IT TAKES TWO TO TANGO

CME1251 Project Based Learning - I

BY

2020510034 YUSUF GASSALOĞLU

2021510046 ERAY KUBİLAY

Outline

INTRODUCTION

PROGRESS SUMMARY

- **Requirements**
- **Task Sharing**
- **Scheduling**
- **Completed Tasks**
- **Incomplete Tasks: Reasons, Explanations**
- **Additional Improvements**

PROBLEMS ENCOUNTERED

ALGORITHMS AND SOLUTION STRATEGIES

SCREENSHOTS

CONCLUSION

REFERENCES

INTRODUCTION

PROGRESS SUMMARY

It Takes Two to Tango

Requirements

- Algortihm that reads the pressed key.
- Algorithms that check movements for each key available.
- Algorithm that checks if the numbers in arrays are single or not.
- Algorithm that checks if there are any matched numbers after each movement.
- Algorithm that randomly fills arrays.
- Algorithm that increases and prints the new score.

Task Sharing

- We actualy finished most of the parts of the project during the sessions that we were given to discuss the week, what we have done. Except for these we did some parts individually then merged and gave the project it’s final form.

- Most parts of the final report, poster and some parts of the algorithm of the project and improvements were done by Yusuf Gassaloğlu.

- Presentation, some parts of the final report and some parts of the algorithm of the project were done by Eray Kubilay.

Scheduling

• We proceeded as planned except for the 3rd week. Because of our midterms we agreed on delaying the tasks on 3rd week to 4th week;

1. Discussing and designing solution alternatives. Creating the necessary variables, structures. Screen. Cursor movement.

2. Initial board with the randomly generated numbers. Moving single number up or down (W/S keys).

3. -------

4. Moving single number left or right (A/D keys). Matching.
   - Remaining parts of the application. Testing/Debugging.

Completed Tasks

- We completed all tasks and made some additional improvements.

Incomplete Tasks

- We completed all tasks.

Additional Improvements

- How to Play shown on screen all the time.
- Colorized texts and board.
- Beep sound comes when two numbers match and disappear.
- Rank System

PROBLEMS ENCOUNTERED

• Our third team member left the project so we did everything as a group of two. We also had our midterms on third week of the project so we could not really find the time to spend on project that week. Except for these we ran into simple code errors but that didn't bother us.

ALGORITHMS AND SOLUTION STRATEGIES

ALGORITHM THAT PRINTS THE INITIAL BOARD WHICH CONTAINS RANDOM NUMBERS BY LOATING THEM RANDOMLY ASWELL

```
// Generating random numbers on the board
for (int i = 0; i < 30; i++)
{
    int numberOnTheTable;
    Random random = new();
    columnOnTheTable = random.Next(0, 30);
    lineOnTheTable = random.Next(0, 3);
    numberOnTheTable = random.Next(1, 4);

    // for check if cell is empty
    if (lineOnTheTable == 0 && firstLine[columnOnTheTable] == " ")
    {
        firstLine[columnOnTheTable] = Convert.ToString(numberOnTheTable);
        Console.SetCursorPosition(columnOnTheTable + 1, 1); Console.WriteLine(numberOnTheTable);
    }
    else if (lineOnTheTable == 1 && secondLine[columnOnTheTable] == " ")
    {
        secondLine[columnOnTheTable] = Convert.ToString(numberOnTheTable);
        Console.SetCursorPosition(columnOnTheTable + 1, 2); Console.WriteLine(numberOnTheTable);
    }
    else if (lineOnTheTable == 2 && thirdLine[columnOnTheTable] == " ")
    {
        thirdLine[columnOnTheTable] = Convert.ToString(numberOnTheTable);
        Console.SetCursorPosition(columnOnTheTable + 1, 3); Console.WriteLine(numberOnTheTable);
    }
    else i--;
}
```

ALGORITHMS THAT CHECK IF THE PRESSED KEY IS ANY OF THE ARROWS AND DO THE MOVEMENT ACCORDINGLY

```
ConsoleKeyInfo cki;          // required for readkey
// Main game loop
while (true)
{
    if (Console.KeyAvailable)
    {   //read key and take action
        cki = Console.ReadKey(true);
        if (cki.Key == ConsoleKey.RightArrow && cursorx < 30) { cursorx++; Console.SetCursorPosition(cursorx, cursory); } // if user presses right arrow
        else if (cki.Key == ConsoleKey.LeftArrow && cursorx > 1) { cursorx--; Console.SetCursorPosition(cursorx, cursory); } // if user presses left arrow
        else if (cki.Key == ConsoleKey.UpArrow && cursory > 1) { cursory--; Console.SetCursorPosition(cursorx, cursory); } // if user presses up arrow
        else if (cki.Key == ConsoleKey.DownArrow && cursory < 3) { cursory++; Console.SetCursorPosition(cursorx, cursory); } // if user presses down arrow
```

ALGORITHM THAT CHECKS AND DOES THE MOVEMENT IF PLAYER PRESSES W (only for third row in this screenshot)

```
else if (cki.Key == ConsoleKey.W) // if user presses W
{
    if (cursory == 3)
    {
        if ((cursorx != 1 && cursorx != 30 && thirdLine[cursorx - 2] == " " && thirdLine[cursorx] == " ") || (cursorx == 1 && thirdLine[cursorx] == " ") || (cursorx == 30 && thirdLine[cursorx - 2] == " "))//
        {
            if (thirdLine[cursorx - 1] != " ")// checks if there is a number in the cell that cursor is located
            {
                if (secondLine[cursorx - 1] == " ")// checks if there is a number in the cell that cursor will be located
                {
                    secondLine[cursorx - 1] = thirdLine[cursorx - 1];
                    thirdLine[cursorx - 1] = " ";
                    for (int i = 0; i < 30; i++) // print the new board
                    {
                        Console.SetCursorPosition(i + 1, 2);
                        Console.WriteLine(secondLine[i]);
                        Console.SetCursorPosition(i + 1, 3);
                        Console.WriteLine(thirdLine[i]);
                    }
                    cursory--;
                }
            }
        }
    }
```

ALGORITHM THAT CHECKS AND DOES THE MOVEMENT IF PLAYER PRESSES A(only for third row in this screenshot)

```
else if ((cki.Key == ConsoleKey.A) && cursorx > 1)// if user presses A
{
    if (cursory == 3)
    {
        if ((cursorx != 1 && cursorx != 30 && thirdLine[cursorx - 2] == " " && thirdLine[cursorx] == " ") || (cursorx == 1 && thirdLine[cursorx] == " ") || (cursorx == 30 && thirdLine[cursorx - 2] == " "))//
        {
            if (thirdLine[cursorx - 1] != " ")// checks if there is a number in the cell that cursor is located
            {
                if (thirdLine[cursorx - 2] == " ")// checks if there is a number in the cell that cursor will be located
                {
                    while (cursorx > 1 && thirdLine[cursorx - 2] == " ")
                    {
                        thirdLine[cursorx - 2] = thirdLine[cursorx - 1];
                        thirdLine[cursorx - 1] = " ";
                        cursorx--;
                    }
                    for (int i = 0; i < 30; i++) // print the new board
                    {
                        Console.SetCursorPosition(i + 1, 3);
                        Console.WriteLine(thirdLine[i]);
                    }
                }
            }
        }
    }
```

ALGORITHM THAT CHECKS AND DOES THE MOVEMENT IF PLAYER PRESSES S(only for first row in this screenshot)

```
else if (cki.Key == ConsoleKey.S) // if user presses S
{
    if (cursory == 1)
    {
        if ((cursorx != 1 && cursorx != 30 && firstLine[cursorx - 2] == " " && firstLine[cursorx] == " ") || (cursorx == 1 && firstLine[cursorx] == " ") || (cursorx == 30 && firstLine[cursorx - 2] == " "))
        {
            if (firstLine[cursorx - 1] != " ")// checks if there is a number in the cell that cursor is located
            {
                if (secondLine[cursorx - 1] == " ")// checks if there is a number in the cell that cursor will be located
                {   // for move value
                    secondLine[cursorx - 1] = firstLine[cursorx - 1];
                    firstLine[cursorx - 1] = " ";
                    for (int i = 0; i < 30; i++) // print the new board
                    {
                        Console.SetCursorPosition(i + 1, 1);
                        Console.WriteLine(firstLine[i]);
                        Console.SetCursorPosition(i + 1, 2);
                        Console.WriteLine(secondLine[i]);
                    }
                    cursory++;
                }
            }
        }
    }
```

ALGORITHM THAT CHECKS AND DOES THE MOVEMENT IF PLAYER PRESSES D(only for third row in this screenshot)

```
else if ((cki.Key == ConsoleKey.D) && cursorx < 30) // if user presses D
{
    if (cursory == 3)
    {
        if ((cursorx != 1 && cursorx != 30 && thirdLine[cursorx - 2] == " " && thirdLine[cursorx] == " ") || (cursorx == 1 && thirdLine[cursorx] == " ") || (cursorx == 30 && thirdLine[cursorx - 2] == " "))
        {
            if (thirdLine[cursorx - 1] != " ")// checks if there is a number in the cell that cursor is located
            {
                if (thirdLine[cursorx] == " ")// checks if there is a number in the cell that cursor will be located
                {
                    while (cursorx < 30 && thirdLine[cursorx] == " ")
                    {
                        thirdLine[cursorx] = thirdLine[cursorx - 1];
                        thirdLine[cursorx - 1] = " ";
                        cursorx++;
                    }
                    for (int i = 0; i < 30; i++) // print the new board
                    {
                        Console.SetCursorPosition(i + 1, 3);
                        Console.WriteLine(thirdLine[i]);
                    }
                }
            }
        }
    }
```

CODE LINE THAT LETS THE USER EXIT THE GAME WHENEVER THEY WANT TO

```
else if ((cki.Key == ConsoleKey.Escape)) { Console.SetCursorPosition(0,40); break; }; // if user presses Esc
```

ALGORITHM THAT PRINTS THE WHOLE SCREEN AFTER EACH MOVEMENT(only for the movements on first row in this screenshot)

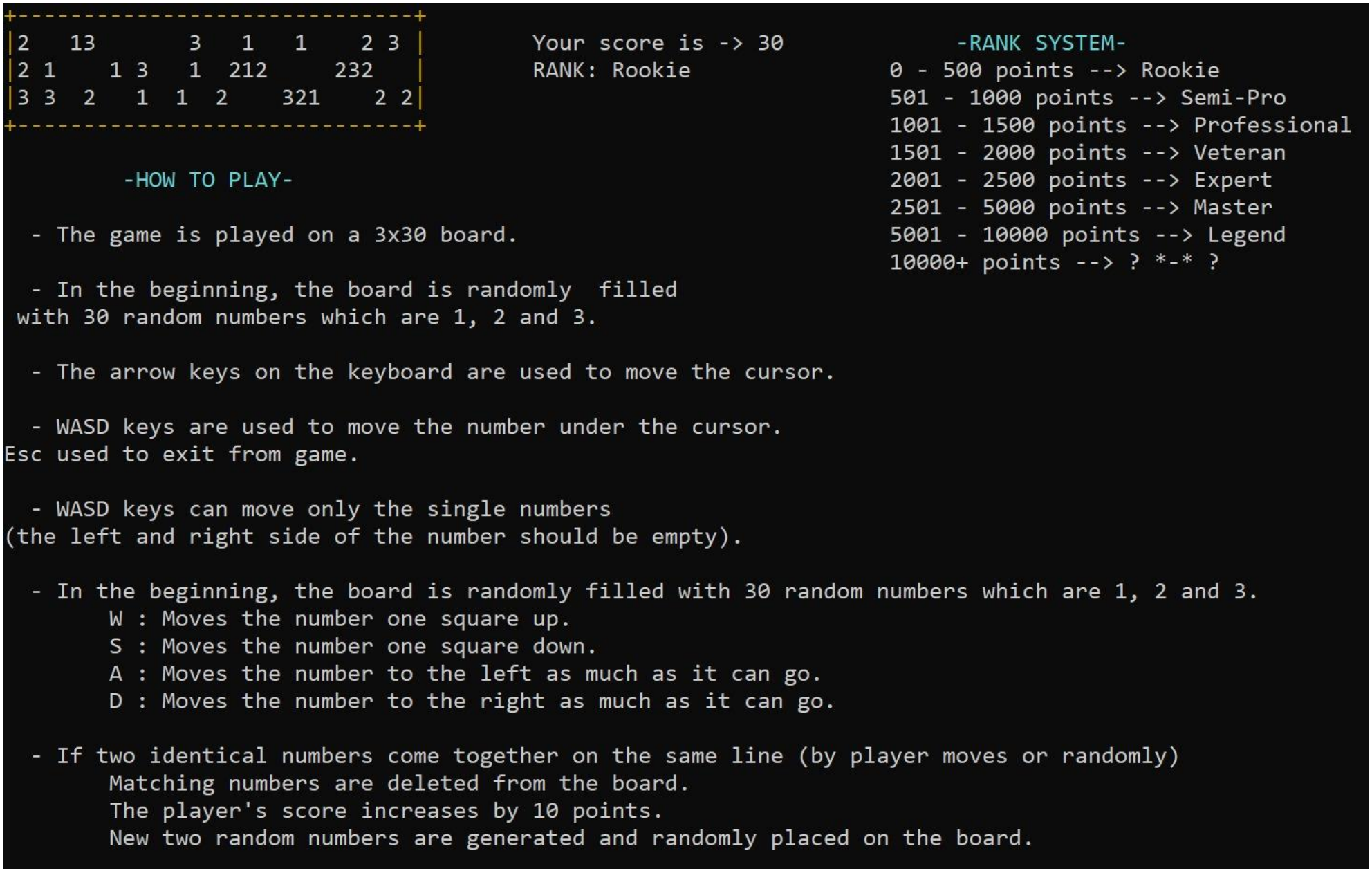
(INCLUDES ADDITIONAL IMPROVEMENT)

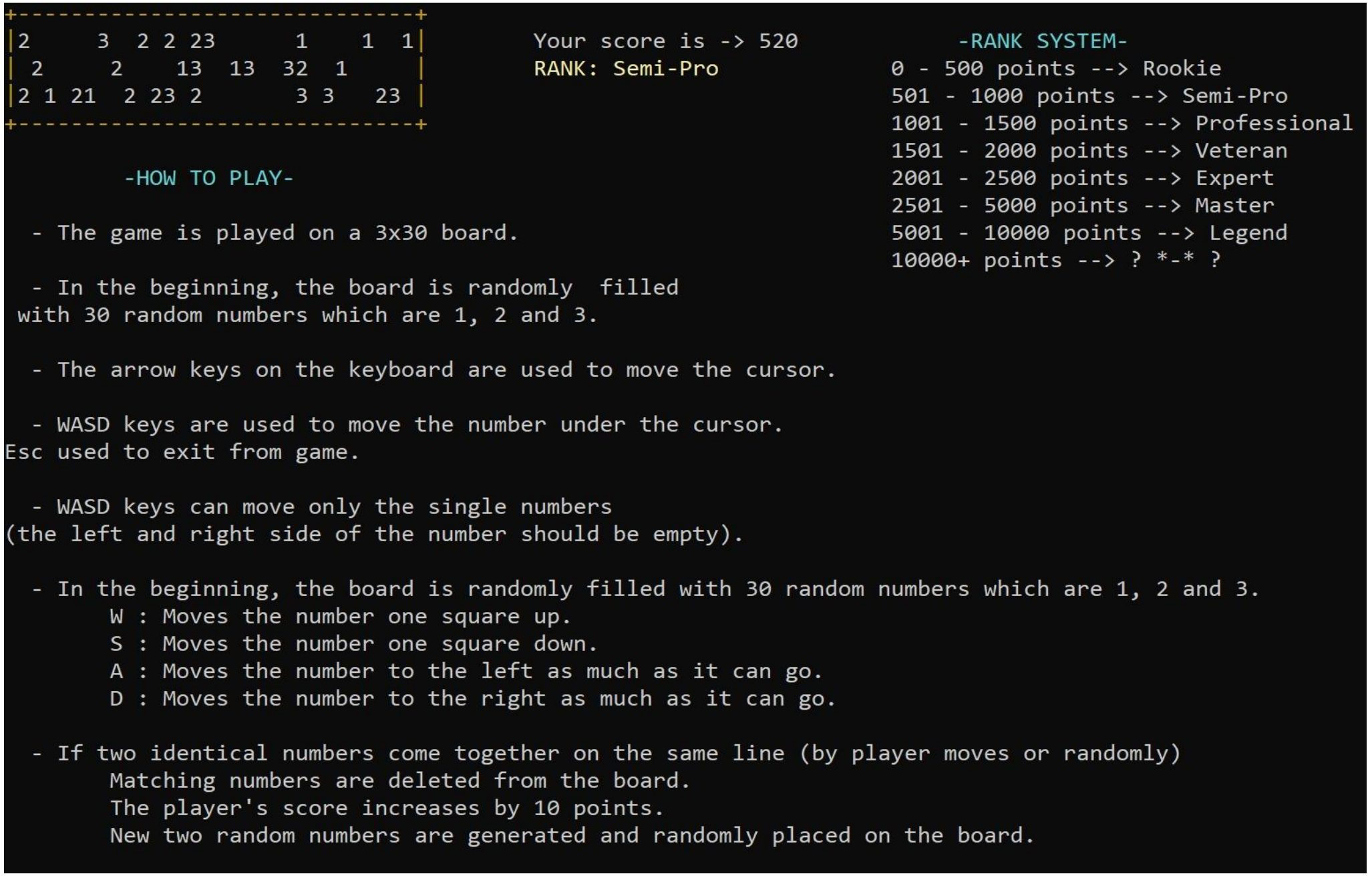
```
for (int j = 0; j < 29; j++)
{   // horizontal check
    if ((firstLine[j] == firstLine[j + 1]) && firstLine[j] != " ")
    {
        firstLine[j] = " "; firstLine[j + 1] = " ";
        for (int i = 0; i < 30; i++) // print the new board
        {
            Console.SetCursorPosition(i + 1, 1);
            Console.WriteLine(firstLine[i]);
        }
        score += 10;
        matchCount += 2;
        if (OperatingSystem.IsWindows()) Console.Beep(1320, 100); //beep sound for matching
        Console.SetCursorPosition(40, 1); Console.WriteLine("Your score is" + " -> " + score);
        Console.SetCursorPosition(40, 2);
        if (score < 500) { Console.WriteLine("RANK: Rookie      "); }
        else if (score > 500 && score < 1000) { Console.ForegroundColor = ConsoleColor.Yellow; Console.WriteLine("RANK: Semi-Pro    "); Console.ResetColor(); }
        else if (score > 1000 && score < 1500) { Console.ForegroundColor = ConsoleColor.DarkYellow; Console.WriteLine("RANK: Professional"); Console.ResetColor(); }
        else if (score > 1500 && score < 2000) { Console.ForegroundColor = ConsoleColor.Cyan; Console.WriteLine("RANK: Veteran     "); Console.ResetColor(); }
        else if (score > 2000 && score < 2500) { Console.ForegroundColor = ConsoleColor.Green; Console.WriteLine("RANK: Expert      "); Console.ResetColor(); }
        else if (score > 2500 && score < 5000) { Console.ForegroundColor = ConsoleColor.Blue; Console.WriteLine("RANK: Master      "); Console.ResetColor(); }
        else if (score > 5000 && score < 10000) { Console.ForegroundColor = ConsoleColor.Magenta; Console.WriteLine("RANK: Legend     "); Console.ResetColor(); }
        else if (score > 10000) { Console.SetCursorPosition(37, 2); Console.ForegroundColor = ConsoleColor.Red; Console.WriteLine("*-* I THINK YOU ARE GOD *-*"); Console.ResetColor(); }
    }
```

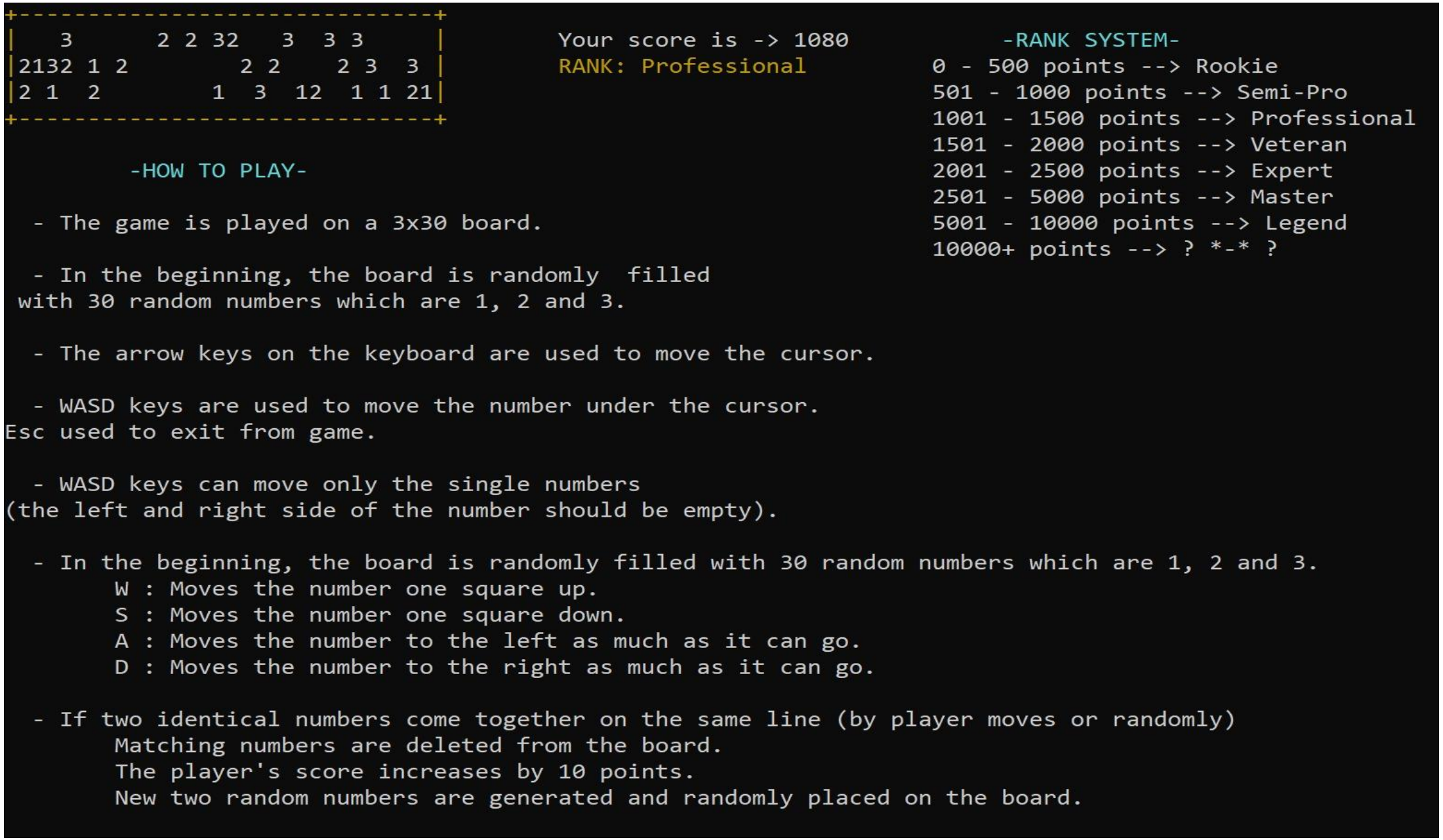
ALGORTIHM THAT PRINTS RANDOM NUMBERS ON BOARD AFTER ANY MATCHING OCCURS

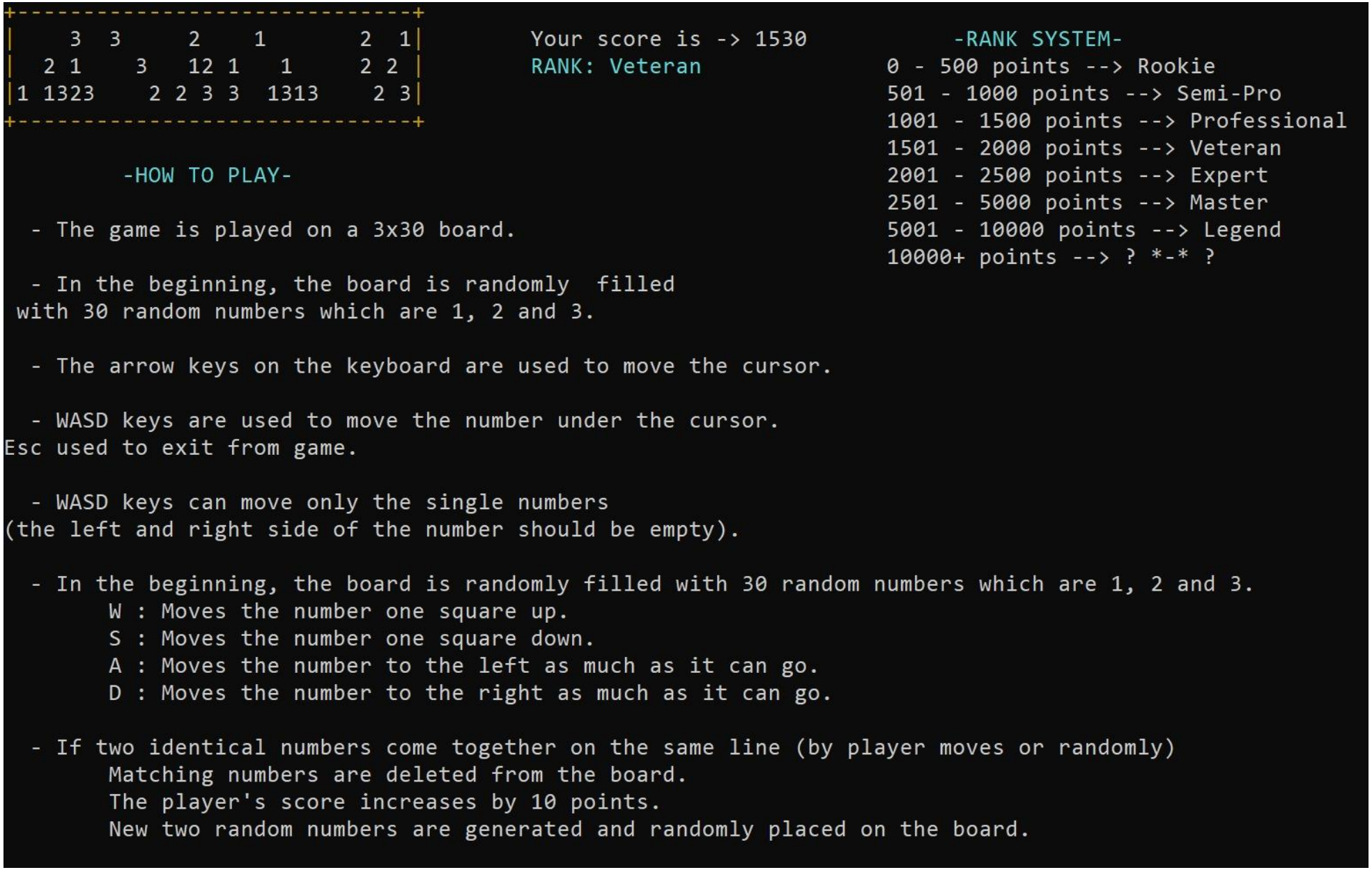
```
for (int i = 0; i < matchCount; i++)
{
    int numberOnTheTable;
    Random random = new();
    columnOnTheTable = random.Next(0, 30);
    lineOnTheTable = random.Next(0, 3);
    numberOnTheTable = random.Next(1, 4);
    // for check if cell is empty
    if (lineOnTheTable == 0 && firstLine[columnOnTheTable] == " ")
    {
        firstLine[columnOnTheTable] = Convert.ToString(numberOnTheTable);
        Console.SetCursorPosition(columnOnTheTable + 1, 1); Console.WriteLine(numberOnTheTable);
    }
    else if (lineOnTheTable == 1 && secondLine[columnOnTheTable] == " ")
    {
        secondLine[columnOnTheTable] = Convert.ToString(numberOnTheTable);
        Console.SetCursorPosition(columnOnTheTable + 1, 2); Console.WriteLine(numberOnTheTable);
    }
    else if (lineOnTheTable == 2 && thirdLine[columnOnTheTable] == " ")
    {
        thirdLine[columnOnTheTable] = Convert.ToString(numberOnTheTable);
        Console.SetCursorPosition(columnOnTheTable + 1, 3); Console.WriteLine(numberOnTheTable);
    }
    else i--;
}
Console.SetCursorPosition(cursorx, cursory);
matchCount = 0;
```

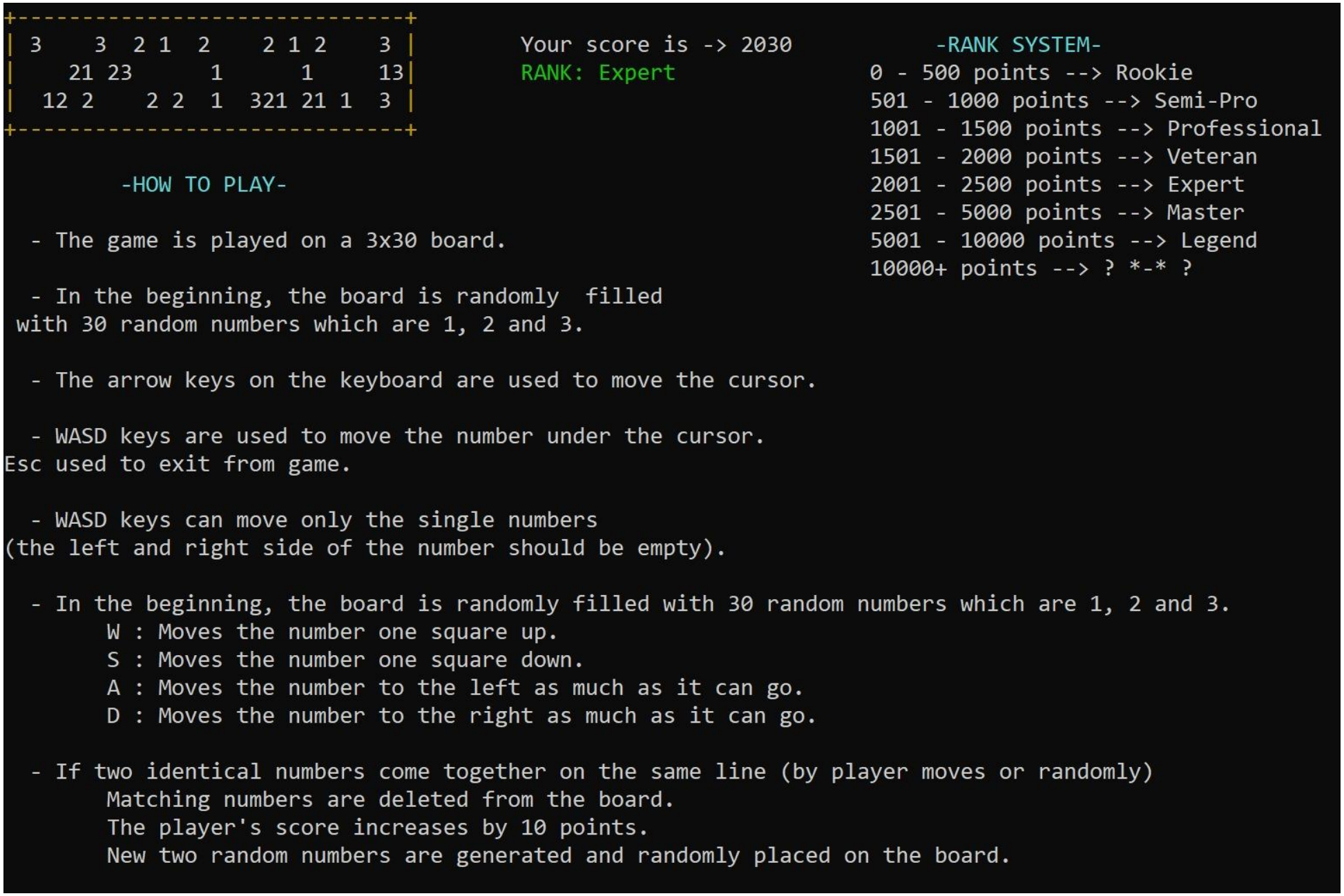
SCREENSHOTS


+------------------------------+
|2   13   3   1   1   2 3 |
|2 1   1 3   1 212   232 |
|3 3 2   1 1 2   321   2 2|
+------------------------------+
Your score is -> 30
RANK: Rookie
-RANK SYSTEM-
0 - 500 points --> Rookie
501 - 1000 points --> Semi-Pro
1001 - 1500 points --> Professional
1501 - 2000 points --> Veteran
2001 - 2500 points --> Expert
2501 - 5000 points --> Master
5001 - 10000 points --> Legend
10000+ points --> ? *-* ?
-HOW TO PLAY-
- The game is played on a 3x30 board.
- In the beginning, the board is randomly filled
with 30 random numbers which are 1, 2 and 3.
- The arrow keys on the keyboard are used to move the cursor.
- WASD keys are used to move the number under the cursor.
Esc used to exit from game.
- WASD keys can move only the single numbers
(the left and right side of the number should be empty).
- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.
W : Moves the number one square up.
S : Moves the number one square down.
A : Moves the number to the left as much as it can go.
D : Moves the number to the right as much as it can go.
- If two identical numbers come together on the same line (by player moves or randomly)
Matching numbers are deleted from the board.
The player's score increases by 10 points.
New two random numbers are generated and randomly placed on the board.


+------------------------------+
|2   3 2 2 23   1   1 1|
| 2   2   13 13 32 1 |
|2 1 21 2 23 2   3 3   23 |
+------------------------------+
Your score is -> 520
RANK: Semi-Pro
-RANK SYSTEM-
0 - 500 points --> Rookie
501 - 1000 points --> Semi-Pro
1001 - 1500 points --> Professional
1501 - 2000 points --> Veteran
2001 - 2500 points --> Expert
2501 - 5000 points --> Master
5001 - 10000 points --> Legend
10000+ points --> ? *-* ?
-HOW TO PLAY-
- The game is played on a 3x30 board.
- In the beginning, the board is randomly filled
with 30 random numbers which are 1, 2 and 3.
- The arrow keys on the keyboard are used to move the cursor.
- WASD keys are used to move the number under the cursor.
Esc used to exit from game.
- WASD keys can move only the single numbers
(the left and right side of the number should be empty).
- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.
W : Moves the number one square up.
S : Moves the number one square down.
A : Moves the number to the left as much as it can go.
D : Moves the number to the right as much as it can go.
- If two identical numbers come together on the same line (by player moves or randomly)
Matching numbers are deleted from the board.
The player's score increases by 10 points.
New two random numbers are generated and randomly placed on the board.


+------------------------------+
| 3   2 2 32   3 3 3 |
|2132 1 2   2 2   2 3 3 |
|2 1 2   1 3 12 1 1 21|
+------------------------------+
Your score is -> 1080
RANK: Professional
-RANK SYSTEM-
0 - 500 points --> Rookie
501 - 1000 points --> Semi-Pro
1001 - 1500 points --> Professional
1501 - 2000 points --> Veteran
2001 - 2500 points --> Expert
2501 - 5000 points --> Master
5001 - 10000 points --> Legend
10000+ points --> ? *-* ?
-HOW TO PLAY-
- The game is played on a 3x30 board.
- In the beginning, the board is randomly filled
with 30 random numbers which are 1, 2 and 3.
- The arrow keys on the keyboard are used to move the cursor.
- WASD keys are used to move the number under the cursor.
Esc used to exit from game.
- WASD keys can move only the single numbers
(the left and right side of the number should be empty).
- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.
W : Moves the number one square up.
S : Moves the number one square down.
A : Moves the number to the left as much as it can go.
D : Moves the number to the right as much as it can go.
- If two identical numbers come together on the same line (by player moves or randomly)
Matching numbers are deleted from the board.
The player's score increases by 10 points.
New two random numbers are generated and randomly placed on the board.


+------------------------------+
| 3 3   2   1   2 1|
| 2 1   3   12 1   1   2 2 |
|1 1323   2 2 3 3 1313   2 3|
+------------------------------+
Your score is -> 1530
RANK: Veteran
-RANK SYSTEM-
0 - 500 points --> Rookie
501 - 1000 points --> Semi-Pro
1001 - 1500 points --> Professional
1501 - 2000 points --> Veteran
2001 - 2500 points --> Expert
2501 - 5000 points --> Master
5001 - 10000 points --> Legend
10000+ points --> ? *-* ?
-HOW TO PLAY-
- The game is played on a 3x30 board.
- In the beginning, the board is randomly filled
with 30 random numbers which are 1, 2 and 3.
- The arrow keys on the keyboard are used to move the cursor.
- WASD keys are used to move the number under the cursor.
Esc used to exit from game.
- WASD keys can move only the single numbers
(the left and right side of the number should be empty).
- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.
W : Moves the number one square up.
S : Moves the number one square down.
A : Moves the number to the left as much as it can go.
D : Moves the number to the right as much as it can go.
- If two identical numbers come together on the same line (by player moves or randomly)
Matching numbers are deleted from the board.
The player's score increases by 10 points.
New two random numbers are generated and randomly placed on the board.


+------------------------------+
| 3 3 2 1 2 2 1 2 3 |
| 21 23 1 1 13|
| 12 2 2 2 1 321 21 1 3 |
+------------------------------+
Your score is -> 2030
RANK: Expert
-RANK SYSTEM-
0 - 500 points --> Rookie
501 - 1000 points --> Semi-Pro
1001 - 1500 points --> Professional
1501 - 2000 points --> Veteran
2001 - 2500 points --> Expert
2501 - 5000 points --> Master
5001 - 10000 points --> Legend
10000+ points --> ? *-* ?
-HOW TO PLAY-
- The game is played on a 3x30 board.
- In the beginning, the board is randomly filled
with 30 random numbers which are 1, 2 and 3.
- The arrow keys on the keyboard are used to move the cursor.
- WASD keys are used to move the number under the cursor.
Esc used to exit from game.
- WASD keys can move only the single numbers
(the left and right side of the number should be empty).
- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.
W : Moves the number one square up.
S : Moves the number one square down.
A : Moves the number to the left as much as it can go.
D : Moves the number to the right as much as it can go.
- If two identical numbers come together on the same line (by player moves or randomly)
Matching numbers are deleted from the board.
The player's score increases by 10 points.
New two random numbers are generated and randomly placed on the board.

```
+------------------------------+
|   32 1      1 31 3  3 1 13 23|
| 1  1   1   2    1    2  2   3|
|  2    3       3  12  2  31  2|
+------------------------------+
```

Your score is -> 2590

RANK: Master

-RANK SYSTEM-

0 - 500 points --> Rookie
501 - 1000 points --> Semi-Pro
1001 - 1500 points --> Professional
1501 - 2000 points --> Veteran
2001 - 2500 points --> Expert
2501 - 5000 points --> Master
5001 - 10000 points --> Legend
10000+ points --> ? *-* ?

-HOW TO PLAY-

- The game is played on a 3x30 board.

- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.

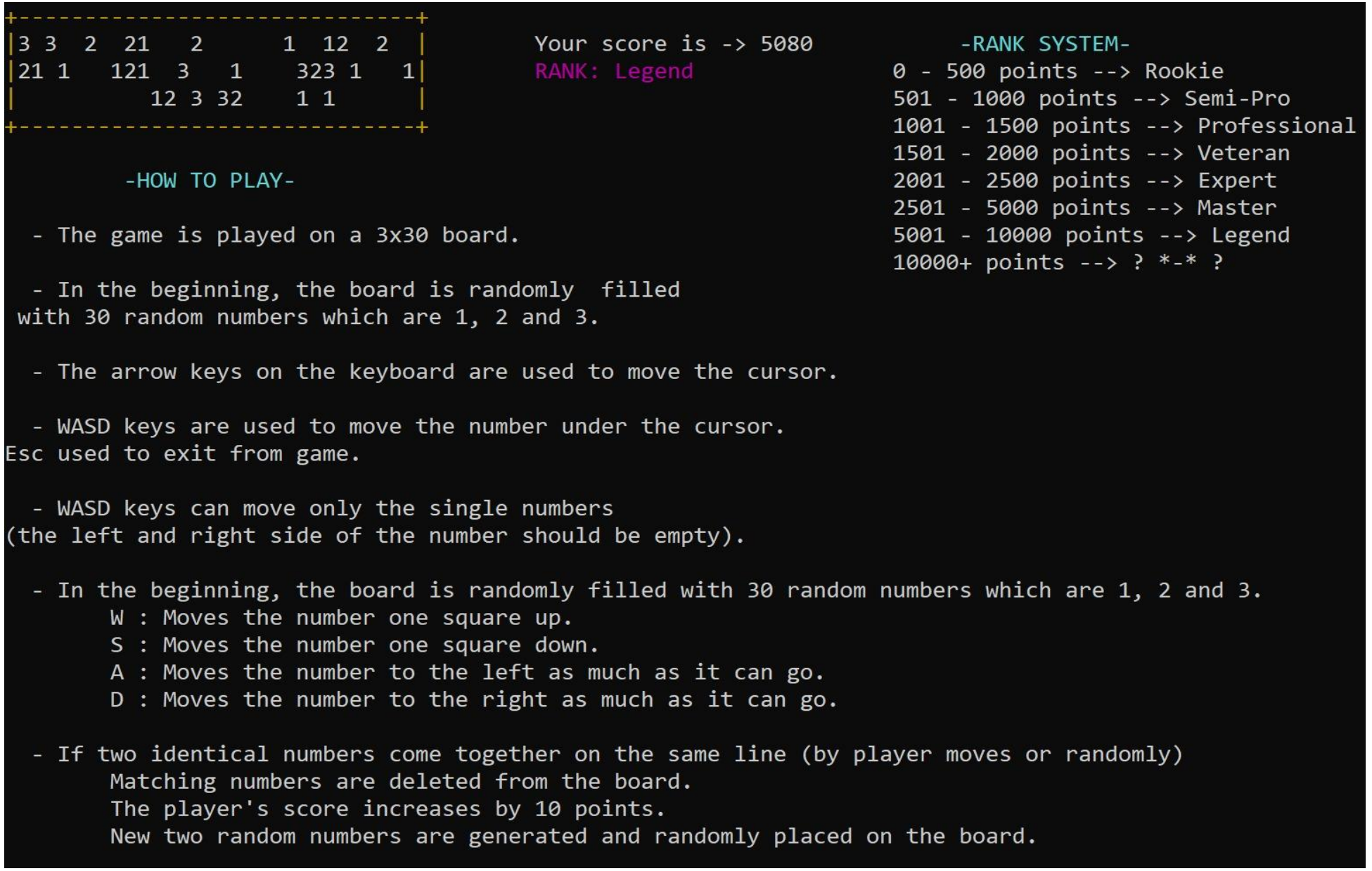
- The arrow keys on the keyboard are used to move the cursor.

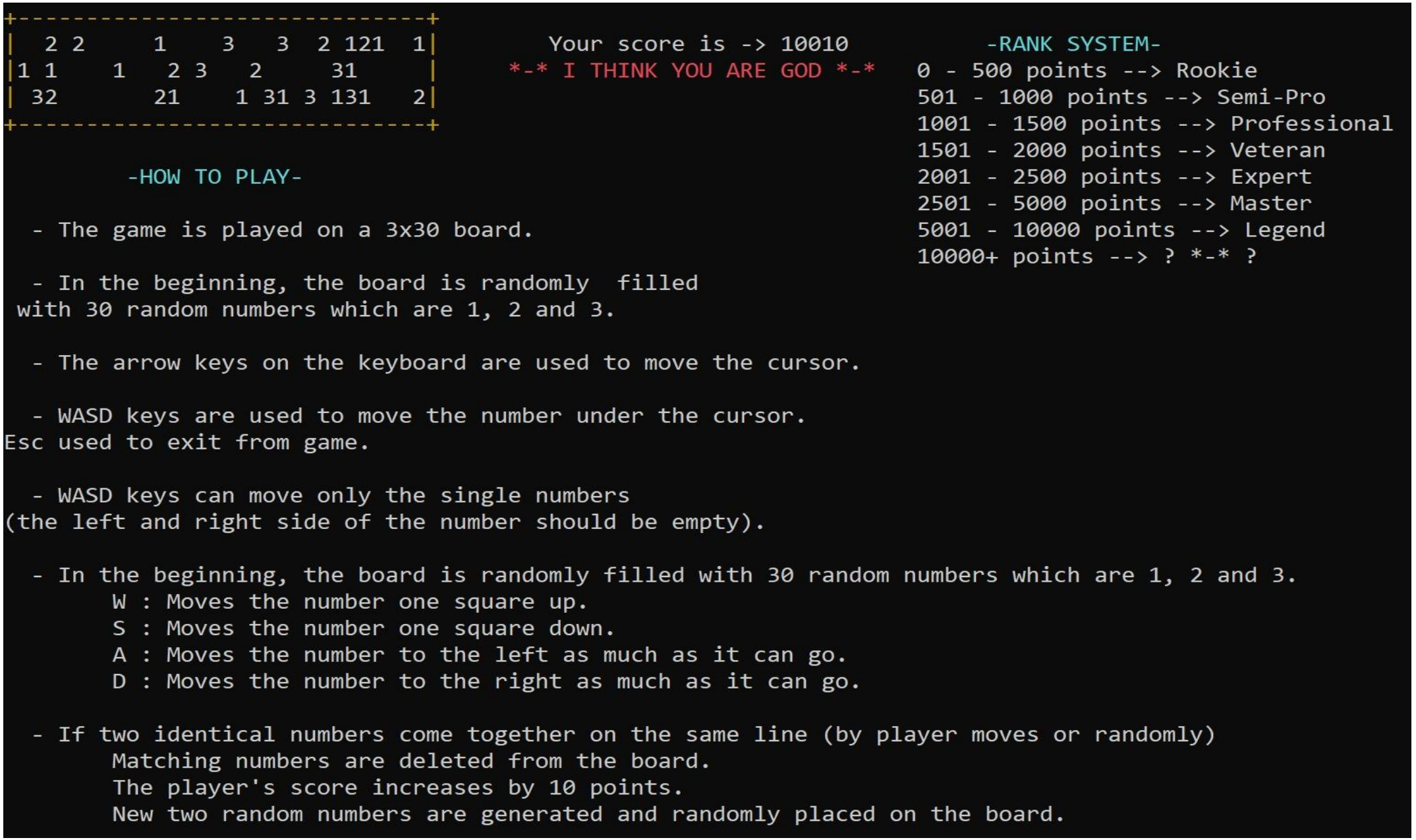
- WASD keys are used to move the number under the cursor.
Esc used to exit from game.

- WASD keys can move only the single numbers
(the left and right side of the number should be empty).

- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.
  - W : Moves the number one square up.
  - S : Moves the number one square down.
  - A : Moves the number to the left as much as it can go.
  - D : Moves the number to the right as much as it can go.

- If two identical numbers come together on the same line (by player moves or randomly)
  - Matching numbers are deleted from the board.
  - The player's score increases by 10 points.
  - New two random numbers are generated and randomly placed on the board.


|3 3 2 21 2 1 12 2 |
|21 1 121 3 1 323 1 1|
| 12 3 32 1 1 |
Your score is -> 5080
RANK: Legend
-RANK SYSTEM-
0 - 500 points --> Rookie
501 - 1000 points --> Semi-Pro
1001 - 1500 points --> Professional
1501 - 2000 points --> Veteran
2001 - 2500 points --> Expert
2501 - 5000 points --> Master
5001 - 10000 points --> Legend
10000+ points --> ? *-* ?
-HOW TO PLAY-
- The game is played on a 3x30 board.
- In the beginning, the board is randomly filled
with 30 random numbers which are 1, 2 and 3.
- The arrow keys on the keyboard are used to move the cursor.
- WASD keys are used to move the number under the cursor.
Esc used to exit from game.
- WASD keys can move only the single numbers
(the left and right side of the number should be empty).
- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.
W : Moves the number one square up.
S : Moves the number one square down.
A : Moves the number to the left as much as it can go.
D : Moves the number to the right as much as it can go.
- If two identical numbers come together on the same line (by player moves or randomly)
Matching numbers are deleted from the board.
The player's score increases by 10 points.
New two random numbers are generated and randomly placed on the board.


+------------------------------+
| 2 2   1   3   3 2 121 1|
|1 1   1   2 3   2   31 |
| 32   21   1 31 3 131   2|
+------------------------------+
Your score is -> 10010
- I THINK YOU ARE GOD *-*
-RANK SYSTEM-
0 - 500 points --> Rookie
501 - 1000 points --> Semi-Pro
1001 - 1500 points --> Professional
1501 - 2000 points --> Veteran
2001 - 2500 points --> Expert
2501 - 5000 points --> Master
5001 - 10000 points --> Legend
10000+ points --> ? *-* ?
-HOW TO PLAY-
- The game is played on a 3x30 board.
- In the beginning, the board is randomly filled
with 30 random numbers which are 1, 2 and 3.
- The arrow keys on the keyboard are used to move the cursor.
- WASD keys are used to move the number under the cursor.
Esc used to exit from game.
- WASD keys can move only the single numbers
(the left and right side of the number should be empty).
- In the beginning, the board is randomly filled with 30 random numbers which are 1, 2 and 3.
W : Moves the number one square up.
S : Moves the number one square down.
A : Moves the number to the left as much as it can go.
D : Moves the number to the right as much as it can go.
- If two identical numbers come together on the same line (by player moves or randomly)
Matching numbers are deleted from the board.
The player's score increases by 10 points.
New two random numbers are generated and randomly placed on the board.

REFERENCES

https://pages.mtu.edu/~suits/notefreqs.html

https://superuser.com/questions/227939/how-to-make-the-pc-speaker-beep-from-the-windows-7-command-prompt

https://docs.microsoft.com/en-us/dotnet/api/system.console.beep

https://www.w3schools.com/cs/cs_arrays.php

https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/

Thanks for listening!