

2019

Software Development Assignment



Hal16829469

Callum Hall

4/2/2019

Contents

Exercise 1.....	2
Player 1.....	2
Player 2.....	3
Exercise 2.....	5
Main method.....	5
Player 2 Loop.....	7
Draw Grid Loop	10
Life Points Method	10
Problems	11

Exercise 1

Player 1

```
static void Main(string[] args)
{
    //Asks the player 1 to input 7 values
    Console.WriteLine("Player 1 please enter 7 values between 1 and 10!");
    //Creates the array
    int[] values = new int[7];
    int number = 0;
    //For loop that allows the player 1 to enter the 7 values
    for (int i = 0; i < values.Length; i++)
    {
        number++;
        Console.Write("Enter number " + number + " > ");
        values[i] = int.Parse(Console.ReadLine());
        if (values[i] > 10 || values[i] < 1)
        {
            Console.WriteLine("Must be between 1 and 10");
            Console.ReadLine();
            Environment.Exit(0);
        }
    }
}
```

try parse?

Abandon
program
for an
invalid
entry?

~~2nd~~ A bit harsh

This is the first part of the code that asks the first player to enter 7 numbers between 1 and 10 using an array to store the inputs and a for loop to let the user input the numbers. This code also makes sure that player 1 inputs a number between 1 and 10 or it will give an error and close the program.

```
for (int i = 0; i < values.Length; i++)
{
    Console.WriteLine(values[i]);
}
```

This code is a temporary test to make sure all the numbers that are input into the array work correctly. From the screenshot below you can see that all the numbers were input correctly.

```
Player 1 please enter 7 values between 1 and 10!
Enter number 1 > 4
Enter number 2 > 6
Enter number 3 > 3
Enter number 4 > 6
Enter number 5 > 7
Enter number 6 > 2
Enter number 7 > 8
4
6
3
6
7
2
8
```

No logic
prior to
coding

Player 2

```
//This is the player 2 section
Console.WriteLine("Player 2 the first number is " + values[0]);
Console.WriteLine("Is the next number higher (h) or lower (l)");

//This loop calculates whether player 2's guess was higher or lower
for (int i = 1; i < values.Length; i++)
{
    string highlow = Console.ReadLine();
    //This is the if for lower
    if (highlow == "l")
    {
        //If player 2 guesses lower then this will tell them it's correct
        if (values[0] > values[i])
        {
            Console.WriteLine("Correct!, the next number is " + values[i]);
            values[0] = values[i];
        }
    }
    else
    {
        //This tells the player that they are wrong
        Console.WriteLine("Wrong, Player 1 wins!");
    }
}
```

This is some of the second part to the code where all the player 2 guesses are calculated. First it checks to see whether player 2 has entered L for lower or H for higher then checks the first number in the array against the second one. If they were correct then the first number is set to the second number in the array and the other number is using the i variable in the for loop so it increases automatically. If player 2 is wrong then it will display some lose text.

Not a good idea
to overwrite your
data in this way
(ie, values[0] is lost!)
Better to use a new
variable.

2 or L ?
h or H ?

```

//This is the if for higher
if (highlow == "h")
{
    //If player 2 guesses higher this will tell them is correct
    if (values[0] < values[i])
    {
        Console.WriteLine("Correct!, the next number is " + values[i]);
        values[0] = values[i];
    }
    else
    {
        //This tells the player they are wrong
        Console.WriteLine("Wrong, Player 1 wins!");
    }
}
}

Console.WriteLine();
//This is if player 2 beats player 1
Console.WriteLine("You win");
Console.ReadLine();

```

This is the rest of the code which does the same as the previous part but for the higher guess. It checks to see if the guess was correct or incorrect then displays the appropriate message. There is also a win message for player 2 if they beat player 1.

Testy?

Exercise 2

Main method

```
static void Main(string[] args)
{
    Console.WriteLine("Player 1 place treasures now:");
    //player 1 x and y arrays for treasure placement
    int[] x = new int[4];
    int[] y = new int[4];

    int lifepoints = 3;

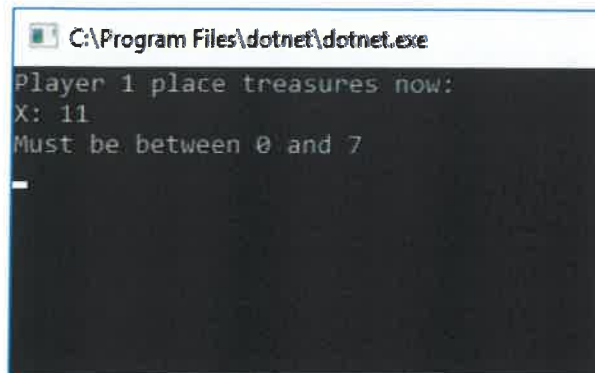
    //player 1 treasure placement loop
    for (int i = 1; i < x.Length; i++)
    {
        //if the x is bigger than 7 and less than 0 show error
        Console.Write("X: ");
        x[i] = int.Parse(Console.ReadLine());
        if (x[i] > 7 || x[i] < 0)
        {
            Console.WriteLine("Must be between 0 and 7");
            Console.ReadLine();
            Environment.Exit(0);
        }

        //if the y is bigger than 7 and less than 0 show error
        Console.Write("Y: ");
        y[i] = int.Parse(Console.ReadLine());
        if (y[i] > 7 || y[i] < 0)
        {
            Console.WriteLine("Must be between 0 and 7");
            Console.ReadLine();
            Environment.Exit(0);
        }
        Console.WriteLine("Treasure " + i + " placed");
    }
}
```

This first part of code controls player 1s inputs and puts them into the array. There is also validation for each x and y guess to make sure the number that has been input is between 0 and 7.

Any use?

This is a test for the player 1 treasure placement. The expected result is that the program will display a message telling the user that it must be between 0 and 7 and then close.



```
C:\Program Files\dotnet\dotnet.exe
Player 1 place treasures now:
X: 11
Must be between 0 and 7
_
```

```
//player 2 guess variables
int[] p2x = new int[9];
int[] p2y = new int[9];
int[] p2correct = new int[4];

//draws the grid after all the treasure is placed
DrawGrid(p2x,p2y, x, y);
Console.WriteLine();
```

These are the player 2 arrays which are used for the grid and the player's lives.



Player 2 Loop

```
//loops that check player 2s guesses and redraws the grid
for (int i = 1; i < 8; i++)
{
    Console.WriteLine("Player 2 guess");
    Console.Write("X: ");
    p2x[i] = int.Parse(Console.ReadLine());
    Console.Write("Y: ");
    p2y[i] = int.Parse(Console.ReadLine());

    //checks is player 2 has found a treasure then redraws the grid
    if (p2x[i] == x[1] && p2y[i] == y[1] && p2correct[1] != 1)
    {
        Console.WriteLine("You found a treasure!");
        p2correct[1] = 1;
        Console.WriteLine();
        DrawGrid(p2x, p2y, x, y);
        Console.WriteLine();
    }
    else
    {
        if (p2x[i] == x[2] && p2y[i] == y[2] && p2correct[2] != 1)
        {
            Console.WriteLine("You found a treasure!");
            p2correct[2] = 1;
            Console.WriteLine();
            DrawGrid(p2x, p2y, x, y);
            Console.WriteLine();
        }
        else
        {
            //if player 2 has not found a treasure then redraws the grid
        }
    }
}
```

This first half of the loop is where the player 2 makes a guess which then goes into an array which then gets checked against the grid to see if the player guessed correctly and if they have, it redraws the grid.

tryParse Int?
Validating
input?

```

else
{
    if (p2x[i] == x[3] && p2y[i] == y[3] && p2correct[3] != 1)
    {
        Console.WriteLine("You found a treasure!");
        p2correct[3] = 1;
        Console.WriteLine();
        DrawGrid(p2x, p2y, x, y);
        Console.WriteLine();
    }
    else
    {
        Console.WriteLine();
        DrawGrid(p2x, p2y, x, y);
        Console.WriteLine("You lose a life");
        lifepoints = lifepoints - 1;
        LifePoints(lifepoints);
    }
}
}

//if that checks to see if player 2 has guessed all locations correctly
if (p2correct[1] == 1 && p2correct[2] == 1 && p2correct[3] == 1)
{
    Console.WriteLine("Player 2 Wins!");
    Console.ReadLine();
    Environment.Exit(0);
}
}

Console.ReadLine();

```

If none of the guesses were correct then player 2 will lose a life point and a message will appear telling them. At the bottom is a loop that checks to see if all player 2s guesses are correct and if they are it will show a win message then close.

This is a test for the drawGrid method. The expected result is that if player 2 guesses correctly there will be an X and if they guess wrong there will be an N and they will lose a life.

```
C:\Program Files\dotnet\dotnet.exe
You found a treasure!

  0  1  2  3  4  5  6  7
0 - - - - - - -
1 - - X - - - -
2 - - - - - - -
3 - - - - - - -
4 - - - - - - -
5 - - - - - - -
6 - - - - - - -
7 - - - - - - -

Player 2 guess
X: 3
Y: 5

  0  1  2  3  4  5  6  7
0 - - - - - - -
1 - - X - - - -
2 - - - - - - -
3 - - - - N - -
4 - - - - - - -
5 - - - - - - -
6 - - - - - - -
7 - - - - - - -

You lose a life
You have 2 life points.

Player 2 guess
X: 
```

Draw Grid Loop

```
//method which draws the grid
static void DrawGrid(int[] p2x, int[] p2y, int[] x, int[] y)
{
    //grid array
    string[,] grid = new string[10, 10];

    //loops that create the grid
    Console.WriteLine(" 0  1  2  3  4  5  6  7");
    for (int i = 0; i < 8; i++)
    {
        Console.Write(i + " ");
        for (int j = 0; j < 8; j++)
        {
            grid[i, j] = "  ";
            for (int f = 0; f < 8; f++)
            {
                //loops that check player 2s guesses to the locations and places an x if they are correct or an N if they are not
                if (p2x[f] == x[1] && p2y[f] == y[1])
                {
                    grid[p2x[f], p2y[f]] = " X ";
                }
                else
                {
                    if (p2x[f] == x[2] && p2y[f] == y[2])
                    {
                        grid[p2x[f], p2y[f]] = " X ";
                    }
                    else
                    {
                        if (p2x[f] == x[3] && p2y[f] == y[3])
                        {
                            grid[p2x[f], p2y[f]] = " X ";
                        }
                    }
                    else
                    {
                        if (p2x[f] == 0 && p2y[f] == 0)
                        {
                            grid[p2x[f], p2y[f]] = "  ";
                        }
                        else
                        {
                            //if the guess is incorrect place an N
                            grid[p2x[f], p2y[f]] = " N ";
                        }
                    }
                }
            }
        }
        Console.WriteLine(grid[i, j] + " ");
    }
    Console.WriteLine();
}
```

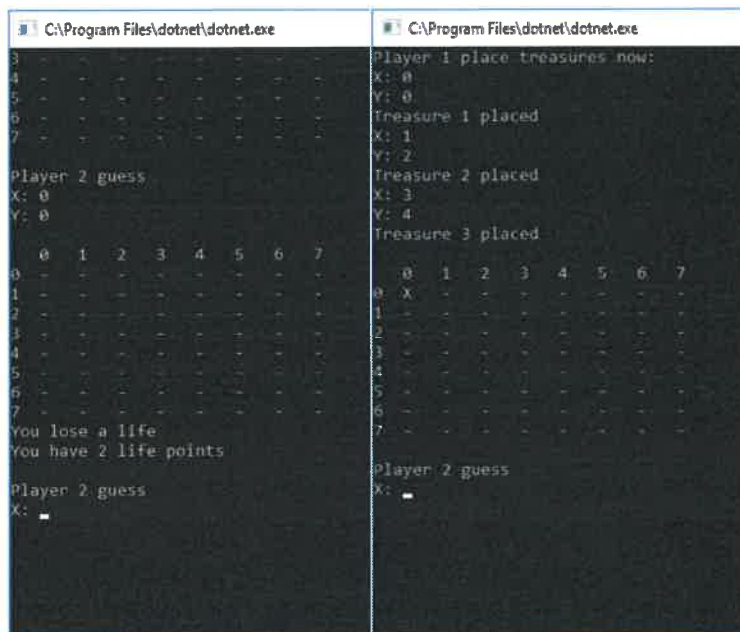
This is a new method called DrawGrid which is used to create the main grid for the game. It uses a multidimensional array to draw all the spaces using two for loops. This method also has loops that check if player 2 has guessed and if they have guessed correctly it will place an X in the spot they chose and if they get it wrong there will be an N placed in the spot they got wrong.

Life Points Method

```
//method that checks player 2s life points to see if player 1 wins or if life points need to be taken
static void LifePoints(int lifepoints)
{
    if (lifepoints <= 0)
    {
        Console.WriteLine("You lose player 1 wins!");
        Console.ReadLine();
        Environment.Exit(0);
    }
    else
    {
        Console.WriteLine("You have " + lifepoints + " life points");
        Console.WriteLine();
    }
}
```

This is also a new method called LifePoints which is used to check the life points of player 2 to see if a life needs to be subtracted and then if not it will display a message with how many life points you have.

Problems



A problem with my program is that if the user guesses 0,0 and its incorrect nothing will happen. Also if player 1 places a treasure at 0,0 then it will be found before player 2 can even guess.

Appendix

Program 1

```
using System;
//callum hall HAL16829469
namespace AssignmentExercise1
{
    class Program
    {
        static void Main(string[] args)
        {
            //Asks the player 1 to input 7 values
            Console.WriteLine("Player 1 please enter 7 values between 1 and 10!");
            //Creates the array
            int[] values = new int[7];
            int number = 0;
            //For loop that allows the player 1 to enter the 7 values
            for (int i = 0; i < values.Length; i++)
            {
                number++;
                Console.Write("Enter number " + number + " > ");
                values[i] = int.Parse(Console.ReadLine());
                if (values[i] > 10 || values[i] < 1)
                {
                    Console.WriteLine("Must be between 1 and 10");
                    Console.ReadLine();
                    Environment.Exit(0);
                }
            }
        }
    }
}
```

```

    }

    Console.WriteLine();
    //This is the player 2 section
    Console.WriteLine("Player 2 the first number is " + values[0]);
    Console.WriteLine("Is the next number higher (h) or lower (l)");

    //This loop calculates whether player 2's guess was higher or lower
    for (int i = 1; i < values.Length; i++)
    {
        string highlow = Console.ReadLine();
        //This is the if for lower
        if (highlow == "l")
        {
            //If player 2 guesses lower then this will tell them it's correct
            if (values[0] > values[i])
            {
                Console.WriteLine("Correct!, the next number is " +
values[i]);

                values[0] = values[i];
            }
            else
            {
                //This tells the player that they are wrong
                Console.WriteLine("Wrong, Player 1 wins!");
            }
        }
        //This is the if for higher
        if (highlow == "h")
        {
            //If player 2 guesses higher this will tell them it's correct
            if (values[0] < values[i])
            {
                Console.WriteLine("Correct!, the next number is " +
values[i]);

                values[0] = values[i];
            }
            else
            {
                //This tells the player they are wrong
                Console.WriteLine("Wrong, Player 1 wins!");
                Console.ReadLine();
                Environment.Exit(0);
            }
        }
    }
    Console.WriteLine();
    //This is if player 2 beats player 1
    Console.WriteLine("You win");
    Console.ReadLine();
}
}
}
}
}

```

Program 2

```

using System;
//callum hall HAL16829469
namespace AssignmentExercise2
{
    class Program

```

```

{
    static void Main(string[] args)
    {
        Console.WriteLine("Player 1 place treasures now:");
        //player 1 x and y arrays for treasure placement
        int[] x = new int[4];
        int[] y = new int[4];

        int lifepoints = 3;

        //player 1 treasure placement loop
        for (int i = 1; i < x.Length; i++)
        {
            //if the x is bigger than 7 and less than 0 show error
            Console.Write("X: ");
            x[i] = int.Parse(Console.ReadLine());
            if (x[i] > 7 || x[i] < 0)
            {
                Console.WriteLine("Must be between 0 and 7");
                Console.ReadLine();
                Environment.Exit(0);
            }

            //if the y is bigger than 7 and less than 0 show error
            Console.Write("Y: ");
            y[i] = int.Parse(Console.ReadLine());
            if (y[i] > 7 || y[i] < 0)
            {
                Console.WriteLine("Must be between 0 and 7");
                Console.ReadLine();
                Environment.Exit(0);
            }
            Console.WriteLine("Treasure " + i + " placed");
        }
        Console.WriteLine();

        //player 2 guess variables
        int[] p2x = new int[9];
        int[] p2y = new int[9];
        int[] p2correct = new int[4];

        //draws the grid after all the treasure is placed
        DrawGrid(p2x, p2y, x, y);
        Console.WriteLine();

        //loops that check player 2s guesses and redraws the grid
        for (int i = 1; i < 8; i++)
        {
            Console.WriteLine("Player 2 guess");
            Console.Write("X: ");
            p2x[i] = int.Parse(Console.ReadLine());
            Console.Write("Y: ");
            p2y[i] = int.Parse(Console.ReadLine());

            //checks is player 2 has found a treasure then redraws the grid
            if (p2x[i] == x[1] && p2y[i] == y[1] && p2correct[1] != 1)
            {
                Console.WriteLine("You found a treasure!");
                p2correct[1] = 1;
                Console.WriteLine();
                DrawGrid(p2x, p2y, x, y);
                Console.WriteLine();
            }
        }
    }
}

```

```

    }
    else
    {
        if (p2x[i] == x[2] && p2y[i] == y[2] && p2correct[2] != 1)
        {
            Console.WriteLine("You found a treasure!");
            p2correct[2] = 1;
            Console.WriteLine();
            DrawGrid(p2x, p2y, x, y);
            Console.WriteLine();
        }
        else
        {
            if (p2x[i] == x[3] && p2y[i] == y[3] && p2correct[3] != 1)
            {
                Console.WriteLine("You found a treasure!");
                p2correct[3] = 1;
                Console.WriteLine();
                DrawGrid(p2x, p2y, x, y);
                Console.WriteLine();
            }
            else
            {
                Console.WriteLine();
                DrawGrid(p2x, p2y, x, y);
                Console.WriteLine("You lose a life");
                lifepoints = lifepoints - 1;
                LifePoints(lifepoints);
            }
        }
    }
}
//if that checks to see if player 2 has guessed all locations
correctly
if (p2correct[1] == 1 && p2correct[2] == 1 && p2correct[3] == 1)
{
    Console.WriteLine("Player 2 Wins!");
    Console.ReadLine();
    Environment.Exit(0);
}
}
Console.ReadLine();
}

//method which draws the grid
static void DrawGrid(int[] p2x, int[] p2y, int[] x, int[] y)
{
    //grid array
    string[,] grid = new string[10, 10];

    //loops that create the grid
    Console.WriteLine("  0  1  2  3  4  5  6  7");
    for (int i = 0; i < 8; i++)
    {
        Console.Write(i + " ");
        for (int j = 0; j < 8; j++)
        {
            grid[i, j] = " - ";
            for (int f = 0; f < 8; f++)
            {
                //loops that check player 2s guesses to the locations and
                places an X if they are correct or an N if they are not
                if (p2x[f] == x[1] && p2y[f] == y[1])

```

```

        {
            grid[p2x[f], p2y[f]] = " X ";
        }
        else
        {
            if (p2x[f] == x[2] && p2y[f] == y[2])
            {
                grid[p2x[f], p2y[f]] = " X ";
            }
            else
            {
                if (p2x[f] == x[3] && p2y[f] == y[3])
                {
                    grid[p2x[f], p2y[f]] = " X ";
                }
                else
                {
                    if (p2x[f] == 0 && p2y[f] == 0)
                    {
                        grid[p2x[f], p2y[f]] = " - ";
                    }
                    else
                    {
                        //if the guess is incorrect place an N
                        grid[p2x[f], p2y[f]] = " N ";
                    }
                }
            }
        }
    }
    Console.WriteLine(grid[i, j] + " ");
}
Console.WriteLine();
}
}

//method that checks player 2s life points to see if player 1 wins or if life
points need to be taken
static void LifePoints(int lifepoints)
{
    if (lifepoints <= 0)
    {
        Console.WriteLine("You lose player 1 wins!");
        Console.ReadLine();
        Environment.Exit(0);
    }
    else
    {
        Console.WriteLine("You have " + lifepoints + " life points");
        Console.WriteLine();
    }
}
}
}
}
}

```

Program 3

```

using System;
//callum hall HAL16829469
namespace AssignmentExercise3
{

```

```

class Program
{
    static void Main(string[] args)
    {
        int player1 = 0;
        int player2 = 0;
        int turn = 0;
        drawGrid(player1, player2, turn);
        for (int i = 0; i < 50; i++)
        {
            turn = 1;
            Console.WriteLine("Player 1 choose where to place an X:");
            player1 = int.Parse(Console.ReadLine());

            drawGrid(player1, player2, turn);

            turn = 2;
            Console.WriteLine("Player 2 choose where to place an O:");
            player2 = int.Parse(Console.ReadLine());

            drawGrid(player1, player2, turn);
        }

        drawGrid(player1, player2, turn);

        Console.ReadLine();
    }

    static void drawGrid(int player1, int player2, int turn)
    {
        int gridSize = 6;
        string[,] grid = new string[6, 6];
        Console.WriteLine();
        Console.Write("  0   1   2   3   4   5");
        for (int i = 0; i < gridSize; i++)
        {
            Console.WriteLine(" ");
            for (int j = 0; j < gridSize; j++)
            {
                grid[i, j] = " - ";
                if (turn == 1)
                {
                    grid[5, player1] = " X ";
                }
                if (turn == 2)
                {
                    grid[5, player2] = " O ";
                }
                Console.Write(grid[i, j] + " ");
            }
        }
        Console.WriteLine();
    }
}

```

