

Proyecto 3: Videojuego Portable

Carlos René Gil Monterroso (19443)
Ricardo Pellecer Orellana (19072)

29 de Abril del 2021

1. Circuito utilizado

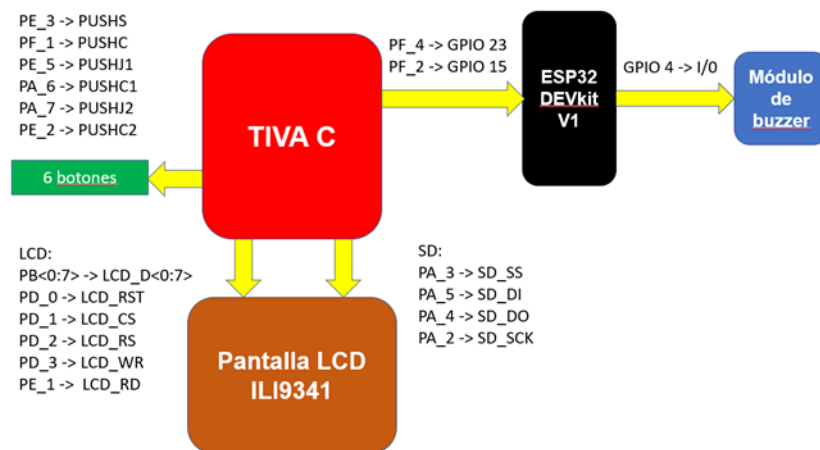
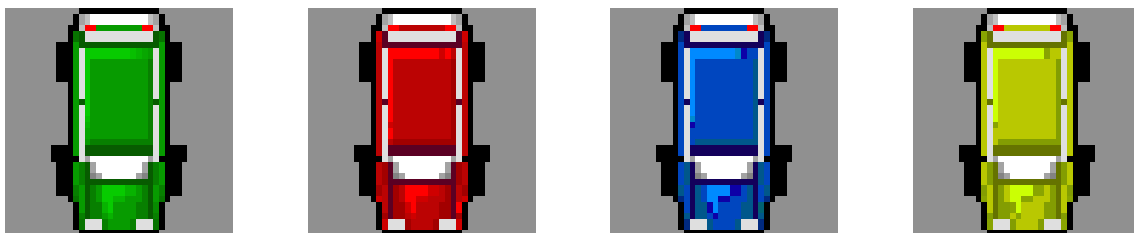


Figura 1: Circuitos Utilizados

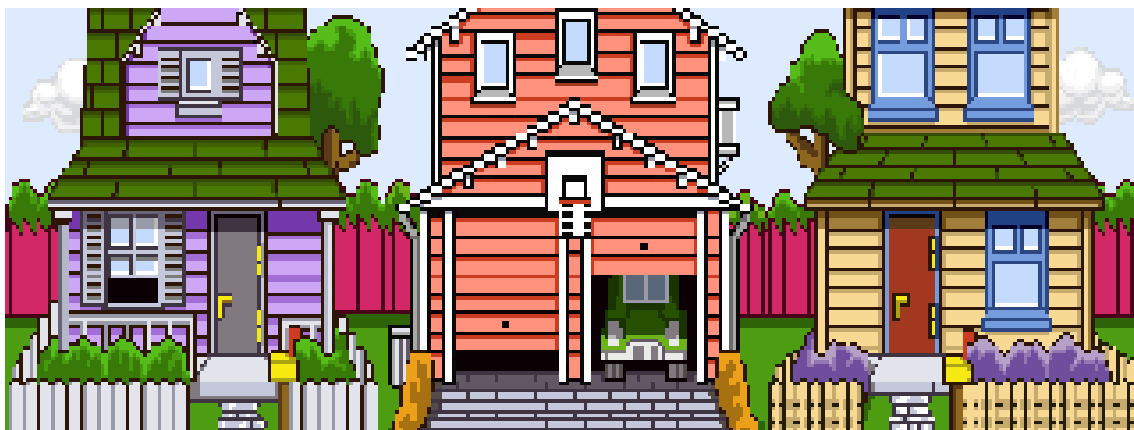
1.1. Subcircuitos

- **Control de Pantalla:** Este circuito este compuesto por la tiva C, 6 pulsadores y la pantalla LCD ILI9341. En este caso, la LCD ILI9341 tiene dos funcionalidades. La primera es como pantalla, para la cual tiene conectado 13 bits a la tiva C (8 bits de datos y 5 bits de control). Y la segunda es como lector de la memoria SD, el cual se conecta con 4 pines a la tiva C, los que corresponden a la comunicación SPI para leer y escribir en la memoria.
- **Sonido:** Este circuito este compuesto por la Tiva C, un ESP32 Devkit V1 y un módulo de buzzer. La tiva C se comunica con el ESP32 enviando dos bits en paralelo, los cuales indican que canción deberá reproducir el ESP con la señal PWM para el buzzer. Es importante mencionar que la razón de que la cantidad de bits enviados por la tiva sea 2, es debido a que el ESP32 debe enviar 3 señales diferentes al buzzer (2 para canciones y 1 para estar en silencio).

2. Gráficos



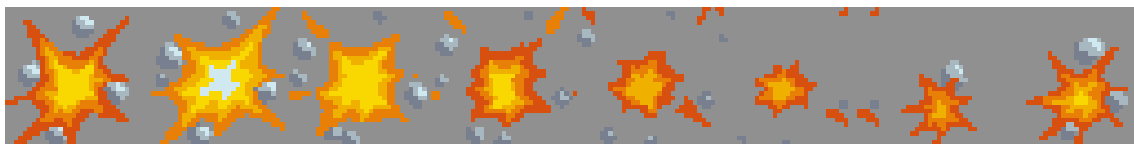
(a) Obstáculos



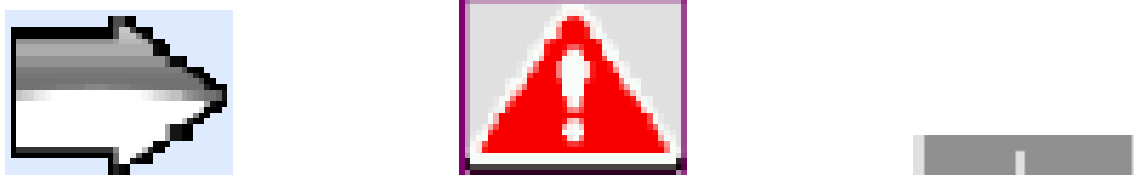
(b) Pantalla de Inicio



(c) Movimientos



(d) Efectos de choque



(e) Gráficos adicionales

Figura 2: Gráficos utilizados

3. Explicación de Funciones Generales

- **Selecciones:** Para realizar las selecciones se utilizó el gráfico de la flecha mostrado en la sección (e) de la Figura 2 y dos botones. Inicialmente, se pinta la flecha al lado de la primera selección deseada. Luego, con uno de los dos botones (conectado al pin PE 3 de la Tiva) se cambia la posición en que se dibuja la flecha y se borra con el color del fondo de pantalla el dibujo preexistente. Luego, con el segundo botón (conectado al pin PF 1 de la Tiva) se realiza la confirmación de la selección; una vez se presiona este botón, una variable de confirmación aumenta, haciendo que pase a la siguiente etapa del juego. Nótese que es importante que cada botón tenga su respectivo antirebote. Además, cabe mencionar que para agregarle dificultad al juego, se programó para que cada cierto tiempo aumente la velocidad con la que estos se mueven hacia el carro principal. Las selecciones que pueden hacerse por pantalla son:
 - Inicio: Comenzar
 - Selección de Modo: 1 Jugador o 2 Jugadores
 - Selección de Carro: Rojo o Azul para cada carro disponible
 - Post Partida: Volver a jugar, no volver a jugar y guardar información en la SD
- **Generación de Obstáculos:** La generación de obstáculos se basó en la generación aleatoria, tanto del color del carro obstáculo como el del carril en el que aparece. Para esta parte, fue importante reconocer que debía de reconocerse a cada vehículo obstáculo como un objeto con color y coordenadas para poder borrarlo y redibujarlo cada cierto tiempo para hacer la ilusión de que se mueve sin cambiar sus propiedades. Para esto, se utilizaron matrices, en las que se colocaron en la fila superior las coordenadas del carril en el que se encuentran y en la fila inferior los colores de los obstáculos codificados a números de 0 a 3. Nótese que para esta parte, que la matriz tuviera un valor de 0 en su coordenada significaba que no había carro, por lo que podía colocarse un nuevo carro obstáculo. Por estos motivos, la matriz definió el máximo de vehículos obstáculos que pueden haber al mismo tiempo -se escogió un máximo de 2, aunque fácilmente podría modificarse para ser más-.
- **Movimiento de los jugadores:** Para los movimientos se utilizaron dos botones por jugador -uno para moverse a la derecha y uno para la izquierda-. Para lograr el movimiento, al presionar un botón, se dibujaba el sprite del movimiento -sección (c) de la Figura 2-, luego se borró el dibujo y se redibujó en el carril siguiente, haciendo la ilusión que el carro cruzó. Es importante tomar en cuenta que se programó de forma que si se está en un carril extremo y se mueve hacia la orilla de la pantalla, el carro de la vuelta completa y aparezca en el otro extremo. Debe mencionarse que con el mismo sprite puede lograrse giros tanto a izquierda como derecha, a pesar de ser solo a la izquierda, gracias al parámetro de flip de la función LCD Sprite.
- **Bloqueo de carriles:** Para agregarle dificultad al juego, se programó para que cada cierto tiempo se vayan bloqueando carriles aleatorios hasta que solo queden 3 disponibles. Se escogió que queden tres para que le quede espacio al jugador de moverse a un lugar seguro aún cuando hay dos obstáculos. En el caso de dos jugadores sucede algo similar, solo que se limita a bloquear un carril por jugador, para que le quede un carril libre para moverse al jugador.
- **Perder:** Un jugador pierde cuando toca a un obstáculo o cuando está sobre un carril bloqueado. Para determinar si está en el mismo carril que un obstáculo o un bloqueo se utiliza la posición del carro principal y se compara con las posiciones de cada obstáculo guardado en su matriz y de cada bloqueo guardado en su variable. Cuando el carro pierde, se muestra la animación de explosión mostrada en la sección (d) de la Figura 2.
- **Guardado en memoria SD:** Para el modo de un jugador, conforme el tiempo pasa, el jugador acumula puntos. Estos puntos son reportados al finalizar la partida y dan la opción a guardarse en la SD al colocarse en la opción de guardado y presionar el botón de confirmación. Esta información es colocada en el archivo de texto High.txt. Por otro lado, para el modo de dos jugadores, no se registra un puntaje, sino quién es el que aguantó más sin chocar. De igual forma, se le da la opción a los jugadores de guardar esta información en la SD con el botón de confirmación en la selección de la pantalla post partida. Esta información se guarda en el archivo de texto 1v1.txt. Es importante mencionar que está programado para que la información solo pueda ser guardada una vez por partida, a pesar de que el botón se presione varias veces, para evitar la saturación de información repetida poco significativa.

4. Vídeo de Presentación y Repositorio

La explicación del trabajo se muestra también en el siguiente video: Proyecto 3: Videojuego Portable.

El repositorio en donde puede encontrarse el proyecto está en el siguiente link de GitHub: Proyecto 3: Videojuego Portable.