

UNIVERSIDAD NACIONAL DE QUILMES

Seminario: Introducción a la Programación de
Microcontroladores con Tecnologías Libres

TRABAJO FINAL



Integrantes:

- Emiliana Romero
- Martin Kopecek

1er Cuatrimestre 2020

ÍNDICE

- Objetivo 3
- Jugabilidad 3
- Materiales 4
- Código Fuente 5
- Esquema Tinkercad 5
- Desarrollo: Soluciones e Inconvenientes 7

OBJETIVO

El objetivo del proyecto es el desarrollo, mediante el simulador Tinkercad de Arduino UNO, del famoso juego SNAKE. Basándonos en la jugabilidad del mismo correspondiente a los celulares Nokia.

JUGABILIDAD

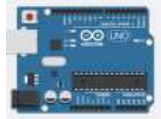


Al comenzar la simulación del emulador:

- Mediante el LCD 16x2 recibiremos la bienvenida al juego. Presionando el botón apartado START, podremos comenzar a jugarlo.
- Tenemos cuatro botones, con los cuales dirigimos el movimiento de la *viborita*.
- Al comer la luz random que se encuentra encendida, el cuerpo de la *viborita* se irá incrementando en uno, al igual que el score.
- El juego finaliza si tocamos el cuerpo de la *viborita* con la cabeza, o nos incrustamos contra algún margen.
- Al finalizar la partida obtendremos en el LCD el score obtenido, el cual denotará la cantidad de luces "*comidas*" durante la misma.

MATERIALES

Para lograr ensamblar el juego utilizamos los siguientes materiales:



- **Arduino Uno R3** Un Arduino UNO



- Una placa de pruebas



- 64 LEDs de color rojo que conforman la matriz de juego de 8x8



- 5 pulsadores: 1 para el START y 4 para la movilidad del snake



- 11 Resistencias de 220w



- Un potenciómetro



- Un LCD 16x2 para presentar el juego y el score final



- 2 registros de desplazamiento



- Un piezo

CÓDIGO FUENTE

A continuación, adjuntamos el código fuente utilizado:

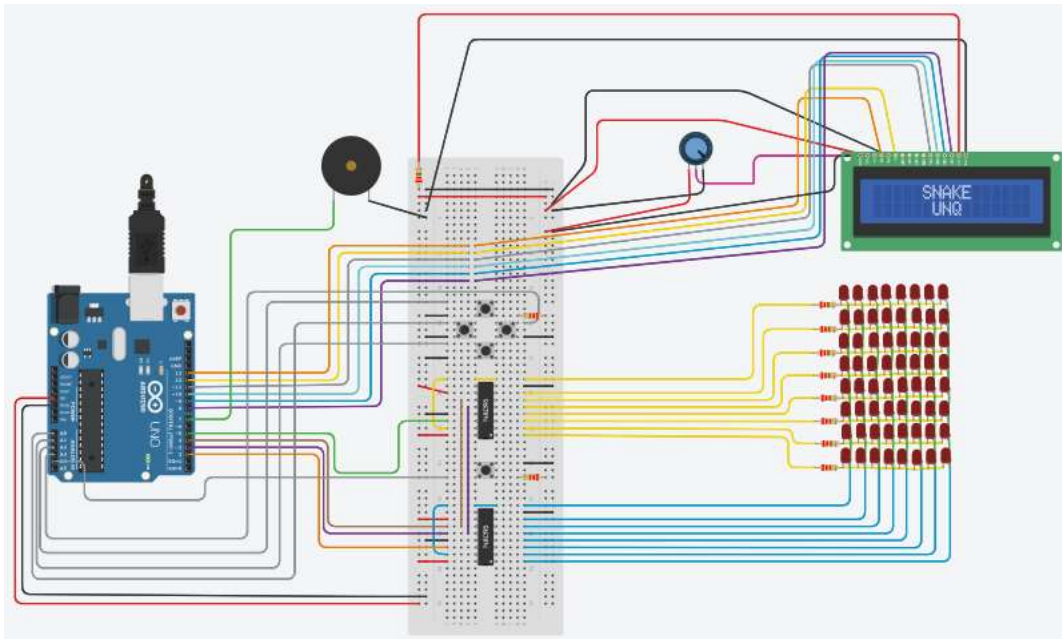


Codigo Fuente
Snake.txt

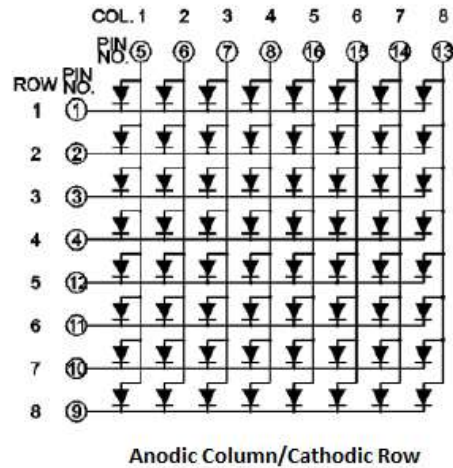
ESQUEMA TINKERCAD

A continuación, dejamos el link a Tinkercad donde se encuentra realizado el esquema de juego. El mismo según el emulador estará disponible durante las próximas 330hs, aunque lo hayamos puesto de forma pública:

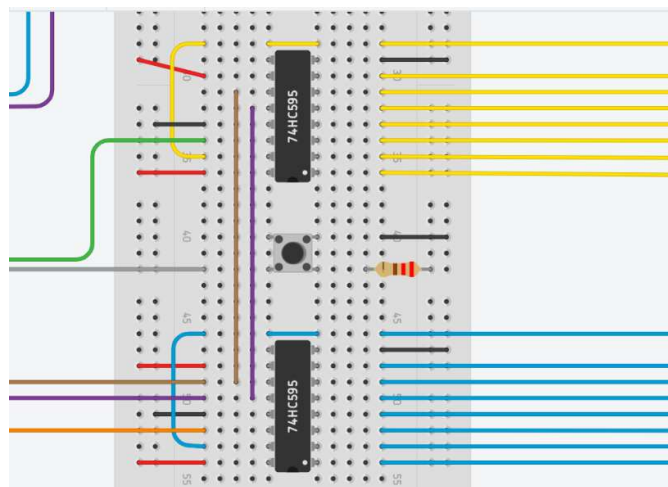
- <https://www.tinkercad.com/things/kEzilHVZEmR-snake-game/editel?sharecode=QfFZQ8zAUVL6mTKVBN7Fqy3uiNObxasC9Ja8U57kvIU>



- Se utilizó el esquema de conexión de **Common anode** para la matriz de leds.



- Para el manejo de la matriz utilizamos 2 registros. En uno conectaremos las columnas (anodos +), y en otro las filas (catodos -). Del registro donde conectamos las columnas, conectamos nuestro clock y reset hacia el Arduino, además del pin de dato; en cuanto al otro registro, clock y reset se conectarán al registro de columnas, y su pin de datos al Arduino. Por tanto, solo se usarán 4 pines del Arduino para toda la matriz.



DESARROLLO: Soluciones e Inconvenientes

Al comienzo imaginamos el juego de una manera distinta a la que terminamos desarrollando: pensamos usar solo botones, los cuales serían destinados al movimiento, pero al iniciar la simulación en las primeras pruebas notamos que necesitábamos uno más para poder iniciar el juego cada vez que quisiéramos, o que se perdiera una partida.

A su vez planteamos realizar una matriz de 10x10 LEDs RGB, aunque decidimos cambiar a una de 8x8 LEDs común, debido a que en principio de 10x10 quedaba demasiado grande, y al intentar mostrar la jugabilidad se nos complicaba con la demostración de las luces encendidas.

Tuvimos algunos inconvenientes que nos llevaron a replantear varias veces la forma de realizar el juego:

- ¿Como conectar la matriz?
 - Decidimos conectar los cátodos de cada luz por fila, y los ánodos por columnas. Nos dimos cuenta que hay dos formas de conectar la matriz y en base a eso su código y conexiones pueden variar.
- Una vez conectada, ¿Como hacemos para que funcione cada LED individualmente?
 - Indagando en ejemplos obtenidos por Jose, y tutoriales, logramos dar esta funcionalidad mediante el código desarrollado.
- Registros de desplazamiento
 - Nos encontramos frente a un problema al no poder hacer funcionar las conexiones, ya que como verán en el esquema utilizamos uno para las filas y otro para las columnas de LEDs. Decidimos vincular ambos, y luego darle la conexión individual al equipo Arduino para de esta manera solucionar el inconveniente.

Con respecto al desarrollo del esquema, y al código, nos fuimos basando en un principio sobre los ejemplos trabajados en clase, ensamblando uno a uno y probandolos individualmente. Además, hemos consultado tutoriales para poder comprender mejor a nivel código el manejo de una matriz con registros de desplazamientos, lo cual a nuestro parecer fue el mayor desafío.