

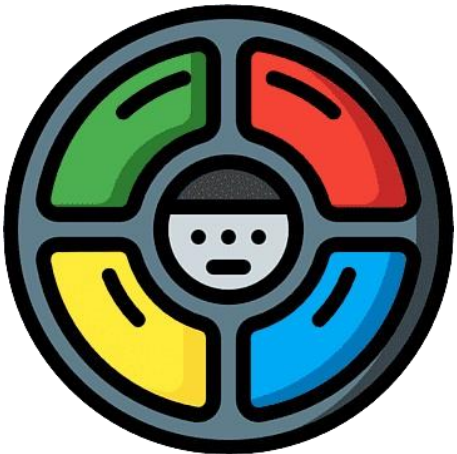
# Simon Say (game)

---

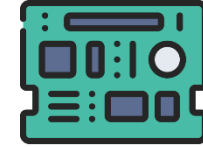
Bar Harush , Matan Katsnelson, Tomer Edelsberg  
Supervisor: Vladi Bodnitsky

# Introducing

Simon is a classic electronic memory game that challenges players to recall and replicate a sequence of colors and sounds. Introduced in the late 1970s, it features a circular console with four colored buttons—red, blue, green, and yellow—each emitting a distinctive tone. The game starts with a simple sequence of lights and sounds, and as players successfully repeat the sequence, it grows progressively more complex. Simon is celebrated for its engaging simplicity and the way it tests and enhances one's memory and concentration skills. Its iconic beeps and flashing lights have made it a memorable piece of gaming nostalgia.



# Implementation Details

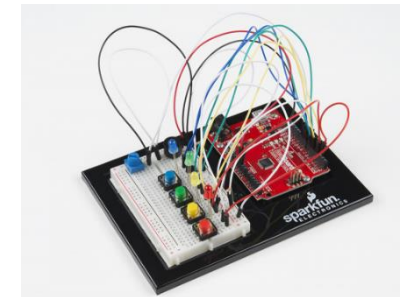
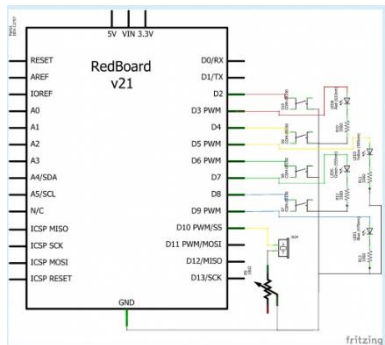


In this course, we explored various foundational topics in microcontroller programming and interfacing which we used in the project. We covered setting PORT to control bits, which allowed us to turn bits on or off (HIGH/LOW), which is essential for toggling components like LEDs. We also examined electrical buttons and the optional use of a Pulldown and Pullup Resistor. The concept of debouncing was introduced to manage the noise in button signals. Polling was discussed as a technique for continuously checking the status of inputs. We delved into handling analog signals and understanding communication protocols, particularly UART, to facilitate data exchange. Lastly, we explored the connection and communication between two microcontrollers, enabling them to work together effectively. With the help of a little self-learning, we added small LCD screens that improve the creation and display of shapes and pixels.

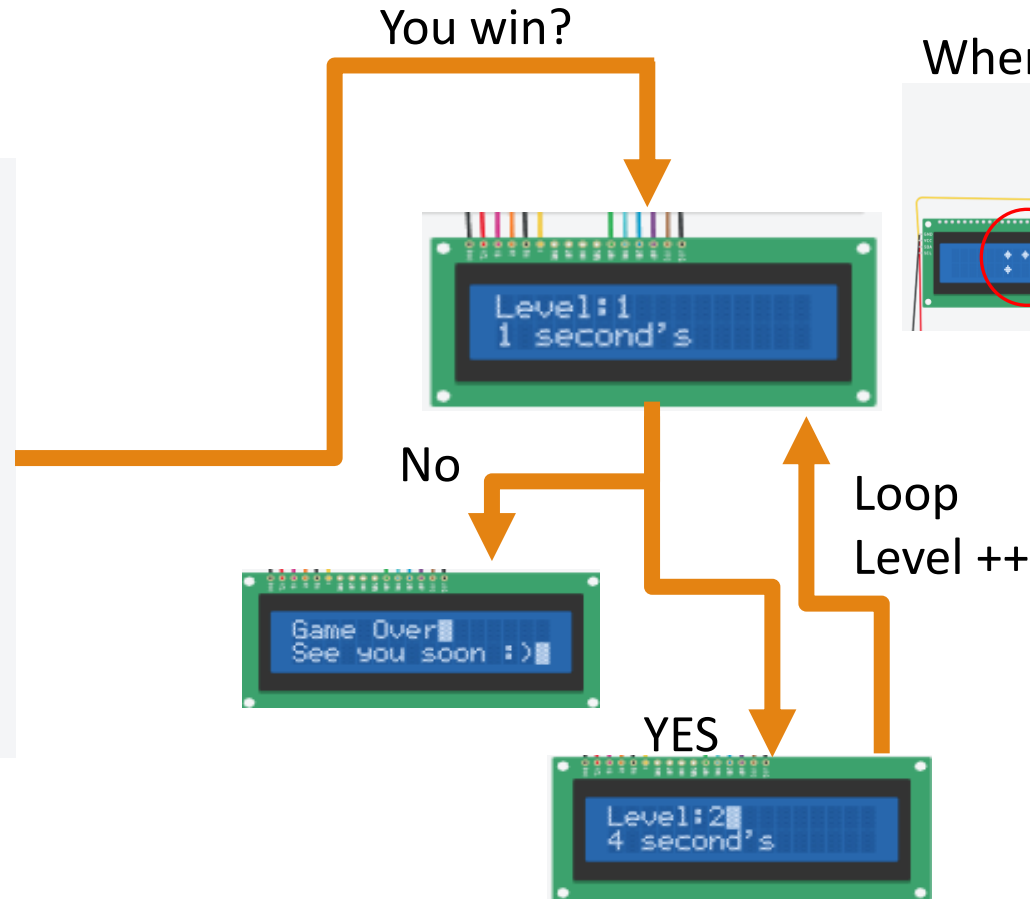
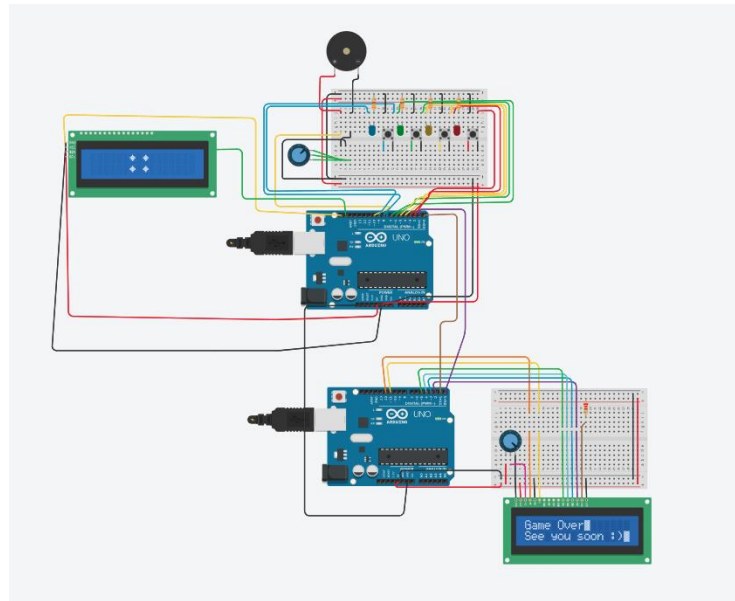
# System Design-Challenges and Learning Points:

At first we used different [tutorials](#) and examples from the net of built models of the game in different forms with the Arduino Uno

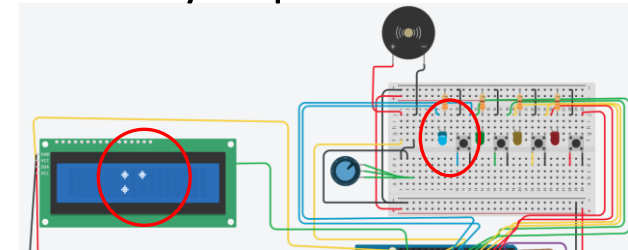
One of the first ones we used is SparkFun Inventor's Kit Experiment Guide



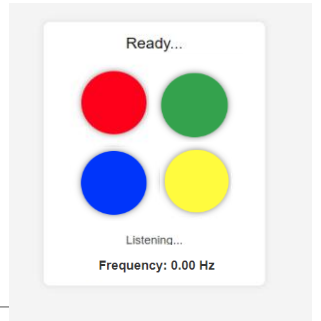
# ScreenShots – Tinkercad



When you press on button



# ScreenShots – From Node.js and UI

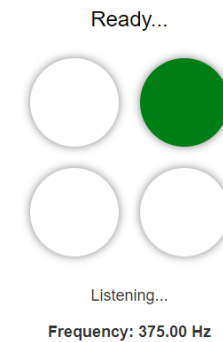
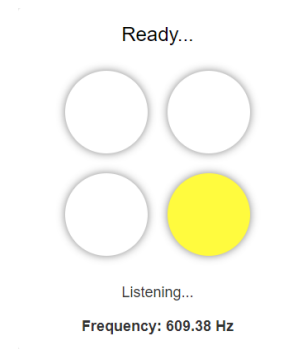
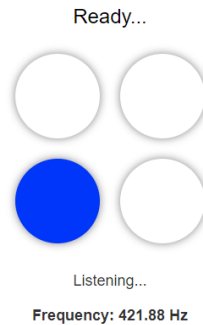
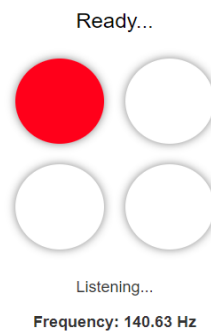


```
PS C:\Users\barex\Downloads\simon say> npm run dev
> simon-say@1.0.0 dev
> nodemon scrape.js

[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node scrape.js`
Server listening at http://localhost:4003
```

We used TONES so that UI would respond according to the pressed button

```
int tones[] = {262, 330, 392, 494}; // Tones for each button (c, e, g, b)
```





# Conclusions for further development

---

- Converting the product into a physical product.
- Creating a cloud-integrated interface for fast data communication.
- Adding additional players.



Click me to see intro

