**Software Requirements Specification**

**for**

**Puzzle Game**

**Prepared by**

***(Leader)* 2023-BS-AI-020 : Mishal Nadeem**

**2023-BS-AI-015 : Eman Tahir**

**2023-BS-AI-021 : Yumna Irfan**

**2023-BS-AI-022 : Abeer Awais**

Table of Contents

1. **Introduction ........................................................................................................ 3  
   1.1 Purpose ......................................................................................................... 3  
   1.2 Intended Audience ....................................................................................... 3  
   1.3 Project Scope ............................................................................................... 3  
   1.4 References .................................................................................................... 3**
2. **Related Work ...................................................................................................... 3**
3. **Technology ......................................................................................................... 4  
   3.1 Tools .............................................................................................................. 4  
   3.2 Libraries ........................................................................................................ 4  
   3.3 Operating Environment ............................................................................... 4**
4. **Methodology ....................................................................................................... 4  
   4.1 Modules ........................................................................................................ 4  
   4.2 Project Features ........................................................................................... 4**
5. **Flowchart ............................................................................................................ 5**
6. **Specific Requirements ...................................................................................... 5  
   6.1 Functional Requirements ............................................................................ 5  
   6.2 Performance Requirements ........................................................................ 5  
   6.3 Interface Requirements ............................................................................... 5  
   6.3.1 User Interface ............................................................................................ 6  
   6.3.2 Hardware Interface .................................................................................... 6  
   6.3.3 Software Interface ..................................................................................... 6**
7. **Benefits ............................................................................................................... 6**
8. **Disadvantages .................................................................................................... 6**
9. **Conclusion ......................................................................................................... 7**

**1. Introduction**

**1.1 Purpose**  
The purpose of this software is to implement a sliding puzzle game, where the player can interactively move tiles on a 4x4 grid to reach the goal state. The game allows the player to make moves, undo previous actions, and check if the puzzle has been solved.

**1.2 Intended Audience**  
The target audience for this software includes:

* Developers and programmers interested in implementing simple puzzle games.
* Users who enjoy solving sliding puzzle games and wish to interact with a console-based application.

**1.3 Project Scope**  
This software will simulate a sliding puzzle game with the following features:

* A 4x4 grid puzzle.
* A user interface that allows moves using keyboard input.
* The ability to undo moves.
* A check to determine if the puzzle is solved.

**1.4 References**

* C++ Programming Language Documentation: <https://en.cppreference.com/w/>
* Stack Data Structure: <https://en.wikipedia.org/wiki/Stack_(abstract_data_type)>

**2. Related Work**

The concept of sliding puzzle games, also known as "15-puzzle," has been implemented in various forms across platforms. Many of these implementations use similar approaches, such as using a 2D array for the puzzle grid and employing a stack for undo functionality. Previous implementations have focused on making the game interactive through graphical user interfaces (GUIs), while the current project is a simple text-based console application for learning purposes.

**3. Technology**

**3.1 Tools**

* **C++ Compiler**: A C++ compiler (e.g., GCC, Clang) will be used to compile the program.
* **Text Editor**: Visual Studio Code, Sublime Text, or any other text editor to write and edit the source code.

**3.2 Libraries**

* The program uses the standard C++ libraries such as iostream for input/output operations.

**3.3 Operating Environment**

* The software can run on any operating system with a C++ compiler installed (Windows, Linux, macOS).
* The software is designed to work in a console or terminal environment.

**4. Methodology**

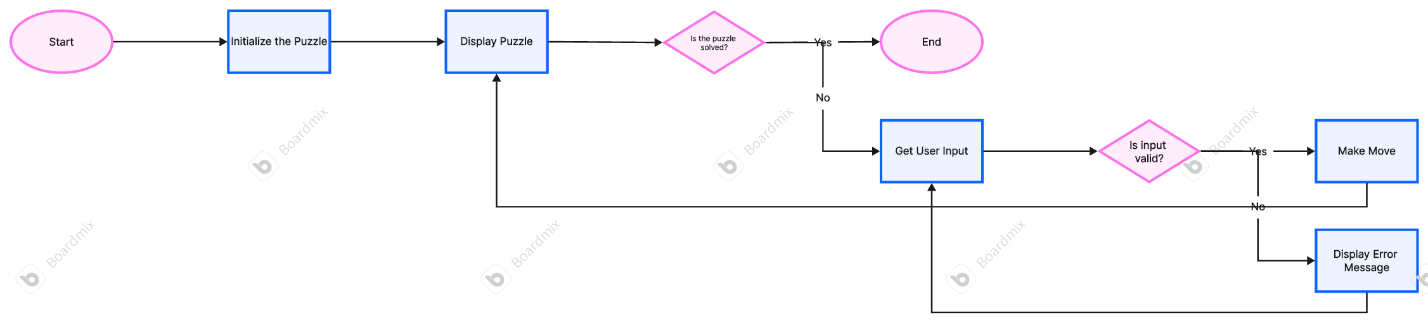
**4.1 Modules**  
The project consists of several modules:

* **Main Module**: Coordinates the game logic and handles user input.
* **Move Management Module**: Handles the logic of stacking moves and undoing them.
* **Puzzle Display Module**: Responsible for printing the puzzle grid to the screen.
* **Game Logic Module**: Contains the functions for validating moves, checking the solved state, and updating the puzzle.

**4.2 Project Features**

* **Sliding Puzzle Game**: The player can move tiles around the 4x4 grid using keyboard inputs (w/a/s/d for movement).
* **Undo Functionality**: The player can undo the last move using the "u" key.
* **Puzzle Solver Check**: The game checks if the current puzzle state matches the goal configuration.

**5. Flowchart**



**6. Specific Requirements**

**6.1 Functional Requirements**

* **Game Initialization**: The system must initialize the puzzle grid with a given start state and a goal state.
* **Move Functionality**: The system must allow the player to move tiles in four possible directions (up, down, left, right).
* **Undo Functionality**: The system must provide the ability to undo the last move.
* **Puzzle Solving Check**: The system must check if the puzzle has reached the goal configuration.

**6.2 Performance Requirements**

* The system should be responsive to user inputs with minimal delay.
* The game should run efficiently without crashing for normal gameplay durations.

**6.3 Interface Requirements**

**6.3.1 User Interface**

* The interface should display the puzzle grid in a readable format, with blank spaces representing the empty tile.
* The system should prompt the user for valid input (e.g., 'w', 'a', 's', 'd', 'u').

**6.3.2 Hardware Interface**

* No specific hardware interfaces are required beyond basic keyboard input for the user.

**6.3.3 Software Interface**

* The system requires the C++ compiler to compile the code and execute the program.
* The system relies on the C++ standard library for input/output operations and memory management.

**7. Benefits**

* **Educational Value**: Users can learn about puzzle-solving algorithms and stack-based data structures.
* **Simple and Accessible**: The text-based interface is easy to use and doesn't require additional graphical resources.
* **Fun and Interactive**: Players can engage with the puzzle in an interactive manner and challenge their problem-solving skills.

**8. Disadvantages**

* **Limited Graphics**: The lack of a graphical user interface may make it less engaging for some users compared to visual puzzle games.
* **Fixed Grid Size**: The grid size is fixed to 4x4, which may limit replayability.

**9. Conclusion**

This puzzle game provides an interactive and educational experience for users interested in solving sliding puzzles. While simple, it demonstrates core programming concepts such as stack data structures, game logic, and user interaction through the console. Future versions could improve the user experience by adding graphics or extending the game with larger grids.