Game Documentation: Dino Game on Arduino

Overview

The Dino Game on Arduino is a simple game implemented using an Arduino board and an LCD display. The objective of the game is to control a dinosaur character to jump over trees displayed on the LCD screen, while avoiding collisions. The game utilizes buttons for player input and provides a score system to track the player's progress.

Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It consists of a microcontroller board and a development environment, making it accessible for beginners and experienced users alike. Arduino boards are widely used for prototyping and building various electronic projects, including games.

Components Required

- Arduino board (e.g., Arduino Uno)
- LCD display compatible with Arduino
- Push buttons for user input
- Jumper wires for connections

Game Description

Upon starting the game, a menu is displayed on the LCD screen, allowing the player to select different options. The available options are "START" and "SCORE." The player can navigate through the menu using the push buttons. Pressing the "START" option initiates the game, while selecting "SCORE" displays the high scores achieved in previous games.

During the game, the dinosaur character (represented by a custom bitmap) moves from left to right on the LCD screen. The player's task is to make the dinosaur jump over the randomly generated tree obstacles (also represented by a custom bitmap) by pressing the corresponding button at the right time. Colliding with a tree ends the game.

The player's score is displayed on the LCD screen during gameplay. The score increases continuously as the game progresses, indicating the player's performance. When the game ends, the final score is shown, and the player is prompted to enter their name using the buttons. The name and score are then saved in a score list.

Code Explanation

The provided code consists of an Arduino sketch that implements the game functionality. Here's an overview of the key components and functions:

Libraries

- **LiquidCrystal**: This library provides functions to control the LCD display.

Custom Bitmaps

- **dino**: Represents the bitmap for the dinosaur character.
- **tree**: Represents the bitmap for the tree obstacle.

Variables and Constants

- **BUTTON_ENTER** and **BUTTON_SELECT**: Pin numbers for the buttons used in the game.
- **MENU_SIZE** and **LCD_COLUMN**: Constants for menu options and LCD display settings.
- **TREE_CHAR** and **DINO_CHAR**: Custom character codes for the tree and dinosaur bitmaps.
- **ALPHABET**: An array of strings representing the alphabet for entering player names.
- Various boolean and integer variables for game state and score management.

Functions

- **setup()**: Initializes the LCD display, creates custom characters, and sets up button pins.
- **loop()**: The main loop of the game, responsible for clearing the LCD and handling the menu.
- **handleMenu()**: Displays the menu and handles menu navigation and option selection.
- **showScore()**: Displays the high scores and allows navigation through the scores using buttons.
- **printScore()**: Prints a specific score and the next score on the LCD.

- **`startGame()`**: Initializes the game state and enters the game loop.
- **`handleGame()`**: Manages the game logic, including tree movement, score update, collision detection, and game over conditions.
- **`handleGameOver()`**: Displays the game over screen and prompts the player to enter their name for score saving.
- **`saveScore()`**: Saves the player's name and score in the score list.
- **`showTree()`**: Displays a tree obstacle at the specified