



LCD-Pacman

Integrantes:

- Cazares Cruz Jeremy Sajid
- Bucio Barrera Oscar Daniel
- Perez Ortiz Saúl
- Acosta Cortes Gerardo

Análisis y Documentación del Código

1. Inclusión de Librerías:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

- `Wire.h`: Librería para la comunicación I2C, usada aquí para comunicarse con la pantalla LCD.
- `LiquidCrystal_I2C.h`: Librería para manejar pantallas LCD que utilizan I2C.

2. Inicialización de la Pantalla LCD:

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

- Crea un objeto `lcd` para la pantalla LCD con dirección I2C `0x27` y dimensiones de 16 columnas por 2 filas.

3. Definición de Caracteres de Pac-Man:

```
byte CharPACMANDERECHA[8] = {...};  
byte CharPACMANIZQUIERDA[8] = {...};
```

- Dos arrays que definen los caracteres personalizados para Pac-Man mirando a la derecha e izquierda. Cada array de 8 bytes representa un carácter de 8x8 píxeles.

4. Declaración de Variables y Constantes:

```
const int botonDerechaPin    = 2;  
const int botonIzquierdaPin = 3;  
const int botonReinicio     = 4;  
bool inicio = true;  
bool moviendoDerecha = true;  
int posicionActual = 0;  
int filaActual = 0;
```

- Establece los pines de los botones y declara variables para controlar el estado del juego, como la dirección de movimiento de Pac-Man y su posición actual.

5. Función `setup()`:

- Inicializa la pantalla LCD, establece los pines de los botones como entradas y muestra el mensaje inicial del juego.

```
void setup()  
{  
    lcd.init(); // Inicializa la pantalla LCD  
    lcd.backlight(); // Enciende la retroiluminación  
    lcd.createChar(0, CharPACMANDERECHA); // Crea el carácter
```

```

lcd.createChar(1, CharPACMANIZQUIERDA); // Crea el carácter personalizado para Pac-Man izquierdo
pinMode(botonDerechaPin, INPUT_PULLUP); // Configura el pin del botón derecho como entrada con pull-up
pinMode(botonIzquierdaPin, INPUT_PULLUP); // Configura el pin del botón izquierdo como entrada con pull-up
pinMode(botonReinicio, INPUT_PULLUP); // Configura el pin del botón de reinicio como entrada con pull-up
mostrarMensajeInicial(); // Muestra el mensaje inicial en la pantalla
}

```

6. Función `loop()`:

- Contiene la lógica principal del juego. Lee los estados de los botones para controlar el movimiento de Pac-Man y reiniciar el juego.

```

void loop() {
    // Si el juego está en el estado inicial...
    // ...y se presiona el botón derecho, empieza a mover a la derecha
    // ...y se presiona el botón izquierdo, empieza a mover a la izquierda
    // Si se presiona el botón de reinicio, muestra el mensaje inicial
}

```

7. Funciones Auxiliares:

- `mostrarMensajeInicial()`: Limpia la pantalla y muestra un mensaje inicial.
- `moverPacman(...)`: Controla el movimiento de Pac-Man en la pantalla, actualizando su posición y dirección.

Estructura y Funcionamiento

- El código se inicia con la inclusión de librerías necesarias para la comunicación con la pantalla LCD.
- Se definen caracteres personalizados para representar a Pac-Man.
- Se inicializan variables y constantes para los botones y el estado del juego.
- En `setup()`, se configura la pantalla LCD y los pines de los botones.
- La función `loop()` contiene la lógica del juego, respondiendo a los botones para mover a Pac-Man y reiniciar el juego.
- Funciones auxiliares ayudan en la visualización y control del juego.

Este código es un excelente ejemplo de cómo interactuar con hardware externo (pantalla LCD y botones) usando Arduino, y será muy útil para ilustrar la sección de implementación de software en tu reporte de práctica.

Código completo

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

byte CharPACMANDERECHA[8] =
{
    B01110,
    B11111,
    B11011,
    B11111,
    B11100,
    B11111,
    B11111,
    B01110
};

byte CharPACMANIZQUIERDA[8] =
{
    B01110,
    B11111,
    B11011,
    B11111,
    B00111,
    B11111,
    B11111,
    B01110
};
```

```

const int botonDerechaPin    = 2;
const int botonIzquierdaPin = 3;
const int botonReinicio     = 4;
bool inicio = true;
bool moviendoDerecha = true;
int posicionActual = 0;
int filaActual = 0;

void setup()
{
    lcd.init();
    lcd.backlight();
    lcd.createChar(0, CharPACMANDERECHA);
    lcd.createChar(1, CharPACMANIZQUIERDA);

    pinMode(botonDerechaPin, INPUT_PULLUP);
    pinMode(botonIzquierdaPin, INPUT_PULLUP);
    pinMode(botonReinicio, INPUT_PULLUP);

    mostrarMensajeInicial();
}

void loop() {
    if (inicio) {
        if (!digitalRead(botonDerechaPin))
        {
            inicio = false;
            moviendoDerecha = true;
            moverPacman(0, 0, true); // Comenzar desde la primera línea
        } else if (!digitalRead(botonIzquierdaPin))
        {
            inicio = false;
            moviendoDerecha = false;
            moverPacman(15, 1, false); // Comenzar desde la segunda línea
        }
    }
}

```

```

if (!digitalRead(botonReinicio))
{
    mostrarMensajeInicial();
    inicio = true;
    delay(500); // Retraso para evitar rebotes del botón
}
}

void mostrarMensajeInicial() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("____PACMAN____");
    lcd.setCursor(0, 1);
    lcd.print("PracticaEmbebido");
}

void moverPacman(int posicionInicial, int filaInicial, bool derecha)
{
    posicionActual = posicionInicial;
    filaActual = filaInicial;
    moviendoDerecha = derecha;

    while (true) {
        lcd.setCursor(posicionActual, filaActual);
        lcd.write(byte(moviendoDerecha ? 0 : 1));
        delay(300);
        lcd.setCursor(posicionActual, filaActual);
        lcd.print(" ");

        if (moviendoDerecha)
        {
            if (posicionActual < 15)
            {
                posicionActual++;
            } else if (filaActual == 0)
            {

```

```

        posicionActual = 0;
        filaActual = 1;
    } else
    {
        mostrarMensajeInicial();
        inicio = true;
        return; // Salir de la función
    }
} else
{
    if (posicionActual > 0)
    {
        posicionActual--;
    } else if (filaActual == 1)
    {
        posicionActual = 15;
        filaActual = 0;
    } else
    {
        mostrarMensajeInicial();
        inicio = true;
        return; // Salir de la función
    }
}

if (!digitalRead(botonReinicio))
{
    mostrarMensajeInicial();
    inicio = true;
    return; // Salir de la función moverPacman
}

// Cambiar la dirección si se presiona el botón contrario
if (moviendoDerecha && !digitalRead(botonIzquierdaPin)) {
    moviendoDerecha = false;
    delay(500); // Retraso para evitar rebotes del botón
}

```

```
    } else if (!moviendoDerecha && !digitalRead(botonDerechaPin))
        moviendoDerecha = true;
        delay(500); // Retraso para evitar rebotes del botón
    }
}
}
```

Resultados





