

Angie Marisela García Ríos
Estefany Cuervo Suárez
Juan Felipe Arroyave Zapata
Electrónica digital II

Índice

1. Identificación del proyecto	2
2. Resumen	2
3. Descripción del hardware	2
4. Descripción del software	3
5. Estructura del código	4
6. Explicación de funciones principales	4
7. Funcion Principal de Ejecucion	5
8. Funciones de Retroalimentación Visual	5
9. Retroalimentacion Auditiva	6
10.Retroalimentacion de Juego	6
11.Procedimiento de Prueba	6
12.Manejo de Errores y Seguridad	7
12.1. Prevención de Errores	7
12.2. Recuperación de Errores	7
12.3. Limitaciones y Consideraciones de Seguridad	7
12.4. Monitorización	8

1. Identificación del proyecto

- **Nombre del proyecto:** Informe de Proyecto - Juego de Obstaculos con Pantalla OLED SSD1306
- **Integrantes del equipo:** Angie Marisela García Ríos, Estefany Cuervo Suárez, Juan Felipe Arroyave Zapata

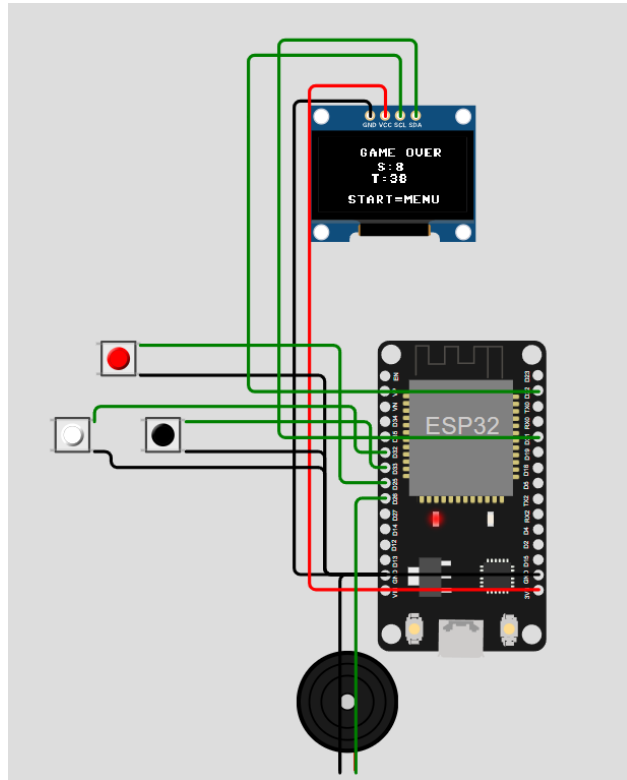
2. Resumen

Este proyecto consiste en el diseño e implementación de un minijuego estilo "Dodger", utilizando la pantalla OLED SSD130, que nos permita observar el minijuego y controlarlo para completarlo. A traves del codigo buscamos implementar diversos modos de juego y un sistema de puntuacion, ademas de asegurarnos que los controles para el minijuego funcionen correctamente para darle al usuario una interfaz para el uso del juego.

3. Descripción del hardware

- **ESP32** (modelo con ADC y pines GPIO)
- **Pantalla OLED SSD1306** (Display Monocromatico de 128x64 pixeles por I2C)
- **Pulsadores** para control conectados a los pines 25, 32 y 33.
- **Buzzer Pasivo** conectadas al pin 26 para la musica y efectos sonoros.
- **Protoboard y cableado** para conexiones y prototipado.

Diagrama de conexión:



Esquema de Wokwi

4. Descripción del software

- **Lenguaje usado:** MicroPython
- **Librerías:** machine, ssd1306, time, random
- **IDE:** Thonny y Wokwi

Flujo general del programa:

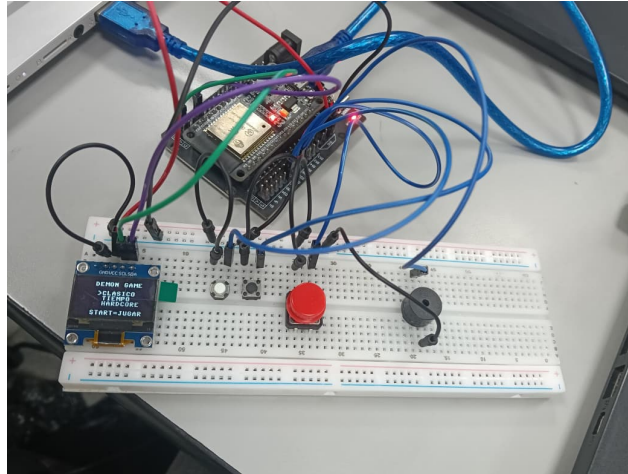
1. Configuración del hardware, los botones y el buzzer
2. Configuración para las variables del juego y la música y efectos de sonido
3. Código para asignar los obstáculos del juego, además del terreno
4. Establecimiento de la función del menú.
5. Configuración del juego y acciones del juego.

5. Estructura del código

- Archivo principal: `main.py`
- No se usan módulos adicionales.
- Código basado en un minijuego para evitar obstaculos acompañado con efectos de sonido y musica, ademas de diferentes dificultades.

6. Explicación de funciones principales

- `draw menu()`: Renderiza el menu principal con la seleccion de modos de juego.
- `draw game()`: Dibuja y recrea el juego en tiempo real con puntaje y obstaculos.
- `draw game over()`: Muestra la pantalla final con los resultados..
- `handle jump()`: Gestiona la fisica del salto con gravedad y velocidad.
- `handle duck()`: Controla el estado de agachado con temporizador.
- `draw player()`: Renderiza el personaje del jugador segun su estado.
- `draw obstacles()`: Renderiza los obstaculos segun su tipo y altura.
- `play menu music()`:Reproduce la cancion principal de juego en un bucle durante el menu.
- `jump sound()`: Efecto sonoro al saltar.
- `duck sound()`: Efecto sonoro al agacharse.
- `game start sound()`: Sonido al empezar el juego
- `beep()`: efecto de sonido de feedback.
- `reset game()`: Reinicia todas las variables del juego.



Evidencia visual del montaje en protoboard

7. Funcion Principal de Ejecucion

- Inicializacion con configuracion del hardware.
- Bucle Principal que se ejecuta continuamente cada 33ms o 30 FPS
- Lectura de Entradas
- Maquina de Estados que gestiona el menu y los estados de juego
- Fisica del Juego que actualiza la posicion del jugador y los obstaculos.
- Deteccion de Colisiones
- Renderizado que dibuja en el OLED segun el estado actual.
- Audio reproduce musica y efectos de sonido.
- Control de tiempo que gestiona temporizadores y dificultad.

8. Funciones de Retroalimentación Visual

- OLED que nos muestra el juego, el menu, el puntaje y el estado del juego.
- Sprites del jugador que cambia al agacharse o saltar, y los obstaculos
- Texto Informativo como los puntajes, el tiempo y los mensajes de estado.

9. Retroalimentacion Auditiva

- Musica Principal durante el menu principal.
- Efectos de acciones como sonidos al saltar o agacharse y al iniciar el juego.
- Sonido de Colision como un sonido grave al llegar al fin del juego.

10. Retroalimentacion de Juego

- Puntaje que incrementa al esquivar obstaculos.
- Velocidad Progresiva como dificultad que aumenta proporcional al puntaje.
- Temporizador que nos muestra el tiempo restante en modo contrarreloj,

11. Procedimiento de Prueba

1. Conectar el ESP32 al computador mediante cable USB.
2. Verificar que el OLED muestre el menu principal correctamente.
3. Confirmar que la cancion principal se este reproduciendo.
4. Presionar el boton DOWN y UP y verificar su funcion en pantalla y deberian hacer un beep.
5. Verificar que los 3 modos de juego sean seleccionables.
6. Probar el inicio del juego y verificar que el jugador aparece junto con el suelo.
7. Probar los controles basicos y la aparicion de obstaculos.
8. Comprobar que la pausa funciona correctamente al igual que la funcion de colision.
9. Asegurarse que el regreso al menu principal funciona y probar los 3 modos de juego.
10. Hacer una prueba de sesion extendida por al menos 5 minutos.
11. Probar el recovery al desconectar y reconectar la alimentacion ademas de verificar el reinicio correcto del menu.

12. Manejo de Errores y Seguridad

12.1. Prevención de Errores

- Validacion del OLED.
- Antirrebote de botones.
- Limites fisicos dentro del juego.
- Validacion de estados para transiciones controladas dentro del juego.

12.2. Recuperación de Errores

- Manejo de excepciones en funciones de audio y graficos.
- Reset completo con la funcion `reset game()` que reinicia todas las variables.
- Recuperacion de estado al seleccionar la pausa o resumen del juego.

12.3. Limitaciones y Consideraciones de Seguridad

- **Protección de hardware:**
 - ESP32 con memoria limitada para buffers grandes
 - Maximo 30 FPS por el procesamiento en micropython
 - El PWM puede causar interferencias con otras operaciones.
 - El OLED tiene una resolucion limitada de 128x64
- **Estabilidad del sistema:**
 - El valor ideal de frecuencia para balancear el sistema son 100 Hz
 - El valor del buffer es idealmente mantenido en 500 muestras
 - Preferiblemente se busca tener maximo 2 filtros activos simultaneamente
- **Consideraciones importantes:**
 - Usar solo 3.3V, nunca 5V o mas
 - El Buzzer no representa riesgo electrico
 - El ESP32 puede llegar a calentarse en sesiones prolongadas
 - Los botones no representan riesgos
 - El sistema depende de una alimentación estable por USB

12.4. Monitorización

- Metricas en tiempo real como FPS implicito
- Visualizacion directa de la grafica
- Gameplay a traves del puntaje, el tiempo y la velocidad.
- Audio segun la musica y los efectos de sonido.
- Sistemas con el estado de los botones y los obstaculos activos