0;

void setup()
{
  // put your setup code here, to run once:
  pinMode(2, INPUT);

  pinMode(3, OUTPUT);
  // pinMode(4, OUTPUT);

}

void loop()
{

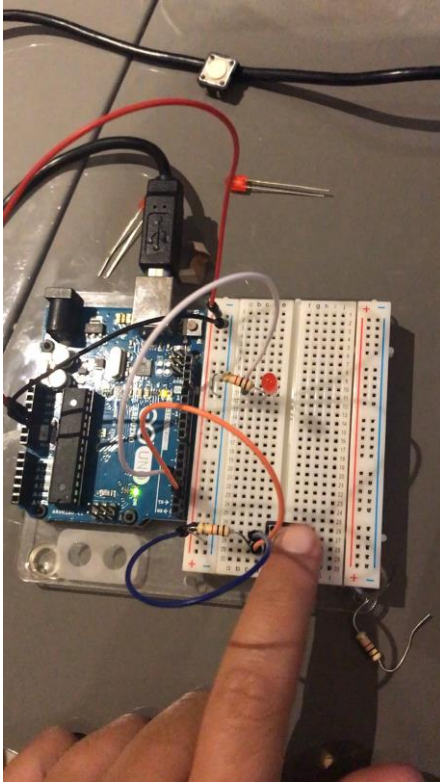
  digitalWrite(3, HIGH);    // turn LED on (High is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(3, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second

}


```

Compiling...
Sketch uses 268 bytes (0%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.

Now, using the very same set up, but a slightly different code (I just added the button press input), I make the LED blink only on button press. Here is a video of it and the updated code:



```
File Edit Sketch Tools Help
Circuit1

int switchState = 0;

void setup()
{
  // put your setup code here, to run once:
  pinMode(2, INPUT);

  pinMode(3, OUTPUT);
  // pinMode(4, OUTPUT);
}

void loop()
{
  if( switchState = digitalRead(2))

  digitalWrite(3, HIGH);    // turn LED on (High is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(3, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

I know that I could have more fun, adding up some LEDs outputs and playing with different delays to make the LEDs blink on different rhythm but unfortunately, I didn't have enough wires to do so. However, I plan to continue that work in class on Monday and add up some more LEDs to the circuit. I've already started to think about how I will update my code to make this work when I'll update my circuit, I just need to test it now:

```
Int switchState = 0; // State of the switch 0 = off
```

```
Void setup()
```

```
{  
pinMode (2, INPUT);  
PinMode (3, OUTPUT);  
pinMode ( 4, OUTPUT); // NEW LED ADDED ON THE 4th SLOT OF ARDUINO  
}
```

```
Void loop ()
```

```
{  
If (switchState = digitalRead(2)){ // if 0 means open circuit and 1 means  
closed circuit, this line of code means "if I press my button, the switchState !=0 which  
is what you can read in input 2 which is 1 (closed circuit)
```

```
DigitalWrite(3, HIGH); //turn in the LED on output 3
```

```
DigitalWrite(4, LOW); //turn off the LED on output 4
```

```
Delay(1000); // one sec delay
```

```
DigitalWrite(3, LOW); //turn off the LED on output 3
```

```
DigitalWrite(4,HIGH); //turn on the LED on output4
```

```
Delay(1000);
```

```
}
```

```
}
```

That code should make the LEDs blinks alternatively, when one is off the other one is one.

Mini game project:

I have started by trying to make a 3 seconds timer with a tricolor lights red orange and green to say "GO" as the mini game can start.

Video miniGame1 shows how I made the light turn on one by one, but I have no control on them (its not button controlled) and they also never turn off.

In the Video miniGame2 you can see that now the light starts only when I press the button and turn off at the end of the 3 second timer. To do so I have just added in the code the "digitalRead(2)" line of code where my button is plugged. To make the light turn off after the 3 second timer I didn't really know how to do it and I think there's probably a better way to do it than the way I decided to proceed but I tried to think logically to solve that problem. I have set up an int, which I change the value every time a new light turns on. When the last light is ON I store a new value for my int, for instance here I said 15. Then, when I reset all my LEDs values to low I just tell the code to do it when my int = 15.

Note that it was not evidently necessary to store value for this int in every LED but I decided to do it so it can give me more control and if I want to have a certain influence on just one led I can by using the value I already stored.

The game: tug of war

I added two buttons at each ends of the circuit. One button for each player.

However, unfortunately I couldn't manage to get the circuit to function the way I had hoped, either all the lights would stay active or not at all. I'm unsure as to if I have accidentally fried the LED's or if there's an error in the code.

Example code below:

```

int switchState = 0;
int switchStatePlayer1 = 0;
int switchStatePlayer2 = 0;
bool ledstate1 = true;
bool ledstate2 = false;
bool ledstate3 = false;
int ledValue = 1;

void setup()
{
    // start button and tricolor start 3 sec count down
    pinMode (2, INPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode ( 5, OUTPUT);

    // player 1 button and LED
    pinMode (7, INPUT);
    pinMode (6, OUTPUT);

    // player 2 button and LED
    pinMode (8, INPUT);
    // pinMode (13, OUTPUT);
    ledstate1 = false;
    ledstate2 = false;
    ledstate3 = false;
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);

```



```

digitalWrite(5, LOW);
digitalWrite(12, HIGH);
digitalWrite(13, HIGH);
switchStatePlayer1 = LOW;
switchStatePlayer2 = LOW;
}
void loop()
{
  //Serial.print("Hi");
  if( switchState = digitalRead(2)){
    ledstate1 = true;
    ledValue = 1;
    if( ledstate1 == true ){
      digitalWrite(3,HIGH);
      delay(1000);
      ledValue = 5;
    }
    if (ledstate2 == false){
      ledstate2 = true;
      digitalWrite(4, HIGH);
      delay(1000);
      ledValue = 10;
    }
    if (ledstate2 == true ){
      ledstate3 = true;
      digitalWrite(5, HIGH);
      delay(1000);
      ledValue = 15;
    }
  }
}

```

```

    }
}
if (ledValue == 15){
    digitalWrite(3, LOW);
    ledstate1 = false;
    digitalWrite(4, LOW);
    ledstate2 = false;
    digitalWrite(5, LOW);
    ledstate3 = false;
    ledValue = 1;
}
// switchStatePlayer1 = digitalRead(7);
// switchStatePlayer2 = digitalRead(8);
if( switchStatePlayer1 == LOW){
    Serial.print ("NOT PLAYING ");
}
else {
    Serial.print ("win");
    digitalWrite(6, HIGH);
}
/* if( switchStatePlayer2 == LOW){
    Serial.print ("NOT PLAYING 2");
}
else {

    Serial.print ("2win");
    digitalWrite(13, HIGH);
}*/

```