

Sofía Valentina Amarilla

Grupo 6 - BMO

Especialidad en Computación, Escuela Técnica N°32 D.E. 14

4to 1ra: Proyecto Informático

Gonzalo Consorti

24 de Noviembre de 2023

Agradecimientos a:

Gonzalo Consorti, me enseñó que no soy buena lidiando con las fechas límites

O trabajando en grupo.

“*El Mono*”, básicamente me hizo todo el proyecto en el lado de diseño,

me salvó de que no tenga un 1 (espero).

El profe Olaso, que ayudó a perforar el display y se ofreció de ayuda para que funcione,

Mi papá, que me ayudó a comprar el display.

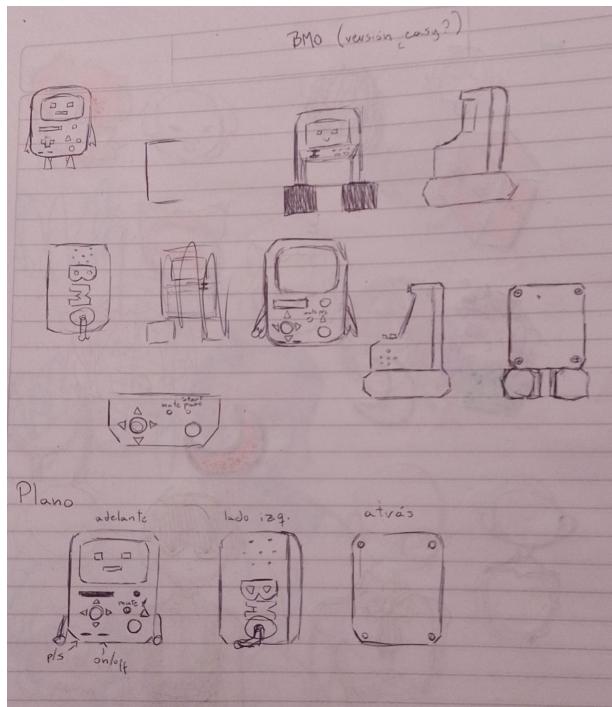
Mi mamá, que me lo pagó.

Y a *mí* por tener paciencia y no pegarme un corchazo.

Introducción

Desde que escuché que había una materia en la que podíamos crear un proyecto y del cual podríamos hacer lo que quisiéramos, no pude imaginar una mejor idea que crear un BMO, siendo que justo en el momento en el que el profesor nos lo mencionó, estaba haciendo una maratón de la famosísima serie “Hora de Aventura”.

Nos había mencionado que dependiendo de lo que se tenga planeado quizás deberían haber algunos ajustes en las funcionalidades, y que además, el límite de pines nos podría limitar y que tendríamos que tener muchas ideas. Pero como ya me había mentalizado de la creación de BMO, ideé estos conceptos.

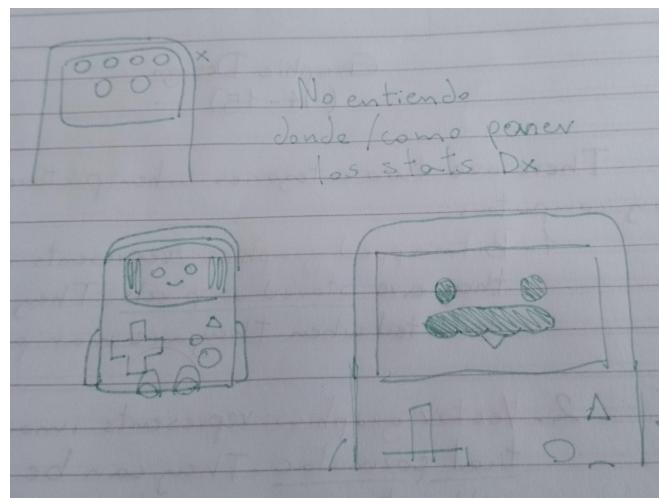


Algunos de mis conceptos que tengo en papel.

La idea principal era un BMO con el que el usuario pueda jugar los juegos clásicos, pero con una pequeña interacción con el BMO. Así que desde ese concepto desarrollé:

- BMO con rueditas: No tiene juegos, pero para compensar tiene rueditas.
- BMO con una mayor forma a un arcade.
- BMO con un buzzer: No tiene juegos pero de vez en cuando se pone a cantar canciones.
- BMO personalidad múltiple: Tiene solo un juego, pero varía entre BMO y fútbol cuando se vuelve a la pantalla principal. Incluso si esto es gracioso, recordé que BMO y Fútbol tienen las mismas caras, y no podrían distinguirse.
- BMO multijugador: BMO con juegos en donde puedan haber dos jugadores, esto significa, dos controladores.
- BMO tamagotchi: Tiene un juego muy simple, pero se le puede dar de comer, podes hablar con el BMO y podes ponerle un bigote para que se convierta en BMO adulto en los diálogos.
- Hay otros conceptos, pero esta lista sería muy larga.

El profesor aceptó el primer concepto que era BMO arcade, así que no sirvió de mucho idear tantos planes. Pero aún así fue divertido dibujarlos y pensarlos.



Registro Semanal

22 de Septiembre de 2023

Elección de display y modelo base 3d

A lo largo de esta semana, tras llevar a cabo una investigación detallada y crear varias diapositivas, la pantalla LCD 12864 de la marca Arduino ha surgido como la opción más idónea para nuestro proyecto "BMO". En el proceso de búsqueda de precios, identifiqué dos opciones disponibles:



[Pantalla Lcd 12864 Impresora 3D Controlador + Adaptador Rampas](#)

Esta alternativa se destaca por su precio económico y viene acompañada de un controlador destinado para impresoras 3D. Tenemos la intención de contar con la asistencia de nuestro profesor para realizar la extracción del controlador, de manera que podamos utilizar únicamente la pantalla en nuestro proyecto.



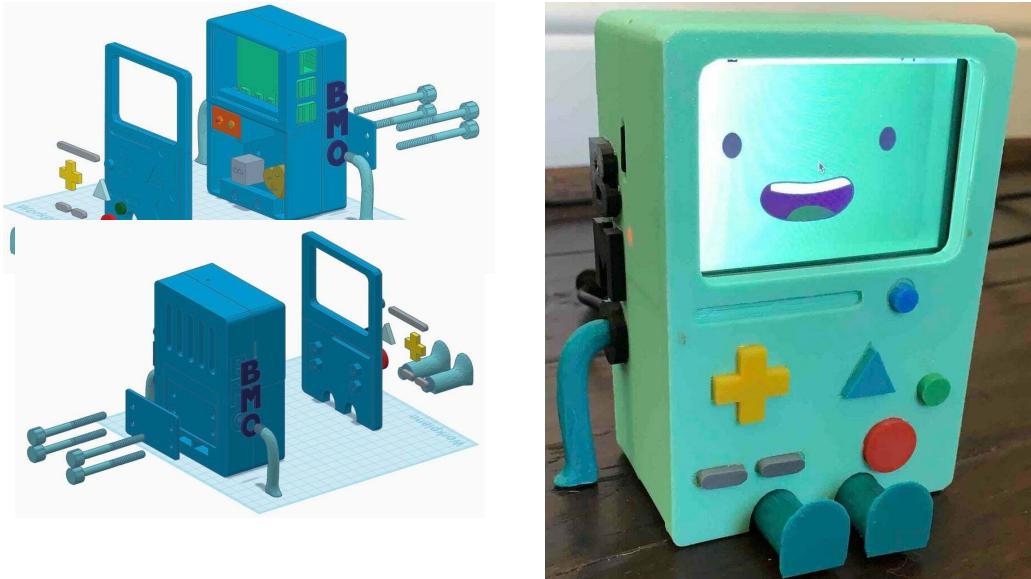
[Display Lcd Gráfico 12864 Backlight Azul 128x64 Arduino Ubot](#)

Aunque la segunda opción presenta un precio más elevado, destaca por ofrecer exclusivamente el display, sin accesorios adicionales. Aunque nuestra preferencia inicial se inclina hacia la primera opción debido a su mayor conveniencia, estamos considerando mantener esta alternativa como válida para una evaluación más detallada y una toma de decisión más informada.



Ambas pantallas comparten una resolución de 128 x 64 píxeles y presentan dimensiones de 93 mm x 70 mm. Cabe destacar que las dimensiones específicas del visor son de 76 mm x 50 mm. Estos detalles proporcionan una visión clara de las similitudes en términos de resolución y tamaño entre ambas opciones.

En nuestro intento de respaldar la elección de esta pantalla, a pesar de su precio más elevado, hemos identificado un modelo 3D que se adapta de manera perfecta a las dimensiones necesarias para nuestro proyecto. Este hallazgo refuerza la viabilidad de esta opción, ya que cumple de manera óptima con las especificaciones y necesidades de nuestro proyecto.



[CAJA RASPBERRY PI 4 - BMO \(ADVENTURE TIME\)](#)

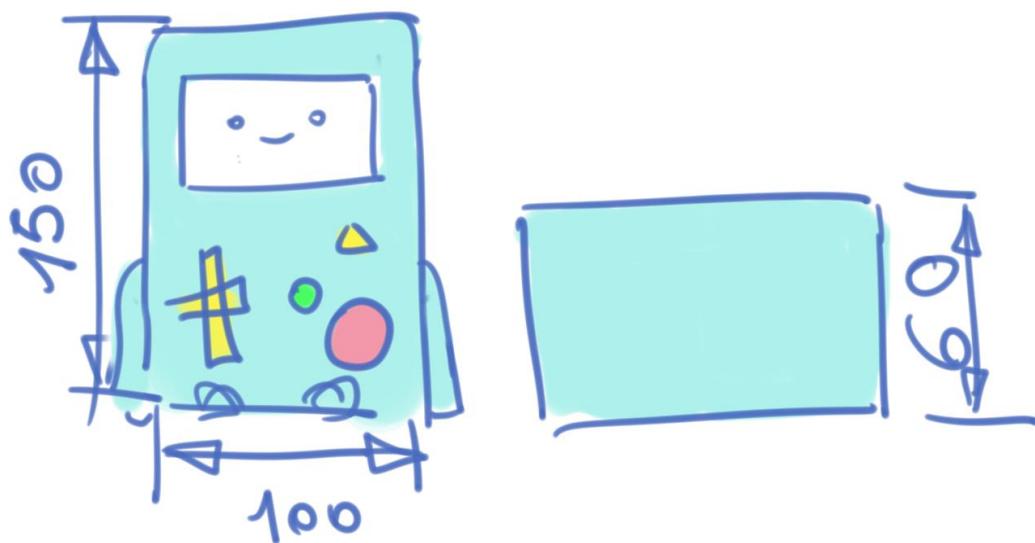
Sin embargo, surge un desafío en la adaptación de este modelo para componentes de Arduino, específicamente para el Arduino Uno. También estamos explorando la posibilidad de ajustar el tamaño del diseño para que sea compatible con la pantalla, que es ligeramente más grande y cuadrada.

25 de Septiembre de 2023

El Mono y los comienzos de la carcasa

Hoy, tras la clase de educación física, nos reunimos con "El Mono" para revisar y discutir el diseño de BMO. Encontró que el diseño era manejable, y hemos obtenido las medidas cruciales de longitud, anchura y la ubicación prevista para los altavoces, entre otras dimensiones esenciales. A pesar de estos avances, aún debemos abordar la parte trasera del diseño, así como los detalles relacionados con los botones y posiblemente la funcionalidad del disco, que actualmente está en duda debido a las limitaciones de pines.

Dado el número de botones requeridos, es probable que debamos realizar ajustes en las dimensiones del diseño, especialmente en lo que respecta a la anchura. Hasta el momento, parece que los componentes no ocuparán demasiado espacio, pero surge una preocupación en relación con el tipo de tornillos que utilizaremos, ya que esto podría afectar significativamente el grosor de BMO. "El Mono" nos ha aconsejado no preocuparnos demasiado por esto por ahora, pero es crucial que consideremos posibles ubicaciones y soluciones para abordar este aspecto en el futuro.



Pequeña ilustración de las medidas.

Por el momento, hemos llegado a un acuerdo en cuanto a las dimensiones de BMO, las cuales serán de 100 mm de ancho, 150 mm de largo y 60 mm de profundidad. Se ha decidido que tanto los brazos como las piernas de BMO tendrán un radio de 10 mm, y estarán posicionados estratégicamente para proporcionar una base estable al robot. Estas medidas servirán como base para avanzar en el diseño y desarrollo de manera consistente con nuestras metas y requisitos.

Pasos para el “Tetrix”

Nuestro profesor, Gonzalo Consorti, me brindó una breve introducción sobre la lógica del juego "Tetris". Sin embargo, debido a las restricciones de tiempo, tengo dudas sobre la posibilidad de implementarla por completo. Las áreas clave que aún deben ser investigadas incluyen:

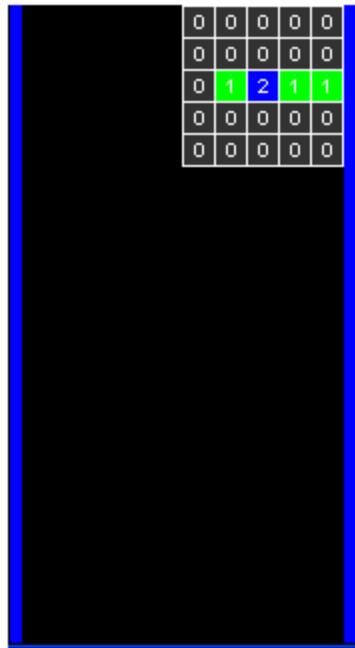
Lógica del Tetris: Comprender la creación, movimientos y matrices de las piezas del juego, así como el diseño del tablero.

Caída de las Piezas: Implementar la activación y desactivación de los píxeles a medida que las piezas descienden. Esto puede requerir pruebas de dirección.

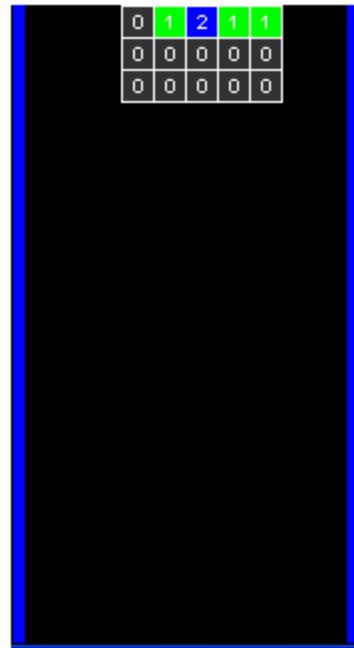
Detección de Colisiones: Identificar si una pieza se superpone a otra y cómo gestionar esta situación.

Eliminación de Filas Completas: Detectar cuando una fila se llena y eliminarla, desplazando las filas restantes hacia abajo.

Wrong initial position



Good initial position (-2, -2)



Existen desafíos adicionales, como la interacción con los bordes del tablero y la detección de posibles errores cuando las piezas se acumulan, eliminando píxeles accidentalmente, creando problemas en el juego. Estas áreas clave son fundamentales para la implementación exitosa del juego Tetris y requerirán una investigación detallada para su ejecución efectiva dentro de nuestras restricciones de tiempo.

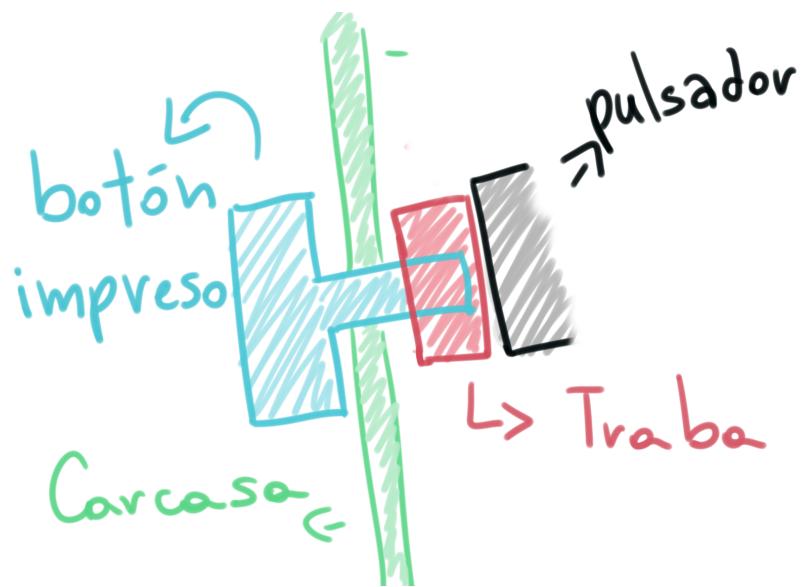
27 de Septiembre de 2023

Medidas de algunos componentes

Durante el proceso, he llevado a cabo meticulosas medidas de los diversos componentes, llegando a la conclusión de que será imperativo emplear exclusivamente pulsadores respaldados por una placa negra. Este enfoque elimina la necesidad de utilizar una protoboard, especialmente

diseñada para los botones ubicados en la parte frontal de la carcasa. Además, he elaborado un detallado boceto que ilustra tanto la estructura del botón como la disposición que estos tendrían en el conjunto.

Como parte de la asesoría proporcionada por el profesor Consorti, se sugirió la incorporación de una pequeña extensión en los botones, acompañada de una traba para facilitar un movimiento suave. Este ajuste se plantea como una medida preventiva para evitar que los botones se desprendan al girar el pequeño robot en un ángulo de 90 grados. La implementación de esta recomendación no sólo fortalecería la integridad del diseño, sino que también añadiría una capa de sofisticación y estabilidad a la funcionalidad del robot.



Representación de lo que hablaba el profesor

29 de Septiembre de 2023

El mono y el botón en forma de cruz

Durante el día de hoy, nuestra concentración fue plenamente hacia el botón de forma de cruz, en especial con las medidas del soporte para estos botones. Esta tarea resultó ser un tanto complicada, ya que el Mono constantemente se confundía con respecto al lado correcto donde se debían encontrar los agujeros para la inserción de los tornillos.

Otro inconveniente que tuvimos fue que, al haber traído conmigo pocos tornillos, siendo que creí que con una o dos alcanzaban para las medidas, parecíamos tener un talento especial para que se nos caiga cada dos por tres. Además que eran tan pequeños que las mediciones del grosor variaba constantemente, añadiendo un nivel de estrés, molestia y complejidad a la labor de este día. Resumiéndolo de forma informal, fue un desastre.

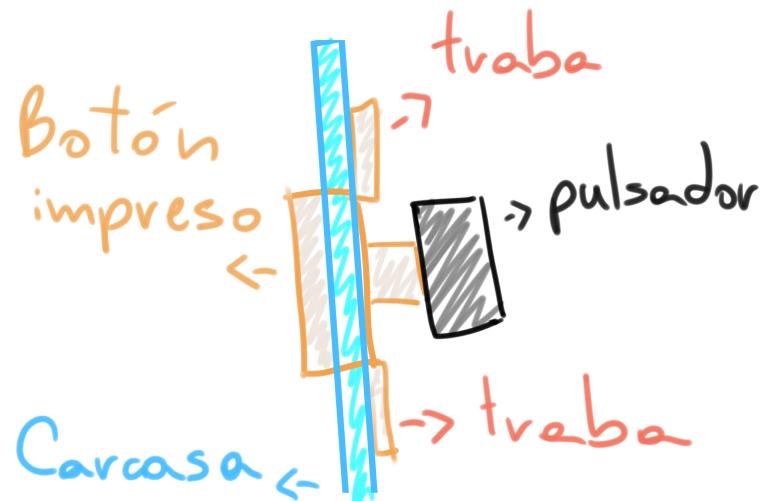


Imagen del progreso del Mono en Autocad.

Otro aspecto que también tomó un considerable tiempo fue la búsqueda de una buena posición para ubicar el agujero destinado al botón en cruz en la carcasa. En un principio no bastaba lugar para los agujeros de los botones con formas geométricas, pero luego, como se

muestra en la imagen, el espacio entre el marco de la carcasa con las dimensiones del pulsador era muy reducido. Este inconveniente se originó al haber tomado únicamente las medidas de la placa y de los pines, pasamos por alto el aumento en tamaño que implica la conexión de cables. La necesidad de cables emergió como un factor que lamentablemente no habíamos tenido en cuenta en nuestras consideraciones iniciales, lo cual añadió una capa adicional de complejidad al proceso.

Posteriormente, al concluir esta fase, opté por abordar el tema de la propuesta que me había sugerido Consorti, de agregar una traba al pulsador mediante una pequeña extensión, permitiendo así un movimiento suave de los botones. Esta modificación se concebía como una medida preventiva para evitar que los botones se desprendieran al girar el pequeño robot en un ángulo de 90 grados. Intenté explicar la idea al Mono, pero al no lograr una comprensión adecuada, decidí recurrir al profesor. Su capacidad imaginativa y creativa supera la mía, o tal vez simplemente se deba a que la claridad de mis explicaciones dejaba mucho que desear. Al final, terminamos optando por la opción del mono, agregando una pequeña traba a los botones para conseguir un resultado mucho más simple, aunque algo frágil.



El resultado que quiere el mono para el botón.

4 de octubre de 2023

Avances con el diseño y el display en persona

Durante la jornada de hoy, dediqué la mayor parte del tiempo colaborando con el Mono, lo que resultó en un significativo avance en el diseño de la carcasa.

En primer lugar, nos centramos en los agujeros ubicados en los laterales del BMO, los cuales resultaron ser bastante simples y no presentaron mayores inconvenientes. No obstante, más tarde me percaté de una disparidad en la cantidad de lados entre el BMO original y el diseño que estábamos creando. Mientras el BMO original parecía tener la forma de un heptágono, el nuestro adoptó la figura de un hexágono. Sin embargo, dado que el Mono ejecutó esta modificación con notable rapidez en comparación con otros elementos de la carcasa, decidí no desilusionarlo y opté por pasar por alto esta diferencia. Esto fue en el primer módulo que pasé con él.

En el siguiente paso, una vez que adquirimos el Display, nos propusimos examinar sus dimensiones. Dado que no lo teníamos físicamente presente, improvisamos y asumimos que el visor estaba ubicado precisamente en el centro, incluso cuando las imágenes en Mercado Libre no mostraban claramente simetría. A pesar de esta limitación, el Mono procedió a incorporar el agujero correspondiente. Sin embargo, observó que debido al display (la placa trasera de color verde), este era ligeramente más grande que la carcasa. En consecuencia, tomamos la decisión de aumentar el ancho en 1 cm adicional, respaldados por la sugerencia del profesor, quien destacó la importancia de proporcionar un margen mayor para proteger el display.

Esta elección también surgió cuando el Mono diseñaba la base destinada a soportar los botones con diversas formas. En el proceso, se percató de que la disposición inicial resultaría demasiado ajustada. Esta observación condujo a la decisión de aumentar el ancho de la carcasa en 1 cm adicional. Sorprendentemente, esta modificación no solo solucionó el problema de espacio, sino que también contribuyó a una estética más armoniosa y equilibrada en el conjunto del diseño.

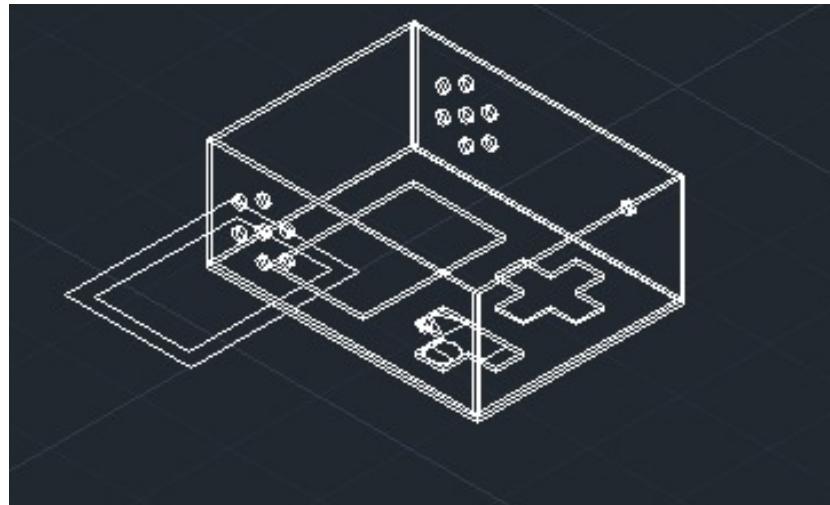


Foto del avance que se logró.

Más tarde, por la noche, al regresar a casa y disfrutar de un momento tranquilo acompañado de una chocolatada caliente para combatir el intenso frío, recibí la inesperada llegada de mi papá con un paquete proveniente de Mercado Libre. El envío, envuelto de manera un tanto descuidada en cartón, estaba resguardado en una bolsa negra que a su vez contenía una pequeña bolsa con burbujas de aire.

Sin perder tiempo, procedí a desenrollar el paquete y me encontré cara a cara con el maravilloso display.



Recreación de la imagen de cuando apenas me llegó el display.

El único inconveniente que encontré con el display que recibí fueron algunos pequeños golpes en el centro de la pantalla. Afortunadamente, cuenta con una delgada lámina que la protegió en cierta medida, aunque parece que no fue completamente suficiente. No obstante, se trata de dos leves abolladuras que pueden ser fácilmente pasadas por alto.

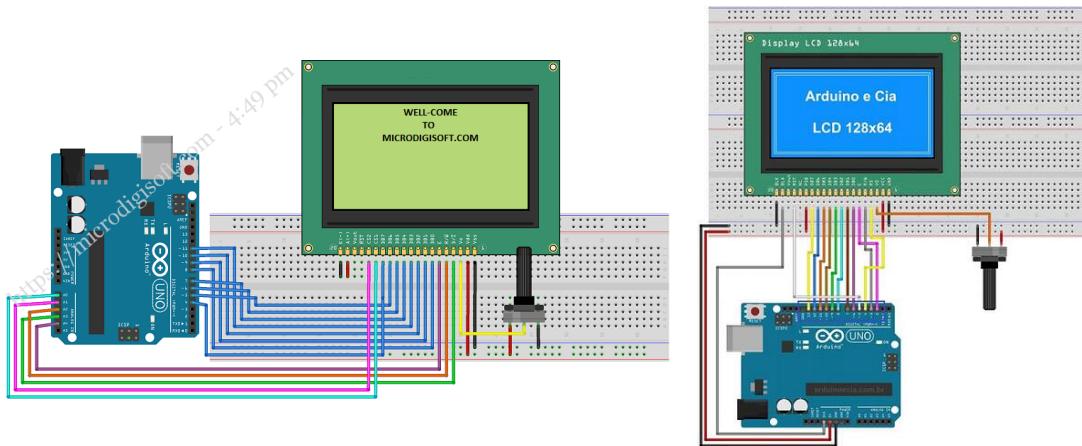
6 de octubre de 2023

Investigación de Display y soportes para los botones (momento del conflicto)

Hoy, al llegar al aula, no perdí tiempo y le mostré al profesor el display. Después de examinarlo detenidamente, señaló que la placa parecía estar soldada solo en la parte de los pines y sugirió que sería fácil de desmontar. Incluso se ofreció a intentarlo por sí mismo en casa, así que decidí confiarle el display para que lo llevara consigo.

La recomendación del profesor de buscar el circuito del display mientras esperamos, aunque bien intencionada, se reveló como un desafío más complejo de lo anticipado. La diversidad de formas de conectar el display, no sólo en términos de circuito sino también en cuanto a las funciones y librerías específicas necesarias, agrega una capa significativa de complejidad a la tarea. Este proceso me ha dejado claro que, a veces, la aparente simplicidad de una sugerencia puede ocultar una profundidad de detalles y consideraciones.

Una lección valiosa que aprendí en este proceso es que el modelo ST7920 con backlight azul tiene variantes, marcadas como A, B y C, cada una con una forma de conexión diferente. Este descubrimiento añade una capa adicional de complejidad al proyecto, ya que ahora debo asegurarme de entender las especificaciones y requerimientos precisos de la variante específica que tenemos entre manos. Además que el nivel de complejidad y absurdo que tienen estos circuitos es asombrosa.

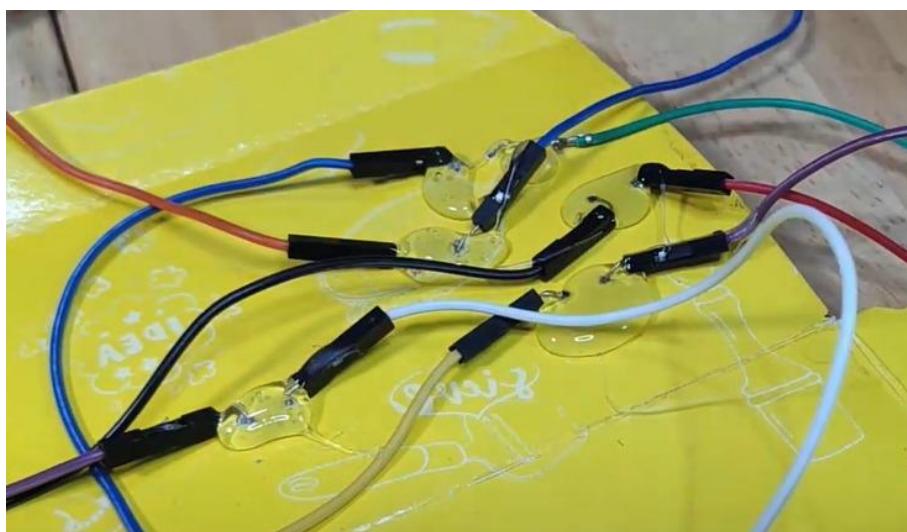


Con esto, en cuenta, el display tiene bastantes complicaciones y precauciones para tomar en cuenta:

- ★ La pantalla podría no funcionar correctamente si no recibe la alimentación adecuada. Hay que verificar si se está suministrando la tensión correcta y que la corriente sea suficiente.

- ★ Algunas de estas pantallas pueden no ser compatibles con las librerías, siendo que muchas están desactualizadas.
- ★ Pueden requerir una cantidad significativa de memoria, especialmente si estás trabajando con gráficos, y hay que asegurarse de que el programa no esté agotando la memoria disponible.
- ★ En general tienen una velocidad muy lenta para la comunicación.

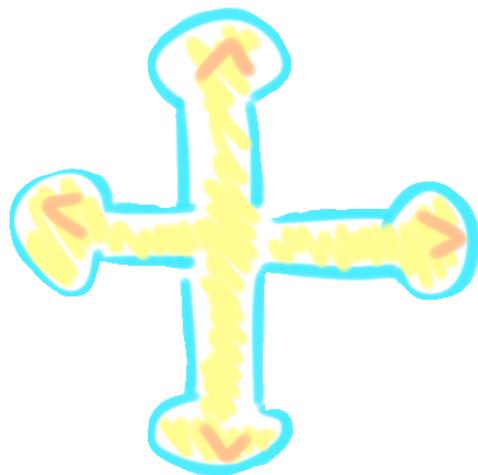
Para no dificultarme la vida al buscar los videos que me fueron más útiles para descubrir esto, decidí crear una lista de reproducción: [Lista de reproducción Projectou](#), Uno de ellos recreaba un pequeño arcade con el mismo display, presentando el juego de "Snake" o "La viberita". La simplicidad y economía del circuito me hicieron sonreír, pero mi expresión cambió al revisar el código fuente y darme cuenta de que no entendía nada. Este momento me recordó la importancia de la humildad en el aprendizaje. Hoy he comprendido que una persona inteligente puede lograr crear lo que desea, incluso sin las herramientas estéticas ideales, siempre y cuando la funcionalidad esté diseñada de manera eficaz y optimizada. Una valiosa lección sobre la importancia de la humildad y la capacidad de aprender de cualquier fuente.



Un circuito humilde.

En este mismo día, continuamos con "Mono" nuestra odisea con el agujero en forma de cruz, pero aquí es cuando mi mundo se desmoronó. La visión que tenía para ese botón era un diseño amarillo saturado, con un grosor sustancial y extremadamente cómodo para los usuarios que disfrutarían interactuar con el robot. Sin embargo, el destino parecía tener otros planes para mí.

Por razones de diseño que no logré comprender del todo, relacionadas con la posición de los tornillos, el soporte y otros elementos, nos vimos obligados a reducir el grosor del botón en forma de cruz. Este cambio comprometió la comodidad que originalmente imaginé para los usuarios. En mi desesperación, busqué la asesoría de mi "abogado", el Sr. Consorti, pero lamentablemente, no había más opción que aceptar este destino impuesto por cuestiones de diseño.



Representación del botón que el Mono quería hacer en compensación.

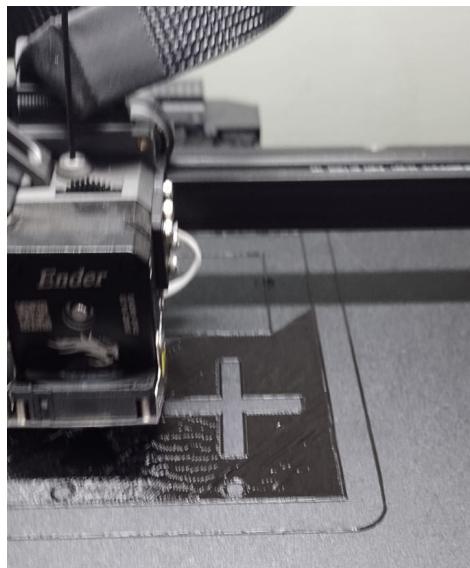
Mis lamentos y esperanzas de que "Mono" mostrara clemencia se desvanecieron. Por cuestiones de diseño, propuso la opción de agregar pequeños círculos al final de los "palos" del botón de cruz para compensar la reducción de grosor. Siendo una persona insegura cuando se trata de una confrontación,

decidí llamar a mi "abogado" nuevamente, quien entre risas explicó a "Mono" que esa solución no sería estéticamente agradable y que mi preferencia era de que no se implementara. Aunque pueda parecer una actitud cobarde, aún me encontraba en duelo por el nuevo grosor del botón y no me sentía lista para enfrentar otro problema de diseño.

8 de octubre de 2023

Comienzo de la impresión

Al llegar al "fab-lab", tuve la emocionante oportunidad de observar el proceso de impresión de la carcasa de mi proyecto durante unos minutos.



No hay mucho más que comentar, además de mi ilusión de ver mi proyecto convirtiéndose en algo más que un concepto o una fantasía que tengo desde chica y que estoy segura que muchos también lo tienen, al ser una persona que se apoya mucho a las series, este proyecto tiene un gran valor emocional para mí. Me siento extremadamente afortunada y agradecida de tener profesores que me apoyan, y que me ayudan en lo que necesite. Incluso si mis demás compañeros no colaboran mucho, realmente no me importa porque cada día se siente

más como un proyecto y reto personal, que incluso si muchas veces me estresa, estas pequeñas cosas hacen que me motive a estudiar y seguir esta especialidad.



11 de octubre de 2023

Poco código y muchas preguntas

A medida que los nervios asociados al diseño gradualmente disminuyen gracias a la confirmación de una carcasa funcional y la preparación de los elementos fundamentales, se deberá comenzar con la etapa de codificación. Reconozco que fue un error de gestión de mi parte haber postergado este paso hasta aproximadamente la mitad del tiempo disponible para el proyecto.

Con el objetivo de aportar un sentido de organización a las preguntas y problemáticas que necesito abordar, elaboraré una lista de puntos específicos y detallaré mis inquietudes en relación a cada uno:

- *Mostrar por consola lo que después se va a ver en el display:* Entiendo que para esto tengo que hacer múltiples “print statements” para ayudarme a visualizar mediante la consola lo que luego estará en el display. Hasta cierto punto me pregunto si esto será efectivo o si no me llevaría mucho tiempo, siendo que la verdad, no queda tiempo para desperdiciar.
- *Buscar alternativas de juegos para el bmo:* Mi habilidad para programar no es tan alta actualmente como para poder crear un juego de tetris, e incluso si lo tuviera, cómo dije anteriormente, el tiempo es limitado y mis compañeros no muestran señal de que estén trabajando, así que tengo que buscar un juego fácil.
- *Iniciar con la animación de la cara del BMO y posteriormente avanzar con algún juego:*
Me encuentro confundida en la cantidad de espacio que toma una animación en el arduino, especialmente con lo grandes que tendrán que ser las matrices siendo que es una pantalla 128 x 64.

Después de reflexionar durante un buen tiempo, me di cuenta de que muchas de las incertidumbres e incógnitas que me han estado preocupando provienen del miedo al futuro. Sin embargo, también comprendí que estancarme en estas preocupaciones no resuelve ninguno de estos problemas. Decidí abordar la situación de frente y le pregunté al profesor cuál sería el juego más fácil de lograr para mi proyecto. Su respuesta fue el clásico "Snake".



Foto del estado curioso en el que estaba Consorti al responder mi pregunta.

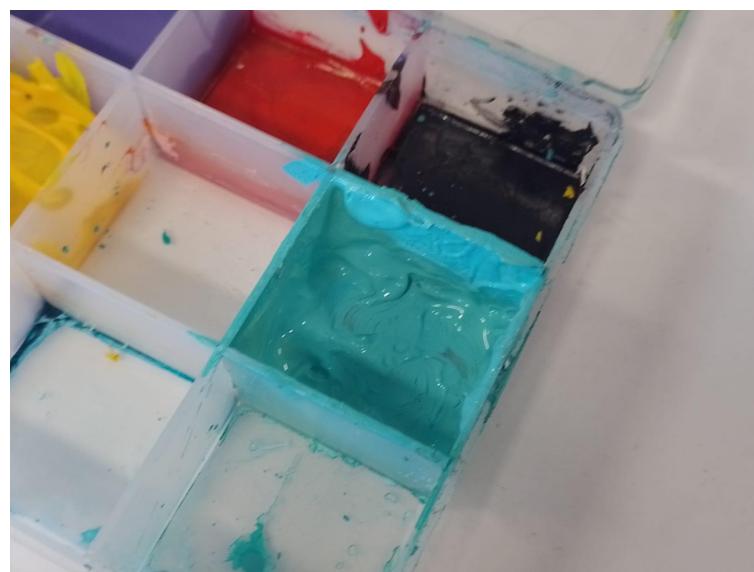
A pesar de haber mencionado previamente a uno de mis compañeros la posibilidad de que él se encargara del código para este juego, las circunstancias actuales y ciertos desacuerdos han complicado esa colaboración. Aunque siento cierta culpa por retirarle el único trabajo que le había asignado, su falta de interés en el progreso del proyecto me lleva a tomar las riendas por mi cuenta. Dada mi naturaleza perfeccionista y determinada, estoy decidida a enfrentar el desafío y crear el código por mí misma. De esta manera, puedo asegurarme de que el código sea armonioso y esté alineado con mis expectativas y visión para el proyecto.

Mi proyecto, después de todo, es una expresión de mi creatividad y dedicación, y estoy lista para asumir la responsabilidad completa de su desarrollo.

Colores para el BMO

Dado que ha comenzado la etapa del caos académico, donde los profesores se centran en evaluar nuestras pocas notas y nos asignan un mínimo de dos trabajos y una presentación, me veo obligada a poner una breve pausa a mi proyecto. A pesar de que esta materia es la que más me motiva, la responsabilidad como estudiante puntual y comprometido me llama a enfocarme en estas tareas académicas.

Sin embargo, para no detener completamente el progreso de mi proyecto, decidí tomar una pequeña medida. Me dirigí a la tienda "Jovi" cercana a mi casa, conocida por proveer útiles para estudiantes, especialmente aquellos de escuela técnica y artistas. Allí, adquirí acrílicos que se asemejan al color característico de BMO. Después de mezclarlos cuidadosamente, este fue el resultado.



El resultado fue uno satisfactorio, en mi parecer, es el color perfecto para BMO.

BMO pintado

Todo el día estuve inmersa en la tarea de pintar a BMO en diferentes rincones de la escuela. A pesar de lo incómodo que resultó no tener manos disponibles debido a la pintura, logré darle la primera capa de color a todo el proyecto. El resultado superó mis expectativas; la impresión fue mucho más agradable de lo que había imaginado. No me percaté de lo plano que dejaría la impresión ni de que la carcasa original, de color negro, no necesitaría una primera capa de pintura blanca.

A pesar de los pequeños inconvenientes y manchas en mi ropa, pelo y cara debido a la logística de moverme con el pincel en la boca, me siento genuinamente orgullosa del resultado. Esta experiencia me hizo sentir como una verdadera artista, sumergida en el proceso creativo. Cada mancha en mi ropa ahora lleva el toque de mi proyecto.

25 de octubre de 2023

El dremel y el display



Al ingresar al aula donde me encontraría con Consorti, me recibió un sonido que me hizo crujir los dientes: era el profesor Olaso, sosteniendo mi display y una Dremel. Consorti explicó que la tarea de desoldar la pantalla resultó ser más complicada de lo esperado, por lo que decidieron utilizar la Dremel para perforar suavemente la placa roja.

El proceso tomó bastante tiempo y resultó divertido observar cómo, al perforar por completo la placa roja, en varias ocasiones rozaba ligeramente la placa verde del display. Al menos, eso les pareció gracioso a los alumnos Maximiliano Passo Koziura y Kevin Gonzalez, mientras yo tenía el Jesús en la boca. Aunque intentaba distraerme del sonido y lo que estaba ocurriendo para poder darle una segunda mano de pintura a la carcasa del BMO, siendo que ya se había secado y solamente faltaba eso para darle un pequeño cierre a la pintura, o por lo menos a la parte frontal de la carcasa.

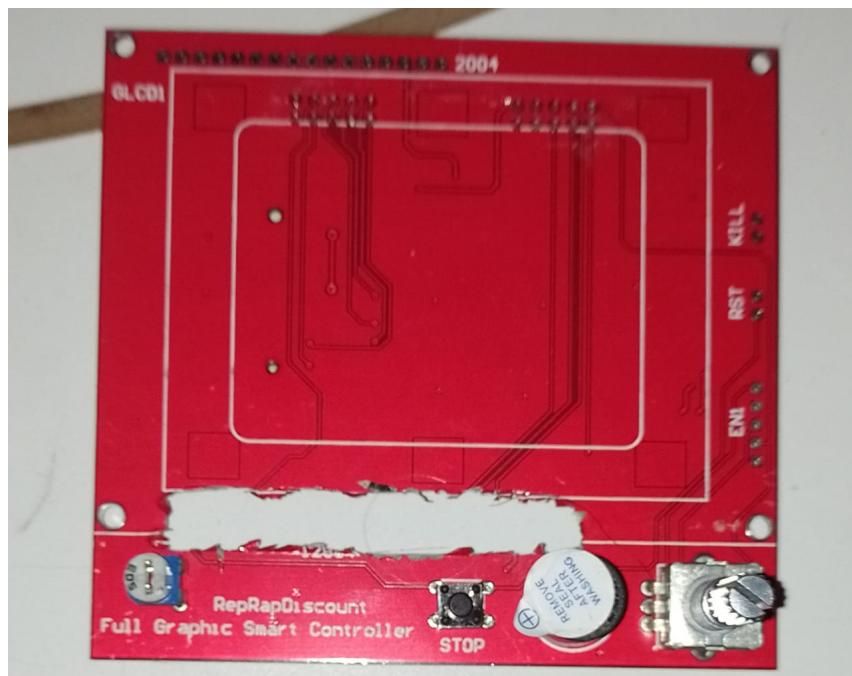


Imagen del resultado de la placa roja.

Después de ese proceso, el profesor Consorti se dedicó a verificar si todos los pines funcionaban correctamente en mi display. Aunque le tomó un poco de tiempo, finalmente llegó a la conclusión de que aún era funcional. Agradecida por el esfuerzo y la dedicación de ambos profesores, ahora me siento lista para avanzar en la parte más esencial de mi proyecto.

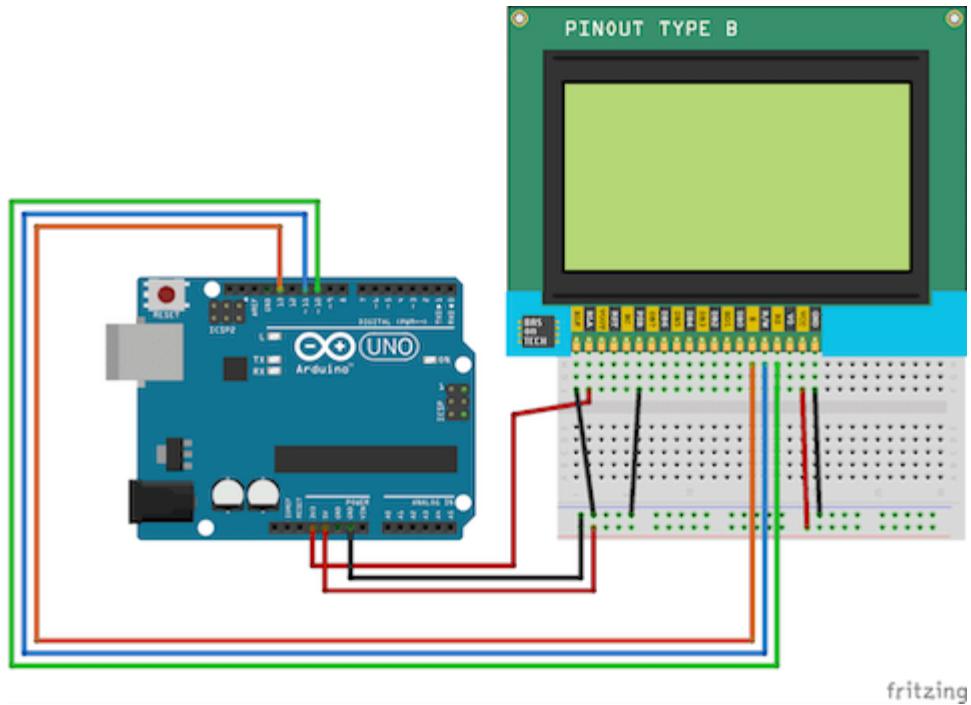
26 de octubre de 2023

BMO en estado crítico

En día de hoy, después de separar el display de la placa, me animé a hacer una prueba para ver si la pequeña pantalla seguía viva o si la Dremel había hecho estragos. Estaba nerviosa porque conectar un Arduino a una protoboard se siente como entrar a terreno desconocido, sobre todo para alguien que está dando sus primeros pasos en esto.

Mi plan era conectar unos botones para ir probando y entender las conexiones, aunque fueran básicas. Pero la idea de no confirmar si el display estaba funcionando el primer día que lo tuve entre mis manos me puso en modo pánico. Con mucho cuidado, desconecté todo y me sumergí en una maratón de videos instructivos para aprender cómo manejar este componente en el que todo mi proyecto depende de que ese display funcione como es debido.

Descubrí un canal de youtube poco reconocido llamado [Bas on Tech](#) presentaba en uno de sus videos el paso a paso de la conexión. Encontrar este recurso resultó un tanto desafiante, dado que la mayoría de los tutoriales se centraban en el uso de un Arduino Mega. Luego, al ir a su [página web](#), tuve la suerte de encontrar una imagen en la que se mostraba el circuito de manera más organizada y detallada



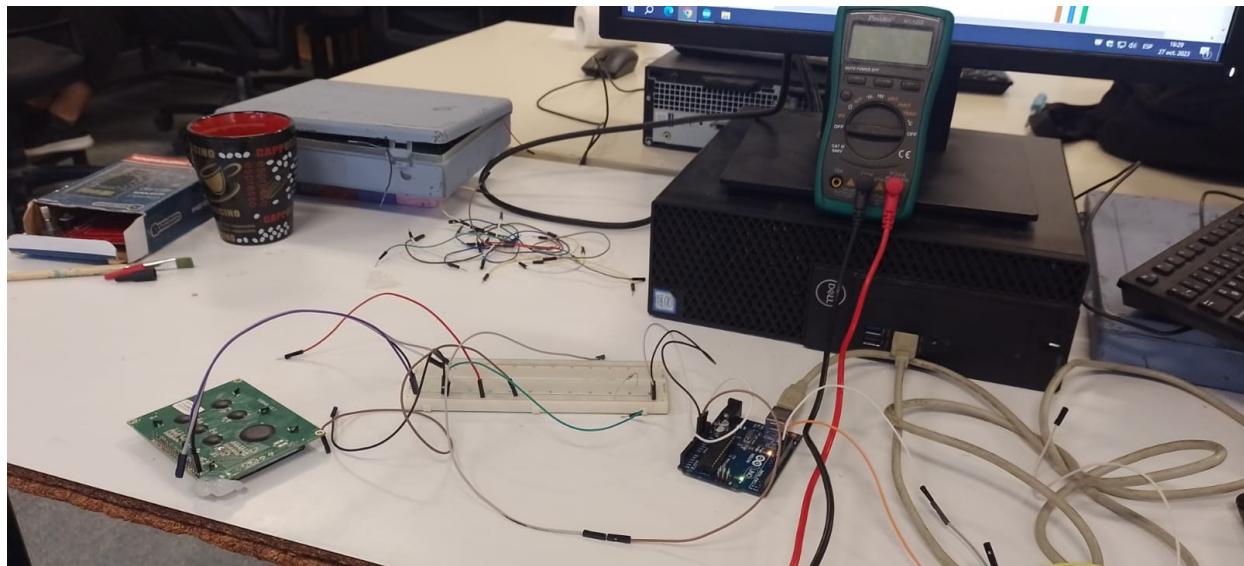
Además, en la página se mencionaba la librería U8G2 del cual hablé previamente de utilizar, así que me parecía una buena idea seguir paso a paso este video, siendo que me dejó en prácticamente en bandeja de plata la forma en la que tenía que utilizarse. Aunque luego de unos minutos me dí cuenta que sería mucho más complicado de lo que creía.

Superando mis miedos, inicié el montaje del circuito, pero me sentí confundida con los pines ubicados en la parte superior izquierda. En ese momento, mi percepción de la perspectiva no estaba funcionando correctamente, y terminé colocando la mayoría de los pines un nivel más arriba de lo debido.

Decidí rendirme con el circuito y agregar el código de Arduino. En la librería U8G2, es necesario especificar explícitamente el modelo del display que se está utilizando y definir los pines que se emplean para el componente. Aunque el video proporcionaba el código necesario,

su implementación fue complicada, siendo que hay cientos de displays. Opté por utilizar una de las animaciones de ejemplo incluidas en la librería.

Posteriormente, llamé al profesor para que me ayude con el inicio con el circuito, especialmente porque varios de los cables no eran de la mejor calidad y se desconectaban con facilidad, lo que complicó y alargó un poco el proceso.



A pesar de haber verificado varias veces que el circuito estuviera correctamente conectado, el display no parecía funcionar. Consorti utilizó un tester para asegurarse de que la corriente pasara correctamente, y todo funcionaba perfectamente cuando lo hacía. Incluso agregó una fuente de alimentación para descartar problemas de energía. Por un breve momento, el display se iluminó completamente blanco, indicando que, al menos, no estaba roto. Consorti concluyó que el problema se encontraba en el código y no en el componente, por lo que debíamos continuar probando códigos para encontrar una solución funcional.

10 de noviembre de 2023

Investigación del Pong

Comencé a investigar sobre algunos de los juegos para comprender la lógica y prepararme por si quiero usar este conocimiento en el futuro, una vez que el año termine y pueda trabajar de manera más tranquila. Al mencionar el Pong, siendo completamente honesta, inicialmente pensé que uno de los jugadores sería automatizado, es decir, que solo habría un jugador. Sin embargo, al revisar los códigos relacionados con un display y un Arduino, noté que todos implicaban la presencia de dos jugadores.

Cuando tenga la oportunidad de hablar con el profesor, tendré que preguntar cuántos pines requerirá el pequeño joystick. Esto es esencial, ya que sería complicado que uno de los jugadores sostenga al BMO y juegue con los botones, mientras que el otro jugador utiliza un joystick. Además, me surgen dudas sobre si se podría conectar el joystick al Arduino en cualquier momento como una extensión o si debe estar conectado constantemente.



Demostración del BMO con los joysticks

Lógica del Pong

Primero, se aborda el funcionamiento de las paletas de los jugadores, las cuales deben realizar movimientos verticales y estar limitadas al eje vertical. Para optimizar esta interacción, se sugiere la utilización de un joystick como dispositivo de control.

Segundo, la dinámica de la pelota se establece con un movimiento lineal horizontal. Al hacer contacto con una de las paredes o con una paleta, la pelota debe rebotar, incorporando así la noción de reflexión en el juego.



Tercero, se aborda la interacción entre la pelota y las paletas. Cuando la pelota entra en contacto con una paleta, su dirección de movimiento debe cambiar de acuerdo con el ángulo de impacto. Si la paleta no logra tocar la pelota y esta pasa más allá, se otorga un punto al jugador contrario.

Cuarto, se introduce la mecánica de puntuación, que se exhibe en la pantalla del juego. Se establece la condición de reinicio cuando un jugador alcanza un cierto número de puntos predeterminado, o se muestra al jugador ganador en caso de alcanzar la victoria.

Quinto, se implementa un nivel de dificultad progresiva. A medida que la pelota realiza más rebotes, se propone aumentar la velocidad del movimiento de la pelota, agregando un elemento de desafío adicional para los jugadores a medida que avanzan en el juego.

18 de Noviembre de 2023

Animación del BMO

En la jornada de hoy, me vi sumido en un estado de gran ansiedad provocada por las elecciones presidenciales, me impedía conciliar el sueño y dejaba mi mente en un constante estado de inquietud. Decidí canalizar esa energía en algo productivo y, tras reflexionar, me pareció que dar inicio a la animación de la cara del BMO sería una excelente manera de redirigir mis pensamientos.

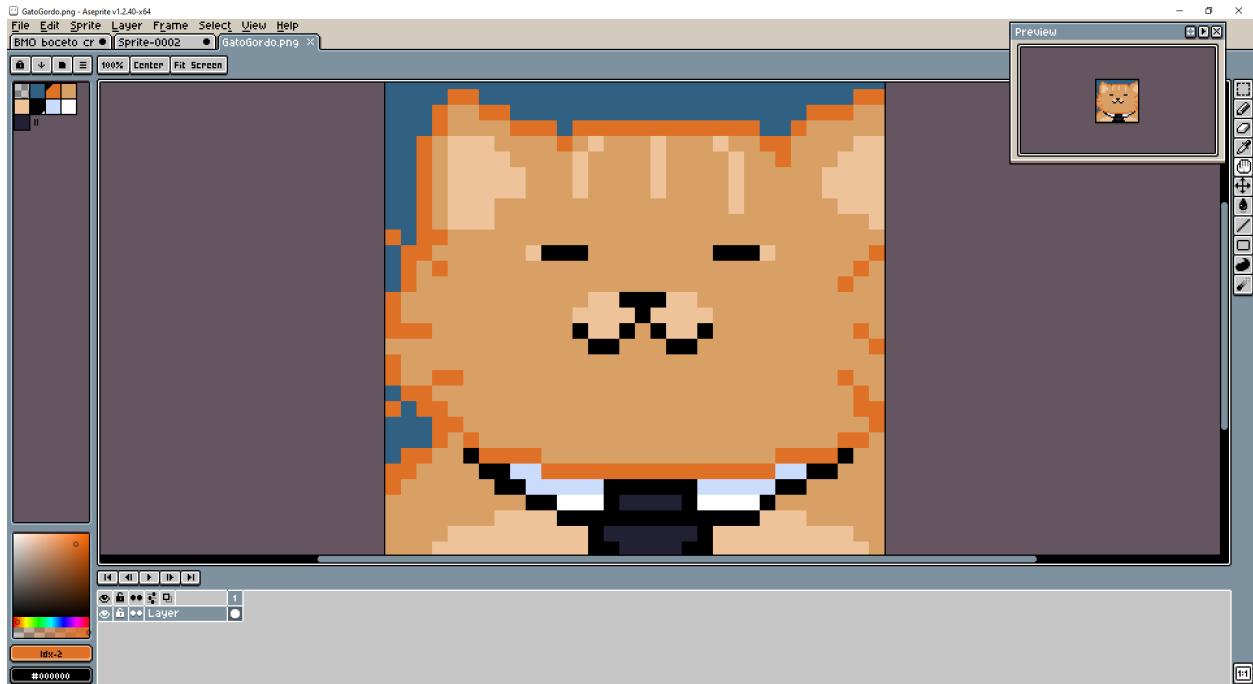
Aunque mi entusiasmo cambió por la decepción ante el color del backlight del display, que en lugar de ser el característico verde del BMO, se mostraba en un tono azul, con píxeles blancos. Sin embargo, en lugar de permitir que esta pequeña inconveniencia afectara mi motivación, decidí centrarme en la animación final.

La idea inicial para esta pequeña animación fue simple: un parpadeo. Por ahora, eso es todo lo que planeo incorporar, ya que me encuentro en la incertidumbre de cuánto espacio consumirán estos frames en las matrices. Es bueno recordar al lector que estamos hablando de una matriz de 128 x 64, no precisamente diminuta. A pesar de su tamaño, la animación será modesta, con alrededor de cinco o seis frames. Pero confío en que incluso esta simple adición dará vida al pequeño robot de manera encantadora.

Mi elección para el programa de dibujo recayó en Aseprite, una aplicación que adquirí recientemente y que resultó ser la elección perfecta para alguien tan inexperto como yo en este terreno. Explorando sus funciones, me di cuenta de que, como sugiere su nombre, está diseñado para sprites y animaciones, no solo ilustraciones estáticas. Este descubrimiento me permitió

visualizar con mayor claridad la velocidad y movimiento que quiero lograr en la animación final.

Definitivamente esta compra me vino como anillo al dedo.



“ Gato Gordo ”

Artista: SoupySoup

- Creado en 18 / 11 / 2023 23:24

La imagen que adjunté anteriormente representa mi primera prueba en el mundo de la ilustración con Aseprite. Fue una especie de prueba inicial para familiarizarme con las herramientas básicas del programa, y debo admitir que me sentí bastante cómodo desde el principio. Sigo siendo un novato en el dibujo, pero al menos para el lado de la ilustración, me aventuré con cierta confianza. Sin embargo, al adentrarme en el terreno de la animación, me di cuenta de que aún tengo mucho por aprender, a pesar de creer que conocía la teoría.



En segundo lugar, surgió la tarea de buscar referencias para la cara del BMO que planeo animar. Este paso resultó ser más complicado de lo que inicialmente imaginé. Al tratarse de una serie animada con un personaje que ha evolucionado a lo largo de los años, encontrar una referencia adecuada se convirtió en un desafío. Las imágenes variaban en perspectiva, algunas mostraban expresiones que no me interesaba dibujar en ese momento, y la diversidad en la forma de los ojos y la boca complicaba aún más la elección. Los ojos cambiaban entre redondos y ovalados, la boca adoptaba formas que iban desde lo redondeado hasta un parecido sorprendente al emoticono ":D", y la distancia entre los ojos y la boca variaba considerablemente.

Ante esta diversidad de referencias, decidí tomar cierta libertad en la interpretación. Me di cuenta de que los propios animadores de "Hora de Aventura", la serie a la que pertenece BMO, parecían tener esa misma libertad creativa al representar al adorable robot. Esto me animó a confiar en mi propia visión y estilo, aportando un toque personal a la animación que estoy emprendiendo.

Debo admitir que estaba un poco oxidada, logré superar los MÚLTIPLES obstáculos y completar la animación en 10 frames, lo que se traduce en 5 frames únicos. Fue un proceso estresante, y a pesar de las dificultades iniciales, estoy satisfecho con el resultado de los ojos.

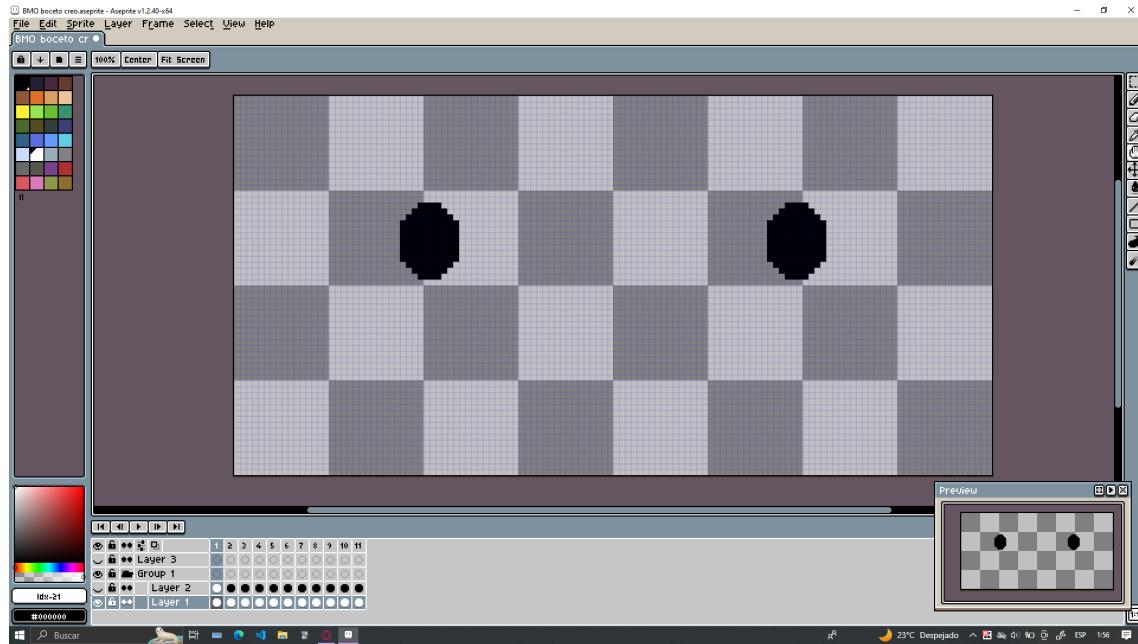


Imagen de los ojos del BMO, mostrando en la parte de abajo la cantidad de fotogramas en la animación

Después de varios intentos de plasmar la boca del pequeño robot en mi ilustración, debo confesar que no logré alcanzar la satisfacción deseada. Ante la creciente frustración, tomé la decisión sensata de tomar una siesta. No solo eso, sino que el reloj marcando las tres de la mañana se hizo evidente, y me encontré inmerso en una somnolencia extrema que hacía que enfrentar un trabajo *algo* desafiante resultara aún más agotador.

Así que, por ahora, mi boca dibujada puede esperar hasta que la luz del día y una mente más fresca me acompañen en este proceso creativo. El proceso artístico toma tiempo, no hay que hacer que se torne en algo que luego nos cree un sentimiento de decepción.

19 de Noviembre de 2023

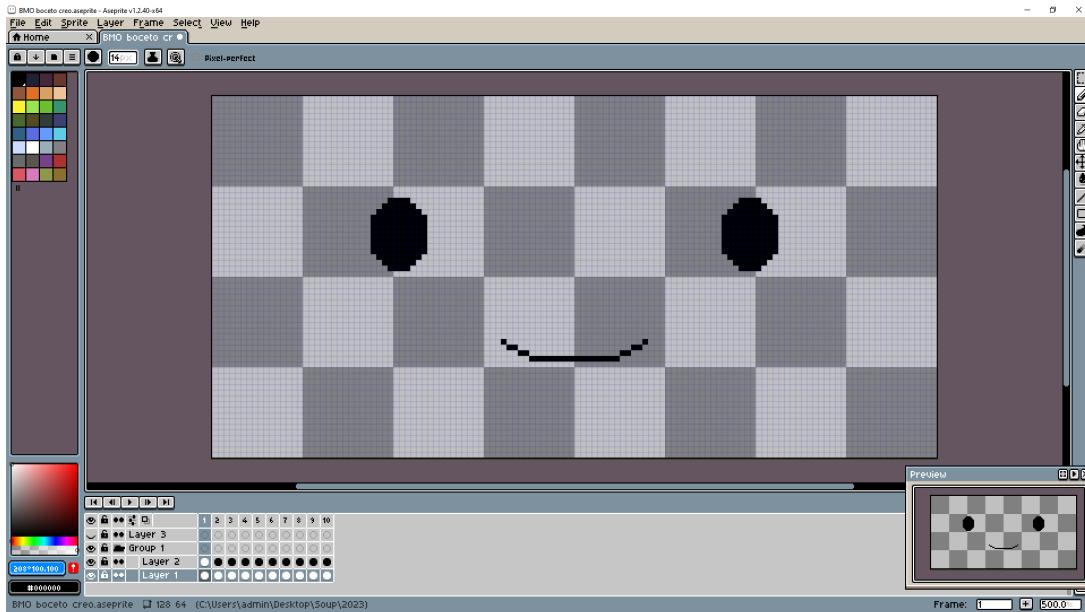
Visualización del BMO y primer boceto de la animación final.

Después de disfrutar de una siesta revitalizante que se extendió por 12 horas, me lancé nuevamente a mi odisea artística. Esta vez, mi desafío era abordar una de las facetas más complicadas para mí al dibujar personajes: la sonrisa. Enfrenté múltiples intentos de dibujar una sonrisa que se viera natural y, al mismo tiempo, fuera perfectamente simétrica. Fue un proceso que desafió mi paciencia.

Después de varios intentos y con cierto grado de frustración, hice una revelación clave: el programa que estaba utilizando tenía una herramienta que simplificaba enormemente la tarea de lograr simetría en los dibujos. Esta herramienta resultó ser un verdadero salvavidas, permitiéndome crear las dos variantes de boca que mejor caracterizan a BMO con mucha más facilidad.

Esta experiencia me enseñó la importancia de explorar las herramientas disponibles en mi entorno creativo. A veces, la solución a nuestros desafíos está más cerca de lo que pensamos, lo que uno aprende “che”. Estoy emocionado de ver cómo esta nueva herramienta transformará mis futuros proyectos y cómo, incluso en los momentos de mayor desafío, siempre hay espacio para el descubrimiento y la mejora.

Primera versión:

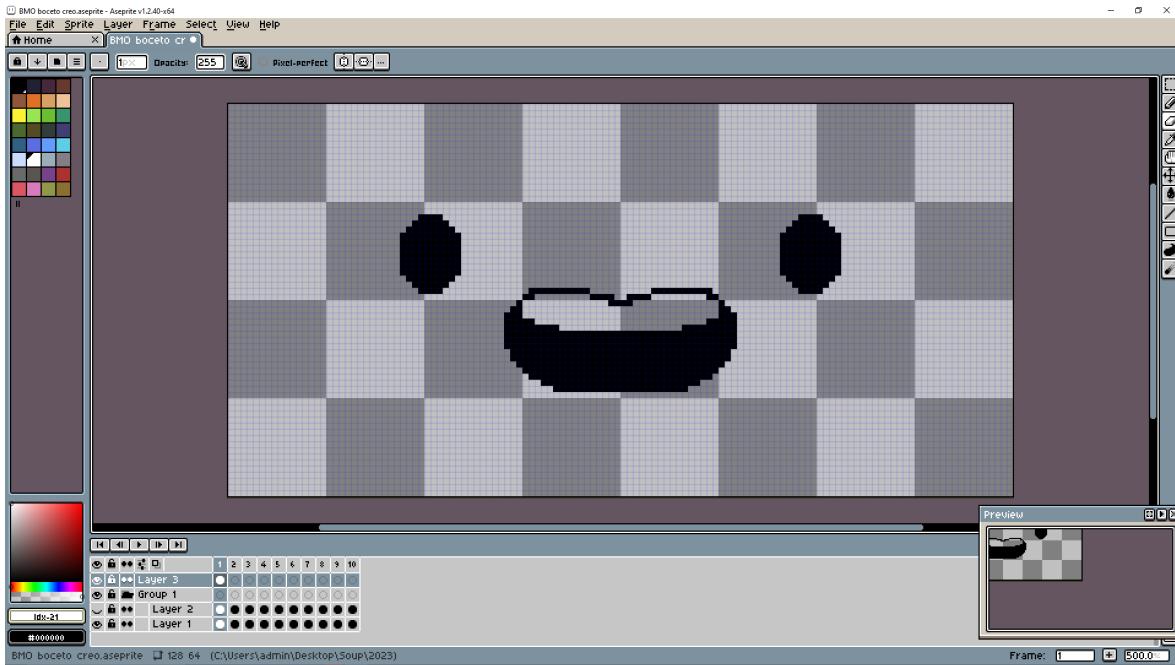


En esta versión, presento una representación bastante simple: una pequeña cara sonriente.

Mi dilema con esta expresión radica en que parece transmitir una especie de psicopatía o la intención siniestra de un robot dispuesto a causar estragos. Sin embargo, al revisar detenidamente mis referencias, me di cuenta de que BMO siempre ha mantenido esa apariencia intrigante. Su mirada fija y la sonrisa que podría interpretarse como maliciosa no le restan encanto. Al contrario, a lo largo de la serie, logra conquistar a la audiencia con su personalidad adorable y su distintiva voz, de manera que la expresión facial pasa a un segundo plano.

Aunque considero la posibilidad de cambiar esa sonrisa por una más parecida a una "U" o una "V", con el propósito de transformar la sensación de que está observando el alma del espectador a simplemente ser un "silly little robot".

Segunda versión:



Esta versión resulta considerablemente más adorable y acogedora, logrando que el personaje cobre vida de una manera encantadora. Sin embargo, al crear esta expresión, olvidé un detalle crucial: el color del backlight es azul y los píxeles son blancos. Esta pequeña omisión podría generar cierta confusión visual y hacer que la representación parezca un tanto extraña.

A pesar de este pequeño tropiezo, estoy considerando la posibilidad de utilizar esta expresión mejorada y perfeccionada para el próximo año. Tal vez, al ajustar los colores y realizar algunos refinamientos, pueda lograr una combinación armoniosa que conserve la ternura recién descubierta, pero que también se adapte a las características visuales específicas de BMO.

En conclusión, me quedaré con la primera versión, quizás eventualmente cambie la boca, pero se vería de esta forma:



Aunque note que la versión actual resultó más psicópata de lo que inicialmente esperaba, he decidido considerarla como el resultado provisional para gestionar eficientemente mi tiempo. Planeo esperar hasta el 22 de noviembre para consultar con el profesor y obtener una tercera opinión. A veces, la perspectiva externa puede proporcionar opiniones constructivas valiosas que podrían ayudarme a afinar y mejorar la expresión de BMO.

20 de Noviembre de 2023

Todo desde 0

Encontré un código simple para mi pantalla y el arduino uno, del juego “Snake”, el problema de encontrar códigos de este juego, es que sinceramente todos eran muy complejos y no llegaba a entenderlos del todo, siendo que en los repositorios había 500 archivos de los cuales mi nivel de c++ no eran suficientes.

Por suerte, no solo estaba solo el código, sino que había una pequeña explicación por cada fragmento de código.

```
#include "U8glib.h"
U8GLIB_ST7920_128X64 u8g(13, 11, 10);
```

El código comienza con el añadido de la librería U8g, la cual es la primera versión de U8G2, el problema con esta librería es que no es tan rápida como la mencionada, pero aún así no hay problema al utilizarla.

Los pines para el display son:

- Pin 13 es para “enable”
- Pin 11 para “Read/Write”
- Pin 10 “ Data or Instruction “

```
#define botonIzq 2
#define botonArr 3
#define botonAb 4
#define botonDer 5
#define botonReset 6
```

En el código original se encontraban solamente cuatro botones (arriba, abajo, izquierda, derecha) pero decidí agregar el botón de reset para acomodarlo a el BMO, siendo que quizás la persona quiera volver a ver su adorable cara y la animación que hice.

```
int comidaX = 6;
int comidaY = 6;
int maxSnakeLargo = 150;
int snakeLargo = 6;
int snakeX[maxSnakeLargo];
int snakeY[maxSnakeLargo];
```

Esto muestra que la comida de la serpiente será de un tamaño de 6x6, a lo largo del juego y que no se podrá cambiar, a comparación del tamaño de la serpiente que su máximo será una longitud de 150 pixeles, esto es porque `int` es para sobreescribirlo y leerlo, mientras que `const int` funciona solo para leer el dato. Esto quizás tenga que cambiarlo, dependiendo del almacenamiento que tenga el arduino.

```
bool movIzq = false;
bool movDer = false;
bool movAr = false;
bool movAb = false;
bool Start = false;
bool Stop = false;
bool arduinoControl = true;
```

Bool tiene dos valores, verdadero o falso. En este caso, todos los botones están en falso, donde falso se define como cero, lo que significa que cuando el sistema está encendido, todos los botones deben estar presionados para que funcione según sus direcciones designadas.

```
void first(){
    u8g.setFont(u8g_font_unifont);
    u8g.drawStr( 12, 15, "J U E G U I T O");
    u8g.drawStr( 25, 35, " D E L A");
    u8g.drawStr( 30, 55, " S E R P I E N T E");
}
void clearlcd(){
    u8g.firstPage();
    do { }
    while( u8g.nextPage() );
}
```

La función `first()` lo que hace es crear una intro / primera pantalla al seleccionar el juego, y muestra a lo largo de la pantalla que es el “jueguito de la serpiente”. `u8g.setFont` se utiliza

para establecer la fuente del texto que está incluida en la biblioteca u8g.drawStr se utiliza para colocar caracteres o texto en la pantalla LCD.

```
void Control(){
    for(int i = snakeLargo; i > 0; i--){
        snakeX[i] = snakeX[i-1];
        snakeY[i] = snakeY[i-1];
    }
    if (movIzq == true){
        snakeX[0] = snakeX[0] - 1;
    }
    else if (movAr == true){
        snakeY[0] = snakeY[0] + 1;
    }
    else if (movAb == true){
        snakeY[0] = snakeY[0] - 1;
    }
    else if (movDer == true){
        snakeX[0] = snakeX[0] + 1;
    }
}
```

La función `void Control()` se encarga de controlar la dirección de la serpiente y sincronizarla con los botones. El uso de `-1` y `+1` se refiere a la dirección de la serpiente.

El movimiento generalmente se realiza en un plano cartesiano, donde la serpiente se mueve a lo largo de los ejes x e y. Los valores negativos y positivos en estos ejes indican direcciones específicas, por ejemplo, hacia la izquierda o derecha en el eje x, y hacia arriba o abajo en el eje y.

Cuando se encuentra una expresión como `snakeX[0] = snakeX[0] - 1`, se está restando 1 a la posición x de la cabeza de la serpiente (`snakeX[0]`). Esto sugiere que la serpiente se está

moviendo hacia la izquierda, ya que la coordenada x disminuirá. Similarmente, cuando se encuentra `snakeY[0] = snakeY[0] + 1`, la serpiente se está moviendo hacia arriba, ya que la coordenada y está aumentando.

```
bool verColisiones(int x, int y){
    for (int i = 0; i < snakeLargo - 1; i++){
        if ((x == snakeX[i]) && (y == snakeY[i])){
            return true;
        }
    }
    return false;
}
```

Esta función `verColisiones` se encarga de comprobar si las coordenadas (x, y) proporcionadas están chocando con alguna parte de la serpiente. Está diseñada para iterar a través de las posiciones de cada segmento de la serpiente y compararlas con las coordenadas dadas.

Dentro del bucle, la condición `if` compara las coordenadas (x, y) con las coordenadas de cada segmento de la serpiente (`snakeX[i]` y `snakeY[i]`). Si se encuentra una coincidencia, significa que las coordenadas dadas están chocando con una parte de la serpiente, y la función devuelve `true`.

```
void Butons() {
    if (arduinoControl) {
        const int botones[] = {botonIzq, botonArr, botonAb, botonDer};
        const char* oneDirections[] = {"Izq", "Arr", "Ab", "Der"};
        bool* directionsFlags[] = {&movIzq, &movAr, &movAb, &movDer};

        for (int i = 0; i < 4; ++i) {
            if (digitalRead(buttons[i]) == HIGH && !(*directionsFlags[i])) {
                delay(200);
                resetearDirecciones();
                *directionsFlags[i] = true;
            }
        }
    }
}
```

```

        Serial.println(oneDirections[i]);
        break;
    }
}
}
}

void resetearDirecciones() {
    movIzq = false;
    movAr = false;
    movAb = false;
    movDer = false;
}

```

La función comienza con una verificación condicional (if (arduinoControl)) para asegurarse de que la función solo se ejecute si la variable arduinoControl es verdadera. Esto sugiere que el control de los botones está condicionado a la existencia o activación de Arduino.

Utiliza un bucle for para iterar sobre los cuatro botones. Dentro de este bucle, se verifica si el botón correspondiente está presionado (digitalRead(buttons[i]) == HIGH) y si la dirección asociada a ese botón no está ya activa (!(*directionsFlags[i])).

Si se cumple la condición de presionar el botón y la dirección no está activa, se introduce un retraso de 200 milisegundos (delay(200)). Este retraso probablemente se incorpora para evitar el procesamiento de múltiples entradas de botones en un período corto, proporcionando cierta estabilidad a la entrada del usuario.

Finalmente, después de procesar un botón, el bucle se rompe con la instrucción break, lo que indica que la función solo procesará el primer botón presionado y luego saldrá del bucle. Esto puede ser una elección de diseño para manejar un solo evento de botón a la vez.

```
void DibujarSnake(){
```

```

for (int i = 0; i < snakeLargo; i++){
    u8g.drawBox(snakeX[i], snakeY[i], 6, 6);
}
}

void DibujarComida(){
    if (OnScreen(comidaX, comidaY)){
        u8g.drawBox(comidaX, comidaY);
    }
}

bool OnScreen(int x, int y){
    if (x >= 0 && x < 128 && y >= 0 && y < 64){
        return true;
    }
    else{
        return false;
    }
}

```

La función DibujarSnake() se encarga de dibujar la serpiente en la pantalla. Utiliza un bucle for para iterar a través de cada segmento de la serpiente. En cada iteración, se utiliza la función u8g.drawBox() para dibujar un cuadro de 6x6 píxeles en las coordenadas correspondientes al segmento actual de la serpiente (snakeX[i] y snakeY[i]).

La función DibujarComida() se encarga de dibujar la comida en la pantalla. Antes de realizar el dibujo, verifica si las coordenadas de la comida (comidaX y comidaY) están dentro de los límites de la pantalla utilizando la función OnScreen(). Si la comida está dentro de la pantalla, se utiliza u8g.drawBox() para dibujar un cuadro en las coordenadas de la comida.

La función OnScreen() es una función de utilidad que verifica si las coordenadas (x, y) están dentro de los límites de la pantalla. Devuelve true si las coordenadas están dentro de los límites (entre 0 y 127 en el eje x, y entre 0 y 63 en el eje y), y false en caso contrario.

```

void darComida() {
    comidaX = random(6, 122);
    comidaY = random(6, 58);
}

void comer(){
    if (snakeX[0] == foodX && snakeY[0] == foodY){
        if(snakeLargo < maxSnakeLargo){
            snakeLargo++;
            u8g.drawBox(snakeX[0], snakeY[0], 6, 6);
        }
        return true;
    }
    else{
        return false;
    }
}

```

Si la cabeza de la serpiente está en la misma posición que la comida (snakeX[0] == comidaX y snakeY[0] == comidaY), y la longitud actual de la serpiente (snakeLargo) es menor que la longitud máxima permitida (maxSnakeLargo), entonces la longitud de la serpiente se incrementa en uno, y se dibuja un nuevo cuadro para representar el segmento de la serpiente en la posición de la cabeza. La función devuelve true para indicar que se ha producido una colisión y la serpiente ha comido.

Si no hay coincidencia con la comida, la función devuelve false.

```

void TouchedItself() {
    if (snakeX[0] < 0 || snakeX[0] >= 128 || snakeY[0] < 0 || snakeY[0] >= 64) {
        u8g.firstPage();
    }

void Ded(){

```

```

u8g.firstPage();
do {
    u8g.setFont(u8g_font_unifont);
    u8g.drawStr(10, 20, "¡Perdiste!");
} while (u8g.nextPage());

resetGame();
}

```

La función TouchedItSelf() verifica si la cabeza de la serpiente ha tocado los límites de la pantalla. Si la coordenada x de la cabeza (snakeX[0]) es menor que 0, mayor o igual a 128, o si la coordenada y de la cabeza (snakeY[0]) es menor que 0 o mayor o igual a 64, entonces se ejecuta u8g.firstPage().

La función Ded() se encarga de manejar la situación en la que el jugador ha perdido el juego. Utiliza un bucle do-while con u8g.nextPage() para dibujar el mensaje "¡Perdiste!" en la pantalla. La función u8g.drawStr() se utiliza para dibujar la cadena de texto en las coordenadas (10, 20). Después de dibujar el mensaje, la función resetGame() se llama, lo que probablemente restablece el estado del juego a sus condiciones iniciales.

```

void resetGame() {
    snakeLargo = 6;
    movIzq = false;
    movDer = false;
    movAr = false;
    movAb = false;
    Start = false;
    Stop = false;

    snakeX[0] = 20;
    snakeY[0] = 20;
    darComida();

    clearlcd();
}

```

```

u8g.firstPage();
do {
    first();
} while (u8g.nextPage());
}

```

La función resetGame() tiene como objetivo restablecer todas las variables y configuraciones del juego a sus valores iniciales, preparándolo para una nueva partida. A continuación, se detalla su funcionamiento:

Comienza reiniciando diversas variables relacionadas con el estado del juego. snakeLargo se fija en 6, indicando probablemente la longitud inicial de la serpiente. Además, se establecen como falsas las variables movIzq, movDer, movAr, movAb, Start y Stop, que probablemente controlan las direcciones de movimiento y el estado del juego.

A continuación, la cabeza de la serpiente se posiciona en (20, 20), lo que sugiere la colocación inicial predeterminada de la serpiente. Luego, se llama a la función darComida() para generar nuevas coordenadas para la comida, asegurando que aparezca en una ubicación diferente después de cada reinicio del juego.

```

void loop() {
    if (OnScreen(snakeX[0], snakeY[0])){
        DibujarSnake();
        DibujarComida();
        if (comer()){
            darComida();
            Butons();
            Control();
        }
        else {
            Butons();
            Control();
        }
    }
}

```

```

if (OnScreen(snakeX[0], snakeY[0])){
    TouchedItself();
    u8g.firstPage();
    do {
        DibujarComida();
        DibujarSnake();
    }
    while (
        u8g.nextPage());
}
else {
    Ded();
}
}

```

Esta es la explicación paso a paso que se encuentra en el loop():

1. Verificación de la posición de la cabeza: Se verifica si la cabeza de la serpiente está dentro de la pantalla utilizando la función OnScreen().
2. Dibujo de la serpiente y la comida: Si la cabeza de la serpiente está dentro de la pantalla, se procede a dibujar la serpiente y la comida utilizando las funciones DibujarSnake() y DibujarComida().
3. Verificación de si la serpiente ha comido: Se verifica si la serpiente ha comido algo utilizando la función comer(). Si la serpiente ha comido, se generan nuevas coordenadas para la comida, y se manejan las entradas de botones y el control mediante las funciones darComida(), Butons(), y Control().
4. Manejo de entrada y control si la serpiente no ha comido: Si la serpiente no ha comido, se manejan las entradas de botones y el control utilizando las funciones Butons() y Control().

5. Verificación de la posición de la cabeza nuevamente: Se vuelve a verificar si la cabeza de la serpiente está dentro de la pantalla.
6. Manejo de colisiones y redibujado en pantalla: Si la cabeza está dentro de la pantalla, se verifica si la serpiente se ha tocado a sí misma mediante la función TouchedItself(). Luego, se limpia la pantalla y se vuelve a dibujar la serpiente y la comida en un bucle utilizando las funciones u8g.firstPage(), DibujarComida(), y DibujarSnake().
7. Manejo de fin de juego: Si la cabeza de la serpiente está fuera de la pantalla, se ejecuta la función Ded(), que probablemente se encarga de mostrar la pantalla de fin de juego.

Esto lo hice el lunes a las 3 AM, así que claramente será el final de la carpeta de campo.

Agradezco a mi mayor lector a lo largo de la historia: Consorti.