**LCD GAME**

**🕹️ Dino LCD Game on ESP32 – Complete Documentation**

**📌 1. Project Overview**

This project is a **side-scrolling survival game** on a 20x4 I2C LCD using an **ESP32**, a **joystick**, **buzzer**, **LED**, and **custom characters**. The player controls a dinosaur that dodges bombs and walls while collecting birds for points and extra lives.

**🔌 2. Circuit Connections (Excel Format)**

**(URL:-https://app.cirkitdesigner.com/project/ac8388aa-4a89-4900-a52d-bbf0698acdea)**

| **Component Name** | **Pin No** | **Destination Component** | **Pin No** | **Special Remark** |
| --- | --- | --- | --- | --- |
| Joystick X-axis | VRx (middle) | ESP32 | GPIO 34 | Analog read for left/right |
| Joystick SW | SW | ESP32 | GPIO 33 | Digital read with INPUT\_PULLUP |
| Buzzer | + | ESP32 | GPIO 25 | Controlled with tone() |
| Buzzer | - | ESP32 GND | GND |  |
| LED (Indicator) | + | ESP32 | GPIO 26 | Blinks on collision |
| LED (Indicator) | - | ESP32 GND | GND |  |
| I2C LCD | SDA | ESP32 | GPIO 21 | I2C communication |
| I2C LCD | SCL | ESP32 | GPIO 22 | I2C communication |
| I2C LCD | VCC, GND | Power | 5V, GND | 5V I2C LCD module |

**🧠 3. Core Features**

* **Player:** Controls a dino character using a joystick.
* **Obstacles:** Moving bombs and walls from right to left.
* **Target (Bird):** Collect birds for score and bonus lives.
* **Game States:** Tracks score, lives, game over condition.
* **Feedback:** Buzzer and LED signals on collisions and rewards.

**🧩 4. Game Elements (Custom Characters)**

| **Character** | **LCD Char** | **Description** |
| --- | --- | --- |
| Dino | byte(0) | Player character |
| Bomb | byte(1) | Hazard, lose life |
| Bird | byte(2) | Bonus, gain score |
| Wall | byte(3) | Hazard, lose life |

**🎮 5. Gameplay Mechanics**

**✅ Controls:**

* **Left/Right:** Move horizontally using joystick X-axis.
* **Jump (Up/Down):** Toggle row using joystick button (SW).

**✅ Objective:**

* **Avoid bombs & walls** — if collided, you lose a life.
* **Catch birds** — gain 1 score point and possibly gain life.

**✅ Game Progression:**

* Every **bird caught** gives 1 score.
* Every **3 birds** → gain 1 life (max 3).
* Every **5 birds** → game speed increases (faster movement).

**🕹️ 6. Code Breakdown**

**setup()**

* Initializes LCD, pins, custom characters.
* Seeds randomness using analog noise from joystick SW pin.

**loop()**

* Regularly handles:
  + Movement reading (handleMovement)
  + Object animation (moveObjects)
  + Collision detection (checkCollision)
  + Display update (drawGame)
* On game over: Shows final screen and halts.

**🧮 7. Game Variables**

| **Variable** | **Purpose** |
| --- | --- |
| playerX, playerY | Position of the dino |
| bombX, bombY | Bomb coordinates |
| birdX, birdY | Bird coordinates |
| wallX, wallY | Wall coordinates |
| score | Player score |
| lives | Player lives |
| gameOver | Game over flag |
| moveDelay | Game speed control |

**🧪 8. Functions Summary**

| **Function** | **Description** |
| --- | --- |
| drawGame() | Updates LCD with game objects and status |
| moveObjects() | Animates obstacles and targets from right to left |
| checkCollision() | Checks for hits and applies effects |
| handleMovement() | Reads joystick input and moves player |

**🔔 9. Enhancements Ideas**

* Add a restart feature using a reset button.
* Store high score in EEPROM.
* Add background music or effects.
* Introduce power-ups or time-limited shields.

**⚠️ 10. Notes**

* Game runs only while gameOver == false.
* Uses tone() for sound; only one tone at a time.
* Be sure your I2C LCD address is 0x27 (verify with I2C scanner).
* Adjust analogRead() thresholds if joystick is too sensitive.