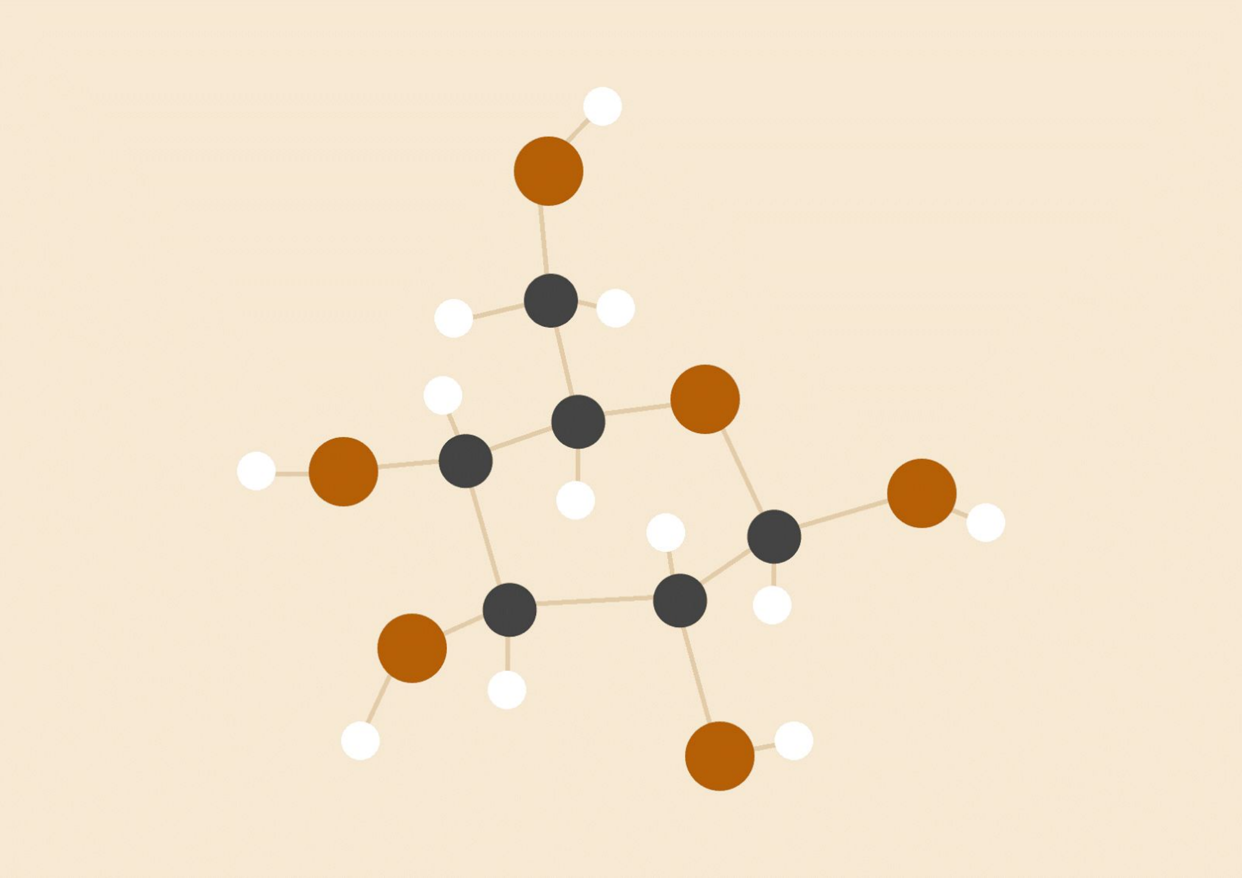
NINJA REFLEX

Do you have what it takes to be a ninja?



Ahmad M. Temoor: ahmadtem@buffalo.edu

Syed Ali Hasan: syedhasa@buffalo.edu

University at Buffalo
The State University of New York

CSE 321 **Embedded Systems**

EXECUTIVE SUMMARY

The 24” x 24” platform powered by a quad-core 64-bit processor running at 1.2 GHz, putting your reflexes to the ultimate test. This is an interactive platform that will not only help you evolve your reflexes but it will do it in a enjoyable yet challenging environment. It is designed in a way in which you can play alone or with as many friends as you want and have a blast either way. Built for all genders, all age groups ranging between toddlers all the way up to seniors and people from any part of the world regardless of the language they speak. The initial implementation of the platform will be at a single location where anyone can come use it but as statistics will be collected and the platform evolves, it can be distributed across the world.

PROJECT OBJECTIVES

Hit the target wall as rapidly as possible, trying to increase your level to the highest possible number within the course of one minute.

REFLEX NINJA will meet the following objectives:

- Initialize a new game
    - All walls will blink thrice
    - A 1 minute timer will be started
    - Player level will be reset to 0

- Display current information in real time
    - Display time left
    - Display current level

- Determine if the target wall, the wall that is lit up, is hit
    - Increment the player level
    - Reset the target wall
    - Update display

- End the game once the timer has stopped
    - All walls will light up
    - Player level will be displayed

PROJECT APPROACH

Python 3 is the programming language that is used to run the backend scripts. As users play, statistics will be parsed and written onto a Json file which will be stored on the external Micro SD card of the Raspberry Pi 3.

We will launch the platform in phases in order to make this the platform that works great with anyone who wants to use it. In order to do this, first we will launch a beta version at a single location where different people can come to and play. This will put the platform to its final test it as it will collect statistics in order to access:

1. validity of existing approach;
2. determine if the users would like to see certain things change;

The application will collect data such as the number of walls hit, frequency and average response time. This information will be stored on the embedded system. After a certain amount of time, this data will be analyzed which will be used to not only make improvements to the existing design but add features such as incorporating different levels of difficulty.

PROJECT DESCRIPTION

The 24” x 24” platform will be designed from wood to provide a sturdy and robust playing arena. The embedded system of choice is the Raspberry Pi 3, due to its impressive specifications and low cost. The screen used to display information could be any monitor or TV that has an HDMI input.

The gameplay is simple; once you power the platform up, all walls will blink three times indicating that the game is about to start. Parallel to which, the screen will display a 1 minute timer and the “Player Level”. The player level is simply the number of times the player has hit the target walls. This will be initialized to 0 and be incremented by 1 every time a target wall is hit. Through the course of this minute, the target wall will change as soon as the current target wall is hit as the player tries to hit as many walls as possible. Once the minute is over, all walls will light up and the player level will be displayed on the screen.

As a player, you can keep playing by yourself, keeping a track of the highest level you have achieved and try to beat your own record. The same concept is extended to multi-players, as different players give it a try, players can keep a ranking based on their highest levels. Thus, the number of players can be as as big or small of a number as desired.

Going through this process, users will be challenged to achieve a higher player level in a fun, possibly addictive manner. The advantages of the game include, yet are not limited to:

- Improved reflexes
- Ability to focus more; improved concentration
- Increased peripheral vision
- Improvement performing under pressure
- Develop social skills while playing with friends

Anyone across the world can greatly benefit from the above and one of the best thing about REFLEX NINJA is that its intuitive design is universal; anyone from anywhere regardless of gender, age, language can get use it and reap the same advantages as anyone else.

DESIGN DETAILS

The design of the Ninja Reflex game and platform involved using several different tools and technologies. A list of tools and technologies is shown below:

1. Raspberry Pi 3
2. Wooden Platform
3. RGB LED Strip
4. Breadboard
5. Connecting Wires
6. Capacitors
7. Transistors
8. MOSFET's
9. LDR's
10. Laser Diodes

All of the above tools were used in conjunction with each other so as to create an aesthetically pleasing and efficiently functioning game. The Raspberry Pi 3, using a script written in Python, interacts with the different sensors such as the LDR's and Laser Diodes, in order to determine the collision of a ball with a wall. It also controls the LED strip which is made to randomly light up once a ball has hit a wall. The platform is housed within a wooden frame that was created in the Machine shop located in Jarvis Hall.

Note: Since a single python script contains the entire backend functionality, UML diagrams were not included.

USER'S MANUAL

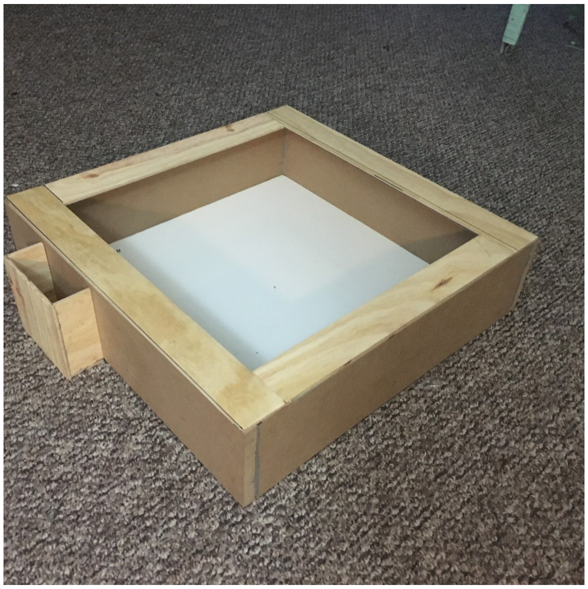
The following steps need to be taken in order for the customer to properly use the product.

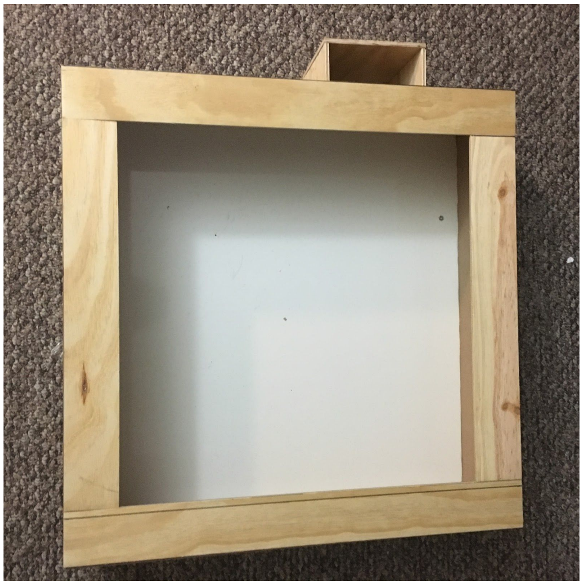
1. First startup the Raspberry Pi 3 and make sure the source of power is connected properly. The game will automatically boot itself up.
2. Wait for the walls to blink three times. This is an indication that the game is about to start.
3. Grab the ball and hold it in the center of the platform and wait for any one of the walls to light up.
4. Once the wall lights up, strike it with the ball as fast as possible.
5. If the correct wall has been striked, the next wall will light up otherwise the game is over. Grab the ball as fast as possible and strike the next wall.
6. This process will consistently repeat for a duration of 1 min after which the round has ended. This will be indicated by all walls being lit up.
7. The player level will be determined by the number of walls hit per minute. This will be displayed on the screen.
8. In order to start the game again, reboot or restart the Raspberry Pi 3 and repeat the same steps.

PROGRAMMERS MANUAL

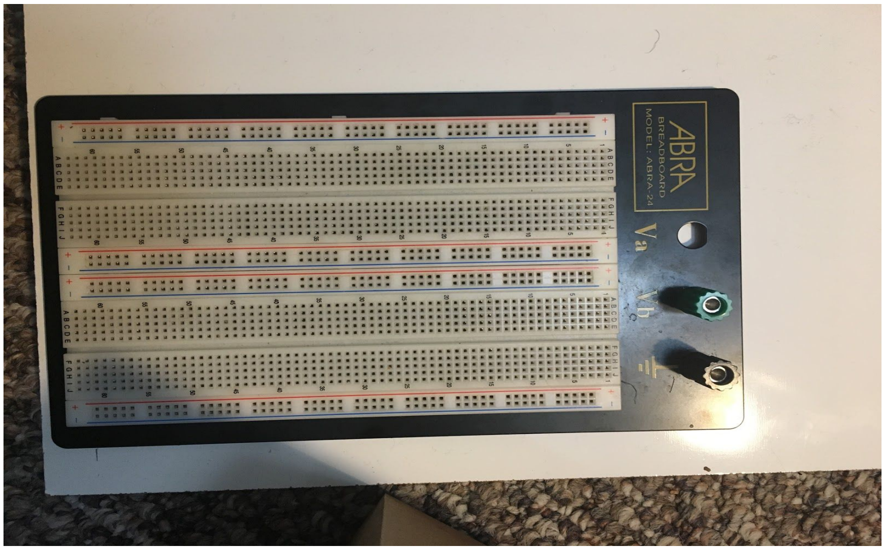
The items needed to create the Ninja Reflex game are as follows:

1. Wooden Platform



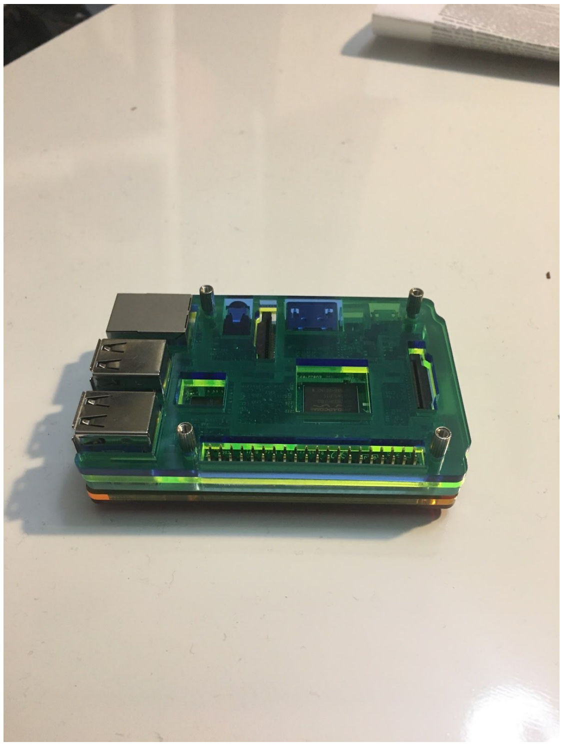


2. Breadboard

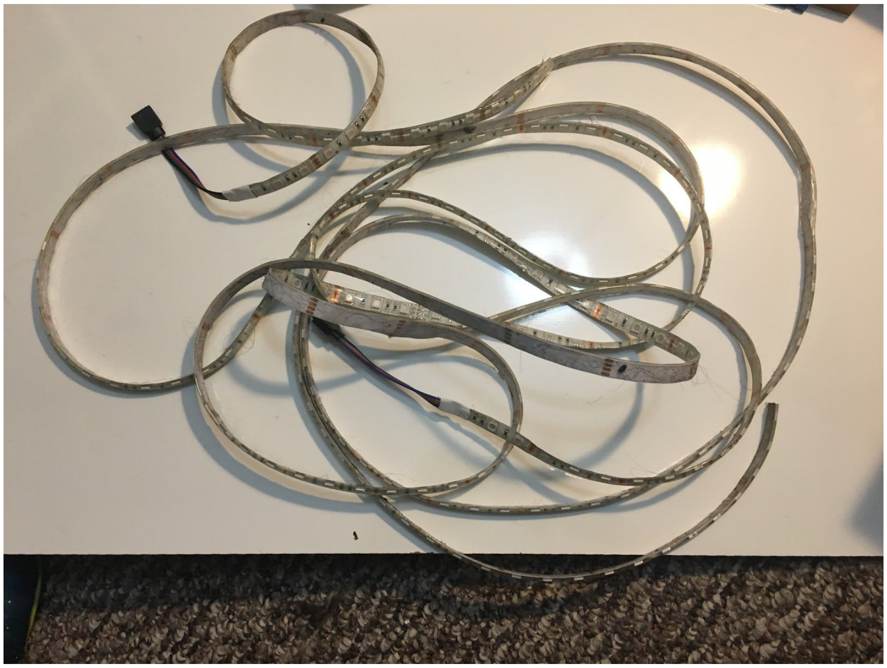


3. Connecting wires

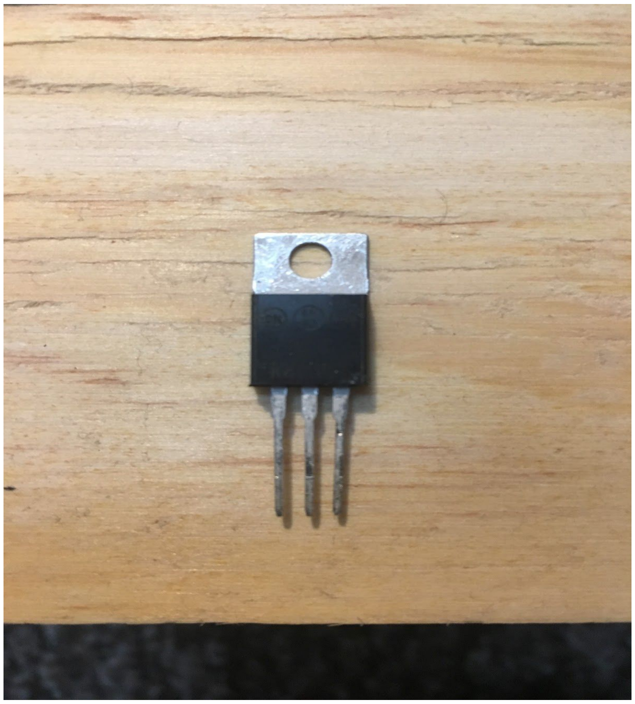
4. Raspberry Pi 3



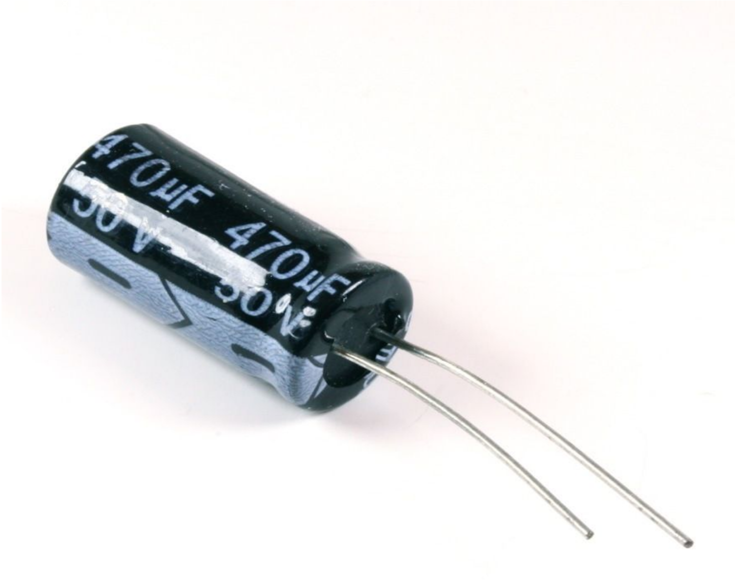
5. RGB LED Strip



6. MOSFET’S



7. Capacitors



8. Transistors

9. Light Dependent Resistor (LDR)

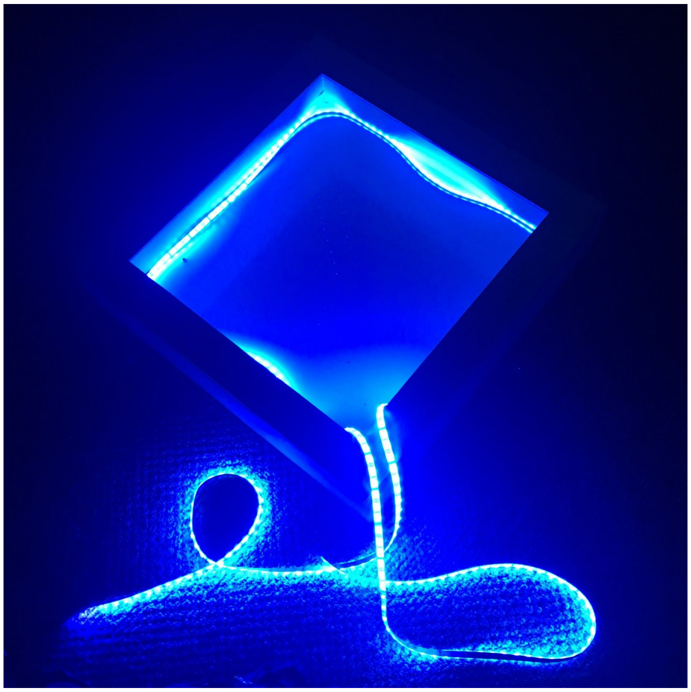
10. Laser Diode

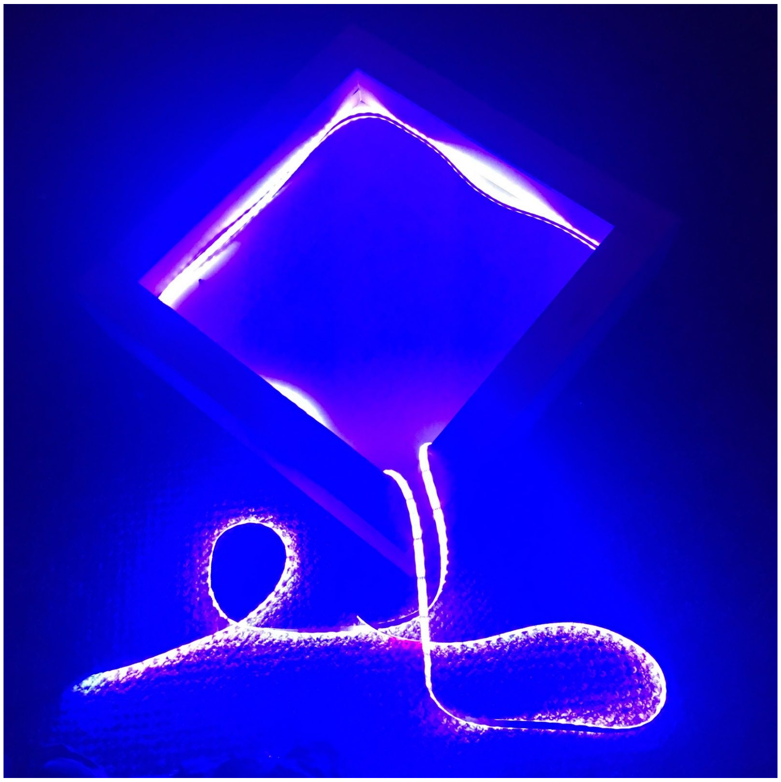
Assembly:

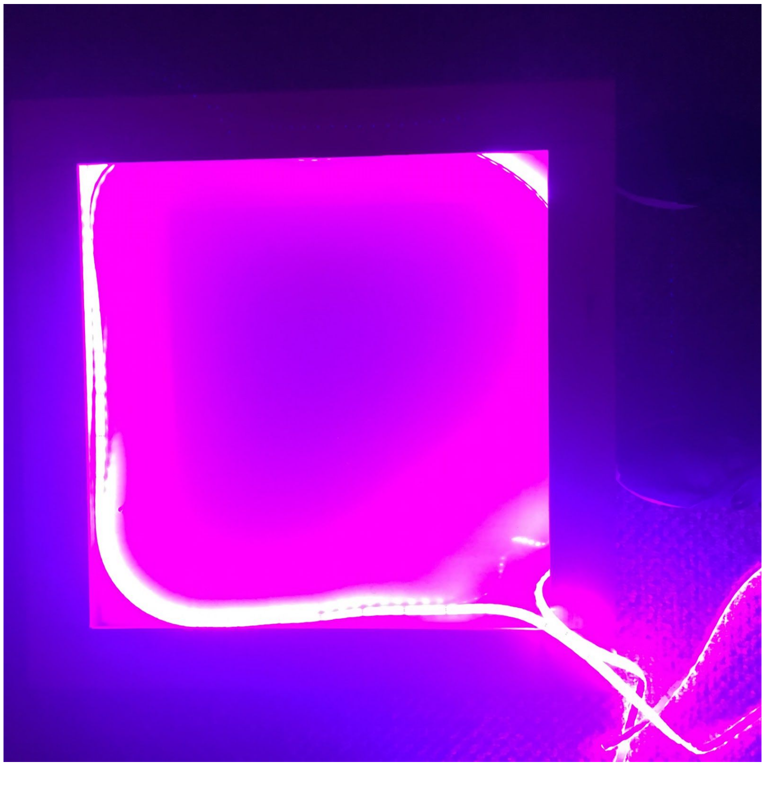
After all the parts listed above have been acquired, the final assembly of the game can be made. The steps for the assembly are as follows:

1. The first step is to embed any required sensors within the wooden platform. The LDR and laser diodes are embedded within each corner of the wall so that a trip wire can be established within the borders of the inside of the square shaped platform. This is used to determine whether a ball has first hit the wall and then allows us to determine if the correct wall has been hit. These LDR's and diodes are both connected to the Raspberry Pi 3 and breadboard.
2. Then the LED strip is attached on the top corners of the wall and roof. This allows the lights to be covered up and only portray the light downwards so as to light up the walls. The LED strip is connected to the Raspberry Pi 3.
3. After this the MOSFET's, capacitors, and transistors are connected to the breadboard and connected to the LED strip in order to create a circuit. These connections along with the Raspberry Pi 3 are used to light up the walls. Because MOSFETS are used, the platform has the potential to eventually control colors based on the RGB value.

The figures below show how once the connections have been made the walls can be lighted up and colors transitioned.







Programming Guideline:

The design philosophy of the backend is the same as the rest of the platform; simple, flexible and universal in the idea that anyone can use it as it is easy to understand. Thus, all the code needed to make platform work is in a single Python 3 script. The following modules are imported:

- gpiozero
    - LightSensor
- pigpio
- time
- random

Manipulating the above libraries, a simple minute circuit is initialized by using the time() method and performing arithmetic operations to get data such as elapsed time. The player level is simply an integer that is incremenented within an “if statement” which is only entered if the target wall is hit. The same concept is extended to turn the wall lights on or off by sending signals to the gpio pins of the Raspberry Pi 3.

REFERENCES

1. https://thepihut.com/blogs/raspberry-pi-tutorials/27968772-turning-on-an-led-with-your-raspberry-pis-gpio-pins
2. http://popoklopsi.github.io/RaspberryPi-LedStrip/#!/
3. https://www.raspberrypi.org/learning/laser-tripwire/