LED Number Guessing Game

This project is a simple number guessing game designed to run on the emu8086 microprocessor emulator. The game uses an LED display (simulated by writing to a specific port) to provide feedback to the player. It challenges players to guess a randomly generated number within a specified number of attempts.

Features

- **Initialization:** Sets up the LED display and prepares the game environment.
- **User Interaction:** Prompts the user to enter the number of attempts they wish to have, ensuring it falls within an acceptable range (1-9).
- **Gameplay:** Players are asked to guess a number between 1 and 9. The game provides feedback for each guess via the LED display and text messages, indicating success or instructing on further attempts.
- **End Game Conditions:** The game concludes either when the player correctly guesses the number or exhausts all attempts. A special message is displayed in each case, and the correct number is shown on the LED display at game over.
- **Input Validation:** Ensures that only valid inputs are accepted for both the number of attempts and the guesses. Invalid inputs trigger error messages and prompt for re-entry.

How to Run

1. **Setup:** Open the emu8086 microprocessor emulator.
2. **Load Program:** Copy and paste the provided code into the emu8086 editor.
3. **Compile:** Use the emulator's compile option to compile the code.
4. **Execute:** Run the compiled program within the emulator.

Controls

- **Number of Attempts:** Enter a single digit (1-9) when prompted for the number of attempts.
- **Guessing:** Enter your guess (a single digit, 1-9) when prompted during the game.
- **Restarting the Game:** To play again, recompile and execute the program as described above.

Troubleshooting

Unexpected or Invalid Inputs

During the gameplay, the program expects numeric input within a specified range (1-9) for both the number of attempts and the guesses. Here's how the program handles unexpected inputs:

- **Number of Attempts:** If the input for the number of attempts is outside the range 1-9 or is not a numeric value, the game will display an error message: "Invalid number of attempts. Please enter a value between 1 and 9." It will then prompt the user to re-enter a valid number of attempts. This ensures that the game does not proceed without a valid input.

- **Guessing Phase:** Similar validation applies to the guessing phase. If a guess is not a single digit within the 1-9 range, the game displays a message: "Any

input other than a digit you enter will prevent you from continuing." This prevents the game from accepting invalid guesses and ensures the integrity of the gameplay.

Corrective Actions

If you encounter these error messages, perform the following actions:

1. **Ensure Numeric Input:** Make sure to enter a single digit (1-9) as instructed by the game prompts. Avoid any letters, symbols, or multiple digits.
2. **Retry Input:** Upon receiving an error message, carefully read the prompt and provide a valid input as per the game's instructions.
3. **Restart the Game:** If for any reason the game does not accept valid inputs or behaves unexpectedly, consider restarting the game. To do this, compile and execute the program again within the emu8086 emulator.

Note

This section should help users understand how the program deals with unexpected or invalid inputs and what actions they can take to resolve any issues encountered.

Credits

This game was developed as a project for the microprocessors course conducted in the computer engineering department of Çukurova University and demonstrates an LED-integrated guessing game that can tolerate user input errors designed on the emu8086 microprocessor emulator.