Aim:

The objective of this assignment was to write a program using object-oriented principles in C++. The program written was the game 2048 with additional rules.

Design:

The game we were required to make consists of functions that will be used frequently. The classes I chose to implement were: Tiles, Game, HallOfFame and PlayerScore.

Tiles

This class consists of one private member variable m_value. This variable is used to keep track of the value inside the tile. The constructor will then initialize this value to zero. Getter and setter methods were used in order to access the member variable. This class provides two other methods, one being the padding method which will apply the appropriate number of spaces before the value of the tile. This was done by checking whether a number is between a certain range, for example between 1 and 9 and would add 3 spaces before the number as the value only consists of one number. This method also displayed the mine tile as an * symbol and the empty tile 0 as no value. The second and final method was the makeTile method which took an int as a parameter. This method generated a random value to be put into a tile. These values were 2, 4 and -1 which symbolizes the mine tile. The input parameter was used to toggle the mine tile as when the game starts there should be no mine tile and this ensures that a mine tile is never displayed in the beginning.

Game

This class is where the main implementation of the game is done. The Game class is derived from the base class Tiles. Two private member variables are in this class, m_score which keeps track of the current score and a four by four array of Tile objects. The constructor initializes the score to 0. The score also has its own setter and getter methods.

- Board Functions
  The functions pertaining to the board are drawBoard which will output the layout of the board and add the tiles according to their position. As this class is derived from the base class Tiles the padding function was used in order to make sure the values were in the appropriate positions.
  The addTile function will as the name implies add a tile on the board. This method will make use of a function from the base class, the makeTile function so that a random number is generated in that position of the board.
- Menu
  This function is used to call the sliding functions using a switch case statement. Since a switch case statement was used the function takes as parameter a char. Each time a slide function is called a tile is also added by calling the addTile function.
- Move Functions ○ moveLeft

This function will slide all the tiles to the left side of the board. It will first loop through the rows from zero till three then the columns will be looped from one till three, these parameters were chosen as the column not considered in the for loop condition will be catered for later on in the code. If the current value is zero, then it will go on to the switch case statement. If the adjacent value is zero (empty), since the current value is non-zero it will be put in the adjacent tile. If the adjacent value is a mine tile, then the current value and the adjacent value are set to zero as per the special rules of the game. For the default case a lastCombined value is used to determine how many tiles got added together. This is used so that if there are three or more tiles next to each other with the same value only two tiles are added at a time. The tiles on the further most left are added first. Then in the case that the current value is a mine tile then the tile preceding the mine gets set to zero as well as the mine tile itself. The score for when tiles get added are calculated by adding the current score to the value of the added tiles. The score for when the mine tile is met is calculated by deducting the current score by the value of the tile that got destroyed. ○ <u>moveRight</u>
The moveRight function is similar to the moveLeft function, the only differences lie in the for loops and the adjacent tile. The rest of the function operates similarly to the moveLeft function.

  - <u>moveUp</u>
    As stated above the functionality is rather similar to the previous two functions, however the difference is in the for loops and the adjacent tile.
  - <u>moveDown</u>
    Once again the functionality of this function is similar to the last three functions and the differences are in the for loops and the adjacent tile.

- Changing index formats used for saving and loading ○ <u>oneDtoTwoD</u>
  This function is used to convert a 1-D array index into a 2-D array index. This is done by taking as input the 1-D array index and dividing it by how many rows there are. As it will be saved as an int the decimal part will be discarded and the number before the decimal is the row position. To get the column position we instead use the modulo and we will obtain the remainder which corresponds to the column position.
  - <u>twoDtoOneD</u>
    This function is used to convert a 2-D array index into a 1-D array index. This is obtained by taking the row position and multiplying it by how many rows there are then adding the column position to it.
- Additional commands ○ <u>Save</u>
  This function is used to save the current board's layout. The user inputs the name of the file where it will be saved, then the name will be concatenated with ".txt" to save it as a text file. Then each time this function is called it will delete any data it had prior and save the new data. If the file is open, initially it will save the score then it will loop through the game board to find non-

zero tiles and convert the 2-D index into a 1-D index and save the 1-D index as well as the value of the tile. If the file was not opened, then an error message will be displayed.

- <u>Load</u>
  This function will load the tile values and their placements as well as the user's score. After the user inputs the name of the saved file the name will be concatenated with ".txt". Then when the file is open since the first line is always the score the program will use substrings to extract the user's score. Then using a while loop it will get the rest of the lines and extract the data required and set the values accordingly using functions like stoi which will convert the string into an integer.
- <u>newGame</u>
  This function will remove all current non-zero values and replace them with zeros so that a brand new game can be played. The score is also set to zero.

- Lose or Win ○ <u>fullBoard</u>
  This function will loop through the matrix and check whether the tile has a value greater than zero, i.e. there are no mine tiles. Since the board has 16 tiles a counter was used so that the entire board is looped through and then a
  Boolean is returned.
  - <u>checkNegative</u>
    This will check if the adjacent up and left tiles are not equal to the current tile. It will also check if the current value is a mine tile or an empty tile.
  - <u>checkPositive</u>
    This will check if the adjacent down and right tiles are not equal to the current tile. This will also check if the current tile is a mine tile or an empty tile.
  - <u>checkEndGame</u>
    This will determine if the player has either lost or won. The purpose of the checkNegative and checkPositive was to make sure that the player cannot perform any other moves which means they have lost. Then it will loop through the board to check if a tile has the value 2048, which means the player won. If neither of those were satisfied then it will continue with the game.

<u>PlayerScore</u>

This class is used for the hall of fame. Since to display the hall of fame the scores have to be in descending order a priority queue was used. To add scores and names into the queue the < operator was overloaded in this class.

The member variables in this class were m_name and m_score and are public to be used with the priority queue later on.

The constructor initialized the member variables. These member variables had getter and setter methods.

HallOfFame

This class inherits functions from the Game class as well as the PlayerScore class. The constructor initializes the PlayerScore to an empty string and zero. This class has two methods, one to save the hall of fame and one to display the hall of fame.

- saveHallofFame

  If a user wins the game they will be prompted to enter their name into the hall of fame. A colon is concatenated to their name and the score is placed at the end.

- displayHall

  This will display the hall of fame. A priority queue object was created, then if the hall of fame file is opened the function will extract the name including the colon and the score. Then the score will be converted into an integer and the setter methods from the PlayerScore class are called. The name and score are then pushed onto the queue and a counter keeps track of how many lines have been read. Then using this counter was used to display the hall of fame in descending order.

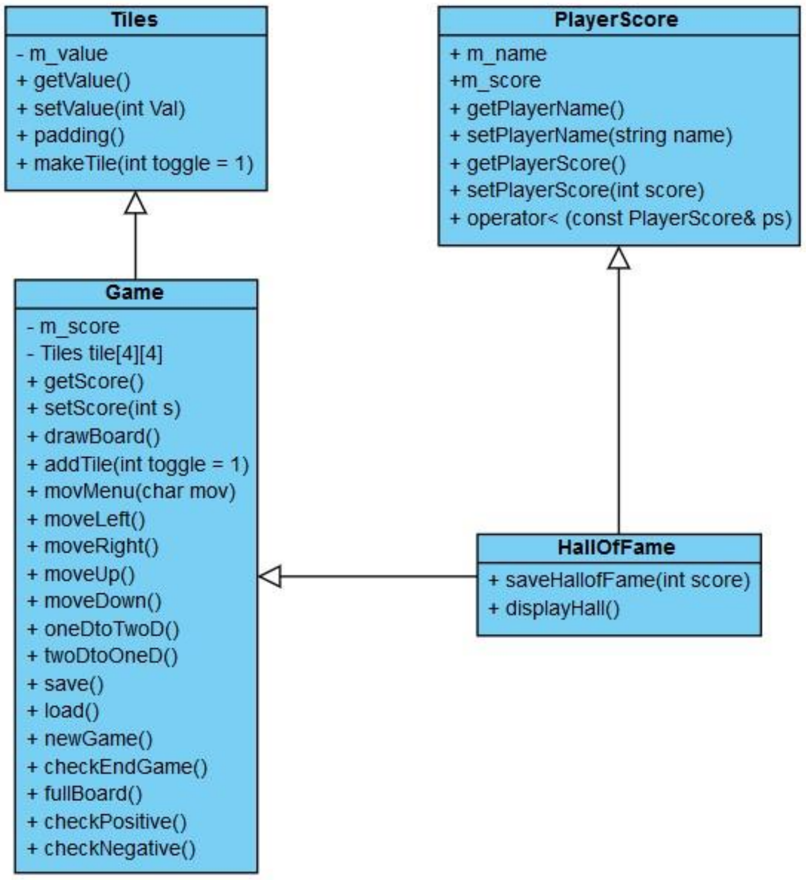


Figure 1: UML diagram

Main()

This is where all the code comes together. Objects for the game and hall of fame classes were made. Srand was used for the functions that used rand. For the game to initially start with two tiles that did not contain a mine tile a zero was put in the parameter and they were called once outside the loop.

The infinite while loop drew the game board and displayed the current score, then the user was asked to enter a character that corresponded to the direction in which they wanted the board to move. If the character was equal to one of the moves then the movMenu function would be called otherwise if the player wanted to start a new game then they would be prompted to make sure that they wanted to do that, same goes for loading and quitting. If the user wanted to display the hall of fame, then it will be displayed for five seconds before returning to asking for input again and the same will happen when they input an invalid character. If the user wanted to save the save function would be called.

Then another switch case is used to check whether the game ended or not, if the function returns a three then it will display that they won and then the saveHallofFame function is called. If it returns a five, then the player lost, and the hall of fame is displayed. If one is returned, then it will break from the case and the loop will start again.

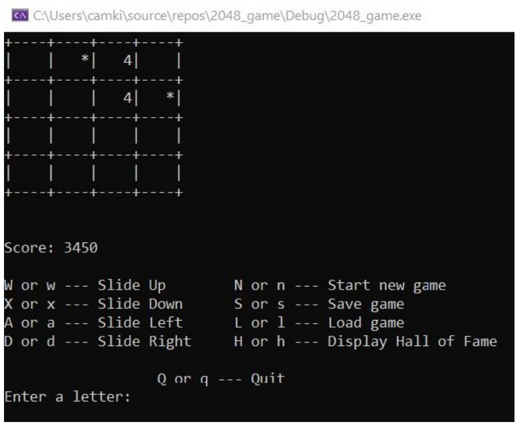
In order to not repeat code a function to display the options for the user was created in the launcher.

Design choices

As this game uses a matrix and the elements of the matrix must be altered, a class was created just for tile specific functions. The game class was created to implement the rules of the game and since the tiles had their own class writing the code was much easier as certain functions did not have to be copied and pasted each time. The inheritance between each class made it easier to keep track of what function was used for what situation and each function was able to be used easily.

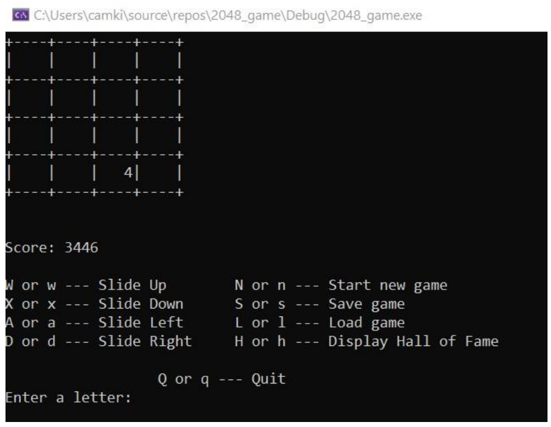
Testing

- Testing the mine tile
  The mine tile should only combine and destroy the tile in the direction of the move unless there are no tiles in that direction, this means that the tile on the other side will be combined with it and destroyed.
  Using the load function this could be easily tested.



```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|    |   *|   4|    |
+----+----+----+----+
|    |    |   4|   *|
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+


Score: 3450

W or w --- Slide Up        N or n --- Start new game
X or x --- Slide Down      S or s --- Save game
A or a --- Slide Left      L or l --- Load game
D or d --- Slide Right     H or h --- Display Hall of Fame

                 Q or q --- Quit
Enter a letter:
```

Figure 2

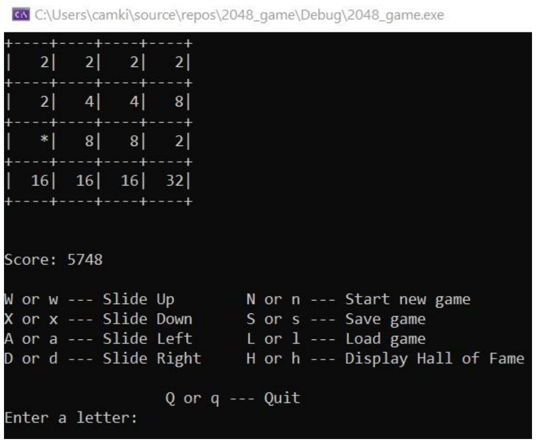


```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|    |    |   4|    |
+----+----+----+----+


Score: 3446

W or w --- Slide Up        N or n --- Start new game
X or x --- Slide Down      S or s --- Save game
A or a --- Slide Left      L or l --- Load game
D or d --- Slide Right     H or h --- Display Hall of Fame

                 Q or q --- Quit
Enter a letter:
```

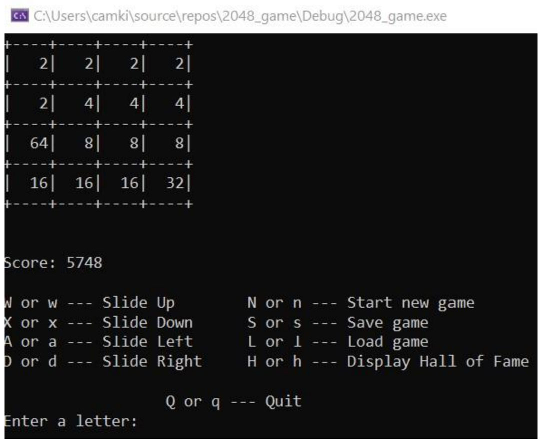
Figure 3

  In Figure 2 the board was moved to the right. Both cases mentioned above appeared to work as intended. The same test was done for other moves.

- Testing for lose case
  To test the case in which there the player lost the board must be full, without mine tiles and without any same adjacent tiles.



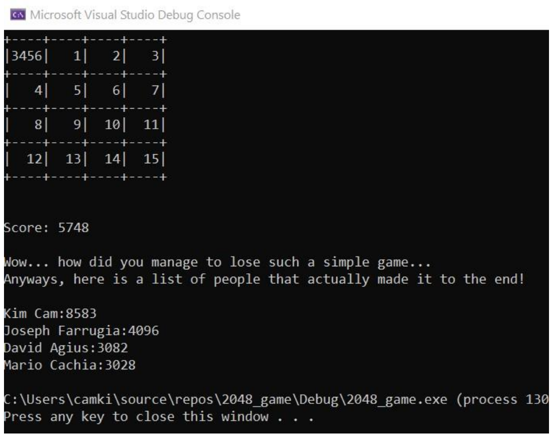
```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|   2|   2|   2|   2|
+----+----+----+----+
|   2|   4|   4|   8|
+----+----+----+----+
|   *|   8|   8|   2|
+----+----+----+----+
|  16|  16|  16|  32|
+----+----+----+----+


Score: 5748

W or w --- Slide Up        N or n --- Start new game
X or x --- Slide Down      S or s --- Save game
A or a --- Slide Left      L or l --- Load game
D or d --- Slide Right     H or h --- Display Hall of Fame

                 Q or q --- Quit
Enter a letter:
```



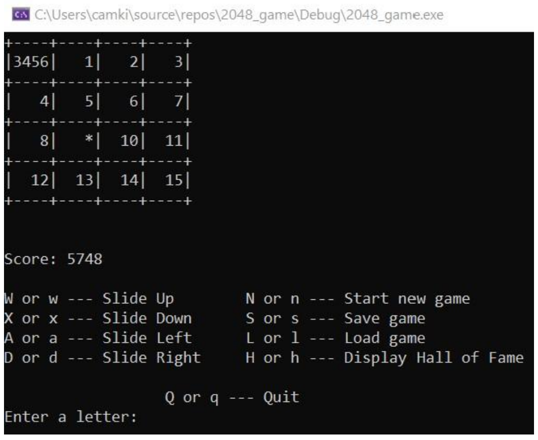
```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|   2|   2|   2|   2|
+----+----+----+----+
|   2|   4|   4|   4|
+----+----+----+----+
|  64|   8|   8|   8|
+----+----+----+----+
|  16|  16|  16|  32|
+----+----+----+----+


Score: 5748

W or w --- Slide Up        N or n --- Start new game
X or x --- Slide Down      S or s --- Save game
A or a --- Slide Left      L or l --- Load game
D or d --- Slide Right     H or h --- Display Hall of Fame

                 Q or q --- Quit
Enter a letter:
```

Figure 4: Tiles can be added and there is a mine, Therefor not a lose condition

Figure 5: Tiles can be added, therefor not a lose condition



```
Microsoft Visual Studio Debug Console
+----+----+----+----+
|3456|   1|   2|   3|
+----+----+----+----+
|   4|   5|   6|   7|
+----+----+----+----+
|   8|   9|  10|  11|
+----+----+----+----+
|  12|  13|  14|  15|
+----+----+----+----+


Score: 5748

Wow... how did you manage to lose such a simple game...
Anyways, here is a list of people that actually made it to the end!

Kim Cam:8583
Joseph Farrugia:4096
David Agius:3082
Mario Cachia:3028

C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe (process 130
Press any key to close this window . . .
```

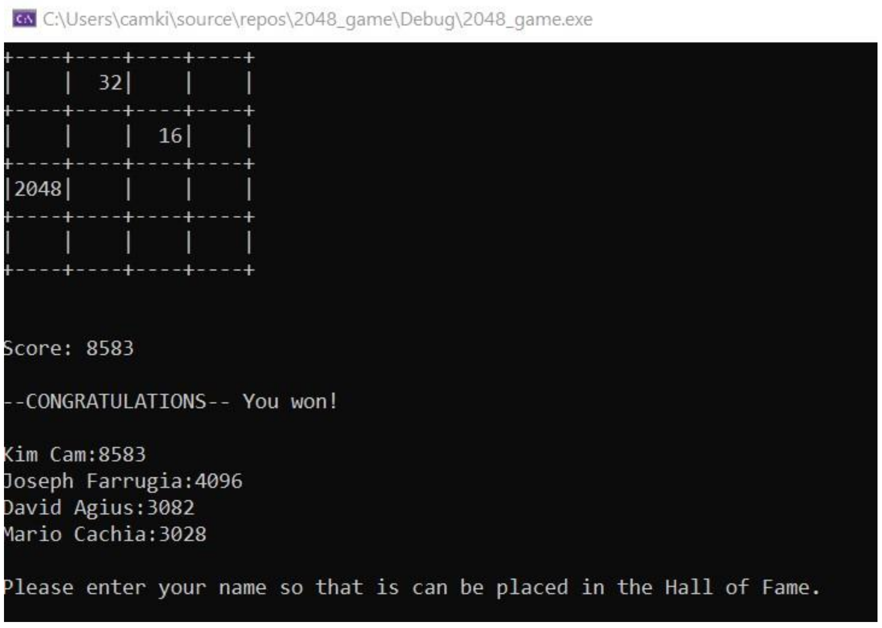


```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|3456|   1|   2|   3|
+----+----+----+----+
|   4|   5|   6|   7|
+----+----+----+----+
|   8|   *|  10|  11|
+----+----+----+----+
|  12|  13|  14|  15|
+----+----+----+----+


Score: 5748

W or w --- Slide Up        N or n --- Start new game
X or x --- Slide Down      S or s --- Save game
A or a --- Slide Left      L or l --- Load game
D or d --- Slide Right     H or h --- Display Hall of Fame

                 Q or q --- Quit
Enter a letter:
```

Figure 6: No tile can be added, therefor lose

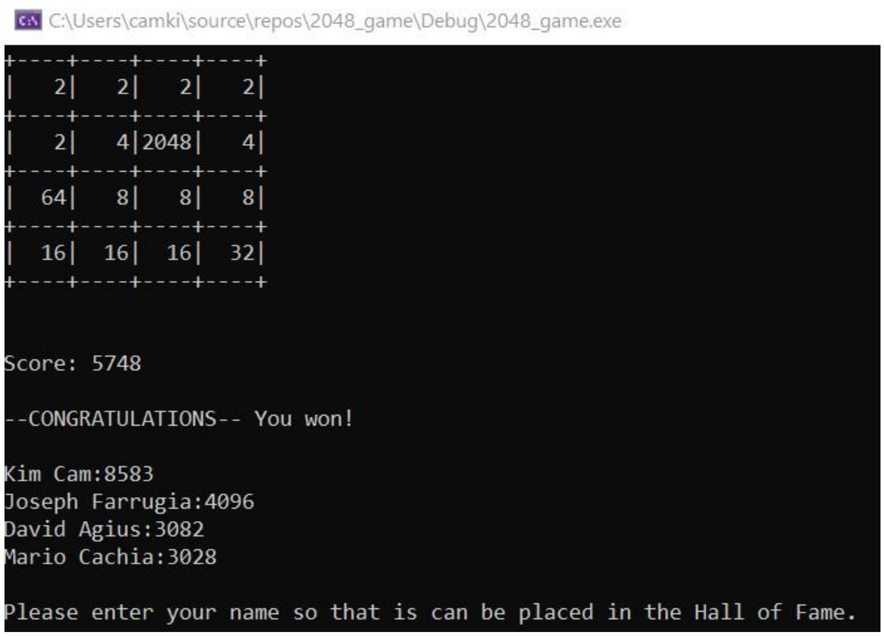
Figure 7: Mine tile in board, therefor not a lose condition

- <u>Testing for win case</u>
  To test if there is a win there must be a tile with the value 2048.



```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|    |  32|    |    |
+----+----+----+----+
|    |    |  16|    |
+----+----+----+----+
|2048|    |    |    |
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+


Score: 8583

--CONGRATULATIONS-- You won!

Kim Cam:8583
Joseph Farrugia:4096
David Agius:3082
Mario Cachia:3028

Please enter your name so that is can be placed in the Hall of Fame.
```

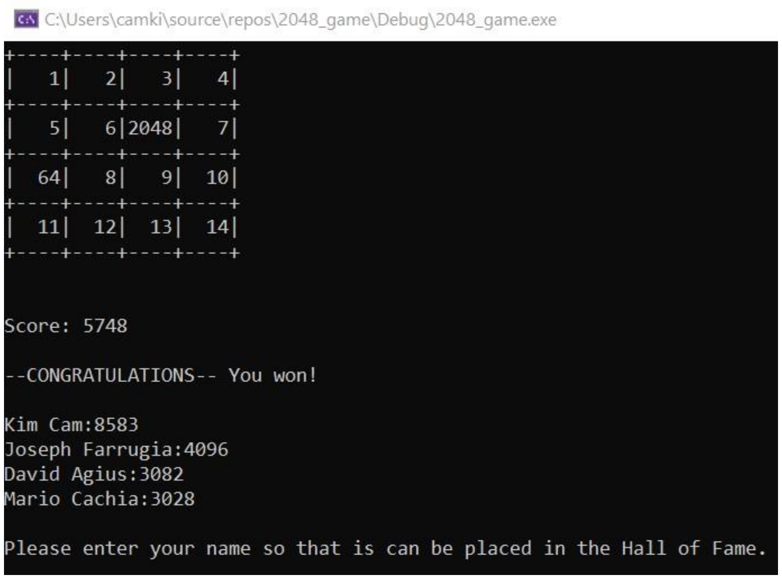
Figure 8: The board is full, but player could have still performed a move if they did not win



C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe

```
+----+----+----+----+
|   2|   2|   2|   2|
+----+----+----+----+
|   2|   4|2048|   4|
+----+----+----+----+
|  64|   8|   8|   8|
+----+----+----+----+
|  16|  16|  16|  32|
+----+----+----+----+


Score: 5748

--CONGRATULATIONS-- You won!

Kim Cam:8583
Joseph Farrugia:4096
David Agius:3082
Mario Cachia:3028

Please enter your name so that is can be placed in the Hall of Fame.
```

Figure 9: Board is full but player would have been unable to perform a move

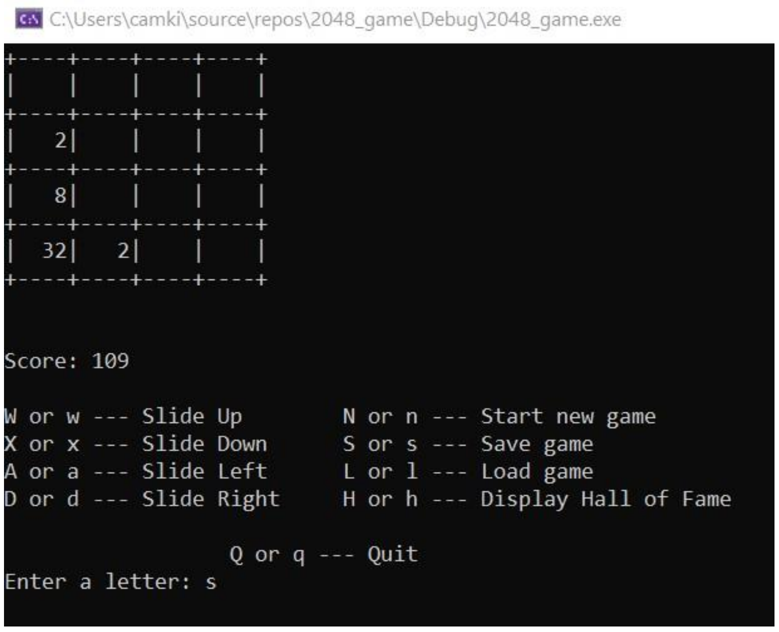


C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe

```
+----+----+----+----+
|   1|   2|   3|   4|
+----+----+----+----+
|   5|   6|2048|   7|
+----+----+----+----+
|  64|   8|   9|  10|
+----+----+----+----+
|  11|  12|  13|  14|
+----+----+----+----+


Score: 5748

--CONGRATULATIONS-- You won!

Kim Cam:8583
Joseph Farrugia:4096
David Agius:3082
Mario Cachia:3028

Please enter your name so that is can be placed in the Hall of Fame.
```

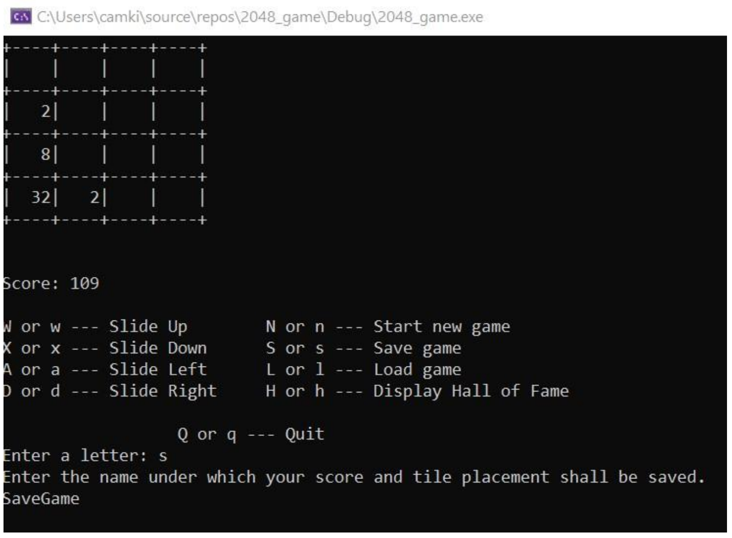
Figure 10: Board is full but player would have been unable to perform another move.

- <u>Testing Save and Load</u>



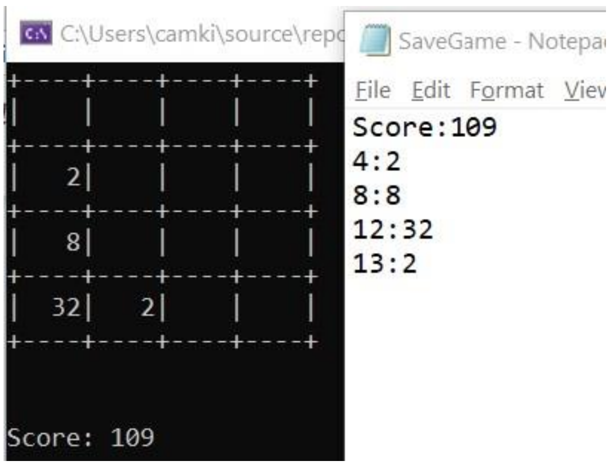
```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|   2|    |    |    |
+----+----+----+----+
|   8|    |    |    |
+----+----+----+----+
|  32|   2|    |    |
+----+----+----+----+


Score: 109

W or w --- Slide Up        N or n --- Start new game
X or x --- Slide Down      S or s --- Save game
A or a --- Slide Left      L or l --- Load game
D or d --- Slide Right     H or h --- Display Hall of Fame

                  Q or q --- Quit
Enter a letter: s
```

Figure 11: Entering s



```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|   2|    |    |    |
+----+----+----+----+
|   8|    |    |    |
+----+----+----+----+
|  32|   2|    |    |
+----+----+----+----+


Score: 109

W or w --- Slide Up        N or n --- Start new game
X or x --- Slide Down      S or s --- Save game
A or a --- Slide Left      L or l --- Load game
D or d --- Slide Right     H or h --- Display Hall of Fame

                  Q or q --- Quit
Enter a letter: s
Enter the name under which your score and tile placement shall be saved.
SaveGame
```

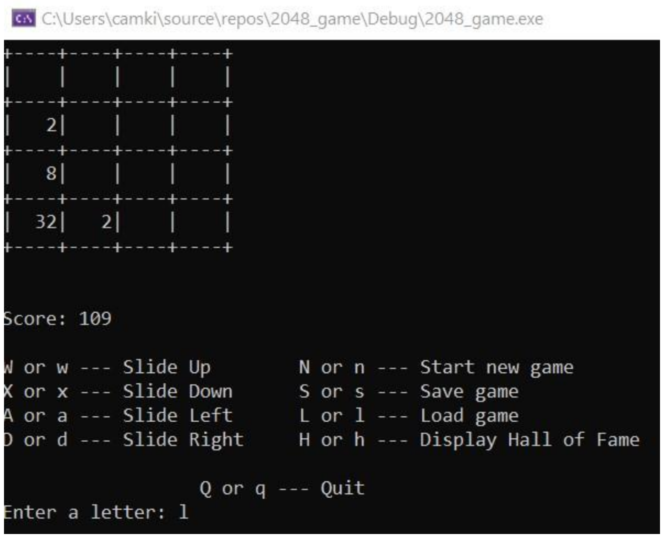
Figure 12: Entering name for saving



```
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|   2|    |    |    |
+----+----+----+----+
|   8|    |    |    |
+----+----+----+----+
|  32|   2|    |    |
+----+----+----+----+


Score: 109
```

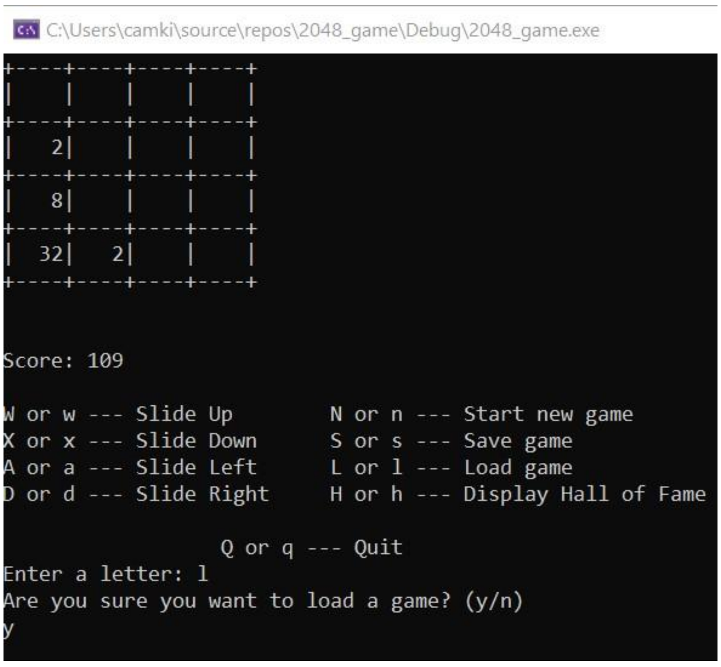
```
Score:109
4:2
8:8
12:32
13:2
```

Figure 13: Board and text file side by side



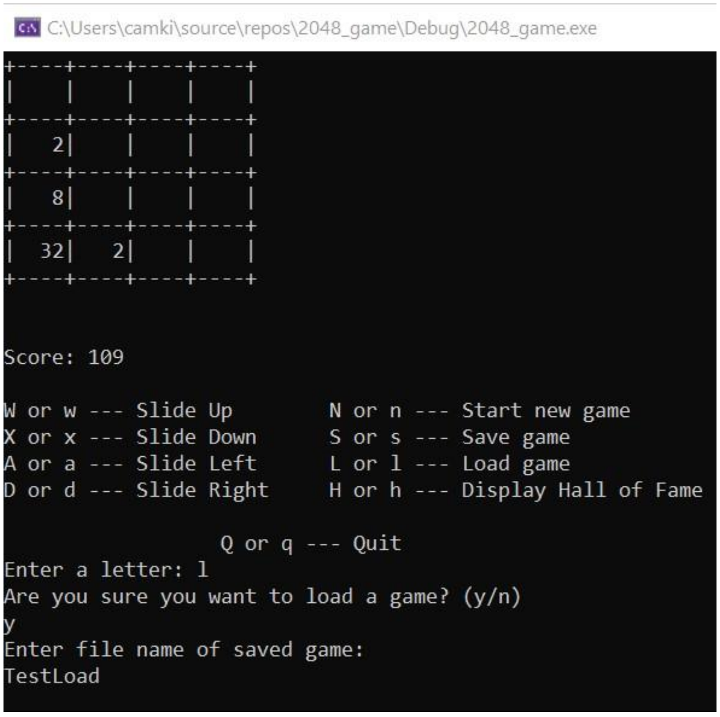
```
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|   2|    |    |    |
+----+----+----+----+
|   8|    |    |    |
+----+----+----+----+
|  32|   2|    |    |
+----+----+----+----+


Score: 109

W or w --- Slide Up         N or n --- Start new game
X or x --- Slide Down       S or s --- Save game
A or a --- Slide Left       L or l --- Load game
D or d --- Slide Right      H or h --- Display Hall of Fame

                  Q or q --- Quit
Enter a letter: l
```

Figure 14: Entering l



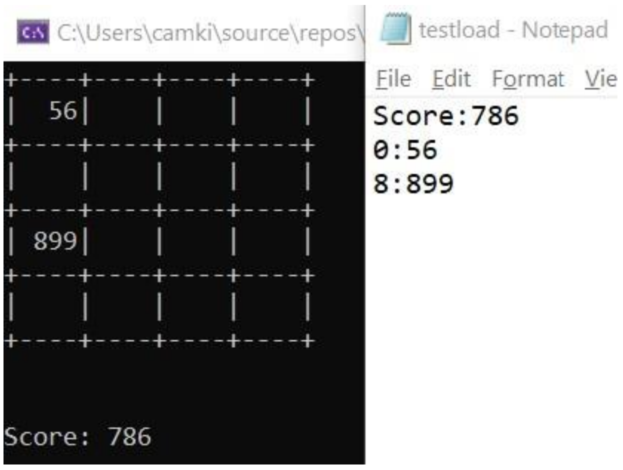
```
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|   2|    |    |    |
+----+----+----+----+
|   8|    |    |    |
+----+----+----+----+
|  32|   2|    |    |
+----+----+----+----+


Score: 109

W or w --- Slide Up         N or n --- Start new game
X or x --- Slide Down       S or s --- Save game
A or a --- Slide Left       L or l --- Load game
D or d --- Slide Right      H or h --- Display Hall of Fame

                  Q or q --- Quit
Enter a letter: l
Are you sure you want to load a game? (y/n)
y
```

Figure 15: Selecting y



```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
|   2|    |    |    |
+----+----+----+----+
|   8|    |    |    |
+----+----+----+----+
|  32|   2|    |    |
+----+----+----+----+


Score: 109

W or w --- Slide Up          N or n --- Start new game
X or x --- Slide Down        S or s --- Save game
A or a --- Slide Left        L or l --- Load game
D or d --- Slide Right       H or h --- Display Hall of Fame

                  Q or q --- Quit
Enter a letter: l
Are you sure you want to load a game? (y/n)
y
Enter file name of saved game:
TestLoad
```

Figure 16: Entering name of game to be loaded

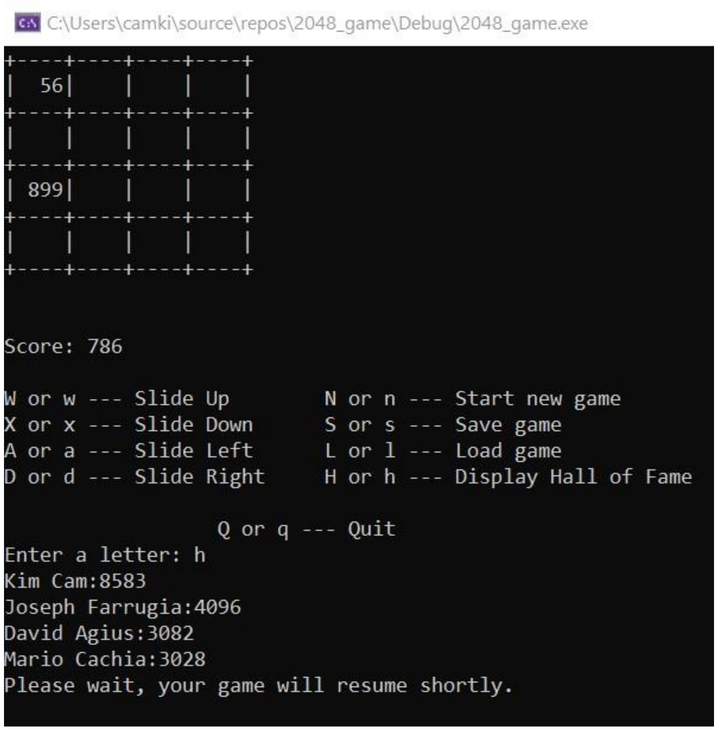


```
C:\Users\camki\source\repos\
+----+----+----+----+
|  56|    |    |    |
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
| 899|    |    |    |
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+


Score: 786
```

```
testload - Notepad
File  Edit  Format  Vie
Score:786
0:56
8:899
```

Figure 17: Board and text file side by side

- <u>Testing Hall of Fame</u>



```
C:\Users\camki\source\repos\2048_game\Debug\2048_game.exe
+----+----+----+----+
|  56|    |    |    |
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+
| 899|    |    |    |
+----+----+----+----+
|    |    |    |    |
+----+----+----+----+



Score: 786

W or w --- Slide Up        N or n --- Start new game
X or x --- Slide Down      S or s --- Save game
A or a --- Slide Left      L or l --- Load game
D or d --- Slide Right     H or h --- Display Hall of Fame

                  Q or q --- Quit
Enter a letter: h
Kim Cam:8583
Joseph Farrugia:4096
David Agius:3082
Mario Cachia:3028
Please wait, your game will resume shortly.
```

Figure 18: Hall of Fame displayed