Jet Shooter Game using ESP32 and OLED

Aim:

The aim of this task is to design and implement a simple 2D shooting game using an ESP32 microcontroller, where a jet (player) shoots down incoming enemies on an OLED screen using push buttons for real-time interaction.

Description:

This task uses the ESP32 and a 128x64 I2C OLED display to build a minimalistic arcade-style shooter game. The player controls a triangular jet that can move left and right using push buttons and shoot bullets to destroy descending enemies. The game involves basic collision detection, object movement, and a reset functionality.

Features:

- 2D jet shooter game rendered on an OLED display using ESP32.
- Four push buttons:

  Move Left
  Move Right
  Shoot
  Restart Game
- Enemy block descends from the top of the screen.
- Player can shoot upward-moving bullets to destroy enemies.
- Collision detection between jet/enemy and bullet/enemy.
- "Game Over" screen on collision with enemy.
- Simple restart mechanism using a dedicated button.

Procedure:

Game Initialization:

- Set up the OLED display with an initial splash screen.
- Initialize jet and enemy positions.
- Configure all buttons as inputs.
- Display startup message before gameplay begins.

Game Logic:

- Use button presses to move the jet or shoot a bullet.
- Update the bullet position if active.
- Continuously move the enemy block downward.
- Check for collisions:
    - Bullet hits enemy → enemy resets.
    - Enemy hits jet → game over state triggered.

Display Refresh:

- Redraw game objects (jet, bullet, enemy) every frame.
- Clear display between frames for proper rendering.
- Show "GAME OVER" message upon collision.

Working:

- The ESP32 initializes the OLED screen and displays a startup message.
- The player's **jet** is represented by a triangle near the bottom of the screen.
- **Push buttons** let the player:

  Move the jet left or right (GPIO 34 and 32).
  Shoot bullets (GPIO 36).
  Restart the game (GPIO 33).

- **Bullets** move vertically upward when fired.
- **Enemies** appear randomly at the top and descend toward the player.
- If a bullet hits the enemy, the enemy is reset.
- If the enemy touches the player, a "Game Over" message is displayed.
- Pressing the restart button resets the game state.

Code:

```
#include <Wire.h>

#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>

// OLED config
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_ADDR 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Button GPIOs
#define BTN_LEFT    34    // Button 1: Move Left
#define BTN_SHOOT   36    // Button 2: Shoot
#define BTN_RESTART 33    // Button 3: Restart Game
#define BTN_RIGHT   32    // Button 4: Move Right

// Jet position
int jetX = SCREEN_WIDTH / 2;
int jetY = SCREEN_HEIGHT - 12;

// Bullet
int bulletX = -10, bulletY = -10;
bool bulletActive = false;

// Enemy
int enemyX = 0;
int enemyY = 0;
```

```
// Game state
bool gameOver = false;
unsigned long lastEnemyMove = 0;
unsigned long lastBulletUpdate = 0;

void resetGame() {
  jetX = SCREEN_WIDTH / 2;
  jetY = SCREEN_HEIGHT - 12;
  bulletActive = false;
  resetEnemy();
  gameOver = false;
}

void resetEnemy() {
  enemyX = random(0, SCREEN_WIDTH - 10);
  enemyY = 0;
}

void setup() {
  Serial.begin(115200);

  if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR)) {
    Serial.println("OLED not found");
    while (true);  // Halt if OLED isn't detected
  }
```

```
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Jet Game Starting...");
  display.display();
  delay(1000); // Just to show splash screen

  pinMode(BTN_LEFT, INPUT_PULLUP);
  pinMode(BTN_SHOOT, INPUT_PULLUP);
  pinMode(BTN_RESTART, INPUT_PULLUP);
  pinMode(BTN_RIGHT, INPUT_PULLUP);

  randomSeed(analogRead(0));
  resetGame();
}

void loop() {
  if (digitalRead(BTN_RESTART) == LOW) {
    resetGame();  // Restart game
    delay(300);   // Debounce delay
    return;
  }

  if (gameOver) {
    showGameOver();
```

```
  return;
}

// Movement
if (digitalRead(BTN_LEFT) == LOW && jetX > 0) jetX -= 5;       // Move left
if (digitalRead(BTN_RIGHT) == LOW && jetX < SCREEN_WIDTH - 10) jetX += 5; // Move right

// Shooting
if (digitalRead(BTN_SHOOT) == LOW && !bulletActive) {          // Shoot bullet
  bulletX = jetX + 5;
  bulletY = jetY;
  bulletActive = true;
}

// Move bullet
if (bulletActive && millis() - lastBulletUpdate > 50) {
  bulletY -= 4;
  lastBulletUpdate = millis();
  if (bulletY < 0) bulletActive = false;
}

// Move enemy
if (millis() - lastEnemyMove > 150) {
  enemyY += 3;
  lastEnemyMove = millis();
  if (enemyY > SCREEN_HEIGHT) resetEnemy();
```

```
  }

  // Bullet hits enemy
  if (bulletActive &&
      bulletX > enemyX && bulletX < enemyX + 10 &&
      bulletY > enemyY && bulletY < enemyY + 10) {
    bulletActive = false;
    resetEnemy();
  }

  // Enemy hits jet
  if (!gameOver &&
      jetX + 10 > enemyX && jetX < enemyX + 10 &&
      jetY + 10 > enemyY && jetY < enemyY + 10) {
    gameOver = true;
  }

  drawGame();
}

void drawGame() {
  display.clearDisplay();

  if (gameOver) {
    display.setTextSize(2);
    display.setCursor(20, 25);
```

```
    display.println("GAME OVER");
  } else {
    // Draw jet
    display.fillTriangle(jetX, jetY + 10, jetX + 5, jetY, jetX + 10, jetY + 10, SSD1306_WHITE);

    // Draw bullet
    if (bulletActive) {
      display.fillRect(bulletX, bulletY, 2, 5, SSD1306_WHITE);
    }

    // Draw enemy
    display.fillRect(enemyX, enemyY, 10, 10, SSD1306_WHITE);
  }

  display.display();
}

void showGameOver() {
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(15, 25);
  display.println("GAME OVER");
  display.display();
}
```

Output:

OLED Display:

- A triangle jet, rectangular bullets, and square-shaped enemy blocks.
- Real-time interaction and movement via push buttons.
- Smooth gameplay with visual feedback on hits and collisions.

Gameplay Events:

- Bullets shoot on button press and disappear if they miss or go offscreen.
- Game ends when enemy collides with the jet.
- Restart button immediately resets the game.