Avishai Cohen ESE 224 Project Report ID# 110542575

**Note:** I used Visual Studio 2015 as my IDE, and some issues may occur if using another IDE.

**Architecture of Design**

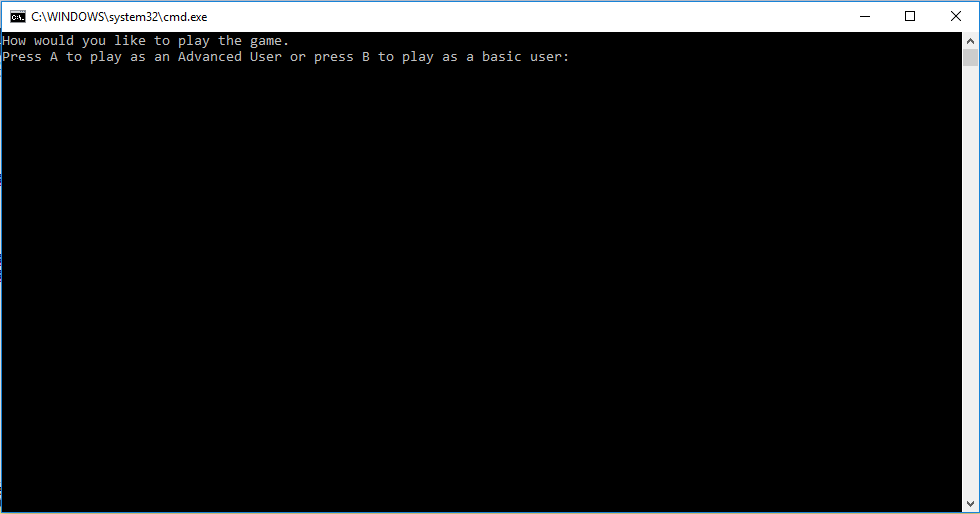
* For this project, I decided to break up my design into 2 different classes. One of my classes was the user class which dealt with most of the users’ options. These are things such as being a basic or advanced user, what the username and password are, and the main menu functionality based on the user type (advanced or basic)…etc. The other class was the Board class which would dealt with all of the games’ board functionality. These are things such as having the board move: left, right, up and down, the undo function, the board layout...etc.

**What each Function Does**

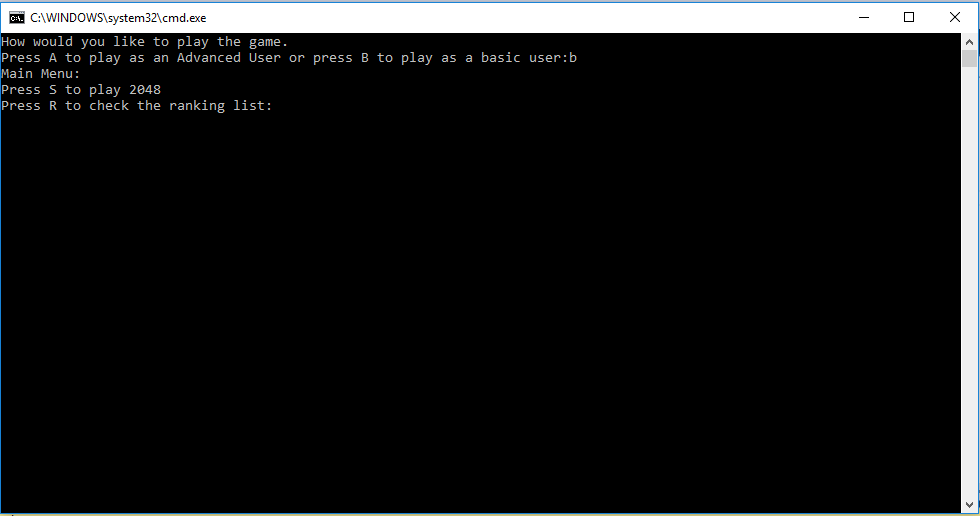
* User class
  + Public
    - void advancedOrBasic();
      * This function asks the user whether they would like to be an advanced user or a basic user
    - char getAdvancedOrBasic();
      * This function will be used by the board class to access the users Game type to determine if they are an advanced user or a basic user
    - string getUserName();
      * This Function will be used by the board class to access the users Username so that it can output their score to their file.
  + Private
    - void signupOrLogin();
      * This function would be called if the user chose to be an advanced user. It would ask them if they wanted to signup or login
    - void signup();
      * If the user decided to signup, this function would be called and would ask the user to enter a username and password to signup. It will also check to make sure the username the user wants isn’t already taken. If the username they chose was taken they are asked to enter a different username.
    - void login();
      * If the user decided to login, this function would be called and would ask the user to enter a username and password to login. It will check to make sure that their login information is correct. If the login information is incorrect it will alert the user of the incorrect login information and ask them if they want to signup or login.
    - void createUserFile();
      * This function will create a user txt file with the name of the user’s username
    - void basicMenu();
      * This function will display the basic menu for a basic user. The basic menu only consists of two options: 1) play the game 2) print out the ranking list
    - void advancedMenu();
      * This function will display the advanced menu for an advanced user. The advanced menu consists of 4 options: 1) play the game 2) print out the game history 3) print out the best record 4) print out the ranking list
    - void printRankingList();
      * This function will find each advanced users best score and rank it based on the score, step and the user’s username, and print out the result
    - void printGameHistory();
      * This function will print out an advanced users game history
    - void printBestRecord();
      * This function will print out an advanced users best record
    - void gameSetup();
      * This function will be called when either user starts a game instantiate on object of the board class and pass the users username and game type into it.
    - void forwardToMenu();
      * This function is called to determine which main menu to call whether the user is an advanced user or a basic user.
* Board
  + Public
    - Board(char gameType, string userName);
      * This overloaded constructer will set up the users board based on their username and game type. The 4x4 board will be created and set all to zero, and 2 random tiles will then be set to 2.
  + Private
    - int randomNumber();
      * This function will generate randomly either a 2 or a 4 and return it.
    - bool placeRandomValue(int value);
      * This function will take an int value as a parameter and will first check the board for any 0’s on the board. It will then randomly select one of these 0’s and replace the 0 with the value passed into the function. If there were no 0’s on the board, the function will return false, otherwise it will return true
    - void move();
      * This function will ask for the user’s input on whether they would like to move up, down, left, right, exit, or undo (if they are an advanced user). After the move is made, the new score, step, and board will be printed out, and will check to see if there are any other moves that can be made.
    - void moveUp();
      * This function will perform the move up action by moving all the numbers up on the board and will check the number below it to see if it is the same number and it should merge. This merge check starts from the top left and ends at the bottom right, going across the rows.
    - void moveDown();
      * This function will perform the move down action by moving all the numbers down on the board and will check the number above it to see if it is the same number and it should merge. This merge check starts from the bottom right and ends at the top left, going across the rows.
    - void moveLeft();
      * This function will perform the move left action by moving all the numbers left on the board and will check the number to the right of it to see if it is the same number and it should merge. This merge check starts from the top right and ends at the top left, going down the columns.
    - void moveRight();
      * This function will perform the move right action by moving all the numbers right on the board and will check the number to the left of it to see if it is the same number and it should merge. This merge check starts from the bottom right and ends at the top left, going up the columns.
    - int getScore();
      * This function will go across the board and return the tile with the highest number, which is the users score.
    - void printBoard();
      * This function will print the board of the user which consist of the user’s name, score, the full 4x4 board, and the controls for the game. The board will have different options depending on if they are a basic or advanced user (The advanced user will have the option to undo while the advanced user doesn’t).
    - bool checkWinOrLose();
      * This function will check if the user can make any moves with the current board. The function loops through each tile from the top left to the bottom right going across the rows, and will check for any potential tiles to merge.
    - void addToStack();
      * This function will only be called if the user is an advanced player. This function will add the board state to the stack after each move.
    - void undo();
      * This function will set the last board’s state to the current board, print out the board, and delete that board from the stack.
    - void addResultsToFile();
      * This function will save the score of the game of an advanced user to their file when the game is completed.

**Output screen shots**

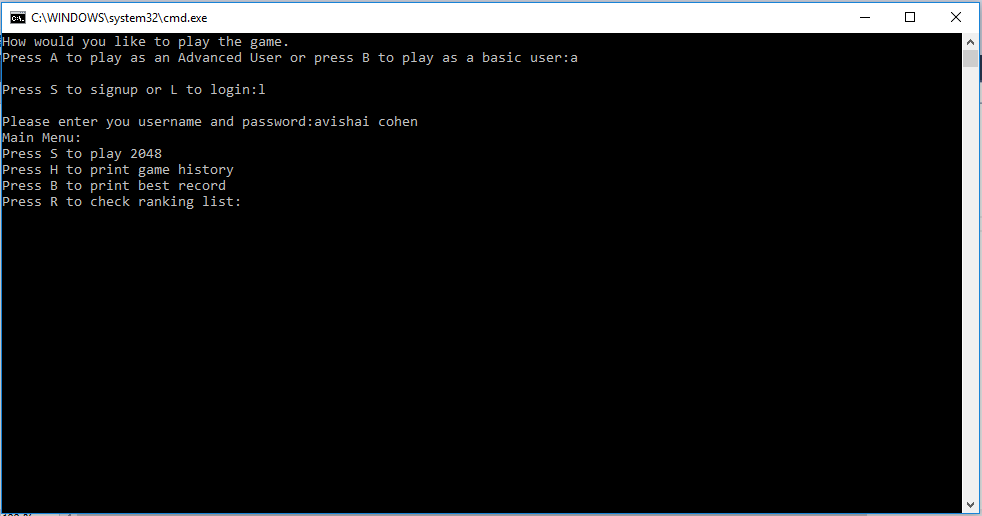
* Welcome Screen

****

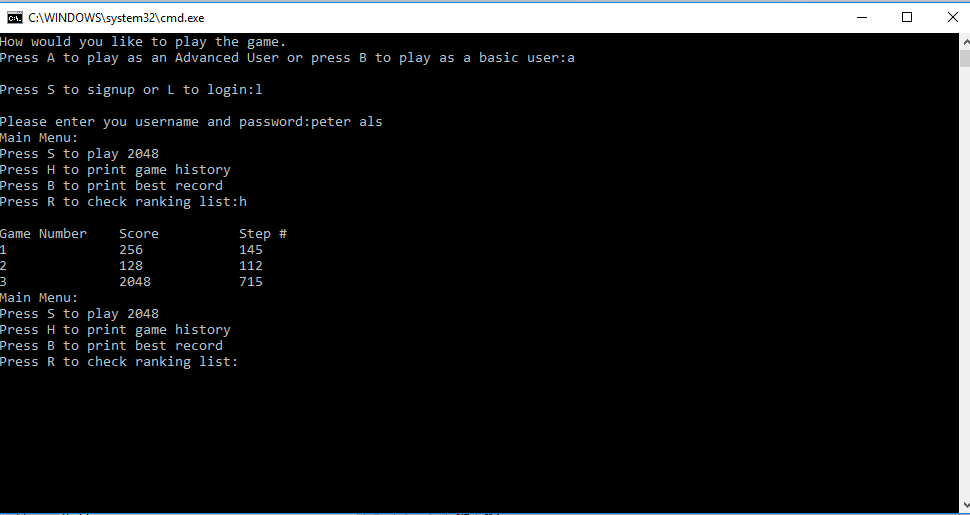
* Basic user Main Menu



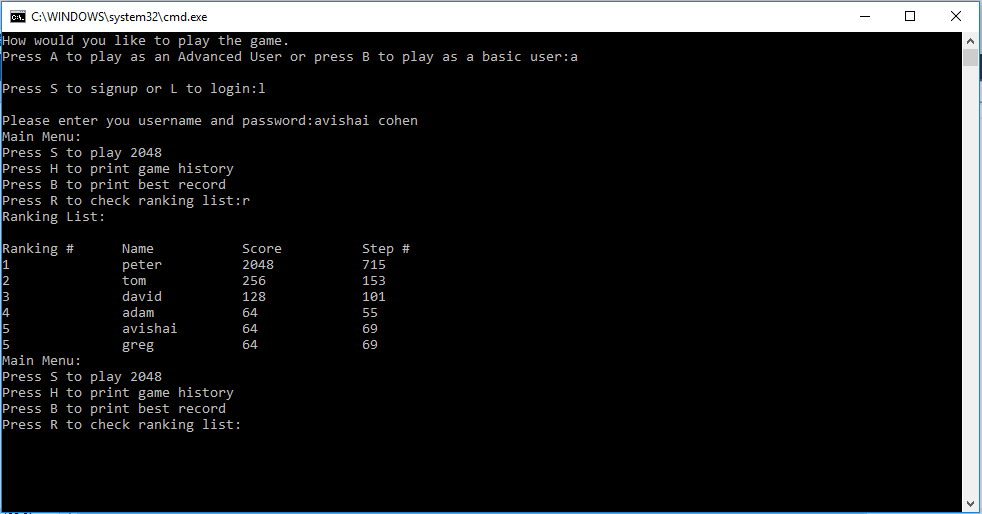
* Advanced user Main Menu



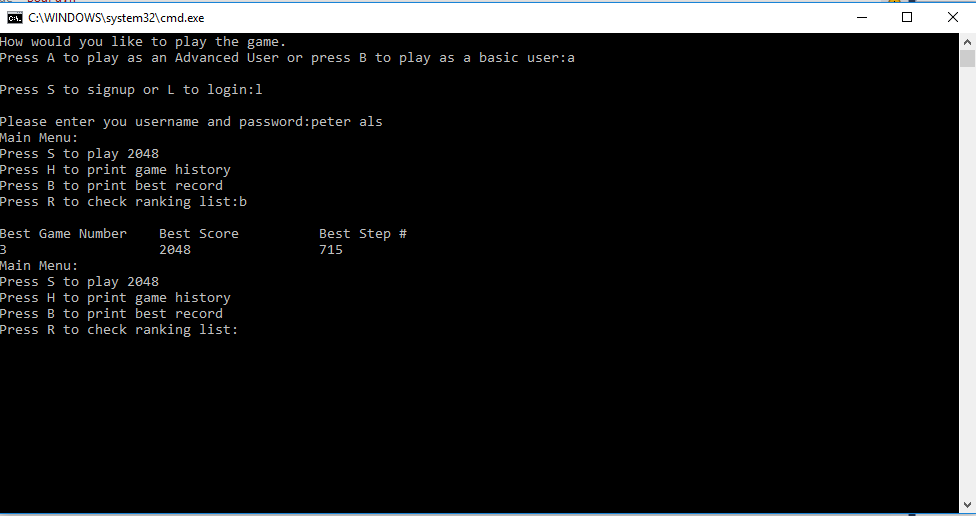
* Game history



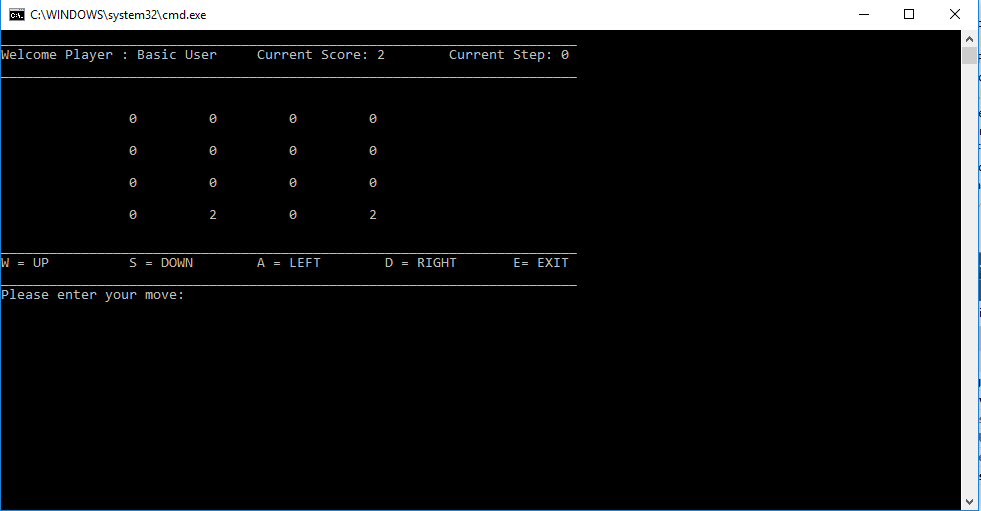
* Ranking list



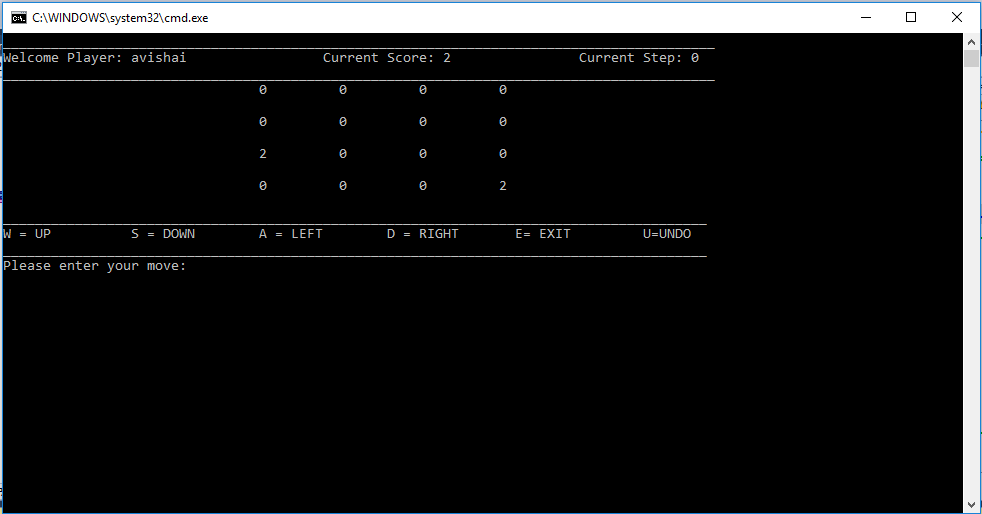
* Best Record



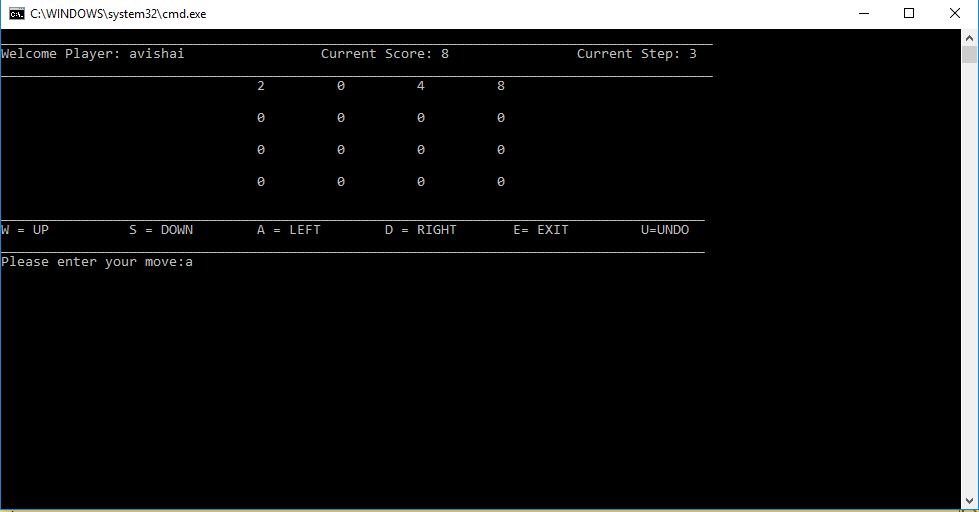
* Basic User Game Board



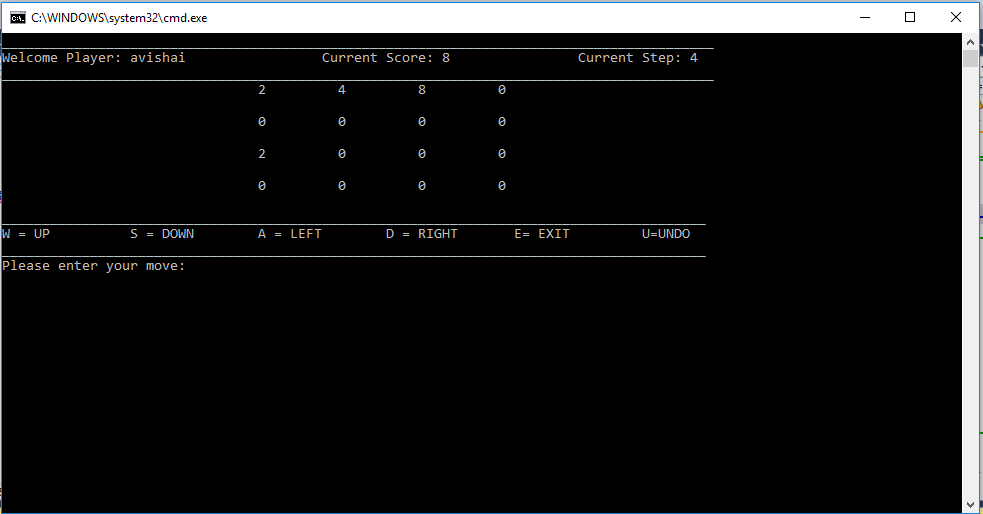
* Advanced User Game Board



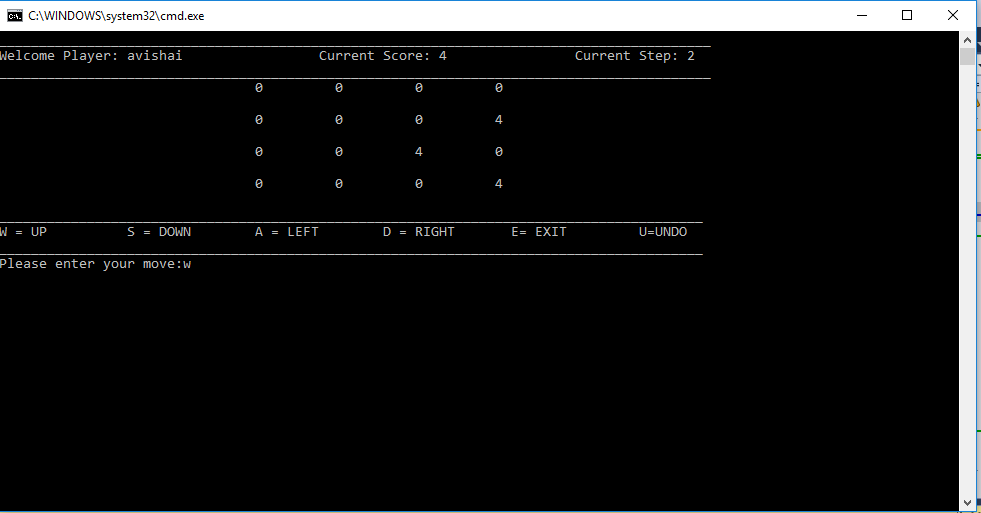
* Move Left
  + Before Move



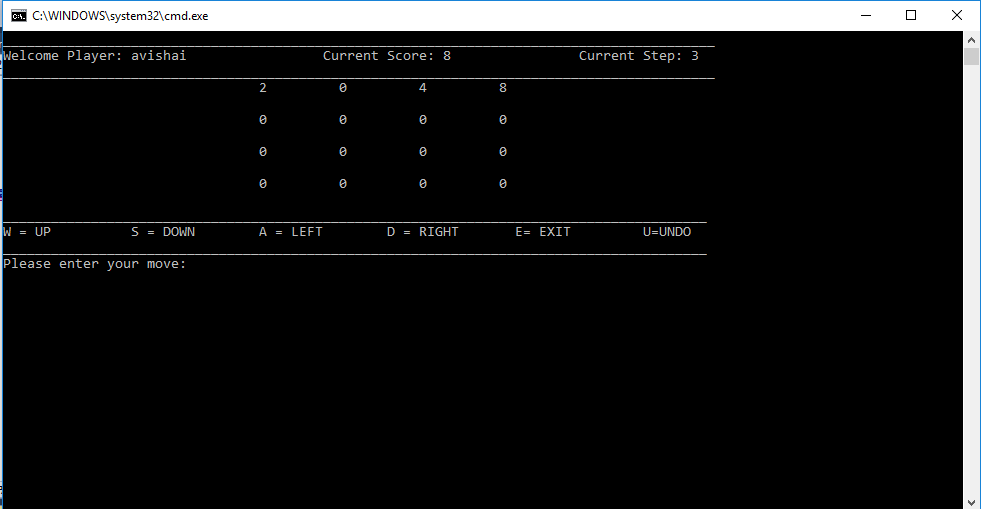
* + After Move



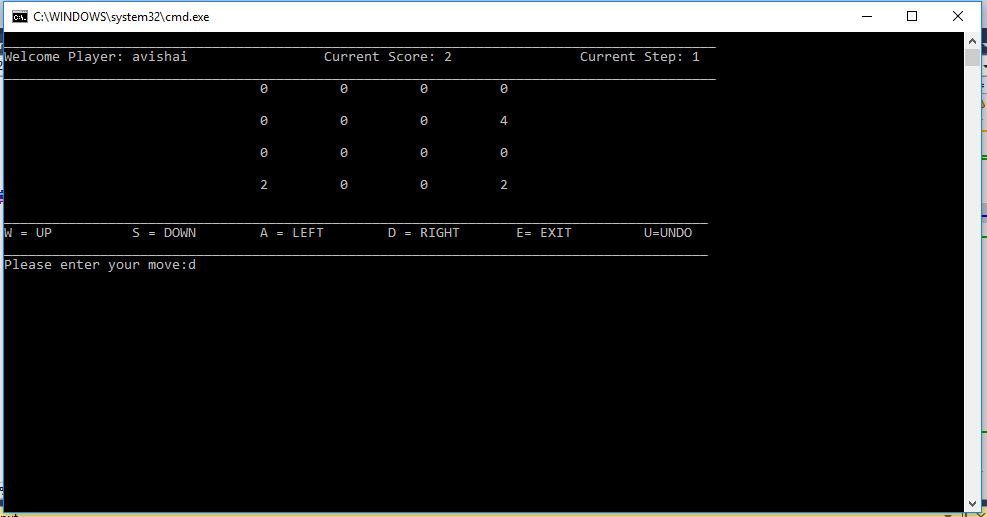
* Move Up
  + Before Move



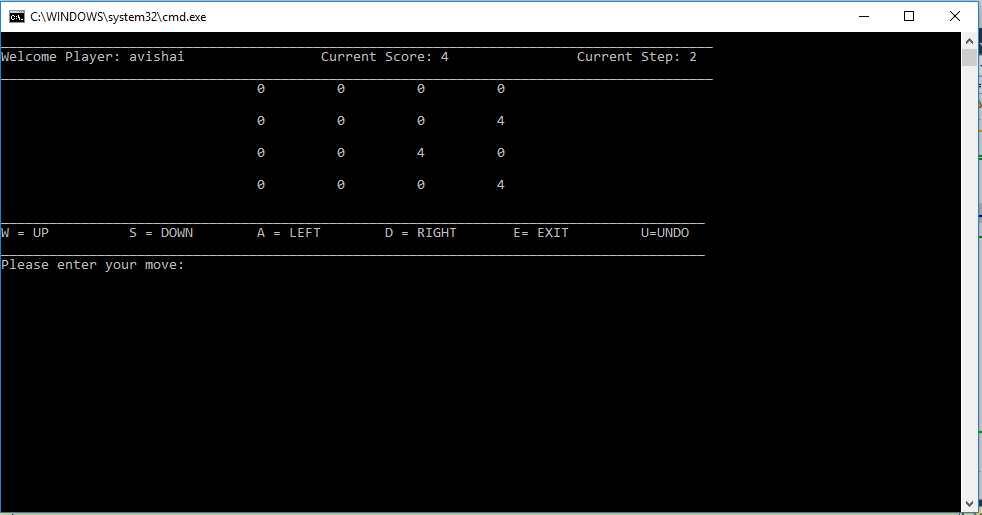
* + After Move



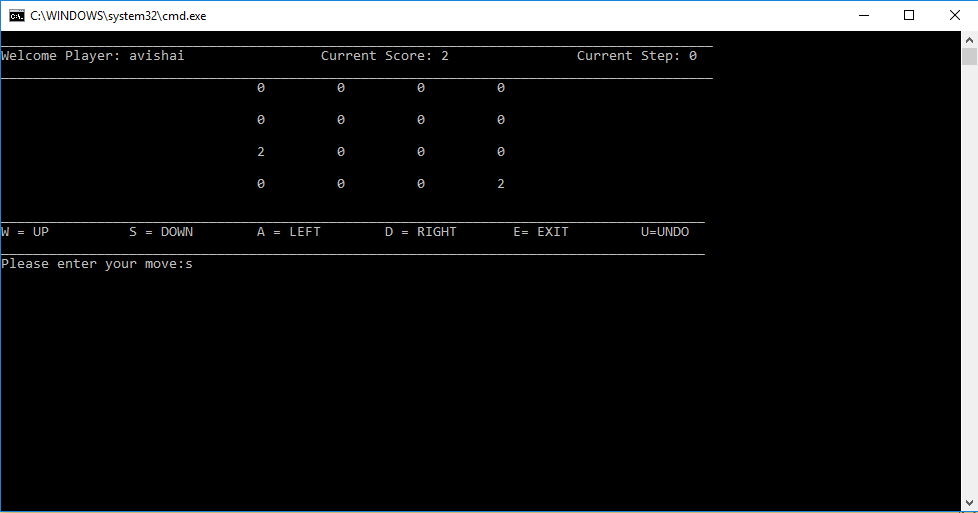
* Move Right
  + Before Move



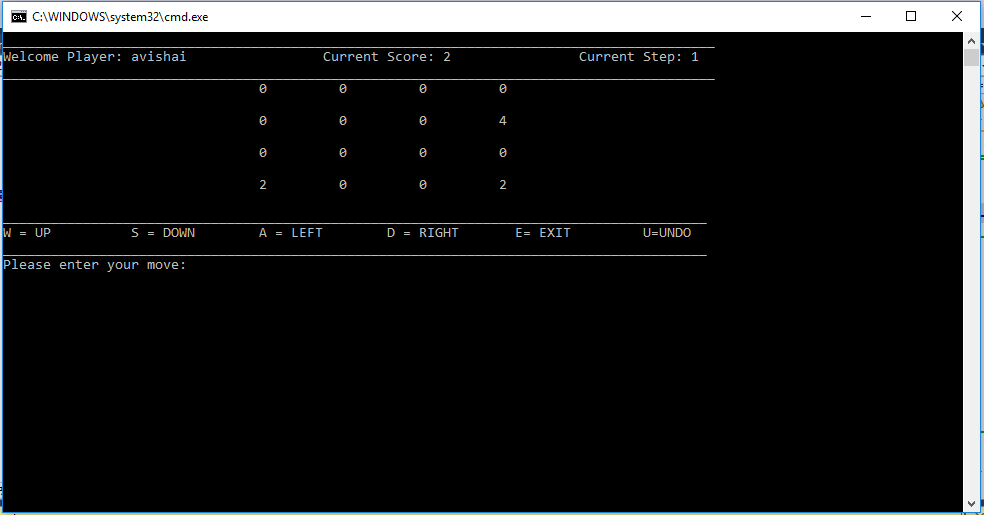
* + After Move



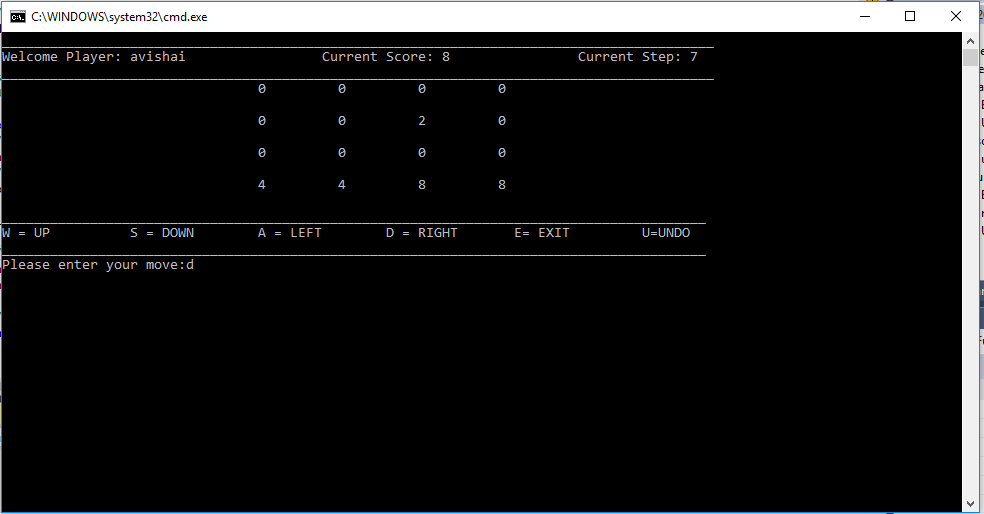
* Move Down
  + Before Move



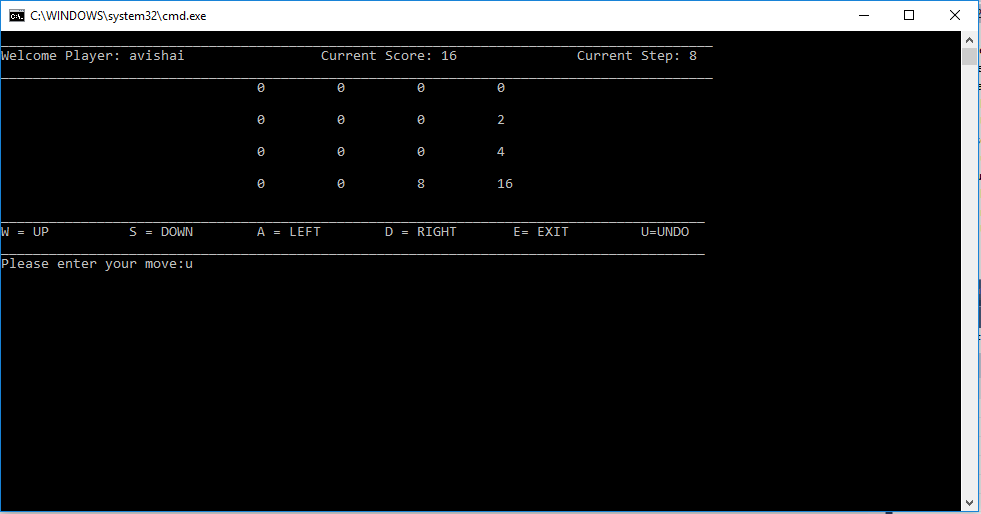
* + After Move



* Undo
  + Original Board



* + Move Right



* + Undo back to Original Board

