ES1101- Introduction to Programming

Mini Project

ARRANGE PUZZLE

Team Number- 40

1. CS21B028- K. Sai Vignesh
2. CS21B051- V. S. K. S. S. Narayana Rao
3. CS21B052- V. Nikhil Chowdary
4. EE21B004- A. Jaya Prakash
5. CS21B011- B. Tharun

**Project Description: -**

Our Project is an Arrange Puzzle Game for the grids of size ranging from 2x2 to 9x9. Initially the input n is taken from user and accordingly numbers ranging from 1 to n2-1 and a space are randomly displayed in a nxn grid. Now, using w, a, s, d keys, the user should move the space to finally arrange the numbers in ascending order with space at the last to win. ‘w’ is used to move the space up, ‘s’ to move it down, ‘a’ to move it left and ‘d’ to move it right. Number of moves is also displayed corresponding to every move. “Invalid Move” is also displayed when the move by the user is not a valid move.

For example, if the input given by user is 3, then a grid like following is displayed: -

|  |  |  |
| --- | --- | --- |
| 5 | 2 | 7 |
| 3 | 1 | 6 |
| 8 | 4 |  |

Now, if the given input is ‘w’, the grid changes to: -

|  |  |  |
| --- | --- | --- |
| 5 | 2 | 7 |
| 3 | 1 |  |
| 8 | 4 | 6 |

If we give input as ‘s’, it displays “Invalid Move”.

If we give input as ‘a’, the grid changes to: -

|  |  |  |
| --- | --- | --- |
| 5 | 2 | 7 |
| 3 |  | 1 |
| 8 | 4 | 6 |

The game ends once the grid is in right order: -

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 |  |

**Input-Output Formats: -**

Input: -

1. Grid Size (Difficulty level) - Integer
2. Navigation Keys - Character – w, a, s, d – Navigates the space button

Output: -

1. New Modified Grid
2. Number of Moves
3. Invalid Move (If the move is not valid)

**Core Functions: -**

1. swaping(int grid[][10], int initial x, int initial y, int final x, int final y)- swaps given initial and final numbers in the grid
2. randomize(int arr[], int n)- randomizes given array
3. correct(int grid[][10], int n)- checks if the current array is the required array to win or not
4. valid(int n, int x, int y, char input)- checks whether the move is valid or not
5. display(int grid[][10], int n)- displays the grid
6. functionality(int grid[][10], char input, int n, int moves)- takes the character input and swaps the elements in array if the move is valid and updates the number of moves accordingly

**Data Formats: -**

1. The Size of Grid - Integer
2. The Grid – 2D Integer Array
3. Given Input- Character
4. Total Number of Moves - Integer

**Challenges: -**

We found it difficult to randomize the numbers in the grid. We overcame it by using some C++ references available online. We also found it difficult to display the two-digit numbers as characters in order to display the “space”. We overcame it by taking all entries as integers and storing the space as 0 in the array and print space wherever 0 is available.