PROJECT ARDUINO CESAR NIETO GONZALEZ

***-Project’s assumptions***

The main part of the program orchestrates the functionality of a digital sliding puzzle game implemented on an Arduino platform. The game involves an initial setup phase where communication with the user is established through a serial interface (UART), and a Liquid Crystal Display (LCD) is initialized. A welcoming message briefly appears on the LCD.

The puzzle board is then set up with a predefined configuration, representing an initial scrambled state. Each tile on the board is labeled with a letter ('A' to 'I'), and one tile is intentionally left empty, allowing for movement.

The program enters a continuous loop, waiting for user input via the UART interface. Users can send commands ('W', 'A', 'S', 'D') to manipulate the empty tile and attempt to rearrange the tiles into the correct order.

Upon receiving a valid command, the program updates the puzzle board and refreshes the LCD to display the current state of the puzzle. The game continuously checks if the puzzle has been successfully solved after each move. If the puzzle is complete, a victory message is displayed on the LCD, prompting the player to press a button to restart the game.

The game restart involves resetting the puzzle to a new random configuration, and the LCD is cleared to begin a new round of the sliding puzzle challenge. In essence, the main part of the program manages user interaction, puzzle state updates, victory conditions, and game resets, providing an engaging and interactive experience for the player.

***-Schematic of system***

Imagen de la pantalla de un celular con la imagen de un video juego

Descripción generada automáticamente con confianza baja

-***Discussion of implementation***

This Arduino code orchestrates a sophisticated and immersive sliding puzzle game, seamlessly blending the intricacies of hardware components and software functionalities inherent to the Arduino ecosystem. It serves as a testament to the versatility of Arduino boards, leveraging GPIO manipulations, UART communication for user input, and the seamless interfacing with an LCD display, powered by the HD44780 controller.

At its core, the code revolves around key functions meticulously designed to govern distinct aspects of the sliding puzzle game. The `moveTile` function takes center stage, deftly choreographing the movement of puzzle tiles in response to user commands ('S', 'A', 'W', 'D'), ensuring a responsive and engaging interaction paradigm. Meanwhile, the `isGameComplete` function introduces an element of challenge, determining whether the puzzle has reached a winning configuration and providing players with a sense of accomplishment.

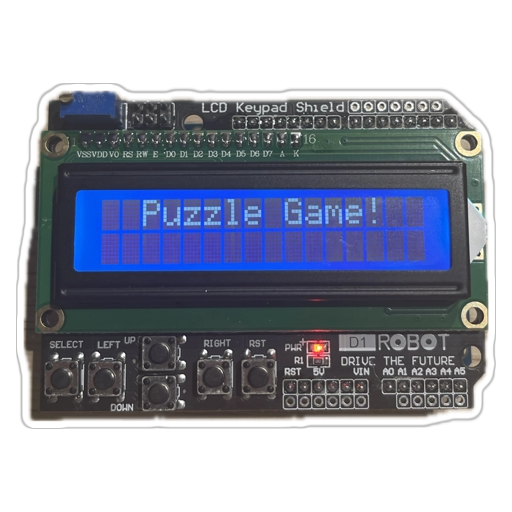
The initial setup phase unveils the intricacies of hardware initialization. The code establishes UART communication channels and configures the LCD screen, laying the foundation for a rich and dynamic user experience. A welcoming message on the LCD screen adds a touch of personalization, creating an inviting atmosphere for players.

The main loop, encapsulated within `while (1)`, serves as the heartbeat of the entire program. It tirelessly awaits user input through UART, allowing players to manipulate puzzle tiles in real-time. The LCD screen becomes an ever-evolving canvas, dynamically reflecting the current state of the puzzle board, delivering instant and visually immersive feedback.

In the realm of victory, the code gracefully handles the scenario. A triumphant message graces the LCD screen, prompting the user to press a key. This strategic pause in the gameplay allows for a celebratory moment before gracefully resetting the puzzle to a new and randomized configuration, ushering in a fresh challenge.

This comprehensive code intricately weaves together the complexities of managing user input, orchestrating puzzle state transitions, and delivering visual feedback through a seamless marriage of UART communication and LCD display interfacing. It transforms an Arduino board into a captivating gaming platform, underscoring the boundless versatility and adaptability of Arduino in the creation of engaging and interactive applications.

***-Pictures***

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

***-Link to gitHUB***

<https://github.com/Kaesaril/ARDUINOPROJECTCESARPK>

***-Summary***

In conclusion, this Arduino code manifests the convergence of hardware and software prowess to bring forth an engaging sliding puzzle game. It artfully integrates Arduino's capabilities, encompassing GPIO manipulation, UART communication, and LCD interfacing. Key functions govern tile movements and game completion, while the setup phase establishes a personalized environment.

The main loop forms the heartbeat, awaiting user input and dynamically updating the LCD display, fostering an immersive experience. Victory prompts a celebratory message, seamlessly transitioning to a new puzzle configuration. Ultimately, this code exemplifies Arduino's adaptability, transforming a microcontroller into a captivating gaming platform.