Whac-a-Mole

Mini arcade game using Arduino

Loga Darius

Group: 30432

Technical University of Cluj-Napoca

1. Introduction

I wanted to recreate the famous arcade game: Whack-a-Mole using Arduino board and components. The purpose of the original game is to collect points by “hitting” moles that appear at random time and frequency using a toy hammer.

My aim for this project is to somehow replicate that “hitting” using buttons, moles appearing at random frequency at certain time thresholds, display the remaining time and score and also play a sound at each threshold and at the end of the game.

2. Design

Using the LCD module, I will try to display a timer starting from a certain time going down to 0 and also the score by pushing buttons corresponding to the each different LED light, which will be the “mole” representation. At certain time thresholds, the LEDs will increase in frequency, which means more “moles” in the same unit of time, one at a time. After each threshold ends, there are 2 or 3 seconds of “pause”, no moles will appear, therefore no points for a better score and also a tone will be played from the buzzer, but, at the end, the tone will be of a different frequency. Below I will provide an image (**Fig. 1**) with the connected components that I used for this project. For example: passive buzzer, LEDs, pushbuttons, resistors (220 and 10k Ω), 10k Ω potentiometer for the LCD display and wires.

3. Implementation

First of all, I tried some ideas in the Tinkercad simulator, but, of course, it worked so slow to test all the functionalities, even when the total time for the game is a little bit over one minute. For the block schematic with the connections I will provide an image (**Fig. 2**) from Tinkercad with all the connections, even though I used a Mega board and the connections are not exactly placed accurate in the simulator as would on the board, but I will insert one (**Fig. 3**) with declarations of the components and their pin location.

From now on, I developed a way to make a countdown timer for the remaining time of the game. I tried using timer interrupts to obtain an incremented time variable used later in the project for defining time thresholds and also for the “pauses”. The real struggle was to find a way to use that variable in a decremental way and by using one formula, I managed to correctly display the countdown. The whole concept is based on pressing the correct button associated with one different colour from LEDs and if pressed right on time, the score will go up by 10, else the red LED, gaining 0 points.

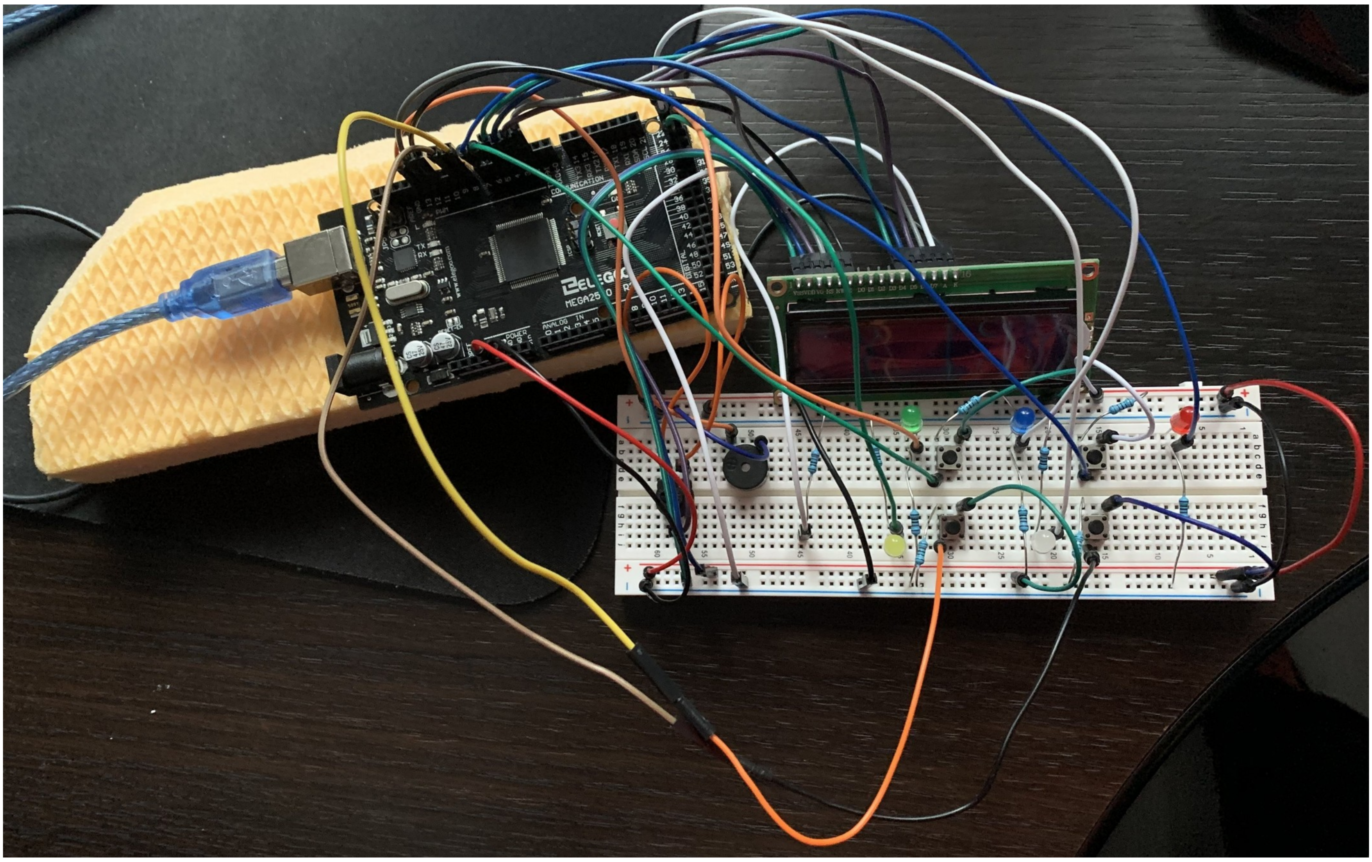


Fig. 1 — Final project

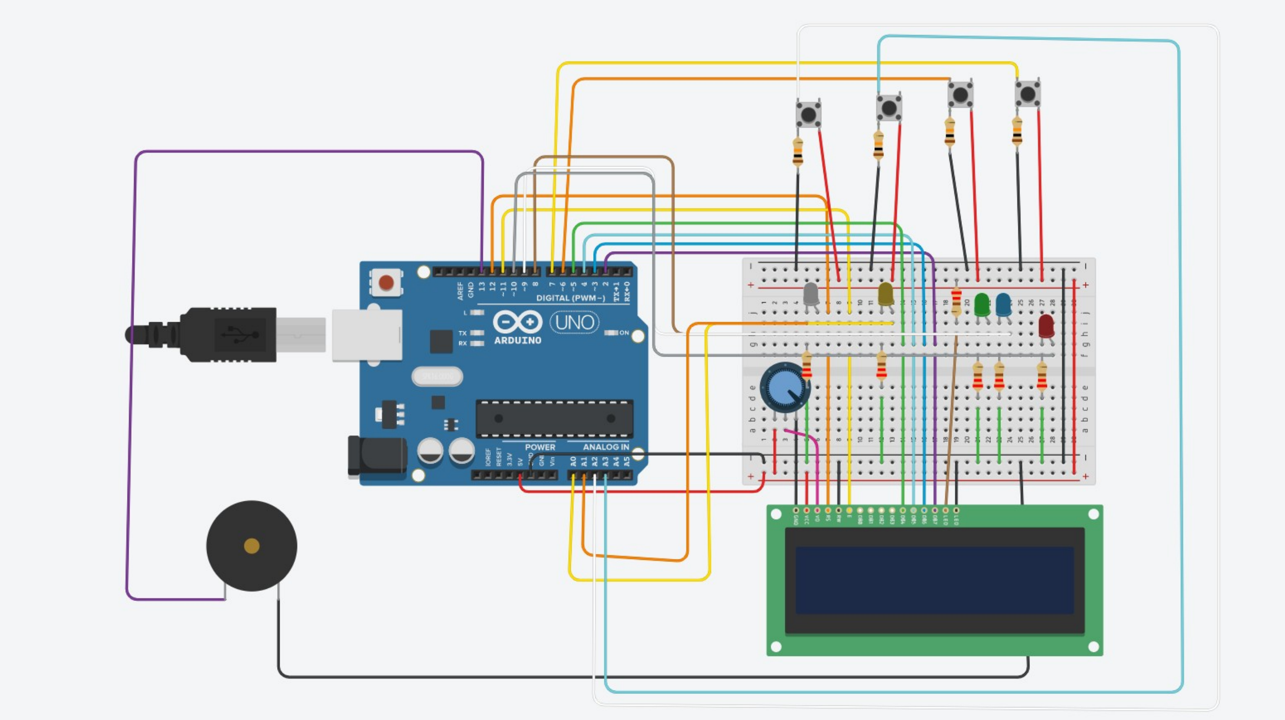
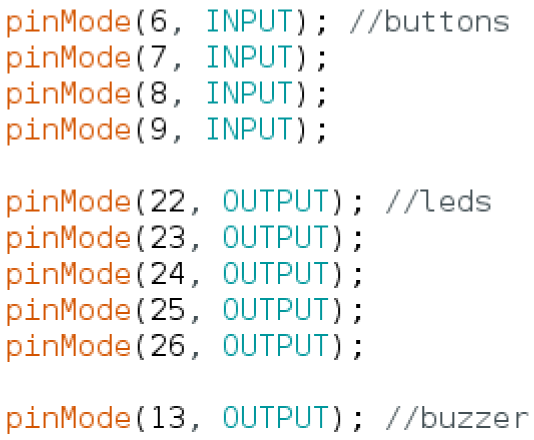


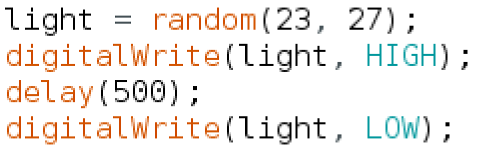
Fig. 2 — Connection scheme



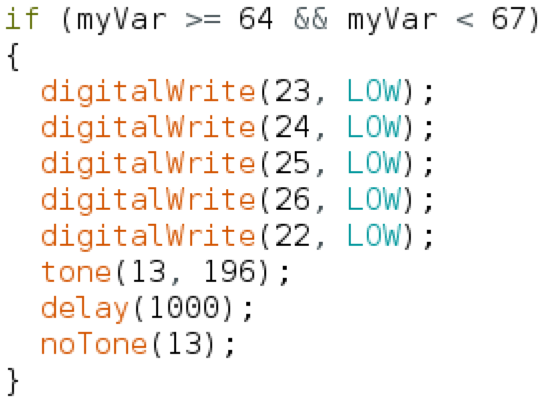
```
pinMode(6, INPUT); //buttons
pinMode(7, INPUT);
pinMode(8, INPUT);
pinMode(9, INPUT);

pinMode(22, OUTPUT); //leds
pinMode(23, OUTPUT);
pinMode(24, OUTPUT);
pinMode(25, OUTPUT);
pinMode(26, OUTPUT);

pinMode(13, OUTPUT); //buzzer
```

Fig. 3 — Components and their associated pins

The "moles", LEDs, flashes at certain time and frequency based on three stages, but at random order. This images represent the random choice of one LED at a time.



```
light = random(23, 27);
digitalWrite(light, HIGH);
delay(500);
digitalWrite(light, LOW);
```



```
if (myVar >= 64 && myVar < 67)
{
  digitalWrite(23, LOW);
  digitalWrite(24, LOW);
  digitalWrite(25, LOW);
  digitalWrite(26, LOW);
  digitalWrite(22, LOW);
  tone(13, 196);
  delay(1000);
  noTone(13);
}
```

Fig. 4 — Last three seconds of the game where the buzzer have a different frequency using delay

4. Results

The mini arcade almost worked as intended except that the random number generated two numbers for every time the LEDs nedeed to flash, but I could get the point for both if I was fast enough and sometimes the buttons would not register that I missed and would not flash the red LED. The faulty generation could be because of the timer interrupt, but I am not that sure and the buttons have a bigger delay that would get past the flashing time. As far as I am concerned, every other functionality works fine.

5. Conclusions

In the end, it was a pleasure to work on such a fun project and also to develop my Arduino skills. For the future, I would like improve this project to work properly and have other features such as: select the desire time, select difficulty, how many stages, and, why not, multiplayer.