

Whack The Light!

Arduino project

By: Nasredene Elymani, Sam Zakeri,
Brandon Hakimi



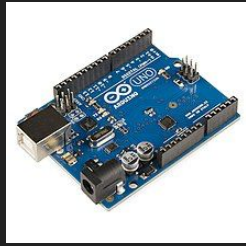


The basis of our game

- 5 sensors, 5 lights
- Once a light pops up on one of the sensors, the user is supposed to hit that light, and if done within a certain time limit, the user will earn a point
- As time goes on and more points are acquired, the time limit for which you must hit the light will shorten and the game will become increasingly difficult
- There will be 3 levels, where each level gets progressively harder than the next
 - Each level gets faster (as in you have less time to press each button), and the third level can have two lights flash at once, and is fastest



Innovation



- There will be two innovative aspects to the whack-a-light game
 - The levels will progressively get harder
 - The player reaches a certain score , the game will proceed to the next level
 - Each level will decrease the time there is to hit the light essentially making it tougher to earn points
 - There will be sound effects
 - Once the user hits the button, a sound effect will be played



The Code

Instantiation

```
const int red = 6; // red led pin
const int blue = 7; // blue led pin
const int white = 8; // white led pin
const int yellow = A1; // yellow led pin
const int green = A2; // green led pin
int randomNumber; // integer to hold the random number value generated
const int button = 9; // start button
int buttonCounter = 0; // starting the button counter value
int buttonState = 0; // initial value to hold button state.
int lastState = 0;
int gameScore = 0; // initial value for score placeholder to make buttons work
// int redButton;
int score = 0; // initialize score keeper for scoreKeeper function
```

Flashing for lights

```
class Flasher
{
  int ledPin; // the number of the LED pin
  long OnTime; // milliseconds of on-time
  long OffTime; // milliseconds of off-time
  int ledState; // ledState used to set the LED
  unsigned long previousMillis; // will store last time LED was updated

  public:
  Flasher(int pin, long on, long off)
  {
    ledPin = pin;
    pinMode(ledPin, OUTPUT);
    OnTime = on;
    OffTime = off;
    ledState = LOW;
    previousMillis = 0;
  }

  void Update()
  {
    unsigned long currentMillis = millis();
```



```
if((ledState == HIGH) && (currentMillis - previousMillis >= OnTime))
{
    ledState = LOW; // Turn it off
    previousMillis = currentMillis; // Remember the time
    digitalWrite(ledPin, ledState); // Update the actual LED
}
else if ((ledState == LOW) && (currentMillis - previousMillis >= OffTime))
{
    ledState = HIGH; // turn it on
    previousMillis = currentMillis; // Remember the time
    digitalWrite(ledPin, ledState); // Update the actual LED
}
}
};

Flasher led1(6, 500, 500);
Flasher ledBlue(7, 500, 500);
Flasher ledWhite(8, 500, 500);
Flasher ledYellow(A1, 500, 500);
Flasher ledGreen(A2, 500, 500);
```


Determination of light up for color

```
if(randGreen == 1){  
    digitalWrite(red, HIGH);  
    digitalWrite(blue, LOW);  
    digitalWrite(white, LOW);  
    digitalWrite(yellow, LOW);  
    digitalWrite(green, LOW);  
}  
if(randGreen == 2){  
    digitalWrite(red, LOW);  
    digitalWrite(blue, HIGH);  
    digitalWrite(white, LOW);  
    digitalWrite(yellow, LOW);  
    digitalWrite(green, LOW);  
}  
if(randGreen == 3){  
    digitalWrite(red, LOW);  
    digitalWrite(blue, LOW);  
    digitalWrite(white, HIGH);  
    digitalWrite(yellow, LOW);  
    digitalWrite(green, LOW);  
}
```

```
if(randGreen == 4){  
  digitalWrite(red, LOW);  
  digitalWrite(blue, LOW);  
  digitalWrite(white, LOW);  
  digitalWrite(yellow, HIGH);  
  digitalWrite(green, LOW);  
}  
if(randGreen == 5){  
  digitalWrite(red, LOW);  
  digitalWrite(blue, LOW);  
  digitalWrite(white, LOW);  
  digitalWrite(yellow, LOW);  
  digitalWrite(green, HIGH);  
}
```