Proyecto:

Simón dice



Alumna:Daiana Rivera

Compañeros de proyecto: Adrian Hidalgo, Leandro Delgado, Rodrigo Acevedo y Ramiro Averbuj

Curso: 4to 2da

Profesor:Gonzalo Consorti

Año: 2024

24/09/24

**Introducción:**

Este proyecto tiene como objetivo recrear el clásico juego de memoria conocido como “Simón dice”, o “Simon Says” en inglés, utilizando el programa Arduino como base para su funcionamiento. Comenzaremos diseñando el modelo en la plataforma en línea Tinkercad, que nos permite visualizar y probar el circuito antes de armarlo físicamente. Para lograr esto, necesitaremos varios componentes electrónicos, como un buzzer, LEDs de colores y botones, que conectaremos a una placa Arduino para darle vida al juego.

**Objetivo:**

El propósito principal de este proyecto es que el juego funcione como el “Simón dice”: el usuario deberá seguir una secuencia de luces que se va complicando con cada nivel. Queremos añadir niveles de dificultad progresiva para hacerlo más desafiante y que el puntaje del jugador, o “Score”, se muestre de manera clara en una pantalla LCD, para que pueda ver su progreso a medida que avanza en el juego.

**¿Qué debo saber para jugar al “Simón dice”?**

En el juego, los participantes deben prestar mucha atención a una secuencia de luces de diferentes colores y luego repetirla en el mismo orden presionando los botones correspondientes. Al iniciar la partida, el nivel de dificultad es bajo: las luces parpadean en un ritmo constante y fácil de seguir. Sin embargo, a medida que se progresa en el juego, las secuencias se vuelven más rápidas y complejas, haciendo que el desafío aumente y poniendo a prueba la memoria y la agilidad mental de los jugadores.

**Proceso de investigación:**

Hoy fue un día de investigación para todo el equipo, de este modo conseguiremos ideas para empezar nuestro proyecto del juego “Simon dice”. Exploramos múltiples fuentes, incluyendo videos de YouTube, documentos en formato PDF y sitios web especializados en proyectos de electrónica y Arduino. Yo fui la primera en enviar material para el proyecto, ya que encontré un video sobre cómo recrear el juego “Simón dice” usando Tinkercad. Aunque el video estaba en inglés, esto no fue un obstáculo para el equipo, ya que tanto Leandro Delgado como yo logramos comprender su contenido y pudimos usarlo como recurso. Este video nos dio una guía inicial sobre los pasos necesarios para diseñar el circuito y configurar los elementos principales en Tinkercad.

Además, mis otros compañeros contribuyeron con muestras de código y proyectos similares, en los que ya estaban establecidas las conexiones y la estructura del juego. Gracias a su aporte, obtuvimos una base sólida para comenzar a programar y entender mejor cómo conectar los componentes electrónicos necesarios. Con toda esta información, logramos avanzar en el proyecto con mayor seguridad y claridad sobre los siguientes pasos.

Esta etapa de investigación nos permitió obtener una visión más completa de cómo otras personas han recreado este juego clásico, y fue la clave para inspirarnos y planificar el nuestro.

En los videos que encontramos, observamos cómo los creadores programaban las secuencias de luces y sonidos, y notamos diferentes maneras de incrementar el nivel de dificultad a medida que el juego avanza. También vimos ejemplos de cómo los desarrolladores integraban una pantalla LCD para mostrar el puntaje, lo que nos dio ideas sobre cómo podríamos mejorar la experiencia visual de nuestro proyecto. Además, algunos creadores compartían trucos sobre la configuración de los LEDs y botones, así como la importancia de calibrar correctamente el buzzer para que los sonidos fueran claros y agradables, reforzando la experiencia interactiva.

Todo el material (links, videos, imágenes) encontrado durante este día de investigación fue guardado en un documento de google a parte donde solo pueden ingresar los miembros de nuestro equipo.

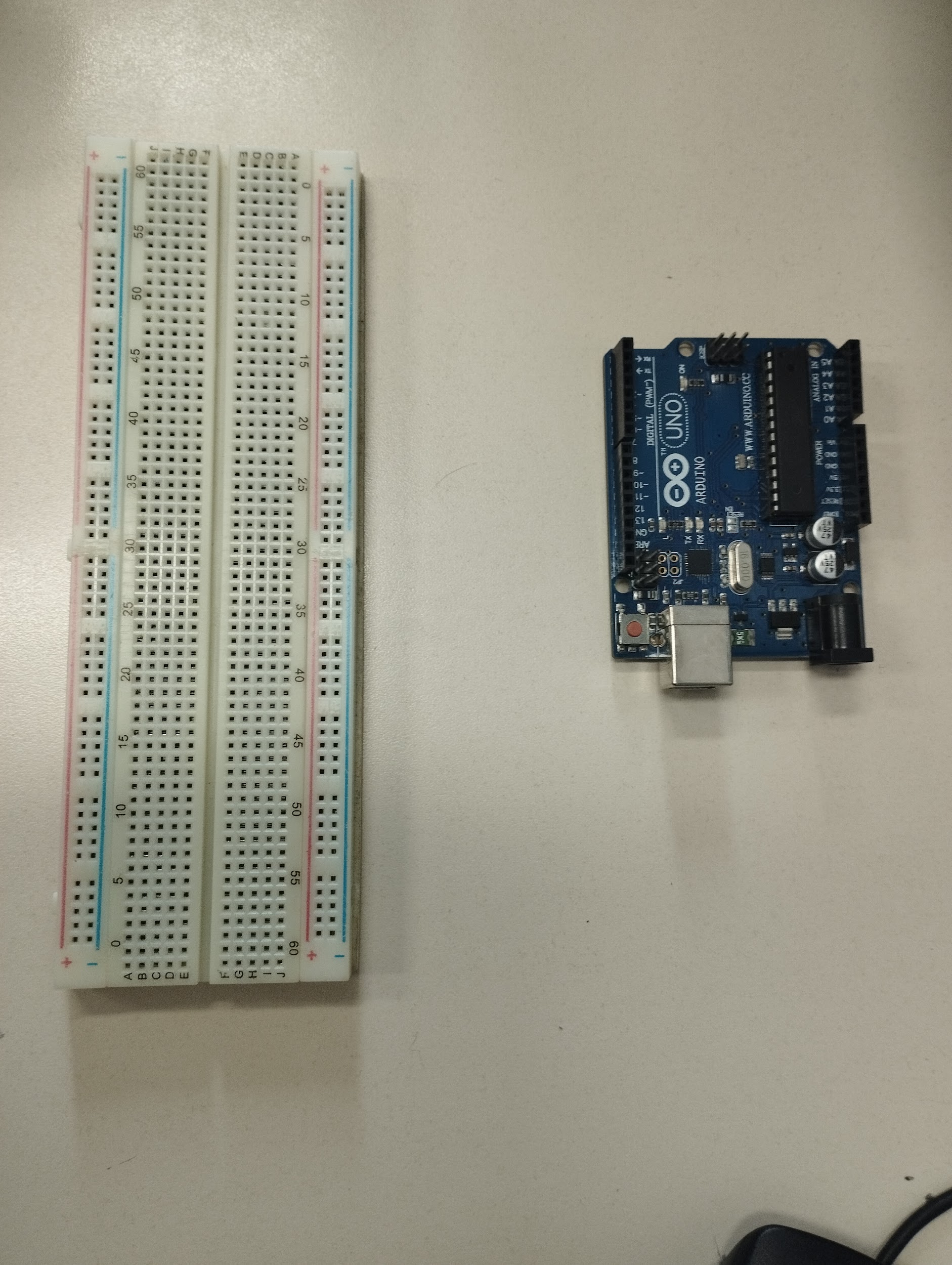
Con las ideas y el conocimiento adquirido, ya estamos listos para comenzar la fase de diseño y programación en Tinkercad, confiando en que estos recursos nos servirán para construir un “Simón dice” que sea funcional.

**Organización - Grupo de WhatsApp:**

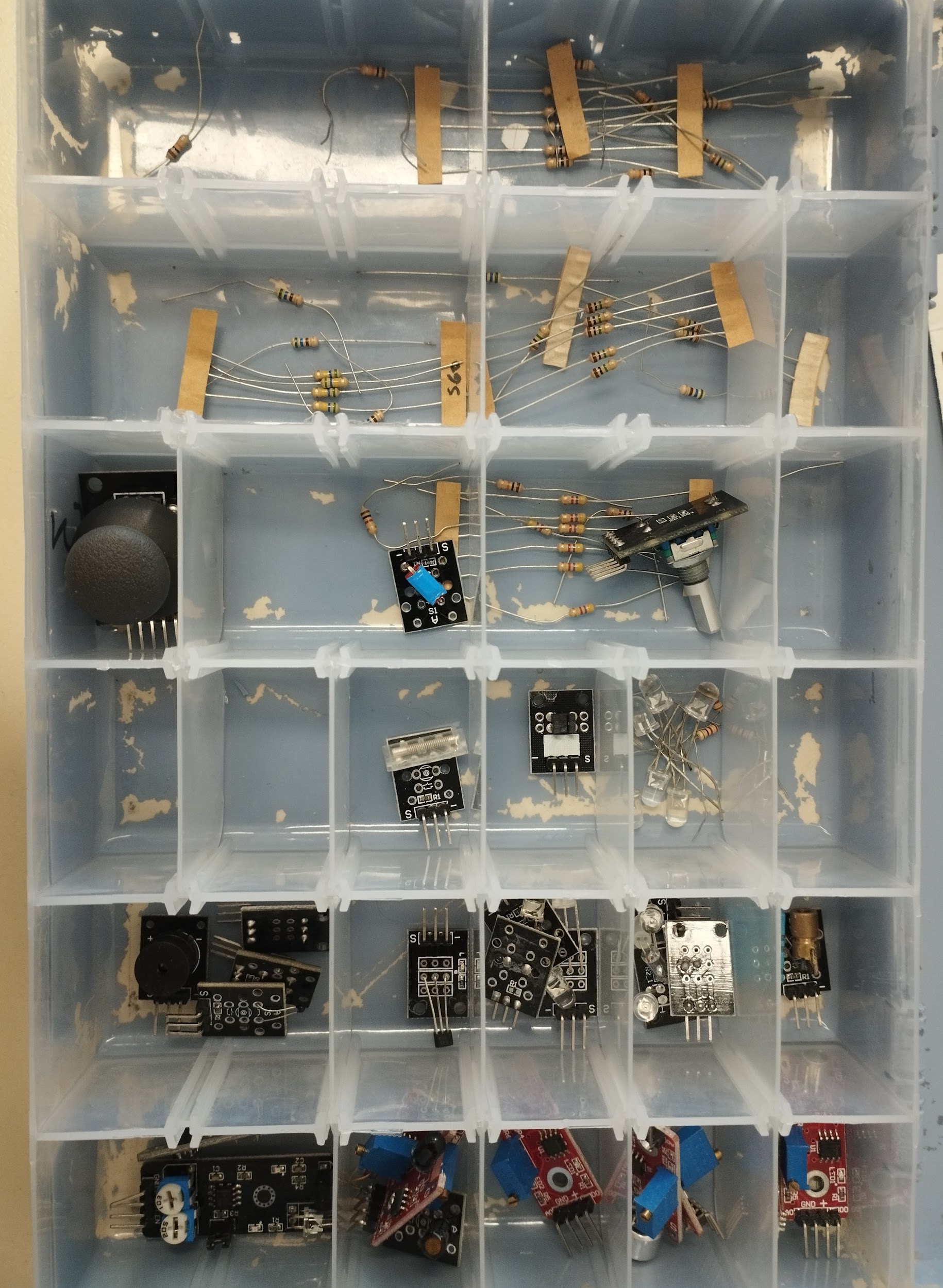
Para mantenernos organizados y asegurarnos de que todos los integrantes del equipo estemos en la misma página, mi compañero Rodrigo Acevedo tomó la iniciativa de crear un grupo de WhatsApp exclusivamente para este proyecto. Este grupo ha sido fundamental para compartir información de forma rápida y eficiente. En él, nos pasamos fotos relacionadas con el progreso del trabajo, intercambiamos ideas para completar la carpeta de campo y discutimos cómo mejorar el diseño y la presentación del proyecto. Además, utilizamos este espacio para hacer un seguimiento continuo del desarrollo del código y de los avances en el modelo 3D. El grupo también nos sirve como un medio de comunicación en tiempo real para resolver dudas o coordinar tareas pendientes, lo que ha sido clave para mantenernos alineados con los objetivos del proyecto. Gracias a esta herramienta, hemos logrado optimizar nuestro tiempo y esfuerzos, trabajando de manera más colaborativa y organizada.

8/10/24

**Entrega de cajas con los componentes:**

Esta mañana llegué a clases sin saber lo que iba a pasar; el profesor nos entregó dos cajas con diversos componentes para el proyecto. En la caja más pequeña recibimos la protoboard, una placa Arduino, y muchos cables necesarios para realizar todas las conexiones.(Estaba todo muy desordenado, tardamos un poquito en separar los materiales como LEDs y botones que estaban ahí).

Por otro lado, la caja más grande contenía variedad de componentes adicionales (los cuales algunos ni entendía que eran, o cual debería de ser su función), como resistencias, sensores y botones, que serían fundamentales para llevar a cabo las distintas funcionalidades del juego “Simón dice”. Por suerte todos los componentes que necesitamos estaban.



No llegamos a ordenar los componentes, solo a revisar si teníamos los que necesitábamos.

**Componentes para nuestro proyecto:**

Llegó el momento de mencionar las definiciones y el funcionamiento de los componentes que usaremos para este trabajo final:

**Buzzer**: El buzzer pasivo, también conocido como piezoeléctrico o simplemente piezo, es un componente electrónico que desempeña un papel crucial en la experiencia interactiva del juego "Simón dice". Su función principal es generar sonidos, lo que permite al usuario recibir retroalimentación sonora en tiempo real durante el desarrollo del juego. A diferencia de los buzzers activos, que emiten un tono constante al aplicarles voltaje, los buzzers pasivos necesitan una señal modulada para producir diferentes tonos, lo que los hace más versátiles para este tipo de proyectos.

**LEDs:** Los LEDs son clave para el juego, ya que representan la secuencia de colores que los jugadores deben memorizar. Se conectan a la placa Arduino y, mediante el código, se encienden y apagan en un patrón específico que los jugadores deben seguir. Me hubiera gustado poner LEDs de color azul, rojo, verde y amarillo, pero no contabamos con esos colores solo color blanco

**Botones:** Los botones permiten al jugador interactuar con el juego, ya que son presionados en el orden correcto para replicar la secuencia de luces que muestra el juego. Son la forma en que el jugador confirma la secuencia que recuerda.

**Potenciómetro:**  El potenciómetro es un componente esencial en el diseño de este proyecto, ya que permite ajustar el contraste de la pantalla LCD. Esto asegura que la información del juego, como las instrucciones o el puntaje, sea claramente visible para los jugadores en todo momento.El potenciómetro actúa como un divisor de voltaje, regulando la intensidad de la señal que controla la visibilidad de los caracteres en la pantalla. Girando su perilla, se puede aumentar o disminuir el contraste hasta alcanzar el nivel óptimo según las condiciones de iluminación del entorno. Este ajuste no solo mejora la experiencia visual del usuario, sino que también evita problemas como caracteres demasiado brillantes o apenas legibles, garantizando que todos los detalles del juego sean fácilmente comprensibles.

**Pantalla LCD:** La pantalla muestra detalles como el nivel actual o el puntaje del jugador. También proporciona mensajes visuales, mejorando la experiencia de juego al mostrar información relevante cuando el jugador gana o pierde. (Realmente pensé que esta pantalla físicamente en la vida real era más grande pero no, es pequeña, no medimos sus lados aun.)

**Lista de materiales:**

* 1 Protoboard
* 1 Placa arduino
* 1 Buzzer pasivo
* 4 LEDs
* 4 Botones
* 9 Resistencias
* 1 Pantalla LCD
* 1 Potenciómetro
* 17 Cables

Todos estos materiales son los que necesitamos si o si para lograr que nuestro proyecto funcione con éxito, aun nos faltan los botones pero seguro los conseguiremos cuando le preguntemos al profesor donde tenemos que comprarlos o adquirirlos.

15/10/2024

**Código por ahora (parte 1):**

| #include <LiquidCrystal.h>  #define L1 8 #define L2 9 #define L3 10 #define L4 11 #define B1 A0 #define B2 A1 #define B3 A2 #define B4 A3 #define BUZZER 12  LiquidCrystal lcd(2, 3, 4, 5, 6, 7);  int pinPulsador = A0; int valorPulsador = 0;  void setup(){   pinMode(L1 , OUTPUT);  pinMode(L2 , OUTPUT);  pinMode(L3 , OUTPUT);  pinMode(L4 , OUTPUT);  pinMode(BUZZER, OUTPUT); }  void loop(){  valorPulsador = analogRead(pinPulsador);  if(valorPulsador < 512){    // En construccion...  }   } |
| --- |

Luego de recibir los componentes, comenzamos a trabajar en el código para implementar el juego en Arduino. Instalamos y configuramos la librería LiquidCrystal para manejar la pantalla LCD en nuestro código, y establecimos las conexiones básicas, basándonos en las conexiones que encontramos en internet.

A continuación, se explicara parte del código inicial hasta el dia de la fecha (15/10):Este es solo el inicio de nuestro código. La idea es que las luces (LEDs) se enciendan en una secuencia aleatoria que el jugador debe repetir usando los botones. Cuando el jugador comete un error, el buzzer emite un sonido para indicar que falló, y el juego reinicia la secuencia. La pantalla LCD muestra el puntaje y el nivel actual, que aumentan con cada ronda completada con éxito. (Por ahora la pantalla LCD no emite nada, pero seguro que pronto si lo hará)

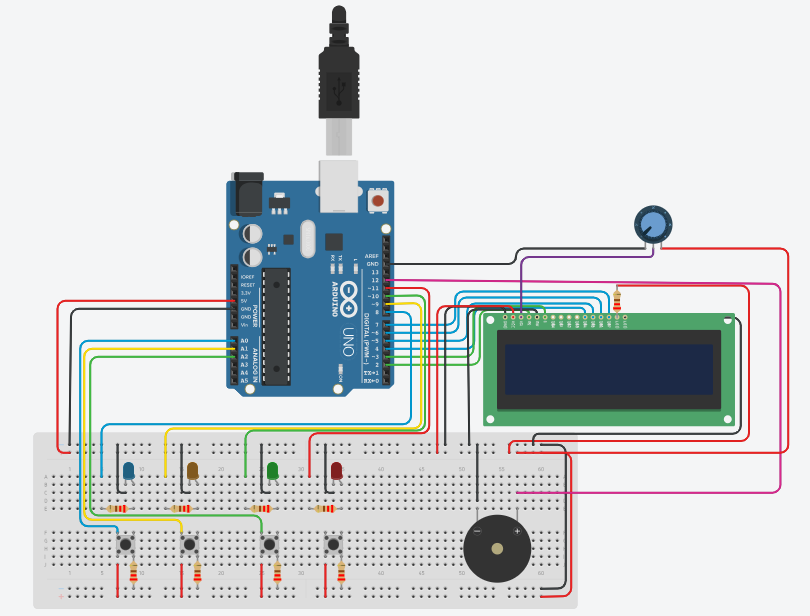
**Consejo del profesor:** El profesor nos sugirió algunas ideas para optimizar la codificación del juego. Nos recomendó usar entradas analógicas, la función map() y un potenciómetro para ajustar el contraste de la pantalla LCD, lo cual permitirá que el texto se vea claramente. Esta orientación nos ayudará a mejorar la experiencia visual del jugador y a facilitar el desarrollo del juego en general.

**¿Qué es el contraste?**

El contraste en una pantalla LCD (Liquid Crystal Display)es la relación entre la intensidad de la luz emitida por la pantalla cuando muestra colores oscuros y la intensidad de la luz emitida cuando muestra colores brillantes. Es fundamental que la pantalla LCD tenga un muy buen contraste y que esté correctamente configurada para la visualización de los datos y que estos sean más fáciles de distinguir

**Conexiones:**

Al final de la clase de hoy (15/10) ya teníamos hechas las conexiones del simon dice basándonos en todas las conexiones que hemos visto durante nuestra investigación.

****

Se han usado todos los materiales requeridos anterior mente mencionados los LEDs son: L1(azul pin 8), L2(amarillo pin 9), L3(verde pin 10), L4(rojo pin 11) y estan configurados como OUTPUT. El buzzer está conectado al pin 12 y también está configurado como OUTPUT.

Ahora me gustaría indagar en la librería que usamos en el código la cual hace posible la visualización del score en este proyecto.

**Investigación sobre la libreria liquidCrystal:**

La librería LiquidCrystal es una biblioteca de software que se puede encontrar en el programa Arduino y Tinkercad, diseñada para simplificar la interfaz con pantallas LCD (Liquid Crystal Display) basadas en el controlador Hitachi HD44780 o compatibles. Estas pantallas son muy comunes y se utilizan para mostrar texto en proyectos de Arduino, como menús, mensajes, instrucciones y cualquier tipo de información visual. En este caso la estaremos utilizando para que el jugador tenga el privilegio y acceso de ver por pantalla su puntaje.

**¿Qué es el controlador Hitachi HD44780?**

El controlador Hitachi HD44780 es un chip de control de pantalla LCD (Liquid Crystal Display) muy común, utilizado en la década de 1980 y comienzos de la década de 1990. Fue creado por una empresa japonesa llamada “Hitachi” y se convirtió en un estándar para la industria de los monitores LCD.

El HD44780 es un controlador de pantalla LCD que nos deja visualizar texto y gráficos en una pantalla LCD de 2 líneas de 16 caracteres cada una. Es un chip muy versátil que puede ser utilizado en una variedad de aplicaciones, desde relojes digitales hasta calculadoras y máquinas de vending (Es un dispositivo automático que vende productos, como bebidas, golosinas, snacks, etc., a cambio de moneda o tarjeta de crédito. ).

**Características de la librería LiquidCrystal:**

Compatibilidad: La librería es compatible con pantallas LCD de 16x2 o 20x4 caracteres, que son las configuraciones más comunes. La pantalla se controla utilizando 4 o 8 pines de datos, además de los pines de control.

Comandos básicos: Con la librería, puedes realizar operaciones como:

-Escribir texto.

-Desplazar texto hacia la izquierda o la derecha.

-Mover el cursor a diferentes posiciones.

-Limpiar la pantalla.

-Crear caracteres personalizados.

-Modo de 4 o 8 bits: La comunicación entre la pantalla y Arduino se puede hacer en dos modos:

-Modo de 4 bits: Utiliza solo 4 pines de datos (más dos pines de control), lo que ahorra pines digitales.

-Modo de 8 bits: Utiliza 8 pines de datos (más dos de control), proporcionando una comunicación más rápida, pero utilizando más pines.

**Facilidad de uso:**

La librería te permite mostrar texto con solo unas pocas líneas de código, sin la necesidad de conocer detalles complejos sobre la comunicación entre el Arduino y la pantalla LCD.

**Ventajas de usar pantallas LCD en proyectos de Arduino:**

Las pantallas LCD son herramientas esenciales en el desarrollo de proyectos con Arduino debido a su versatilidad y facilidad de uso. A continuación, se detallan algunas de sus principales ventajas:

1. Interactividad MejoradaLas pantallas LCD permiten mostrar información en tiempo real, como resultados de sensores, mensajes de error, menús de opciones, y estados del sistema. Esto mejora significativamente la experiencia del usuario, ya que le brinda un medio visual para interactuar con el proyecto de manera directa y clara.
2. Versatilidad y FlexibilidadSe pueden integrar fácilmente en diversos tipos de proyectos, desde sistemas de control y monitoreo hasta videojuegos interactivos. Además, están disponibles en varios tamaños y formatos, lo que las hace adecuadas tanto para proyectos compactos como para aplicaciones más avanzadas.
3. Ahorro de Pines del ArduinoLas pantallas LCD pueden configurarse en el modo de 4 bits, lo que reduce la cantidad de pines digitales necesarios en el Arduino. Esto es especialmente útil en proyectos donde se requieren múltiples conexiones para otros componentes, como sensores, botones o LEDs.
4. Personalización y Control VisualGracias a su capacidad de mostrar texto y números, las pantallas LCD permiten personalizar los mensajes según las necesidades del proyecto. Por ejemplo, en un juego como *Simón Dice*, pueden mostrar el nivel actual, las instrucciones o el puntaje del jugador, añadiendo un toque profesional al diseño.

**Código y Funciones Clave Utilizadas**

En el desarrollo del proyecto, es fundamental utilizar funciones específicas de Arduino para controlar tanto la pantalla LCD como otros componentes integrados. Algunas de las funciones más relevantes son:

1. **analogRead()** Esta función se utiliza para leer valores de pines analógicos, como un potenciómetro conectado al circuito. En este caso, el potenciómetro se emplea para ajustar el contraste de la pantalla LCD, permitiendo que el texto sea más legible según las condiciones de iluminación del entorno.

**Ejemplo:**

| int contraste = analogRead(A0); |
| --- |

1. **map()** La función map() es utilizada para convertir un rango de valores en otro que sea más adecuado para los componentes del proyecto. En el caso de la pantalla LCD, transforma los valores del potenciómetro (0-1023) al rango necesario (0-255) para controlar el contraste.

**Ejemplo:**

| int valorMapeado = map(contraste, 0, 1023, 0, 255); |
| --- |

1. **digitalWrite()** Esta función permite encender y apagar los LEDs en el orden y tiempo correcto según la secuencia del juego. A medida que los niveles avanzan, las luces parpadean más rápido, aumentando la dificultad y poniendo a prueba la memoria del jugador.

**Ejemplo:**

| digitalWrite(ledVerde, HIGH); delay(500); digitalWrite(ledVerde, LOW); |
| --- |

22/10/2024

Hoy, dos de mis compañeros de clase, quienes también forman parte de mi equipo, comenzaron a trabajar en los bocetos para nuestro proyecto de *Simón Dice*. Utilizaron el programa **Tinkercad** para diseñar modelos 3D que darán forma al dispositivo. Uno de los aspectos más interesantes de su propuesta fue la idea de incluir una pantalla LCD con un soporte similar al de los celulares. Sin embargo, se concentraron tanto en esta idea que no consideraron la posibilidad de pegar algunos componentes directamente con silicona caliente, algo que habría simplificado el diseño y la construcción del modelo físico.

Mientras debatíamos sobre los diseños, uno de mis compañeros explicó su visión de manera bastante detallada:

*“Básicamente, mi idea del modelo 3D es que funcione como una caja con una tapa que tenga algunos huecos estratégicamente ubicados. Estos huecos permitirán que sobresalgan los cuatro botones, los cuatro LEDs y la pantalla LCD. La pantalla se mantendría fija gracias a un soporte similar a los que se usan para sostener celulares. De esta manera, la caja y su tapa no solo protegerían el circuito interno, sino que también harían que el dispositivo sea funcional sin necesidad de un diseño excesivamente complejo.”*

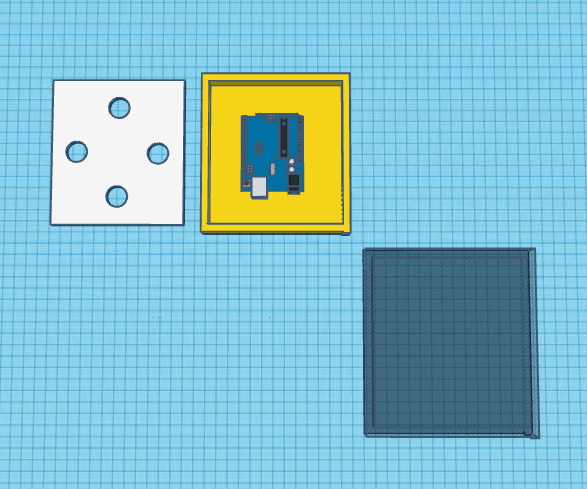
Aunque la propuesta sonaba muy interesante y bien pensada, noté algo importante que no habían considerado: si las medidas de los componentes físicos reales estaban correctamente reflejadas en el modelo 3D. Les pregunté si habían tomado en cuenta las dimensiones reales de cada componente y si las habían escalado adecuadamente en Tinkercad. Su respuesta fue un tanto preocupante, pero al mismo tiempo graciosa:

*“No, jajaja.”*

Por supuesto, este pequeño descuido significará que tendremos que dedicar tiempo en la próxima clase para medir cuidadosamente todos los componentes físicos: los botones, los LEDs, la pantalla LCD e incluso las dimensiones internas y externas de la caja. Solo así podremos asegurarnos de que el diseño 3D sea funcional y que todos los elementos encajen correctamente sin problemas.

Este tipo de situaciones nos recordó lo importante que es trabajar de manera organizada y considerar cada detalle desde el inicio. Aunque crear un modelo 3D puede ser emocionante, no sirve de mucho si no se basa en datos precisos. Ahora, con esta experiencia, hemos aprendido la importancia de combinar creatividad con planificación técnica, algo que será esencial para completar nuestro proyecto exitosamente.

**Este es el modelo 3D que creamos hoy:**



En el diseño, el soporte para el Arduino está representado en color amarillo. Esta pieza tiene un diseño sólido y práctico que asegura que la placa Arduino se mantenga fija y protegida dentro del dispositivo. A la derecha del soporte, se encuentra otra pieza de color blanco que incluye cuatro orificios estratégicamente ubicados. Estos orificios permitirán que los botones del juego sobresalgan de la carcasa, facilitando la interacción del usuario con el dispositivo. Además, hemos incorporado una tapa negra transparente que cubrirá todo el circuito, con el propósito de protegerlo del polvo y otros elementos externos que podrían afectar su funcionamiento.

La elección de los colores en el modelo no fue al azar; cada detalle del diseño fue pensado para combinar funcionalidad y estética. El color amarillo del soporte hace que el Arduino sea fácilmente identificable dentro del dispositivo, mientras que la tapa negra le da un toque profesional y moderno al conjunto. Además, al ser transparente, permitirá ver los LEDs iluminándose durante el juego, lo cual es un detalle visual muy llamativo y acorde con la temática interactiva de nuestro proyecto.

Mientras tanto, mis otros dos compañeros continuaban trabajando en el código del proyecto, realizando modificaciones importantes en las variables y añadiendo funciones clave para garantizar el correcto funcionamiento de todos los componentes electrónicos. Entre los ajustes más destacados, estuvieron afinando la programación de los LEDs, que se encenderán en secuencia según las reglas del juego. También implementaron funciones específicas para controlar el buzzer, que emitirá sonidos distintivos en cada nivel del juego, y la pantalla LCD, que mostrará información crucial como el puntaje del jugador y el nivel actual.

Este proceso de diseño y programación ha requerido mucho trabajo en equipo, ya que cada miembro del grupo aporta sus habilidades para completar diferentes partes del proyecto. Por un lado, el diseño del modelo 3D nos permitió visualizar cómo se verá el dispositivo una vez ensamblado y prever posibles ajustes antes de pasar a la etapa física. Por otro lado, el avance en la programación asegura que cada componente esté correctamente integrado y funcione como se espera.

Aunque aún tenemos detalles por pulir, como verificar que las dimensiones del modelo 3D coincidan con las medidas reales de los componentes físicos y optimizar el código para hacerlo más eficiente, el progreso de hoy ha sido significativo. La combinación de creatividad, diseño técnico y desarrollo de software nos está acercando cada vez más a nuestro objetivo: crear un *Simón Dice* completamente funcional, interactivo y atractivo tanto en su diseño como en su ejecución.

En las próximas clases planeamos unir ambos frentes de trabajo: ajustar el modelo 3D basado en los avances del código y asegurarnos de que todos los elementos encajen perfectamente, tanto física como lógicamente. Estoy muy emocionada por ver cómo se encenderán los leds y como todos estos elementos se unen para dar vida a nuestro proyecto final.

5/11/24

**Código por ahora (parte 2):**

Luego de recibir los componentes y trabajar en las primeras implementaciones del código, hemos logrado avanzar considerablemente en la estructura y funcionalidad del juego. A continuación, se describe detalladamente lo que ha cambiado y lo que se ha mantenido desde la versión inicial del 15/10 hasta la última actualización(5/11) :

#### **Aspectos que permanecieron igual**

Librería LiquidCrystal:  
Seguimos utilizando la librería LiquidCrystal para manejar la pantalla LCD. Aunque se modificaron los pines de conexión, la funcionalidad básica sigue en pie. Esto refleja la importancia de mantener una herramienta robusta para mostrar información relevante como el nivel y el tiempo.

Idea central del juego:  
El propósito del juego no cambió: los LEDs se iluminan en una secuencia aleatoria, y el jugador debe replicarla usando los botones. El buzzer sigue indicando errores, y la pantalla LCD muestra el nivel del jugador.

Uso de pines:  
Aunque los números de pines cambiaron, seguimos asignando entradas y salidas específicas para manejar los LEDs, botones y el buzzer.

**Cambios y mejoras en el código**

Reorganización y definición de pines:

Se optimizó el manejo de pines asignándolos a constantes claras, mejorando la legibilidad y escalabilidad del código. Además, los pines de entrada y salida ahora están organizados por letras (A, B, C, D), lo que permite identificar fácilmente las conexiones.  
 Línea modificada:

| #define BUZZER 7 #define ENTRADA\_A A0 #define ENTRADA\_B A1 #define ENTRADA\_C A2 #define ENTRADA\_D A3 #define SALIDA\_A 2 #define SALIDA\_B 3 #define SALIDA\_C 4 #define SALIDA\_D 5 |
| --- |

Nueva estructura para manejar la secuencia y niveles:

Ahora el juego cuenta con dos arreglos: secuencia y secuenciaUsuario, que almacenan la secuencia generada por el juego y la ingresada por el usuario, respectivamente. Se introdujo la variable nivelActual para controlar el progreso del jugador y velocidad para ajustar la dificultad.

Líneas agregadas:

| int nivelActual = 0; int velocidad = 500; const int NIVEL\_MAX = 50; int secuencia[NIVEL\_MAX]; int secuenciaUsuario[NIVEL\_MAX]; |
| --- |

Función:

**generaSecuencia:** Esta función genera una secuencia aleatoria que el jugador debe seguir, utilizando randomSeed y random. Esto garantiza una experiencia única en cada partida.  
  
 Nueva función:

| void generaSecuencia() {  randomSeed(millis());  for (int i = 0; i < NIVEL\_MAX; i++) {  secuencia[i] = random(2, 6);  } } |
| --- |

Funciones para manejar la interacción:

**muestraSecuencia:** Muestra la secuencia generada en los LEDs y emite tonos específicos según el LED encendido.

**leeSecuencia:** Compara la secuencia ingresada por el usuario con la generada por el juego. Si el usuario comete un error, llama a la función secuenciaError; si acierta, llama a secuenciaCorrecta.

Fragmento de **muestraSecuencia:**

| void muestraSecuencia() {  for (int i = 0; i < nivelActual; i++) {  digitalWrite(secuencia[i], HIGH);  delay(velocidad);  digitalWrite(secuencia[i], LOW);  delay(200);  } } |
| --- |

Fragmento de **leeSecuencia:**

| void leeSecuencia() {  for (int i = 0; i < nivelActual; i++) {  if (secuenciaUsuario[i] != secuencia[i]) {  secuenciaError();  return;  }  }  secuenciaCorrecta(); } |
| --- |

Indicaciones de error y acierto:

**secuenciaError:** Enciende todos los LEDs, reproduce una melodía de error y reinicia el nivel.

**secuenciaCorrecta:** Incrementa el nivel y reduce la velocidad, mostrando el nuevo nivel en la pantalla LCD.

Nueva función:

| void secuenciaError() {  digitalWrite(SALIDA\_A, HIGH);  delay(250);  melodiaError();  nivelActual = 0; } |
| --- |

Tiempo (función experimental): Se intentó agregar una función para mostrar el tiempo transcurrido en la pantalla LCD, aunque requiere mejoras para ser funcional.  
  
 Línea agregada:

| lcd.print(millis() / 1000); |
| --- |

El código ha evolucionado de un esquema inicial a una estructura robusta, con funciones específicas para manejar las secuencias, los niveles, y la interacción del jugador. Se han añadido arreglos, funciones y nuevas configuraciones de hardware que permiten integrar todos los componentes del juego. Aún quedan aspectos por perfeccionar, como la funcionalidad del tiempo, pero el avance logrado demuestra un gran progreso en la implementación del juego.

### **Mejoras Pendientes**

Funcionalidad de **tiempo**:

Es necesario mejorar el método para que registre y muestre el tiempo correctamente en la pantalla LCD.

Niveles y Velocidad:

A medida que el nivel aumenta, la velocidad se reduce, lo cual aumenta la dificultad del juego, pero puede necesitar ajuste para que la dificultad sea progresiva.

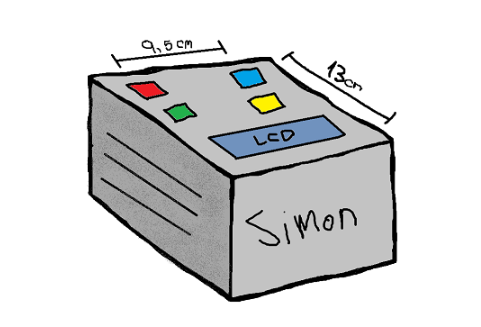
Este código ya cuenta con una estructura sólida, utilizando funciones para dividir tareas y mejorar la legibilidad, aunque aún existen detalles a mejorar para que funcione completamente como se desea.

**Charla con el profesor sobre el modelo 3D:**Hoy, enfrentamos un gran desafío relacionado con la mejora del modelo 3D de nuestro proyecto. Mis compañeros estaban muy frustrados porque, aunque sabían que el modelo necesitaba ajustes importantes, no tenían una idea clara de cómo realizarlos. Uno de los principales problemas era cómo crear un hueco o dejar un espacio adecuado para que se pudiera conectar un cable USB al Arduino y asegurar que el dispositivo funcionará correctamente. En ese momento, recordé que nuestro profesor había compartido previamente nos había enseñado otro juego con un diseño básico que podría servirnos como inspiración.

El diseño que nos mostró consistía en una especie de caja con dos botones en la parte superior y una pantalla LCD, que representaba una simulación del clásico juego "PONG". Este modelo era sencillo pero funcional y, sobre todo, adaptable a nuestras necesidades. Inmediatamente le pregunté al profe si me podía enseñar nuevamente ese jeugo y me dejo, le tomé una foto y la compartí en el grupo de WhatsApp para que todos pudieran verla y basarse en ese diseño para crear el nuestro.

Mi compañero Adrián Hidalgo, con su creatividad, decidió usar el programa Paint en Windows para crear un boceto basado en la foto que mandé por whatsapp. Mi compañero Hidalgo logró hacer un muy lindo boceto del juego, pero lo tiene que modificar aún para que quede mejor, las medidas van a ser de 9,5 cm de ancho × 13 cm de largo.

El boceto de mi compañero es este:



Este boceto sirvió como un punto de partida visual para entender cómo podría verse nuestro modelo final. Personalmente, me gustó mucho el diseño que propuso, ya que lograba combinar funcionalidad y estética de una manera efectiva. Sin embargo, aún quedaba por resolver la cuestión de las dimensiones reales de las piezas. Era fundamental medirlas con precisión para garantizar que todas las partes encajaran correctamente en el modelo final.

Gracias a esta colaboración y al apoyo del profesor, logramos superar un obstáculo importante. Este proceso nos ayudó a reafirmar la importancia de buscar referencias, trabajar en equipo y utilizar las herramientas que tenemos a nuestra disposición para avanzar en el proyecto de manera más eficiente.

12/11/24

**Código por ahora (parte 3):**

Nuevas Funcionalidades y Cambios Introducidos:

1. Librería **Entropy**:

Se agrega esta librería para mejorar la generación de números aleatorios usando Entropy.random() como semilla para randomSeed.

| #include <Entropy.h> Entropy.initialize(); int semilla = Entropy.random(); randomSeed(semilla); |
| --- |

Esto mejora la aleatoriedad en la generación de secuencias.

1. Variable **tiempoInicio**:

Se agrega para rastrear el tiempo transcurrido desde un punto inicial, utilizada en nuevas funcionalidades.  
unsigned long tiempoInicio;

1. Función **muestraTiempo**:

Nueva función que muestra en la pantalla LCD el tiempo transcurrido desde el inicio.

| void muestraTiempo() {  lcd.setCursor(0, 0);  lcd.print("Time: ");  lcd.print((millis() - tiempoInicio) / 1000); //muestra segundos  lcd.print(" segs"); } |
| --- |

1. Reinicio del Tiempo en Errores:

En secuenciaError, ahora se reinicia el tiempo cuando se detecta un error.

| tiempoInicio = millis(); //reinicia el tiempo al fallar |
| --- |

1. Código Completamente Comentado:

Se comenta gran parte del contenido del setup y loop original, probablemente para pruebas o simplificación temporal.

#### **Código Modificado o Agregado (Líneas Cambiadas):** A continuación, solo las líneas afectadas:

| #include <Entropy.h> //nueva libreria agregada  unsigned long tiempoInicio; //nueva variable global  void setup() {  Serial.begin(9600);  Entropy.initialize();  int semilla = Entropy.random(); //uso de Entropy para generar la semilla  randomSeed(semilla);  Serial.print("Semilla: ");  Serial.println(semilla);  //codigo comentado temporalmente }  void secuenciaError() {  tiempoInicio = millis(); //reinicia el tiempo al fallar  nivelActual = 0;  lcd.clear();  lcd.setCursor(0,1);  lcd.print("Nivel: ");  lcd.print(nivelActual);  delay(200); }  void muestraTiempo() {   lcd.setCursor(0, 0);  lcd.print("Time: ");  lcd.print((millis() - tiempoInicio) / 1000); //calcula y muestra los segundos transcurridos  lcd.print(" segs"); } |
| --- |

#### **Resumen de los Avances:**

* Se introdujo una mejora significativa en la generación de números aleatorios con la librería Entropy.
* Se agregó una nueva funcionalidad para medir y mostrar el tiempo transcurrido en pantalla (muestraTiempo).
* El código fue parcialmente simplificado, probablemente para pruebas o depuración, comentando secciones que aún no se utilizan.
* El manejo de errores ahora incluye un reinicio del cronómetro, haciéndolo más robusto.

Este avance demuestra una transición hacia un código mejor construido, con mayor foco en la interacción del usuario mediante tiempos y mejoras en la calidad del juego al usar una semilla de aleatoriedad mejorada.

Bueno, ahora voy a hablarles sobre la librería Entropy la cual nos ayudará mucho en este proyecto.

**Investigación sobre la libreria liquidCrystal:**

La biblioteca **Entropy** es una herramienta para usar con Arduino que ayuda a generar números aleatorios de verdad. Estos números no son como los de una calculadora (que son predecibles), sino completamente impredecibles, como lanzar una moneda o tirar dados.

### 

### ¿Cómo funciona?

Cuando le pides a tu Arduino que use la biblioteca***Entropy***, esta aprovecha pequeñas variaciones en el funcionamiento interno de la placa (como el tiempo que le lleva hacer algo) para crear números al azar. Es como si la placa estuviera lanzando muchos dados invisibles y usara los resultados para darte un número.

### ¿Para qué sirve?

* Si estás haciendo un juego y necesitas algo impredecible, como el lanzamiento de un dado.
* Para seguridad, como contraseñas aleatorias.
* Simulaciones donde necesitas que los resultados cambien cada vez que pruebas.

### ¿Qué tiene de especial?

La mayoría de los números "aleatorios" que usamos en una computadora no son realmente aleatorios. Son generados siguiendo un patrón escondido. Entropy utiliza el hardware del Arduino para obtener números que son más cercanos a la verdadera aleatoriedad.

### Ventajas

* Es fácil de usar, solo necesitas unas pocas líneas de código.
* Funciona bien para juegos, simulaciones y proyectos de Arduino donde quieras algo impredecible.

**Cómo estuvo la clase hoy dia (12/11)**

Esta semana la mayoría de todos los grupos, ya estaban ensamblando sus proyectos y algunos incluso tenían sus prototipos funcionando no se si del todo bien pero no juzgare ya que ni el mio esta armado. Esto, honestamente, me generó cierta presión. Sentía que nuestro equipo debía avanzar más rápido para no quedarnos atrás y para no entregar todo a última hora. Aunque estábamos haciendo lo mejor posible, los contratiempos no tardaron en aparecer.

El primer obstáculo surgió cuando intentamos ensamblar el Arduino. Para nuestra sorpresa, necesitábamos nueve resistencias, pero solo encontramos cuatro en nuestra caja de materiales, eso fue muy raro ya que la primera vez que nos dieron las cajas habían suficientes resistencias. Fue un momento de tensión, hasta que un compañero decidió revisar y consultar a los otros equipos y ver las otras cajas para encontrar más resistencias. Afortunadamente, logramos reunir las nueve resistencias necesarias.

Mientras tanto, Acevedo se concentró en replantear su modelo 3D . Se dio cuenta de que necesitaba modificarlo porque los botones que estábamos usando eran demasiado pequeños y no encajaban correctamente y eran muy incómodos de usar por lo diminutos que estos eran. Esto nos hizo reflexionar sobre la importancia de ajustar los detalles físicos para que todo funcione en armonía y para que no quede desalineado.

Por otro lado, nos enfrentamos a otro desafío: no teníamos una pantalla LCD, una pieza sumamente esencial para el proyecto ya que por esta pantalla vamos a mostrar el “Score”. Sin embargo, junto con mi compañero Hidalgo, conseguimos una de otro grupo que estaba trabajando en un proyecto llamado "el jardín inteligente". Fue un momento gracioso, ya que ellos tenían dos pantallas y amablemente nos dieron una, espero que les vaya bien con su proyecto sinceramente.

Alrededor de las 10:00 a.m., por pedido de mis compañeros el profesor se acercó a nuestro grupo para evaluar nuestro progreso. Nos sugirió que usáramos un buzzer pasivo en lugar del que habíamos seleccionado inicialmente, ya que era más adecuado para nuestro circuito y codigo. Justo cuando pensábamos que las cosas no podían mejorar, el profesor apareció con un juego de botones mucho mejores: eran más grandes, de mejor calidad, ¡y además tenían colores! colores que iban bien con la temática de nuestro juego (rojo, azul, amarrillo, verde). Esto nos emocionó bastante, ya que estos botones encajarían mejor en nuestro modelo 3D y facilitarían la interacción con el dispositivo, estoy muy feliz.

Con los nuevos materiales en mano, nos pusimos a ajustar todo en la placa Arduino para que coincidiera perfectamente con el diseño de las conexiones en tinkercad. Fue una jornada intensa, llena de desafíos, pero también de aprendizajes y trabajo en equipo. Al final del día, aunque aún queda mucho por hacer, sentimos que estamos un paso más cerca de lograr un proyecto funcional y bien diseñado.

19/11/24

**Medición y avances en el modelo 3D**

Hoy, mis compañeros y yo nos dedicamos a medir todas las piezas necesarias para realizar el modelo 3D, basándonos en el boceto que mi compañero Hidalgo había preparado y diseñado anteriormente. Este boceto nos sirvió como guía para asegurarnos de que las medidas fueran precisas y todos los componentes encajaran correctamente en el diseño final.

Junto con mi compañero Acevedo, realizamos las mediciones de los componentes. Trabajamos en equipo para garantizar que no se nos escapara ningún detalle. Después, compartí las medidas en el grupo de WhatsApp para que todos tuvieran acceso rápido a esta información y pudieran agregarlas en su carpeta de campo. A continuación, detallo las medidas que tomamos:

* ***Protoboard:***16x5,8 cm; Largo: 3,5 cm.
* ***Placa Arduino:*** 7,7x5,3 cm; Largo: 13 mm.
* ***Pantalla LCD:*** 8x3,5 cm; Largo: 2 cm.

**Nota personal:**Me siento bastante cansada, pero estoy contenta con los avances que logramos mis compañeros. Sin embargo, prefiero trabajar en la carpeta de campo en casa porque aquí, con tanto ruido y distracciones, me cuesta concentrarme y mucho.

26/11/24

**Última semana: cerrando el proyecto**Esta es la semana más crucial de todo el proceso. Era momento de cerrar todos los detalles, desde la presentación hasta el circuito y los ajustes finales en el diseño 3D. El ambiente estaba cargado de nervios y ruido porque sabíamos que todo debía quedar perfecto para la presentación final de la proxima semana. Mientras tanto, otros compañeros estuvieron trabajando en el código. Decidieron rehacerlo porque el profesor les indicó que era necesario cambiar las posiciones de los botones y luces para optimizar el diseño y su funcionamiento. Este ajuste fue un poco complicado, pero entendimos que era esencial para mejorar el proyecto.

### **Reflexión sobre el proyecto**

A lo largo de esta semana, enfrentamos varios desafíos. Uno de los problemas más significativos fue que el buzzer pasivo no funcionaba correctamente, lo que nos hizo invertir tiempo en buscar soluciones. Sin embargo, nos organizamos como equipo para hacer más eficaz el trabajo de buscar soluciones y al mismo tiempo no retrasarnos con lo demás pendiente, compartiendo ideas y distribuyendo las tareas para avanzar lo más rápido posible. Por ejemplo, cuando el profesor nos sugirió usar un buzzer pasivo, rápidamente adaptamos nuestro enfoque.

También hubo momentos difíciles, como cuando se fue el internet, lo que nos dejó sin acceso a recursos en línea y justo en ese momento los chicos estaban por probar el arduino. Sin embargo, el profesor aprovechó ese tiempo para conversar con todos incluso los otros grupos. El profesor nos dijo cosas para tener en cuenta para la presentación final. Tenemos que incluir y hablar sobre estos temas :

* Por qué elegimos este proyecto.
* Los problemas que enfrentamos.
* Cómo desarrollamos cada etapa.
* Los cambios que hicimos en las funciones y la organización.
* Una descripción del producto final.

En general, aunque fue una mañana intensa, el trabajo en equipo y la organización fueron clave para superar los retos y lograr que el proyecto tomara forma. Nos sentimos orgullosos del esfuerzo, del buen trabajo en equipo y estamos listos para presentar nuestro trabajo con confianza. Espero que en la presentación de mañana 3/12/24 nos vaya genial, y que seamos capaces de demostrar todo lo trabajado en este proyecto.

28/11/24

**No es martes pero si jueves**

El 28 de noviembre no fue martes, pero fue un día en el que hubo avances. Mi compañero Ramiro se quedó este Jueves 28/11 en la escuela para continuar trabajando en los detalles que quedaban pendientes. Con mucha dedicación, logró hacer funcionar el código correctamente, algo que fue una gran satisfacción para todos. Aunque el buzzer pasivo sigue sin funcionar correctamente, los LEDs y los botones respondieron perfectamente por suerte. Este avance nos dio la motivación necesaria para seguir puliendo los últimos aspectos como la presentación y la preparación de nuestros diálogos para esta.

Ramiro incluso nos mandó un video por whatsapp donde se veía que el código y el circuito corrían sin problemas. Fue emocionante ver el fruto de nuestro trabajo funcionando. Sin embargo, por cuestiones de tiempo, no alcanzamos a imprimir el diseño 3D que habíamos preparado, pero no pasa nada ya que el profesor dijo que no nos preocupamos ya que posiblemente el diseño no iba a estar impreso para el dia de la presentación ya que hay fila para utilizar la impresora 3D de la escuela.

Uno de los mayores retos ahora es probar los 50 niveles de nuestra simulación del juego "Simón Dice." Este es un desafío que exige paciencia y mucha concentración, ya que queremos confirmar si la melodía final de victoria se reproduce correctamente al completar todos los niveles, pero a la vez tememos confundirnos y tener que volver a pasar todos los niveles. Esta melodía sería el broche de oro para cerrar nuestra experiencia interactiva, y estamos ansiosos por ver si todo funciona como planeamos. Ninguno de nosotros tiene tiempo para hacer eso, así que le pediré ayuda a un familiar para que pase los niveles, espero que no se vaya a estresar jaja.

A continuación, compartiré los testimonios de mis compañeros sobre su trabajo, junto con mi experiencia y aportes al proyecto.

### **Mi contribución - Daiana:**

### A lo largo del proyecto, me encargué de organizar a mis compañeros, fui responsable de recolectar y estructurar los testimonios, de cada uno para incluirlos en nuestra presentación, que diseñe usando Canva. Fue un desafío creativo porque quería que la presentación reflejara no solo el esfuerzo técnico sino también el trabajo en equipo y la dedicación de todos.

### También medí todas las piezas físicas del circuito, un trabajo que requirió precisión para que los diseños 3D fueran exactos. Mi objetivo principal era mantenernos enfocados en avanzar y no dejar las tareas para último momento. Creo que mi esfuerzo por mantener la motivación en el grupo ayudó a que todo siguiera su curso, incluso cuando enfrentamos algunos contratiempos.

### **El trabajo de mis compañeros**

Ramiro:"Mi principal tarea fue buscar información sobre cómo integrar el LCD al circuito. Esto era clave para que todo funcionara correctamente. También ayudé en la parte final del código, ya que al cambiar la estructura física del proyecto, tuvimos que ajustar aspectos como las luces. Además, participé en el ensamblaje físico del circuito y realicé varias pruebas para asegurarme de que todo estuviera bien. Incluso me quedé en la escuela hasta las 4 de la tarde para verificar que el código funcionara correctamente. ¡Y funcionó! 😁"

Rodrigo:"Comencé buscando ejemplos en línea para tener una referencia sólida que nos ayudara a estructurar el proyecto. Después, trabajé en armar un circuito de prueba en Tinkercad, lo cual fue muy útil para visualizar cómo debía conectarse todo. También utilicé la misma plataforma para crear el modelo 3D del diseño, ajustándolo constantemente según las necesidades que iban surgiendo. Fue un trabajo que requirió paciencia y creatividad, pero estoy muy satisfecho con el resultado."

Adrián:"Mi aporte principal fue realizar los bocetos iniciales del proyecto y colaborar en la escritura del código. Siempre traté de aportar ideas que mejoraran el diseño y de solucionar cualquier error que surgiera en el camino. Además, trabajé en conjunto con Acevedo para asegurarnos de que el diseño 3D reflejara nuestras ideas de manera precisa. Fue un proceso muy enriquecedor, lleno de aprendizaje."

Lean:"Trabajé junto con Ramiro para definir las características que queríamos incluir en el juego y luego me encargué de codificarlas. También colaboré en las conexiones físicas y en Tinkercad, asegurándonos de que todo funcionara en ambas plataformas. Fue un reto, pero al final logramos superar los problemas y hacer que el circuito funcionara."

2/12/24

**Última anotación**Esta es la última vez que escribiré en esta carpeta. Me gustó mucho hacerlo, ya que soy alguien que disfruta de escribir y tener las cosas organizadas. Hoy, mi compañero Leandro Delgado hizo los últimos retoques al código. Nos comentó que le costó mucho, incluso diría demasiado, arreglar el código, pero finalmente lo logró y pudo añadir la melodía de acierto. Estoy muy feliz por él y por su logro. Además, sin querer, modificó un poco la melodía de error. Sin embargo, asegura que no se escucha mal, así que decidimos dejarla así para no arriesgarnos a romper o arruinar el código.

**Presentación**Estamos sumamente ansiosos por la presentación de mañana. Nuestro plan es iniciar con la presentación de las diapositivas en Canva, luego cada uno de nosotros hablará sobre su experiencia y contestará las preguntas planteadas por el profesor:

* ¿Por qué elegimos este proyecto?
* ¿Qué problemas enfrentamos?
* ¿Cómo desarrollamos cada etapa?
* ¿Qué cambios hicimos en las funciones y en la organización?
* ¿Cómo es el producto final?

Finalmente, mostraremos nuestro circuito en funcionamiento. Sinceramente, espero aprobar, ya que no quiero tener que venir en diciembre. Todo mi grupo dio lo mejor de sí, y eso es algo que aprecio enormemente.

**Conclusión**Amé hacer este trabajo y tuve la suerte de estar en un equipo genial, donde todos trabajamos y aportamos lo mejor de nosotros. Esto se notó especialmente en mis compañeros Rodrigo Acevedo, Adrián Hidalgo, Leandro Delgado y Ramiro Averbuj, quienes pusieron muchísima dedicación en el proyecto. Gracias a este tipo de trabajos prácticos, aprendimos mucho más sobre cómo organizarnos y trabajar de forma autónoma. Ojalá que mañana todo salga bien para nosotros y también para los demás equipos, que realmente se merecen que les vaya bien.

**Referencias:**

**Videos de YouTube:**

[Narenthan Narenthan](https://www.youtube.com/@narenthannarenthan7869). (2020 dic 12). *HOW TO CREATE A SIMON SAYS GAME WITH AN LCD DISPLAY*[Video]. YouTube.<https://www.youtube.com/watch?v=I7odCmkVMqY>

[Robot UNO](https://www.youtube.com/@RobotUNO) (2020, octubre 28*). MINIJUEGO con ARDUINO para MEJORAR LA MEMORIA || SIMON DICE || [proyecto explicado paso a paso]*

[Video]. YouTube.<https://youtu.be/Kk6Hax4D6YI?si=H2Aollln8KKz7p5c>

JUAN JOSE. (5 MESES). *PROYECTOS CON ARDUINO - JUEGO SIMÓN DICE*

[Video]. YouTube.<https://youtu.be/VxUzrWZxYkU?si=00cxEw7u0xaMyzVL>

**Paginas webs:**

Arduino Forum. (2023). *Simon dice y pantalla LCD.*<https://forum.arduino.cc/t/simon-dice-y-pantalla-lcd/969762>

Robot Uno. (2021). *Mini juego con Arduino: Simon dice.*<https://robotuno.com/minijuego-con-arduino-simon-dice/>

Escuela Superior de Ingeniería de la URJC. (2019). *Juego Simon dice.*<https://blogs.etsii.urjc.es/dseytr/juego-simon-dice/>

**Inteligencias artificiales aplicadas:**

OpenAI. (2023)*. ChatGPT: Modelo conversacional de inteligencia artificial*.<https://chatgpt.com/>

Microsoft. (2023). *Copilot: Herramienta asistiva para desarrolladores*.<https://copilot.microsoft.com/chats/EDnbsYrLkWhenybJCTbSg>

Perplexity AI. (2023). *Plataforma de búsqueda y respuesta*.<https://www.perplexity.ai/>