TRIANSH

CME1251 Project Based Learning - I

BY

2020510034 YUSUF GASSALOĞLU

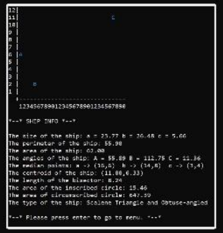
2017510042 HİLAL NUR IŞIK

2021510015 KORAY BORA BOZCA

Outline

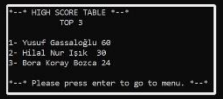
PROGRESS SUMMARY

TRIANSH



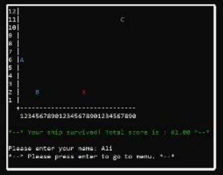
```
12|
11|                  C
10|
9 |
8 |
7 |
6 |A
5 |
4 |
3 |
2 |   B
1 |
  +------------------------------
   123456789012345678901234567890

*--* SHIP INFO *--*

The size of the ship: a = 23.77 b = 26.48 c = 5.66
The perimeter of the ship: 55.90
The area of the ship: 62.00
The angles of the ship: A = 55.89 B = 112.75 C = 11.36
The median points: a -> (16,8)  b -> (14,8)  c -> (3,4)
The centroid of the ship: (11.00,6.33)
The length of the bisector: 8.24
The area of the inscribed circle: 15.46
The area of circumscribed circle: 647.39
The type of the ship: Scalene Triangle and Obtuse-angled

*--* Please press enter to go to menu. *--*
```

The player creates a triangle battleship and tries to dodge from a random shot. If the ship survives, the player gets a point which is the area of the ship. The game area is 30 * 12 units. If the player gets a score that is equal to a score in the table, the name of the player is placed under the old one.



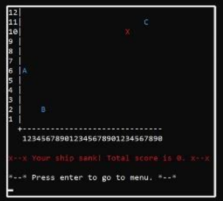
```
*--* HIGH SCORE TABLE *--*
          TOP 3

1- Yusuf Gassaloğlu 60
2- Hilal Nur Işık  30
3- Bora Koray Bozca 24

*--* Please press enter to go to menu. *--*
```

A competitive game that calculates ship properties excellent



```
12|
11|                     C
10|
9 |
8 |
7 |
6 |A
5 |
4 |
3 |
2 |   B         X
1 |
  +------------------------------
   123456789012345678901234567890

*--* Your ship survived! Total score is : 62.00 *--*

Please enter your name: Ali
*--* Please press enter to go to menu. *--*
```

An animated and "colorful" battleship game.
The bombs drop with an animation



```
12|
11|                        C
10|                    X
9 |
8 |
7 |
6 |A
5 |
4 |
3 |
2 |   B
1 |
  +------------------------------
   123456789012345678901234567890

X--X Your ship sank! Total score is 0. X--X

*--* Press enter to go to menu. *--*
```

Don't be too brave. Randomly thrown bomb can explode on your head.

Hilal Nur
ISIK

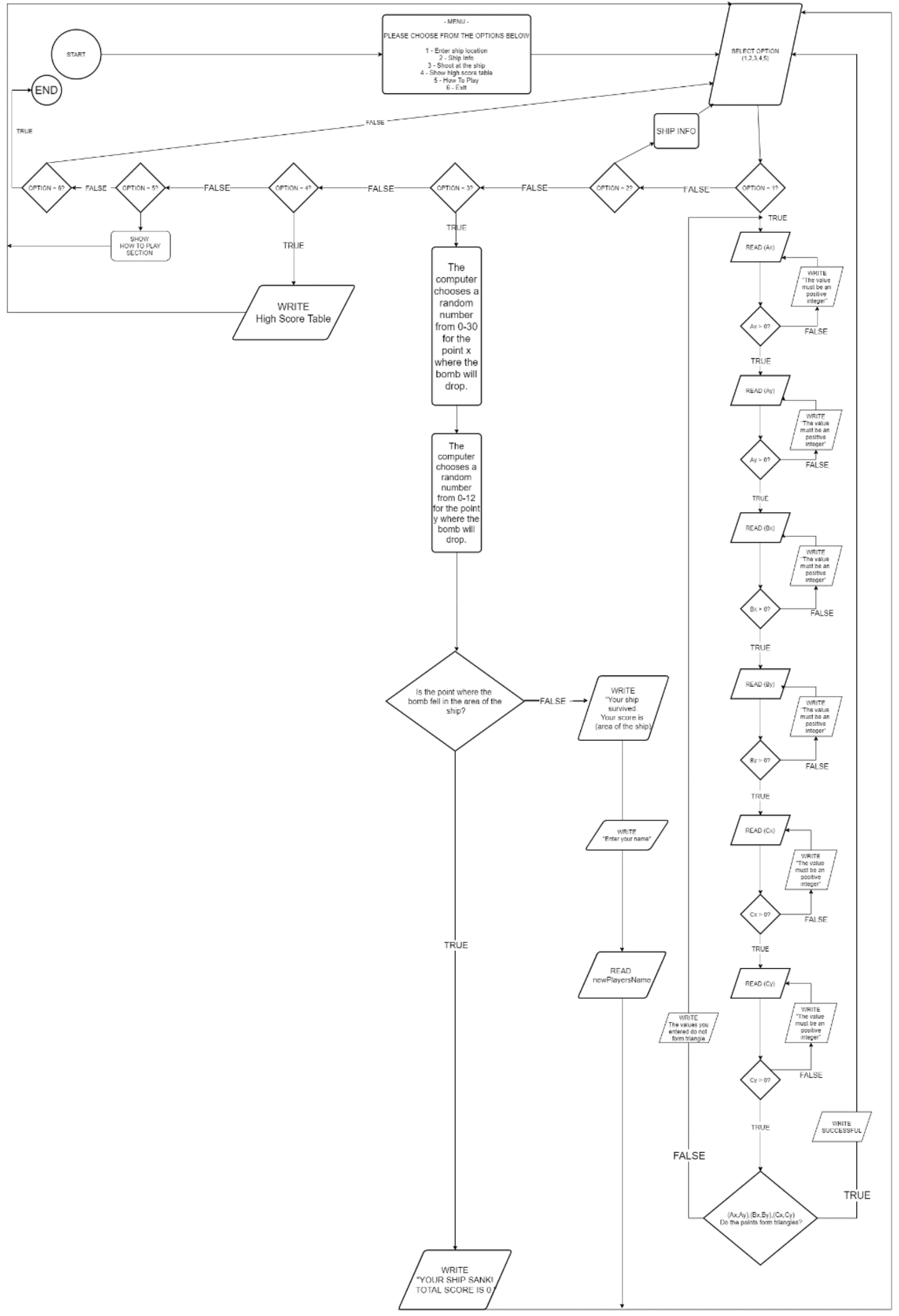
Yusuf
GASSALOGLU

Koray Bora
BOZCA

Console battleship game

TRIANSH

DEUCENG


START
END
- MENU -
PLEASE CHOOSE FROM THE OPTIONS BELOW
1 - Enter ship location
2 - Ship Info
3 - Shoot at the ship
4 - Show high score table
5 - How To Play
6 - Exit
SELECT OPTION (1,2,3,4,5)
SHIP INFO
OPTION = 1?
OPTION = 2?
OPTION = 3?
OPTION = 4?
OPTION = 5?
OPTION = 6?
TRUE
FALSE
SHOW HOW TO PLAY SECTION
WRITE High Score Table
The computer chooses a random number from 0-30 for the point x where the bomb will drop.
The computer chooses a random number from 0-12 for the point y where the bomb will drop.
Is the point where the bomb fell in the area of the ship?
WRITE "Your ship survived. Your score is (area of the ship)
WRITE "Enter your name"
READ newPlayersName
WRITE "YOUR SHIP SANK! TOTAL SCORE IS 0."
READ (Ax)
Ax > 0?
READ (Ay)
Ay > 0?
READ (Bx)
Bx > 0?
READ (By)
By > 0?
READ (Cx)
Cx > 0?
READ (Cy)
Cy > 0?
WRITE "The value must be an positive integer"
(Ax,Ay),(Bx,By),(Cx,Cy) Do the points form triangles?
WRITE The values you entered do not form triangle
WRITE SUCCESSFUL

Requirements

- Algortihm that takes inputs from player.
- Algorithms that calculates asked ship properties.
- Algorithm that shoots to ship.
- Algorithm that writes high score table.

Task Sharing

- We all go individually and compare what we have done. Then we improve the best and choose it.
- Presentation , flowchart, poster and algorithm of the project made by Yusuf Gassaloğlu.
- Code that draws coordinate system written by Koray Bora Bozca. (after progress report)
- Ship info requirements searched by Hilal Nur Işık.

Scheduling

- We proceeded as planned;

1. Understanding the game. Discussing and designing solution alternatives. Drawing flowchart.

2. Creating the necessary variables, screen. Drawing the ship. Calculating basic properties of the ship.

3. Calculating advanced properties of the ship. Triangle validation. Shooting.

4. Menu. Playing the game with all rules. High score table operations. Remaining parts of the game.

Completed Tasks

- We completed all tasks and made additions.

Incomplete Tasks

- We completed all tasks.

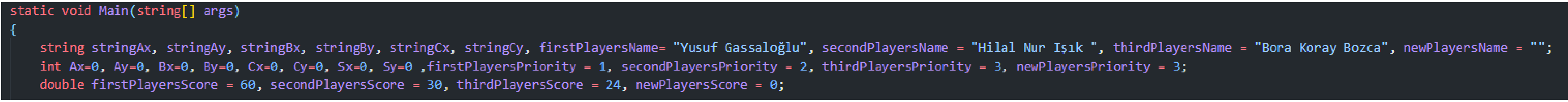
Additional Improvements

- How to Play menu option.
- Alternative algorithm ways in the code by comment lines.
- Colorized view.
- Shooting Animation.

PROBLEMS ENCOUNTERED

- We ran into simple code errors but that didn't bother us.

ALGORITHMS AND SOLUTION STRATEGIES



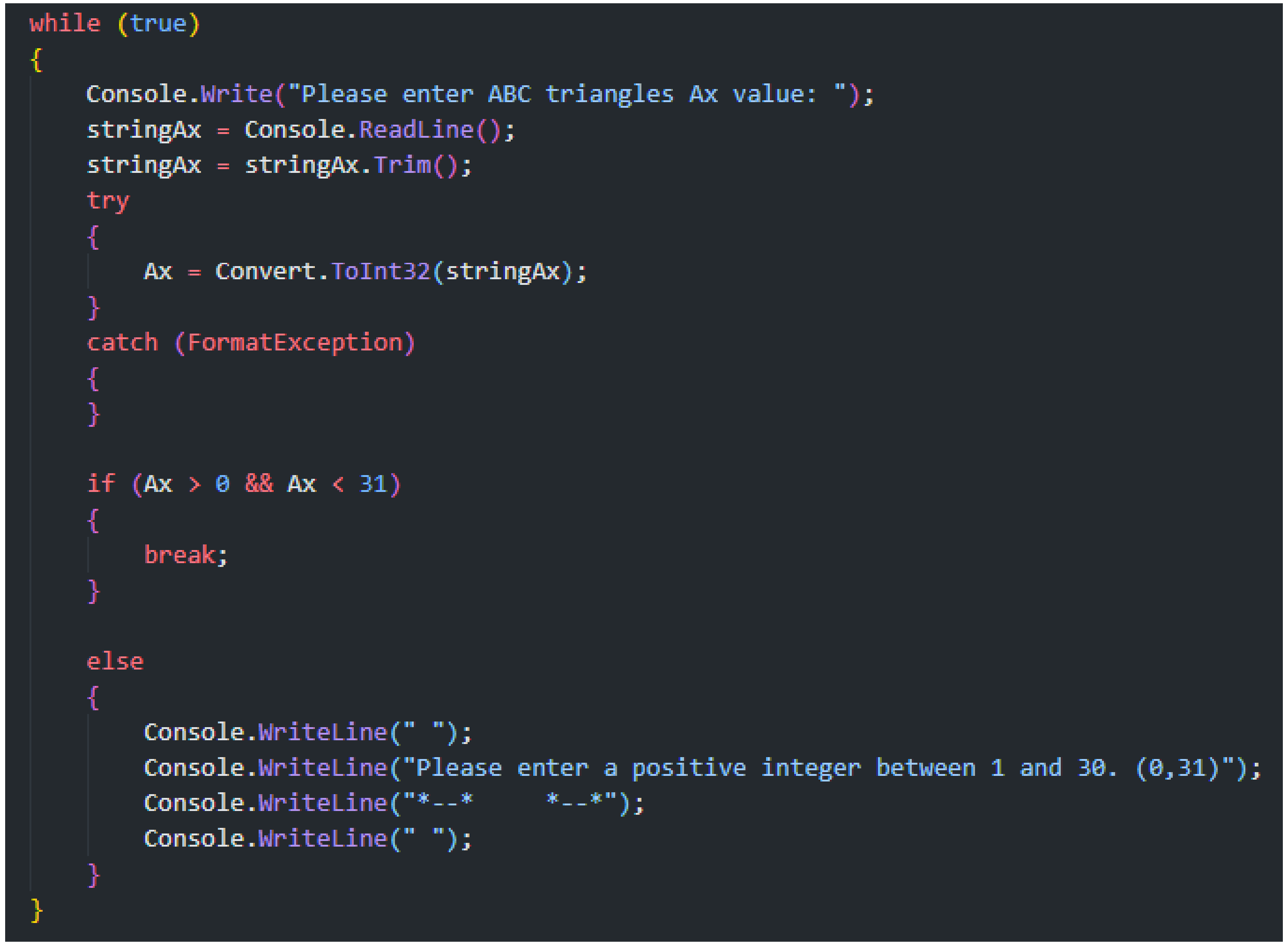
```
static void Main(string[] args)
{
    string stringAx, stringAy, stringBx, stringBy, stringCx, stringCy, firstPlayersName= "Yusuf Gassaloğlu", secondPlayersName = "Hilal Nur Işık ", thirdPlayersName = "Bora Koray Bozca", newPlayersName = "";
    int Ax=0, Ay=0, Bx=0, By=0, Cx=0, Cy=0, Sx=0, Sy=0 ,firstPlayersPriority = 1, secondPlayersPriority = 2, thirdPlayersPriority = 3, newPlayersPriority = 3;
    double firstPlayersScore = 60, secondPlayersScore = 30, thirdPlayersScore = 24, newPlayersScore = 0;
```

We keep the code running faster by keeping less data in memory.

MENU ALGORITHM

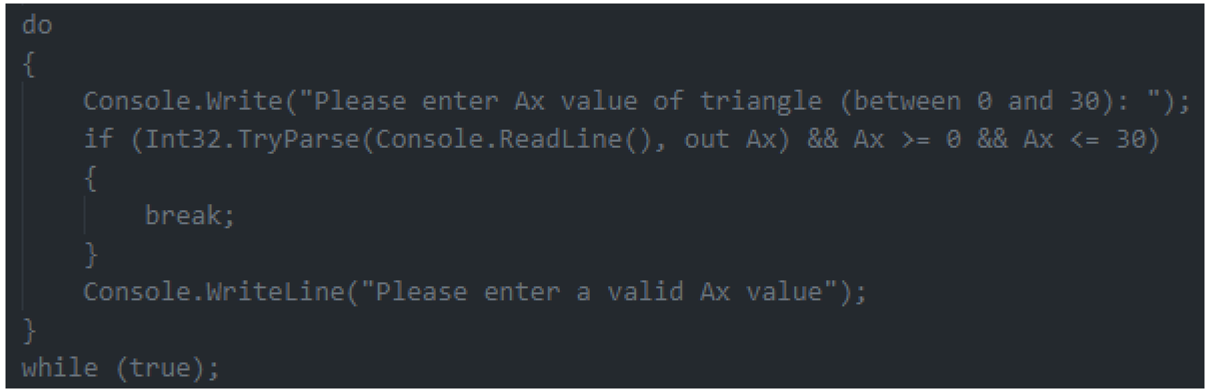
WE USED LOOP FOR MENU. IT ENDS IF THE PLAYER SELECTS EXIT OPTION.

ALGORTIHM THAT GETS INPUTS



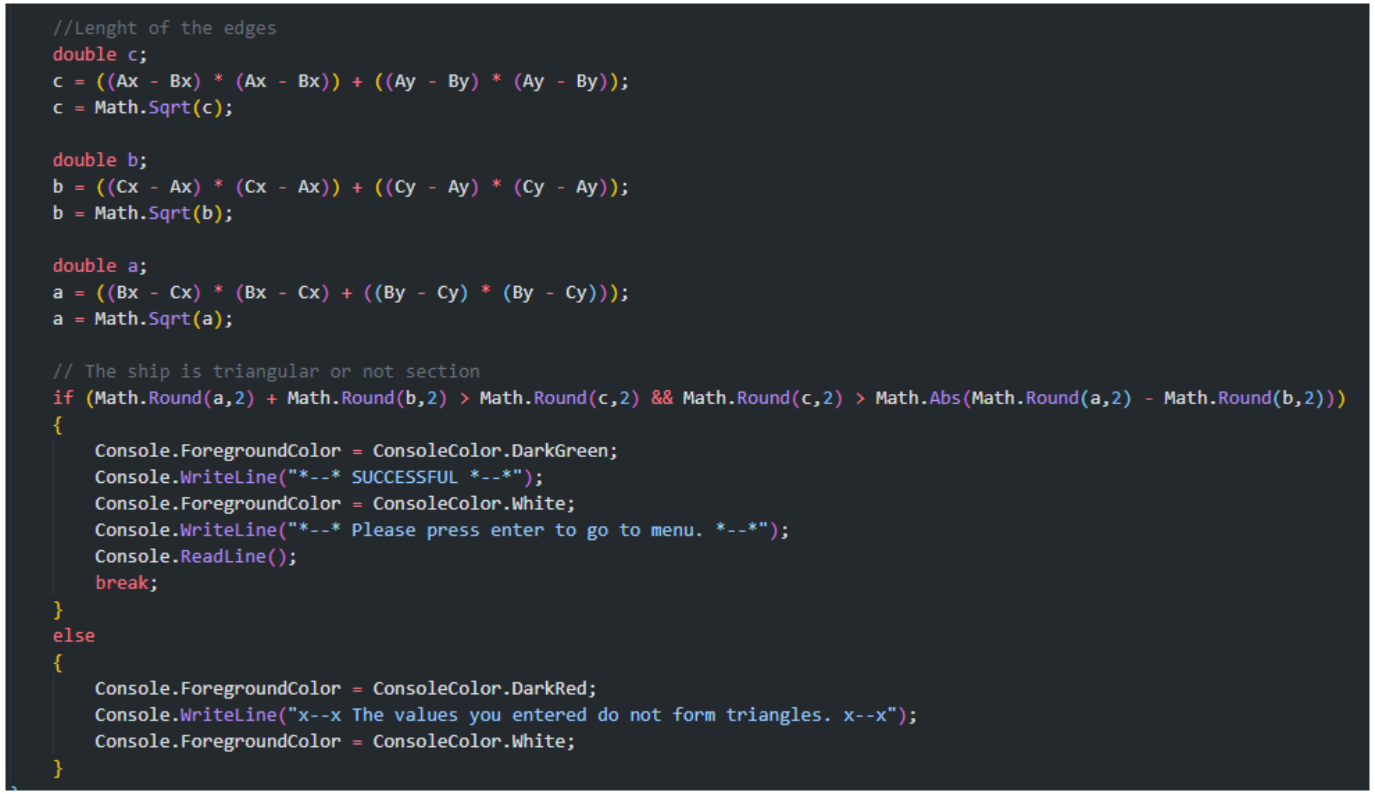
```
while (true)
{
    Console.Write("Please enter ABC triangles Ax value: ");
    stringAx = Console.ReadLine();
    stringAx = stringAx.Trim();
    try
    {
        Ax = Convert.ToInt32(stringAx);
    }
    catch (FormatException)
    {
    }

    if (Ax > 0 && Ax < 31)
    {
        break;
    }

    else
    {
        Console.WriteLine(" ");
        Console.WriteLine("Please enter a positive integer between 1 and 30. (0,31)");
        Console.WriteLine("*--*     *--*");
        Console.WriteLine(" ");
    }
}
```

ALTERNATIVE ALGORTIHM THAT GETS INPUTS

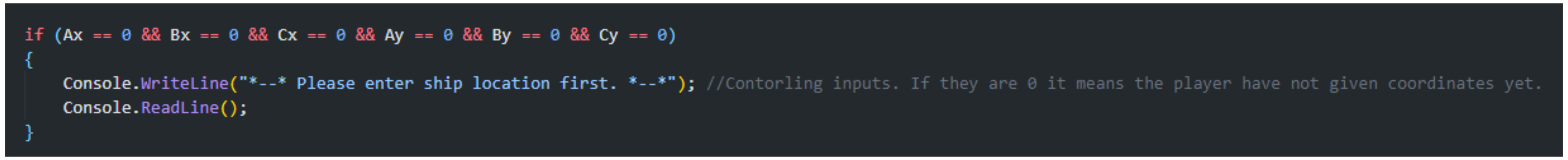


```
do
{
    Console.Write("Please enter Ax value of triangle (between 0 and 30): ");
    if (Int32.TryParse(Console.ReadLine(), out Ax) && Ax >= 0 && Ax <= 30)
    {
        break;
    }
    Console.WriteLine("Please enter a valid Ax value");
}
while (true);
```

ALGORITHM THAT CONTROLS IF THE SHIP IS TRIANGULAR

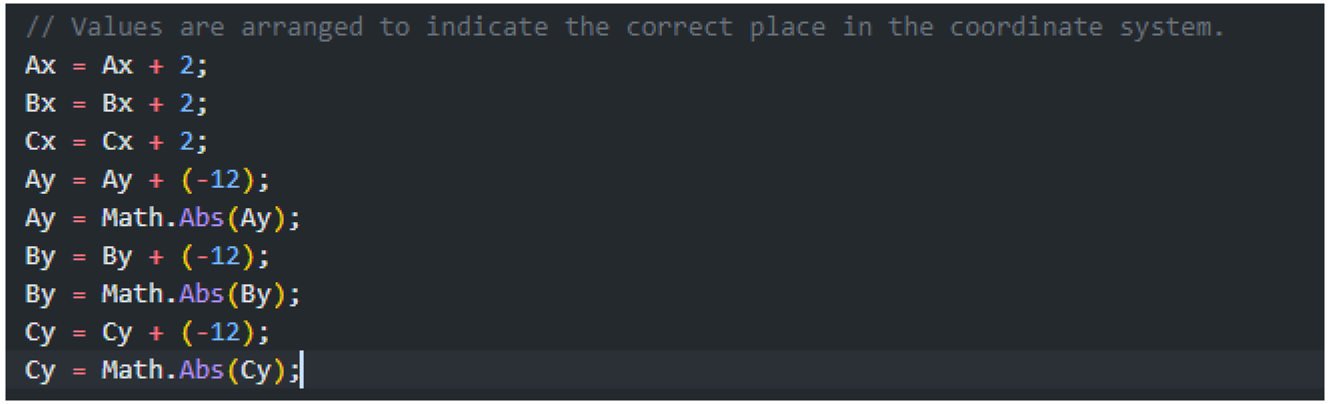

```
//Lenght of the edges
double c;
c = ((Ax - Bx) * (Ax - Bx)) + ((Ay - By) * (Ay - By));
c = Math.Sqrt(c);

double b;
b = ((Cx - Ax) * (Cx - Ax)) + ((Cy - Ay) * (Cy - Ay));
b = Math.Sqrt(b);

double a;
a = ((Bx - Cx) * (Bx - Cx) + ((By - Cy) * (By - Cy)));
a = Math.Sqrt(a);

// The ship is triangular or not section
if (Math.Round(a,2) + Math.Round(b,2) > Math.Round(c,2) && Math.Round(c,2) > Math.Abs(Math.Round(a,2) - Math.Round(b,2)))
{
    Console.ForegroundColor = ConsoleColor.DarkGreen;
    Console.WriteLine("*--* SUCCESSFUL *--*");
    Console.ForegroundColor = ConsoleColor.White;
    Console.WriteLine("*--* Please press enter to go to menu. *--*");
    Console.ReadLine();
    break;
}
else
{
    Console.ForegroundColor = ConsoleColor.DarkRed;
    Console.WriteLine("x--x The values you entered do not form triangles. x--x");
    Console.ForegroundColor = ConsoleColor.White;
}
```

RETURNS TO THE MENU IF THE PLAYER HAS NOT ENTERED THE COORDINATES YET



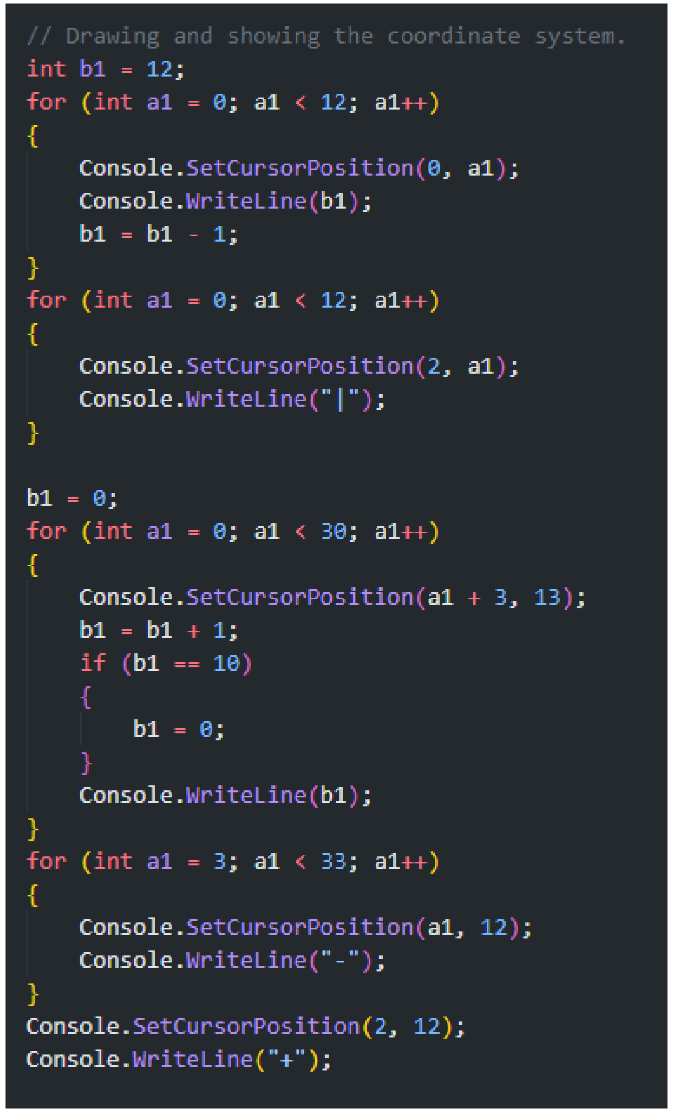
```
if (Ax == 0 && Bx == 0 && Cx == 0 && Ay == 0 && By == 0 && Cy == 0)
{
    Console.WriteLine("*--* Please enter ship location first. *--*"); //Contorling inputs. If they are 0 it means the player have not given coordinates yet.
    Console.ReadLine();
}
```

VALUES ARE ARRANGED TO INDICATE THE CORRECT PLACE IN THE COORDINATE SYSTEM



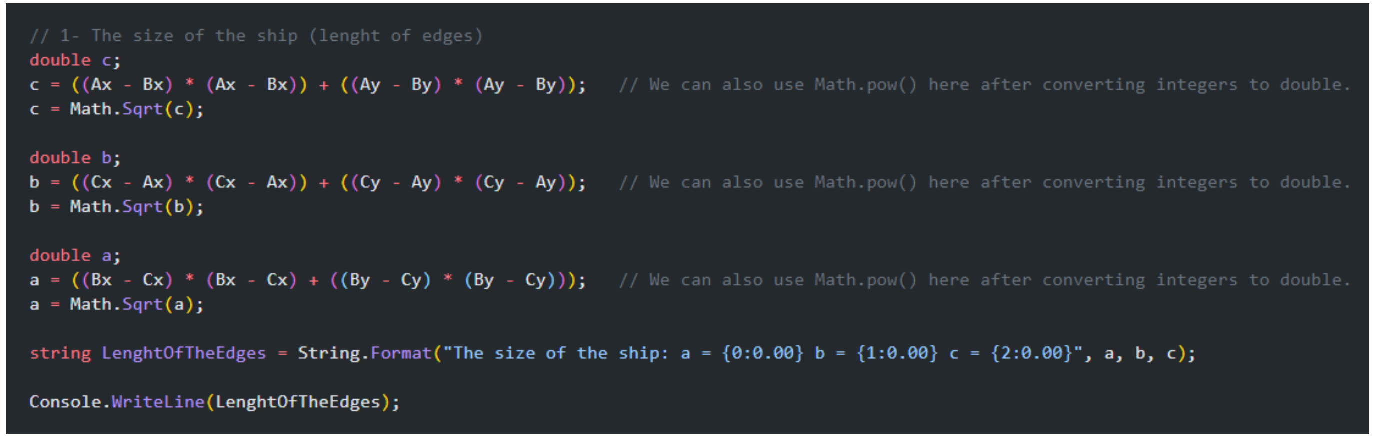
```
// Values are arranged to indicate the correct place in the coordinate system.
Ax = Ax + 2;
Bx = Bx + 2;
Cx = Cx + 2;
Ay = Ay + (-12);
Ay = Math.Abs(Ay);
By = By + (-12);
By = Math.Abs(By);
Cy = Cy + (-12);
Cy = Math.Abs(Cy);
```

ALGORITHM THAT DRAWS AND SHOWS COORDINATE SYSTEM



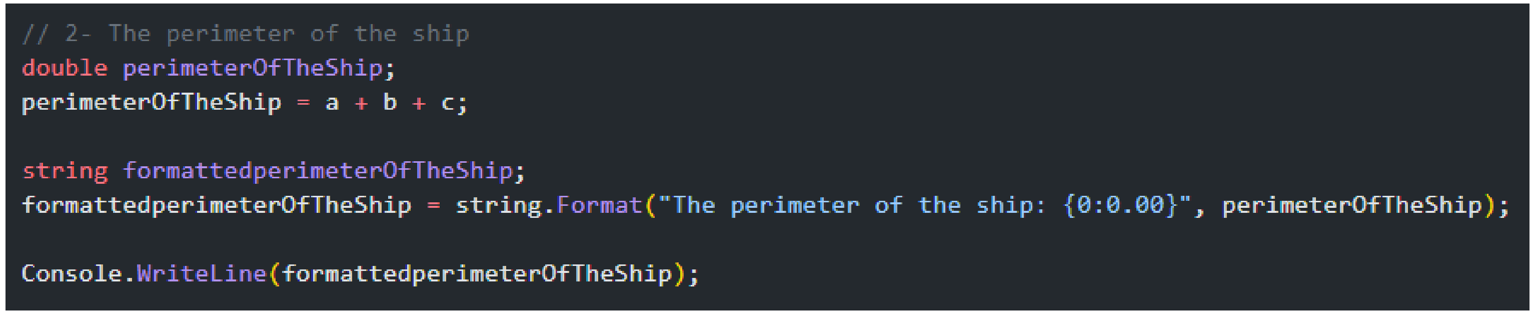
```
// Drawing and showing the coordinate system.
int b1 = 12;
for (int a1 = 0; a1 < 12; a1++)
{
    Console.SetCursorPosition(0, a1);
    Console.WriteLine(b1);
    b1 = b1 - 1;
}
for (int a1 = 0; a1 < 12; a1++)
{
    Console.SetCursorPosition(2, a1);
    Console.WriteLine("|");
}

b1 = 0;
for (int a1 = 0; a1 < 30; a1++)
{
    Console.SetCursorPosition(a1 + 3, 13);
    b1 = b1 + 1;
    if (b1 == 10)
    {
        b1 = 0;
    }
    Console.WriteLine(b1);
}
for (int a1 = 3; a1 < 33; a1++)
{
    Console.SetCursorPosition(a1, 12);
    Console.WriteLine("-");
}
Console.SetCursorPosition(2, 12);
Console.WriteLine("+");
```

THE SIZE OF THE SHIP ALGORITHM (LENGHT OF THE EDGES)



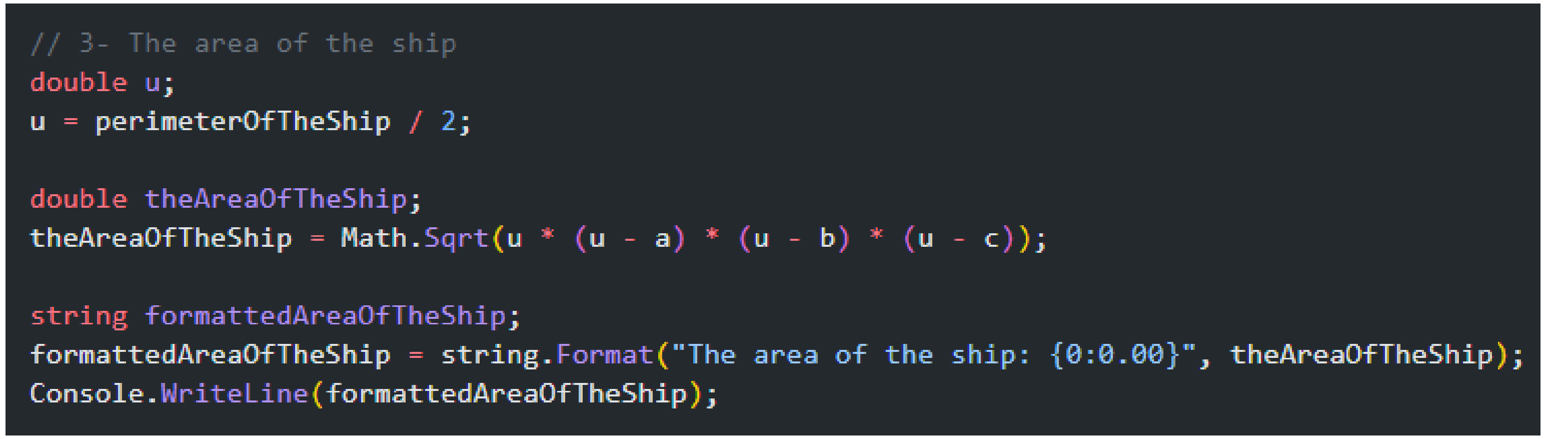
```
// 1- The size of the ship (lenght of edges)
double c;
c = ((Ax - Bx) * (Ax - Bx)) + ((Ay - By) * (Ay - By));   // We can also use Math.pow() here after converting integers to double.
c = Math.Sqrt(c);

double b;
b = ((Cx - Ax) * (Cx - Ax)) + ((Cy - Ay) * (Cy - Ay));   // We can also use Math.pow() here after converting integers to double.
b = Math.Sqrt(b);

double a;
a = ((Bx - Cx) * (Bx - Cx) + ((By - Cy) * (By - Cy)));   // We can also use Math.pow() here after converting integers to double.
a = Math.Sqrt(a);

string LenghtOfTheEdges = String.Format("The size of the ship: a = {0:0.00} b = {1:0.00} c = {2:0.00}", a, b, c);

Console.WriteLine(LenghtOfTheEdges);
```

THE PERIMETER OF THE SHIP ALGORTIHM



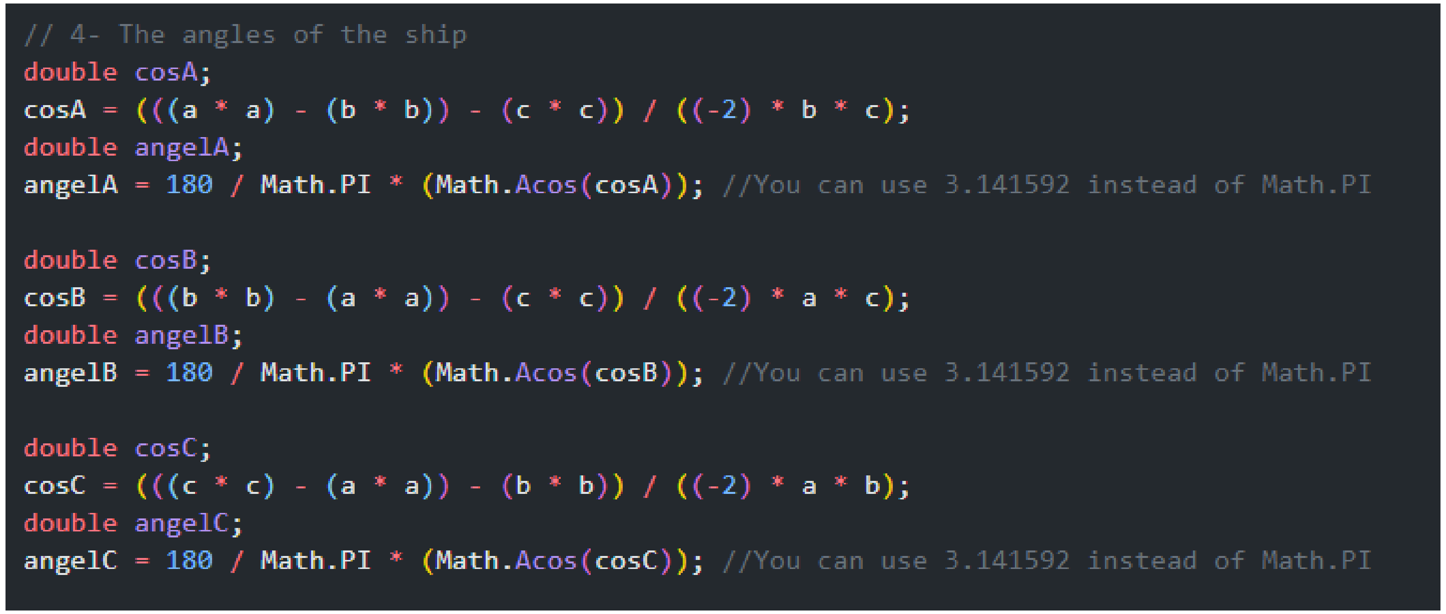
```
// 2- The perimeter of the ship
double perimeterOfTheShip;
perimeterOfTheShip = a + b + c;

string formattedperimeterOfTheShip;
formattedperimeterOfTheShip = string.Format("The perimeter of the ship: {0:0.00}", perimeterOfTheShip);

Console.WriteLine(formattedperimeterOfTheShip);
```

THE AREA OF THE SHIP ALGORITHM



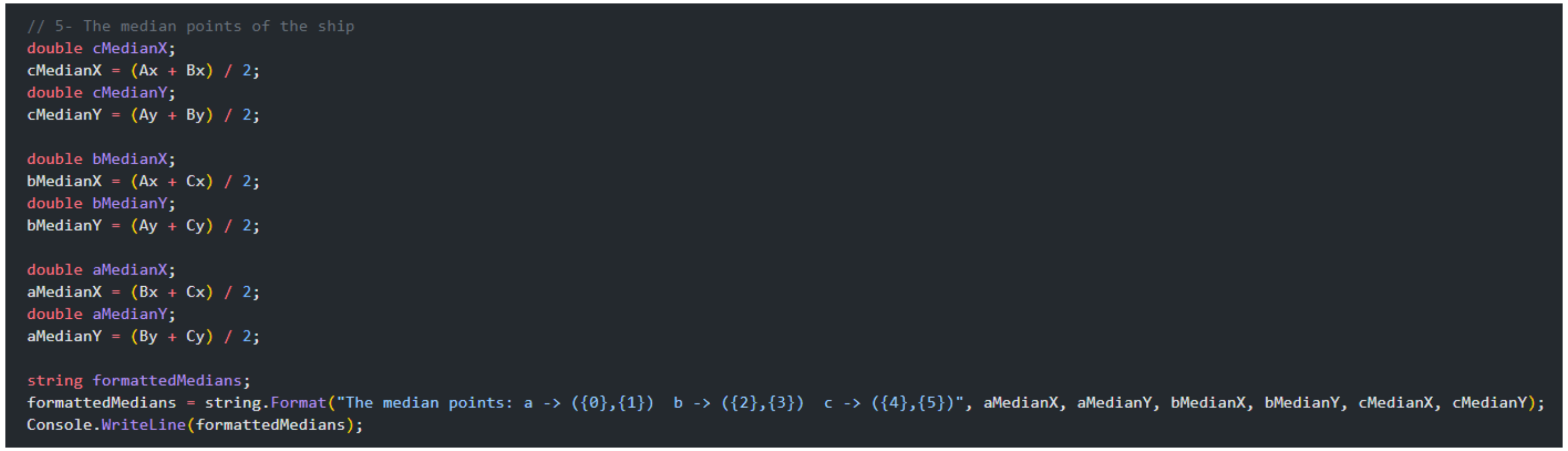
```
// 3- The area of the ship
double u;
u = perimeterOfTheShip / 2;

double theAreaOfTheShip;
theAreaOfTheShip = Math.Sqrt(u * (u - a) * (u - b) * (u - c));

string formattedAreaOfTheShip;
formattedAreaOfTheShip = string.Format("The area of the ship: {0:0.00}", theAreaOfTheShip);
Console.WriteLine(formattedAreaOfTheShip);
```

THE ANGLES OF THE SHIP ALGORITHM



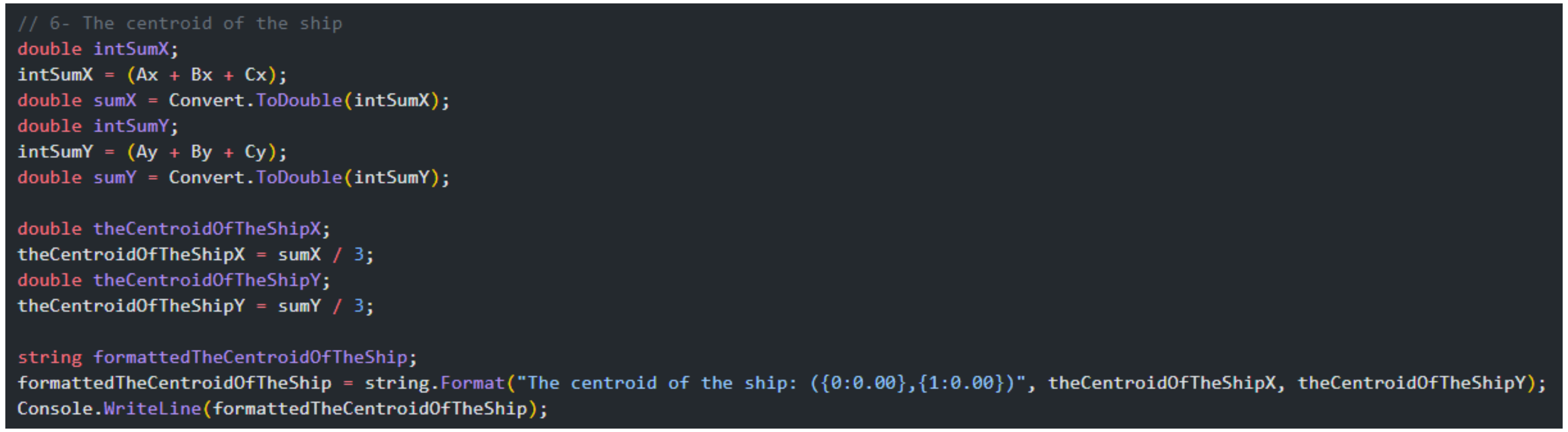
```
// 4- The angles of the ship
double cosA;
cosA = (((a * a) - (b * b)) - (c * c)) / ((-2) * b * c);
double angelA;
angelA = 180 / Math.PI * (Math.Acos(cosA)); //You can use 3.141592 instead of Math.PI

double cosB;
cosB = (((b * b) - (a * a)) - (c * c)) / ((-2) * a * c);
double angelB;
angelB = 180 / Math.PI * (Math.Acos(cosB)); //You can use 3.141592 instead of Math.PI

double cosC;
cosC = (((c * c) - (a * a)) - (b * b)) / ((-2) * a * b);
double angelC;
angelC = 180 / Math.PI * (Math.Acos(cosC)); //You can use 3.141592 instead of Math.PI
```

THE MEDIAN POINTS OF THE SHIP



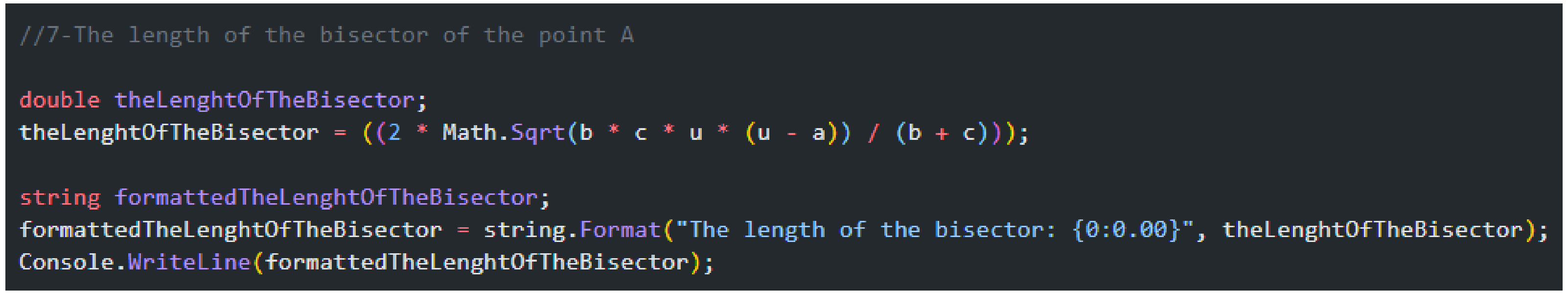
```
// 5- The median points of the ship
double cMedianX;
cMedianX = (Ax + Bx) / 2;
double cMedianY;
cMedianY = (Ay + By) / 2;

double bMedianX;
bMedianX = (Ax + Cx) / 2;
double bMedianY;
bMedianY = (Ay + Cy) / 2;

double aMedianX;
aMedianX = (Bx + Cx) / 2;
double aMedianY;
aMedianY = (By + Cy) / 2;

string formattedMedians;
formattedMedians = string.Format("The median points: a -> ({0},{1})  b -> ({2},{3})  c -> ({4},{5})", aMedianX, aMedianY, bMedianX, bMedianY, cMedianX, cMedianY);
Console.WriteLine(formattedMedians);
```

THE CENTROID OF THE SHIP ALGORITHM



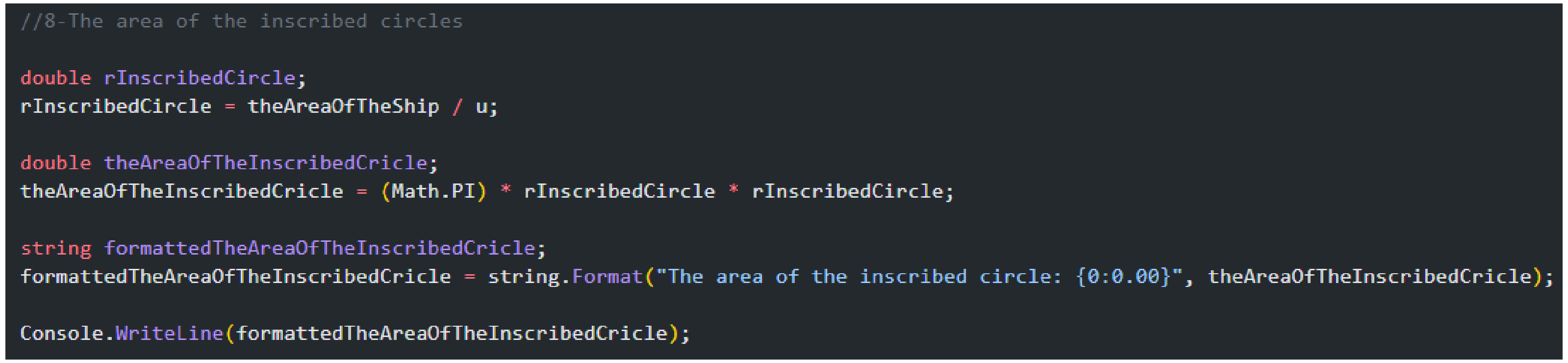
```
// 6- The centroid of the ship
double intSumX;
intSumX = (Ax + Bx + Cx);
double sumX = Convert.ToDouble(intSumX);
double intSumY;
intSumY = (Ay + By + Cy);
double sumY = Convert.ToDouble(intSumY);

double theCentroidOfTheShipX;
theCentroidOfTheShipX = sumX / 3;
double theCentroidOfTheShipY;
theCentroidOfTheShipY = sumY / 3;

string formattedTheCentroidOfTheShip;
formattedTheCentroidOfTheShip = string.Format("The centroid of the ship: ({0:0.00},{1:0.00})", theCentroidOfTheShipX, theCentroidOfTheShipY);
Console.WriteLine(formattedTheCentroidOfTheShip);
```

THE LENGHT OF THE BISECTOR OF THE POINT A ALGORITHM



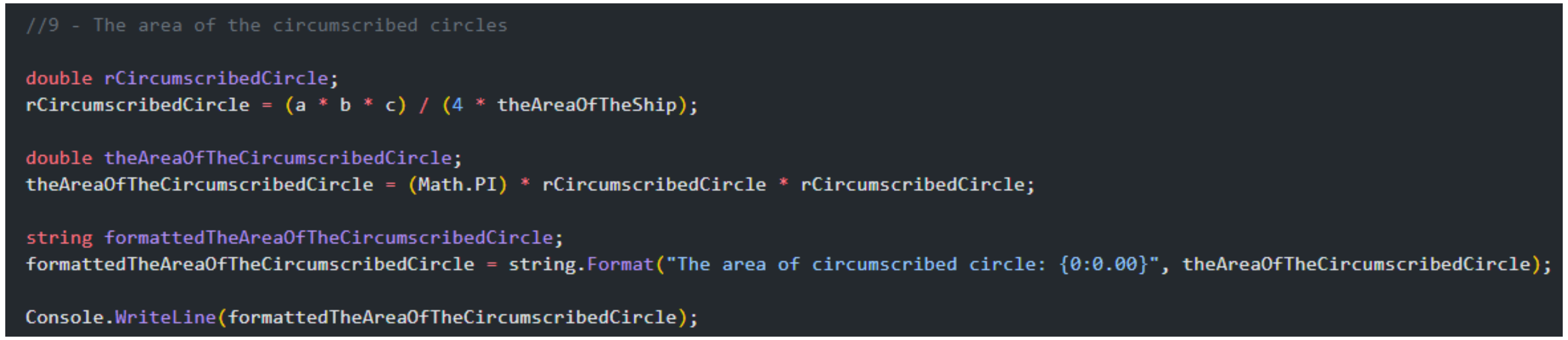
```
//7-The length of the bisector of the point A

double theLenghtOfTheBisector;
theLenghtOfTheBisector = ((2 * Math.Sqrt(b * c * u * (u - a)) / (b + c)));

string formattedTheLenghtOfTheBisector;
formattedTheLenghtOfTheBisector = string.Format("The length of the bisector: {0:0.00}", theLenghtOfTheBisector);
Console.WriteLine(formattedTheLenghtOfTheBisector);
```

THE AREA OF THE INSCRIBED CIRCLES ALGORITHM

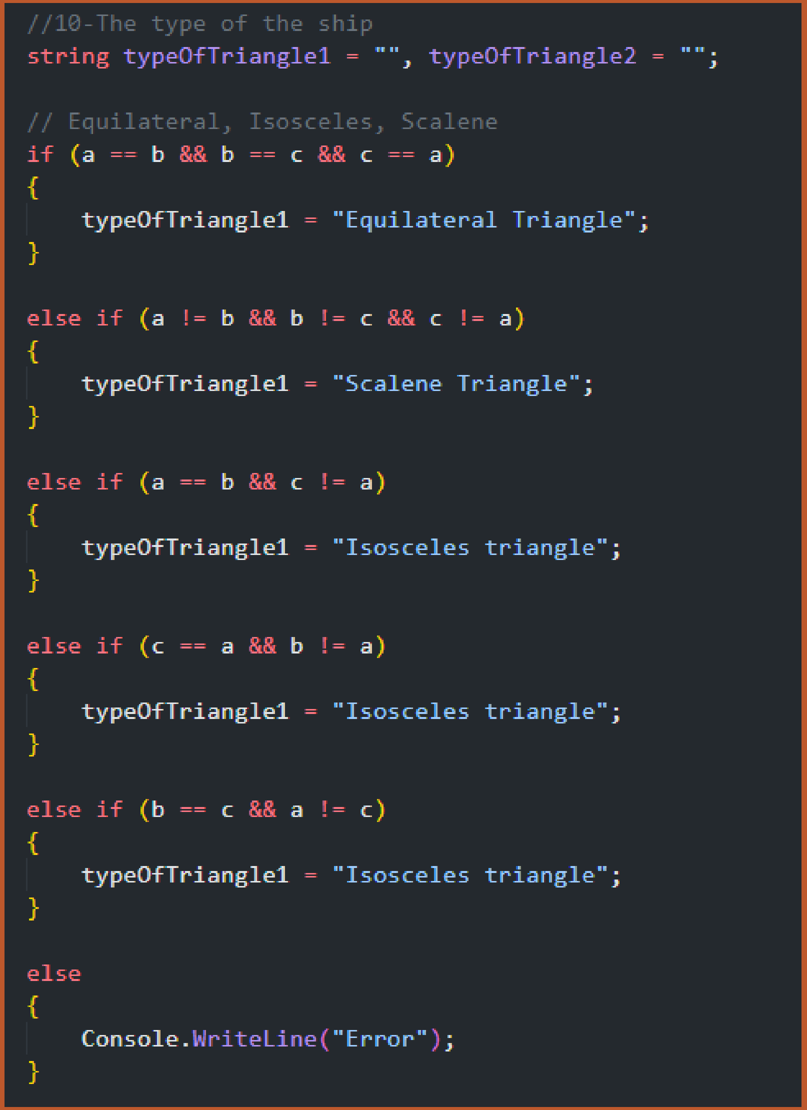


```
//8-The area of the inscribed circles

double rInscribedCircle;
rInscribedCircle = theAreaOfTheShip / u;

double theAreaOfTheInscribedCricle;
theAreaOfTheInscribedCricle = (Math.PI) * rInscribedCircle * rInscribedCircle;

string formattedTheAreaOfTheInscribedCricle;
formattedTheAreaOfTheInscribedCricle = string.Format("The area of the inscribed circle: {0:0.00}", theAreaOfTheInscribedCricle);

Console.WriteLine(formattedTheAreaOfTheInscribedCricle);
```

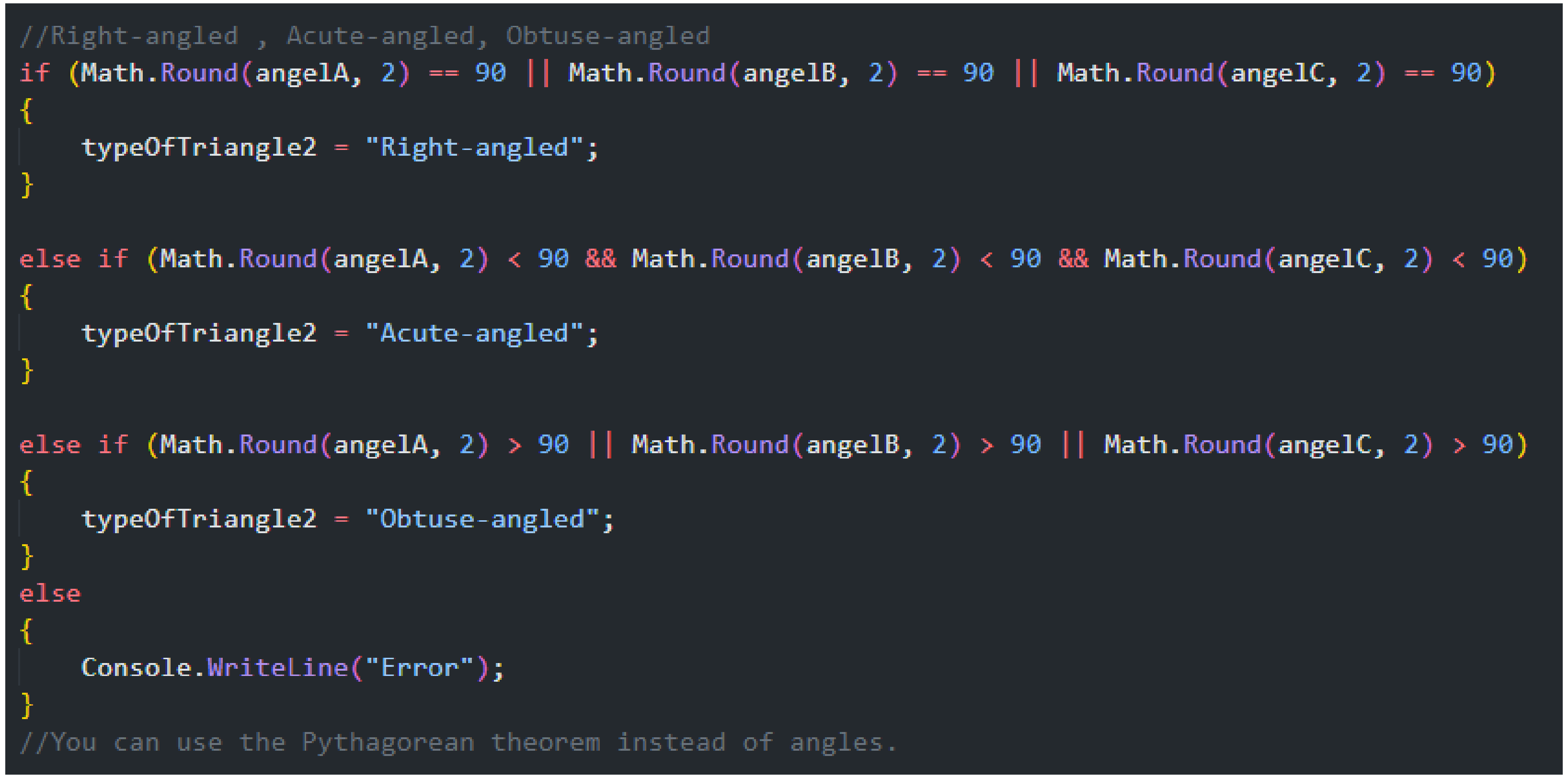
THE AREA OF THE CIRCUMSCRIBED CIRCLES ALGORITHM



```
//9 - The area of the circumscribed circles

double rCircumscribedCircle;
rCircumscribedCircle = (a * b * c) / (4 * theAreaOfTheShip);

double theAreaOfTheCircumscribedCircle;
theAreaOfTheCircumscribedCircle = (Math.PI) * rCircumscribedCircle * rCircumscribedCircle;

string formattedTheAreaOfTheCircumscribedCircle;
formattedTheAreaOfTheCircumscribedCircle = string.Format("The area of circumscribed circle: {0:0.00}", theAreaOfTheCircumscribedCircle);

Console.WriteLine(formattedTheAreaOfTheCircumscribedCircle);
```



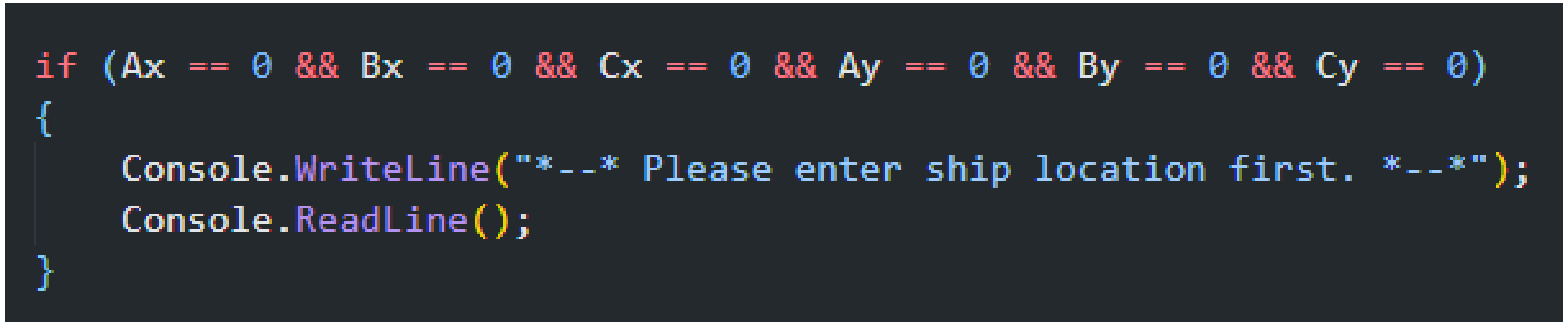
```
//10-The type of the ship
string typeOfTriangle1 = "", typeOfTriangle2 = "";

// Equilateral, Isosceles, Scalene
if (a == b && b == c && c == a)
{
    typeOfTriangle1 = "Equilateral Triangle";
}

else if (a != b && b != c && c != a)
{
    typeOfTriangle1 = "Scalene Triangle";
}

else if (a == b && c != a)
{
    typeOfTriangle1 = "Isosceles triangle";
}

else if (c == a && b != a)
{
    typeOfTriangle1 = "Isosceles triangle";
}

else if (b == c && a != c)
{
    typeOfTriangle1 = "Isosceles triangle";
}

else
{
    Console.WriteLine("Error");
}
```



```
//Right-angled , Acute-angled, Obtuse-angled
if (Math.Round(angelA, 2) == 90 || Math.Round(angelB, 2) == 90 || Math.Round(angelC, 2) == 90)
{
    typeOfTriangle2 = "Right-angled";
}

else if (Math.Round(angelA, 2) < 90 && Math.Round(angelB, 2) < 90 && Math.Round(angelC, 2) < 90)
{
    typeOfTriangle2 = "Acute-angled";
}

else if (Math.Round(angelA, 2) > 90 || Math.Round(angelB, 2) > 90 || Math.Round(angelC, 2) > 90)
{
    typeOfTriangle2 = "Obtuse-angled";
}
else
{
    Console.WriteLine("Error");
}
//You can use the Pythagorean theorem instead of angles.
```

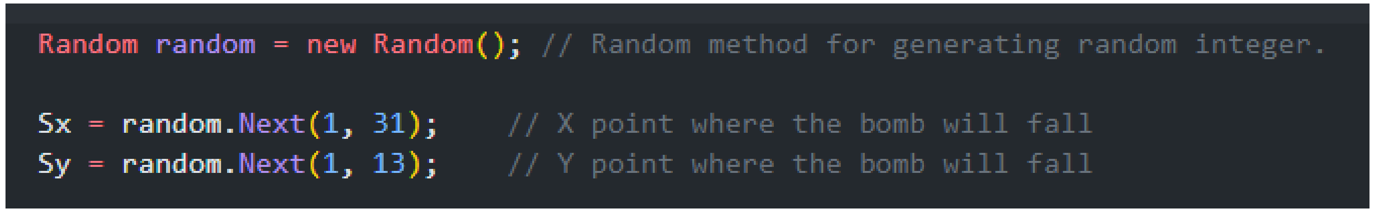
THE TYPE OF THE SHIP ALGORITHM

RETURNS TO THE MENU IF THE PLAYER HAS NOT ENTERED THE COORDINATES YET

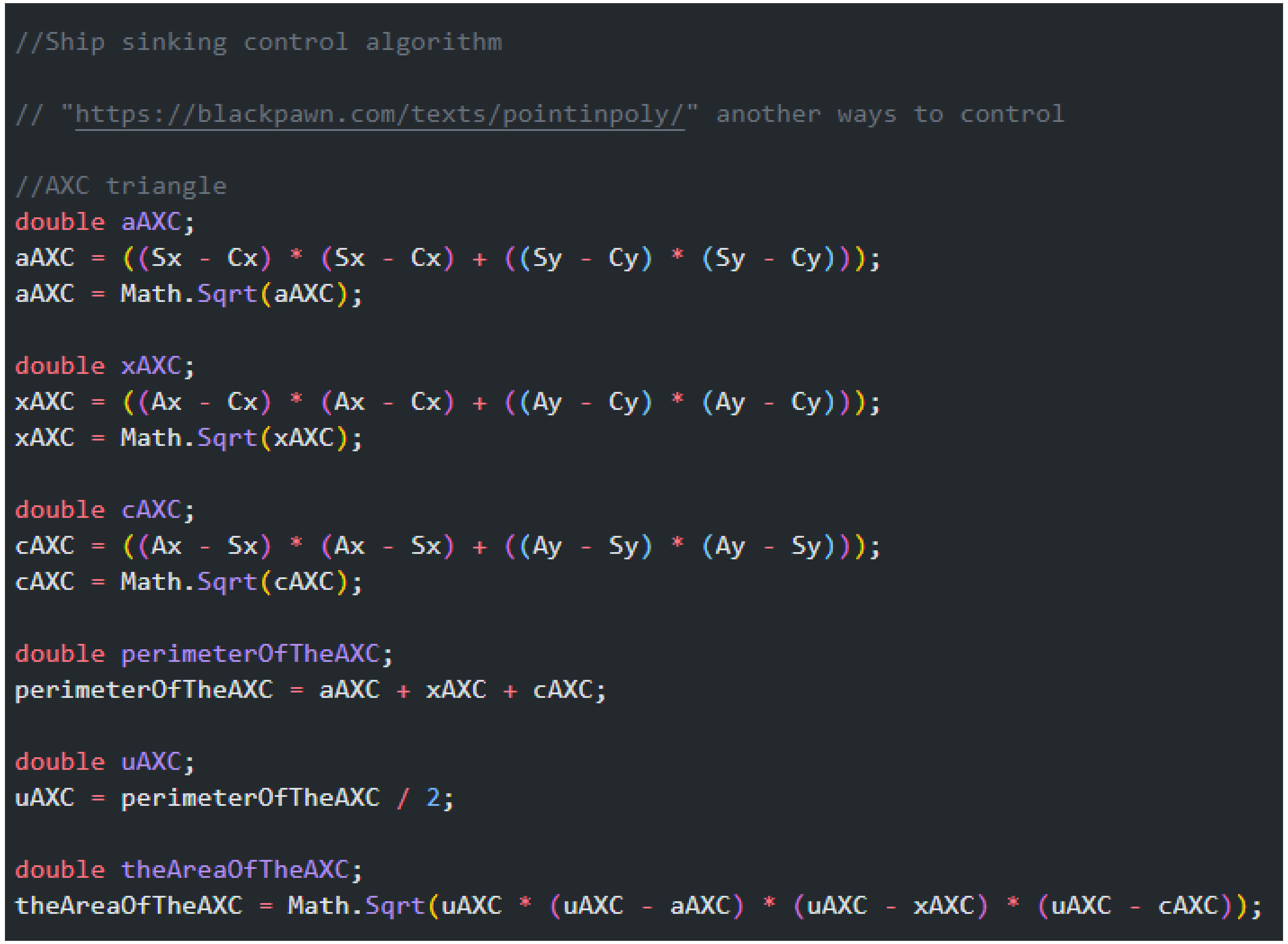


```
if (Ax == 0 && Bx == 0 && Cx == 0 && Ay == 0 && By == 0 && Cy == 0)
{
    Console.WriteLine("*--* Please enter ship location first. *--*");
    Console.ReadLine();
}
```

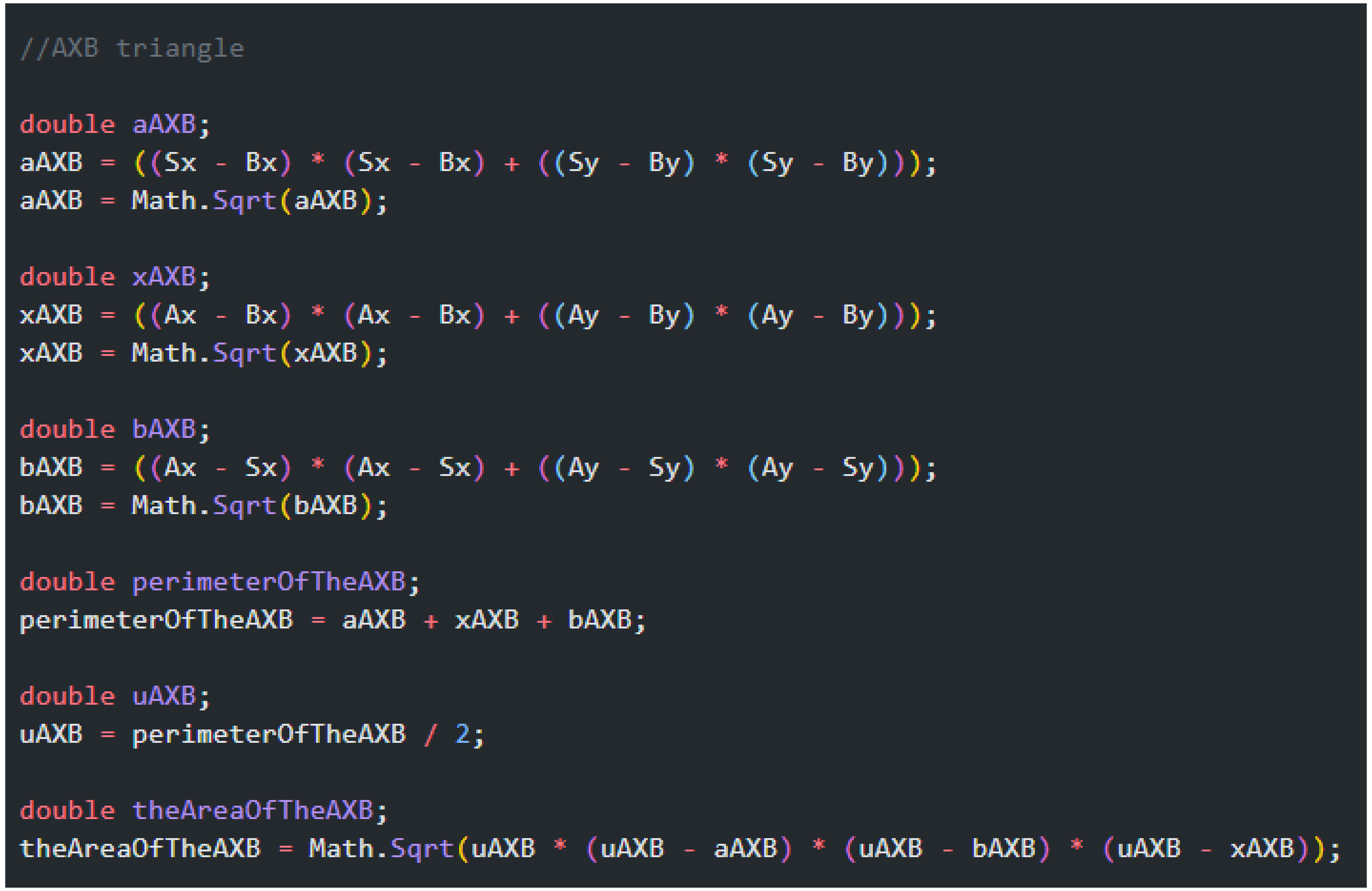
RANDOM ALGORITHM



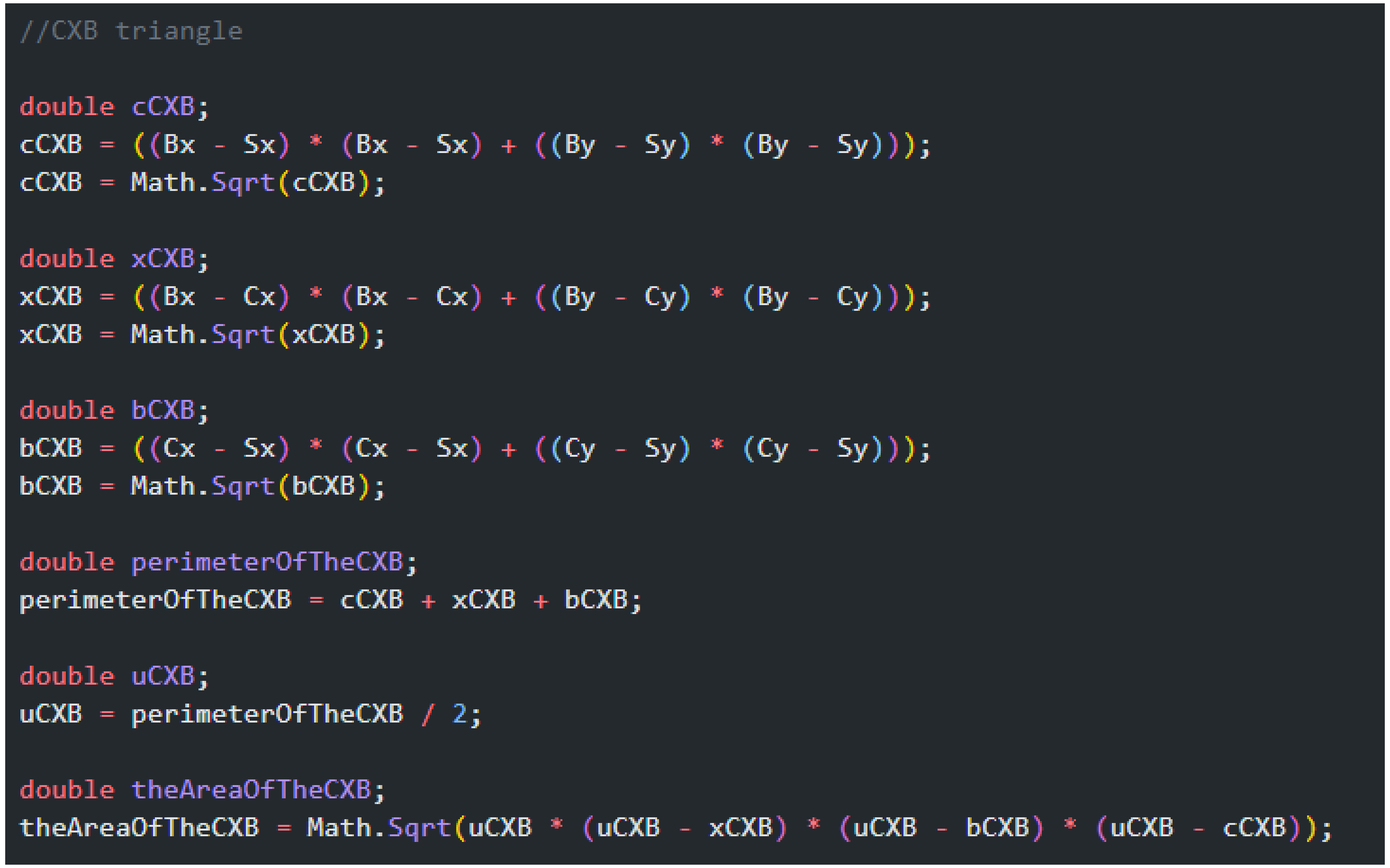
```
Random random = new Random(); // Random method for generating random integer.

Sx = random.Next(1, 31);    // X point where the bomb will fall
Sy = random.Next(1, 13);    // Y point where the bomb will fall
```



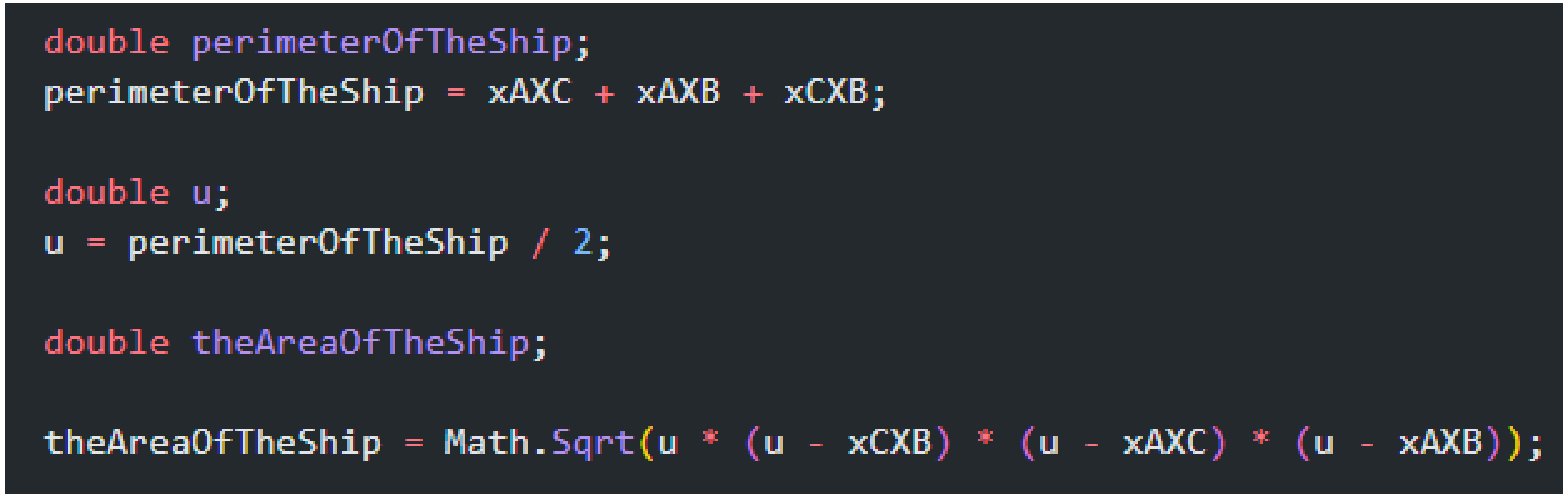
```
//Ship sinking control algorithm

// "https://blackpawn.com/texts/pointinpoly/" another ways to control

//AXC triangle
double aAXC;
aAXC = ((Sx - Cx) * (Sx - Cx) + ((Sy - Cy) * (Sy - Cy)));
aAXC = Math.Sqrt(aAXC);

double xAXC;
xAXC = ((Ax - Cx) * (Ax - Cx) + ((Ay - Cy) * (Ay - Cy)));
xAXC = Math.Sqrt(xAXC);

double cAXC;
cAXC = ((Ax - Sx) * (Ax - Sx) + ((Ay - Sy) * (Ay - Sy)));
cAXC = Math.Sqrt(cAXC);

double perimeterOfTheAXC;
perimeterOfTheAXC = aAXC + xAXC + cAXC;

double uAXC;
uAXC = perimeterOfTheAXC / 2;

double theAreaOfTheAXC;
theAreaOfTheAXC = Math.Sqrt(uAXC * (uAXC - aAXC) * (uAXC - xAXC) * (uAXC - cAXC));
```



```
//AXB triangle

double aAXB;
aAXB = ((Sx - Bx) * (Sx - Bx) + ((Sy - By) * (Sy - By)));
aAXB = Math.Sqrt(aAXB);

double xAXB;
xAXB = ((Ax - Bx) * (Ax - Bx) + ((Ay - By) * (Ay - By)));
xAXB = Math.Sqrt(xAXB);

double bAXB;
bAXB = ((Ax - Sx) * (Ax - Sx) + ((Ay - Sy) * (Ay - Sy)));
bAXB = Math.Sqrt(bAXB);

double perimeterOfTheAXB;
perimeterOfTheAXB = aAXB + xAXB + bAXB;

double uAXB;
uAXB = perimeterOfTheAXB / 2;

double theAreaOfTheAXB;
theAreaOfTheAXB = Math.Sqrt(uAXB * (uAXB - aAXB) * (uAXB - bAXB) * (uAXB - xAXB));
```



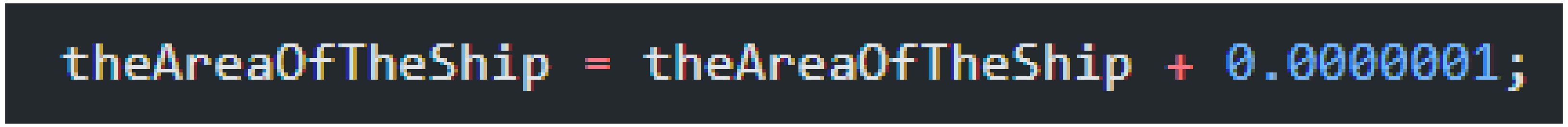
```
//CXB triangle

double cCXB;
cCXB = ((Bx - Sx) * (Bx - Sx) + ((By - Sy) * (By - Sy)));
cCXB = Math.Sqrt(cCXB);

double xCXB;
xCXB = ((Bx - Cx) * (Bx - Cx) + ((By - Cy) * (By - Cy)));
xCXB = Math.Sqrt(xCXB);

double bCXB;
bCXB = ((Cx - Sx) * (Cx - Sx) + ((Cy - Sy) * (Cy - Sy)));
bCXB = Math.Sqrt(bCXB);

double perimeterOfTheCXB;
perimeterOfTheCXB = cCXB + xCXB + bCXB;

double uCXB;
uCXB = perimeterOfTheCXB / 2;

double theAreaOfTheCXB;
theAreaOfTheCXB = Math.Sqrt(uCXB * (uCXB - xCXB) * (uCXB - bCXB) * (uCXB - cCXB));
```



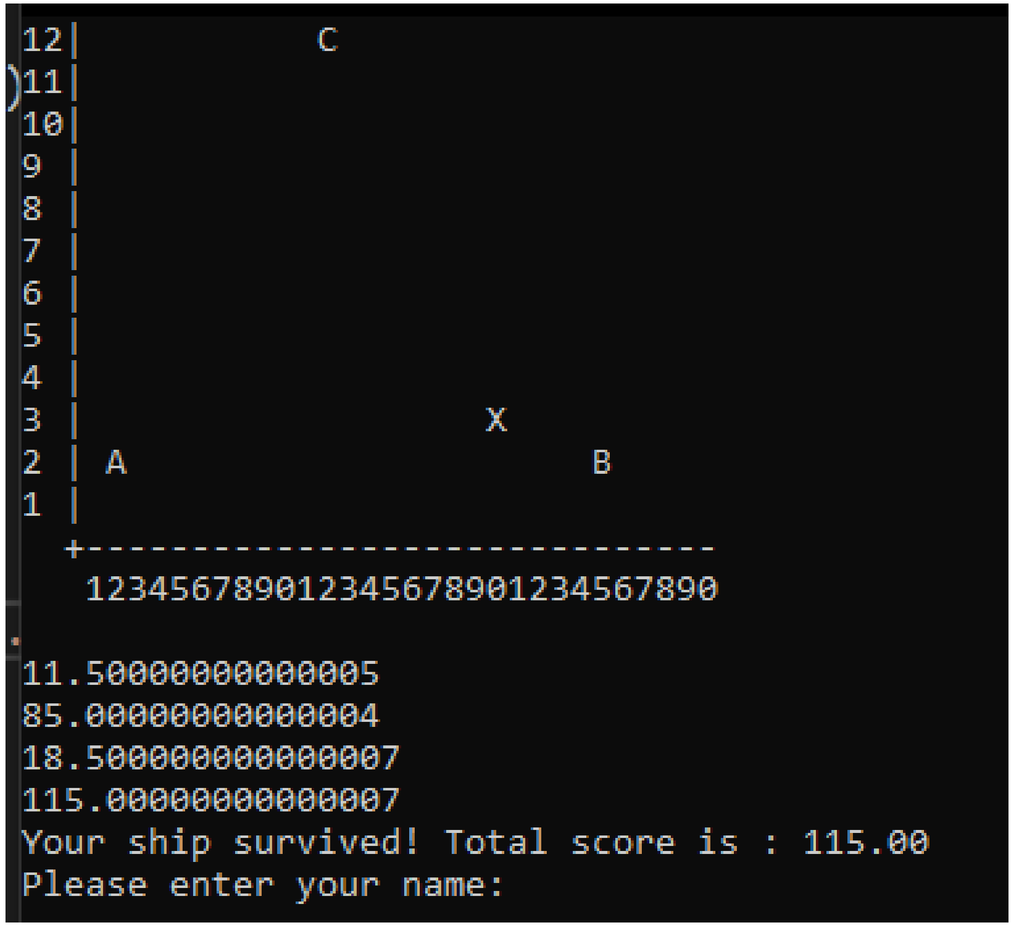
```
double perimeterOfTheShip;
perimeterOfTheShip = xAXC + xAXB + xCXB;

double u;
u = perimeterOfTheShip / 2;

double theAreaOfTheShip;

theAreaOfTheShip = Math.Sqrt(u * (u - xCXB) * (u - xAXC) * (u - xAXB));
```

ONE OF THE MOST IMPORTANT CODE IN THIS GAME

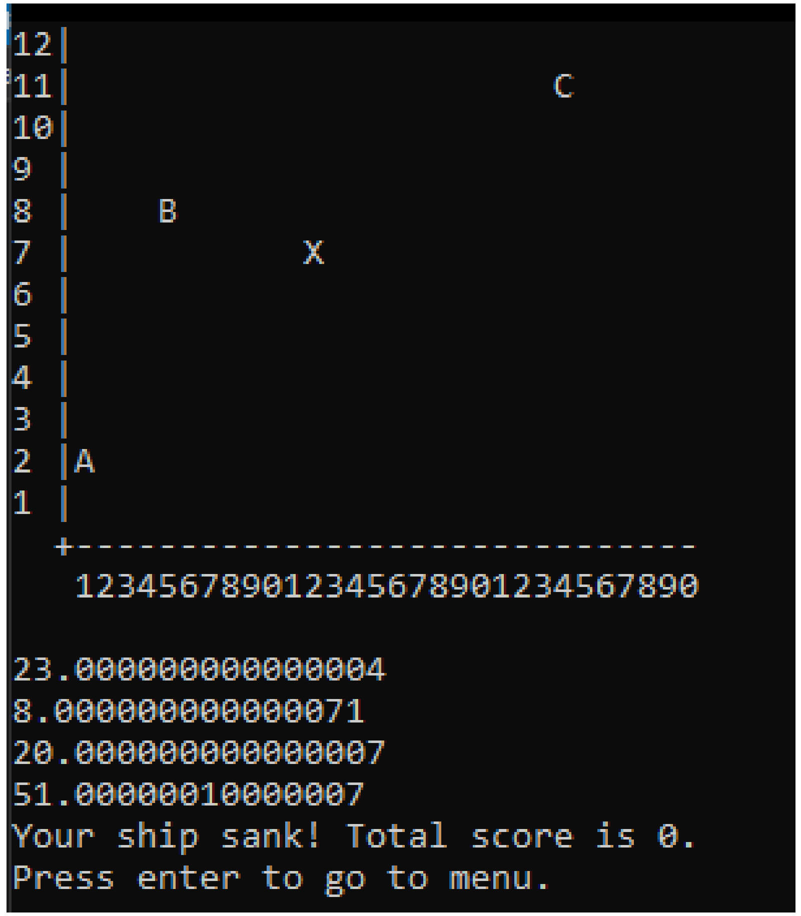
This code fixes double data types problems that occures after dot.



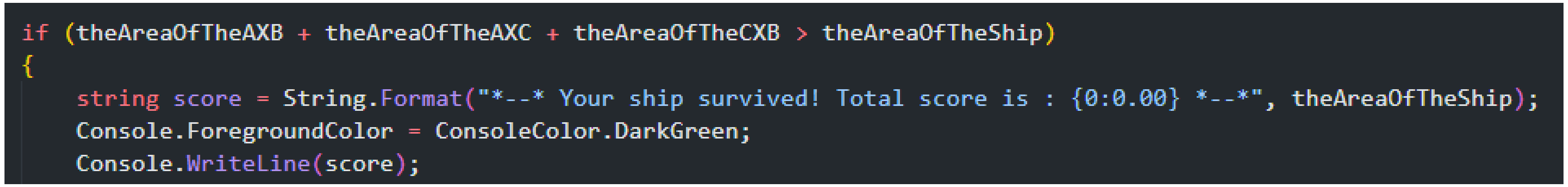
```
theAreaOfTheShip = theAreaOfTheShip + 0.0000001;
```

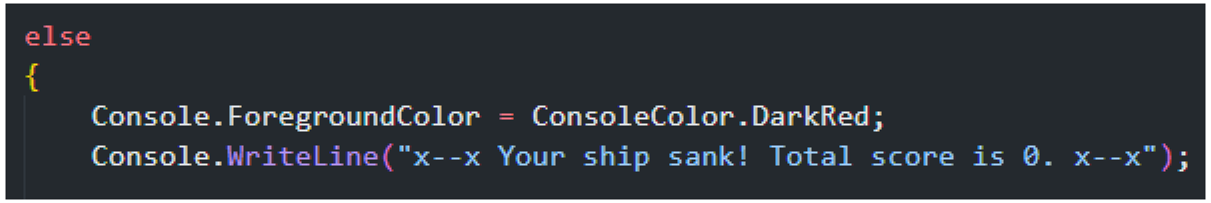


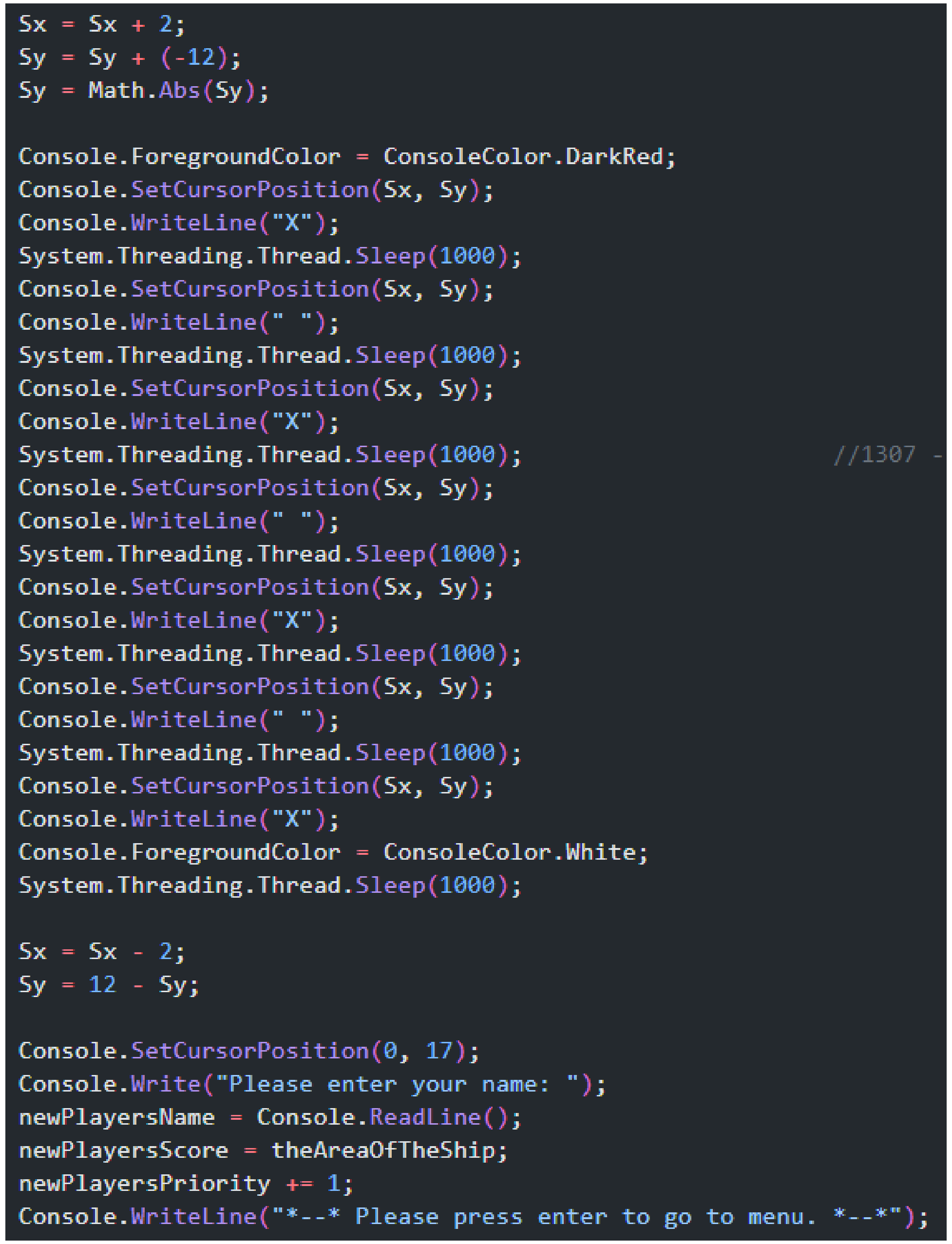
```
12|           C
11|
10|
9 |
8 |
7 |
6 |
5 |
4 |
3 |                   X
2 | A                      B
1 |
  +------------------------------
   123456789012345678901234567890

11.50000000000005
85.00000000000004
18.500000000000007
115.00000000000007
Your ship survived! Total score is : 115.00
Please enter your name:
```



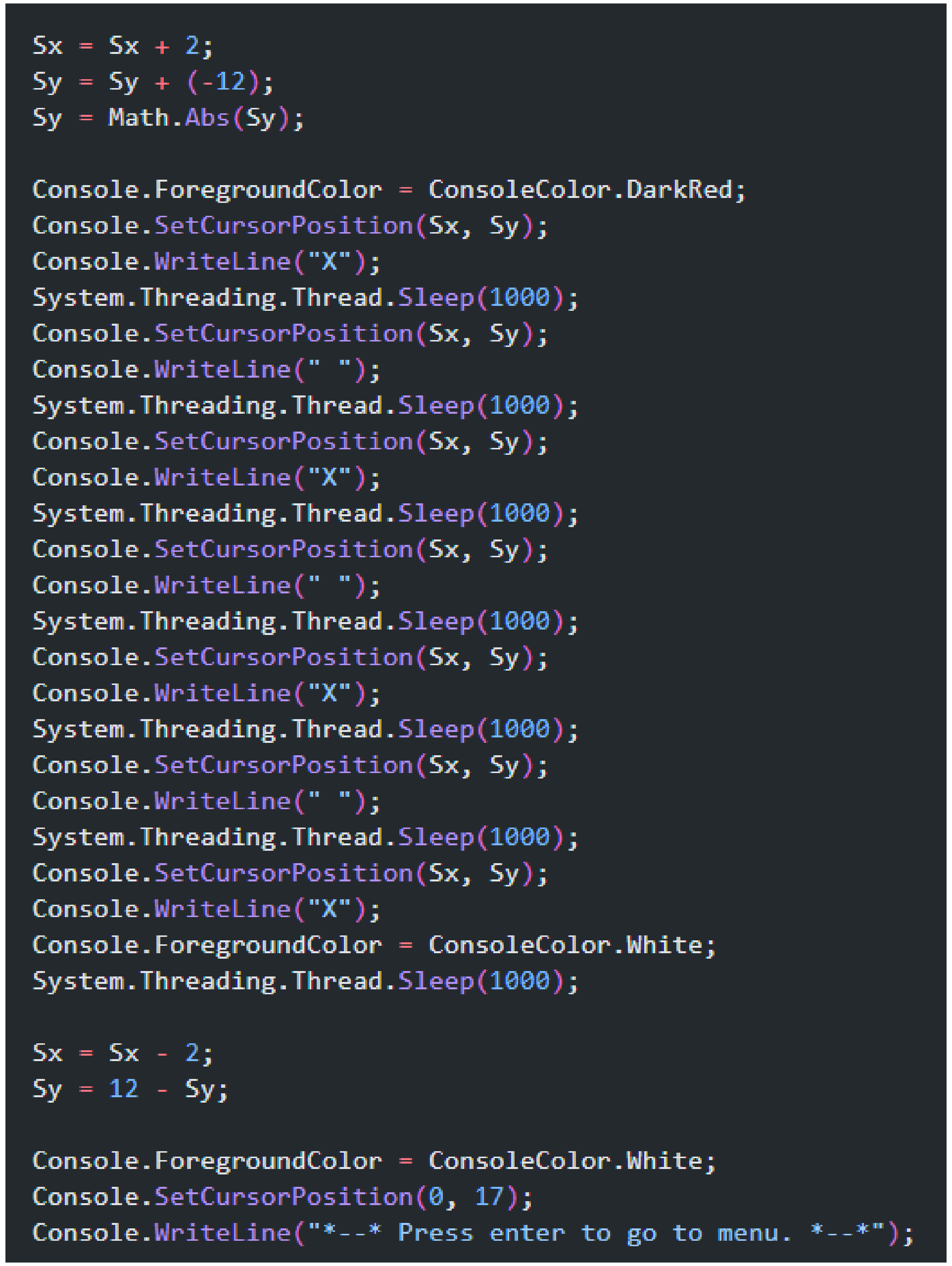
```
12|
11|                       C
10|
9 |
8 |    B
7 |           X
6 |
5 |
4 |
3 |
2 |A
1 |
  +------------------------------
   123456789012345678901234567890

23.000000000000004
8.000000000000071
20.000000000000007
51.00000010000007
Your ship sank! Total score is 0.
Press enter to go to menu.
```


```
if (theAreaOfTheAXB + theAreaOfTheAXC + theAreaOfTheCXB > theAreaOfTheShip)
{
    string score = String.Format("*--* Your ship survived! Total score is : {0:0.00} *--*", theAreaOfTheShip);
    Console.ForegroundColor = ConsoleColor.DarkGreen;
    Console.WriteLine(score);
```

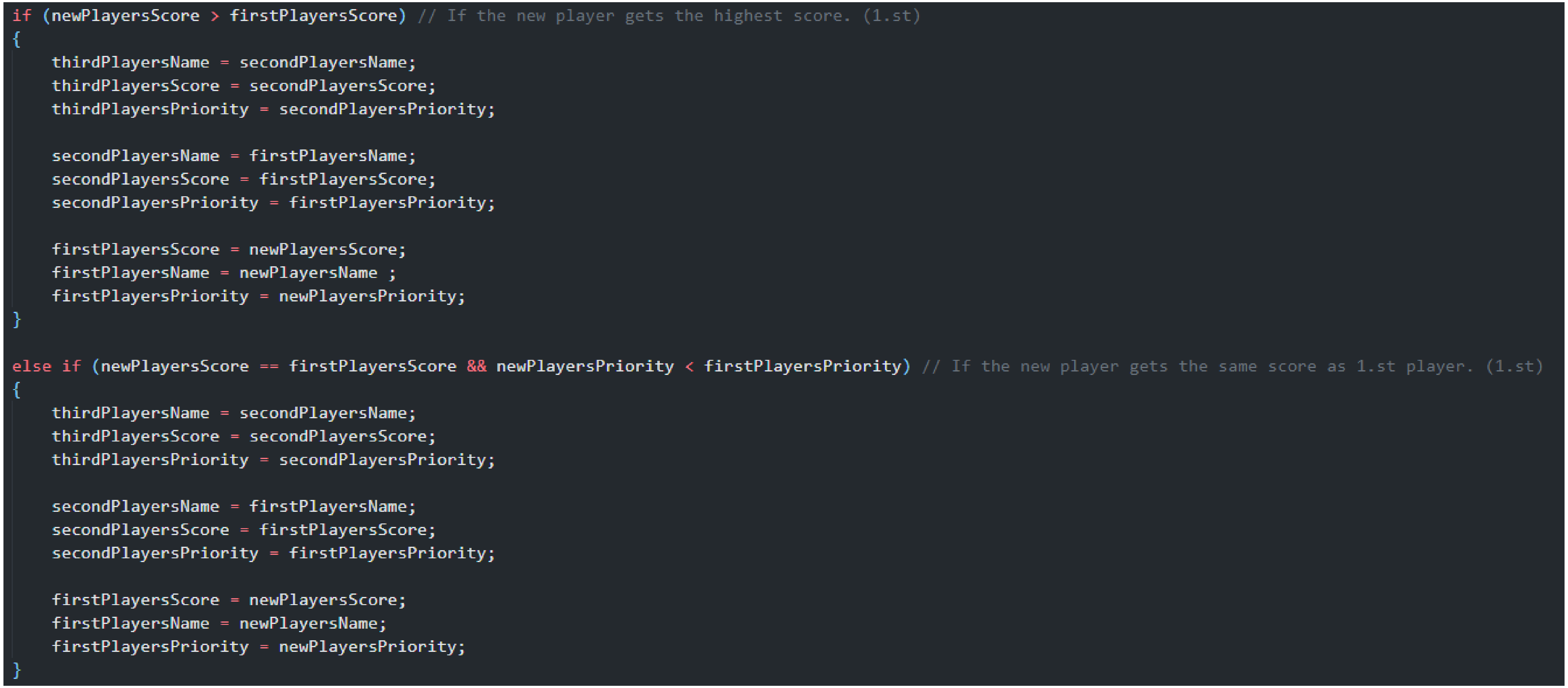

```
else
{
    Console.ForegroundColor = ConsoleColor.DarkRed;
    Console.WriteLine("x--x Your ship sank! Total score is 0. x--x");
```



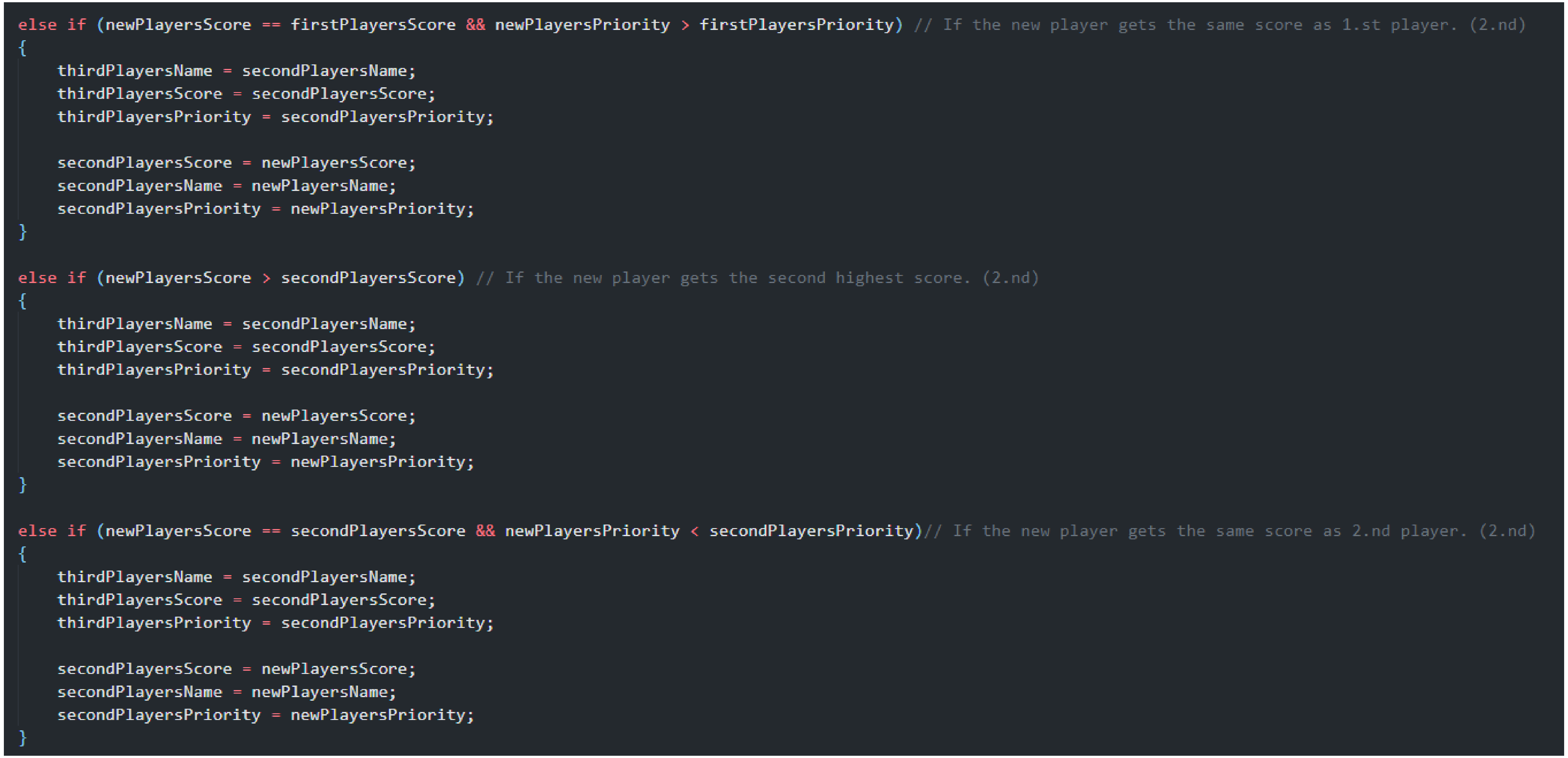
```
Sx = Sx + 2;
Sy = Sy + (-12);
Sy = Math.Abs(Sy);

Console.ForegroundColor = ConsoleColor.DarkRed;
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine("X");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine(" ");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine("X");
System.Threading.Thread.Sleep(1000);                        //1307 -
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine(" ");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine("X");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine(" ");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine("X");
Console.ForegroundColor = ConsoleColor.White;
System.Threading.Thread.Sleep(1000);

Sx = Sx - 2;
Sy = 12 - Sy;

Console.SetCursorPosition(0, 17);
Console.Write("Please enter your name: ");
newPlayersName = Console.ReadLine();
newPlayersScore = theAreaOfTheShip;
newPlayersPriority += 1;
Console.WriteLine("*--* Please press enter to go to menu. *--*");
```



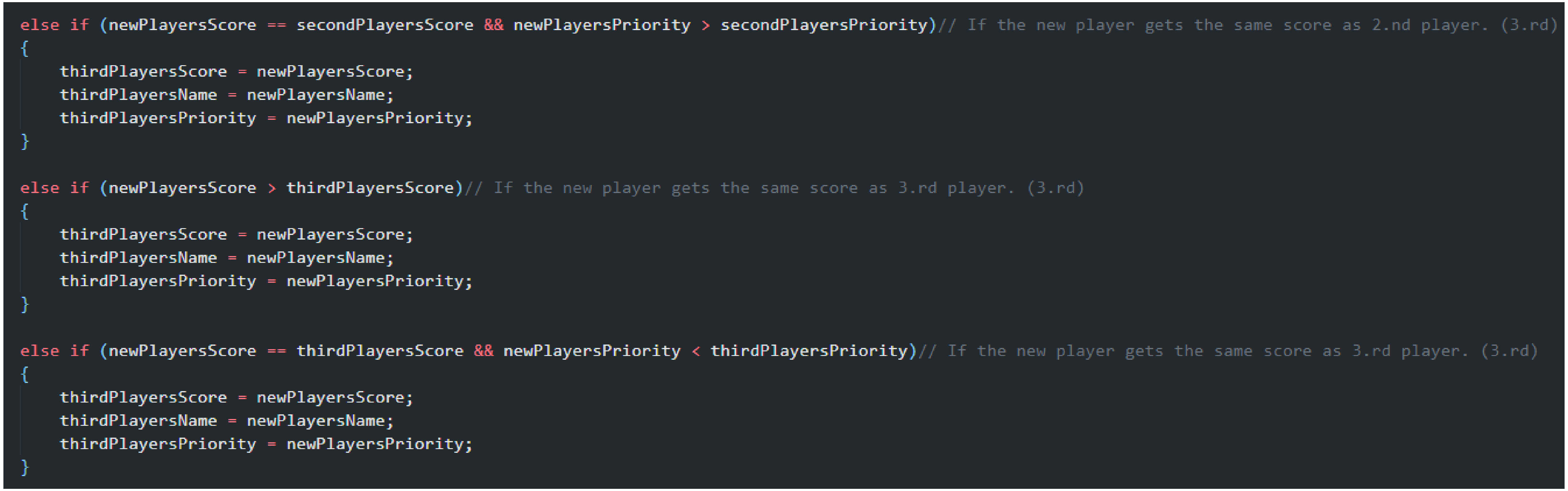
```
Sx = Sx + 2;
Sy = Sy + (-12);
Sy = Math.Abs(Sy);

Console.ForegroundColor = ConsoleColor.DarkRed;
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine("X");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine(" ");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine("X");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine(" ");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine("X");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine(" ");
System.Threading.Thread.Sleep(1000);
Console.SetCursorPosition(Sx, Sy);
Console.WriteLine("X");
Console.ForegroundColor = ConsoleColor.White;
System.Threading.Thread.Sleep(1000);

Sx = Sx - 2;
Sy = 12 - Sy;

Console.ForegroundColor = ConsoleColor.White;
Console.SetCursorPosition(0, 17);
Console.WriteLine("*--* Press enter to go to menu. *--*");
```



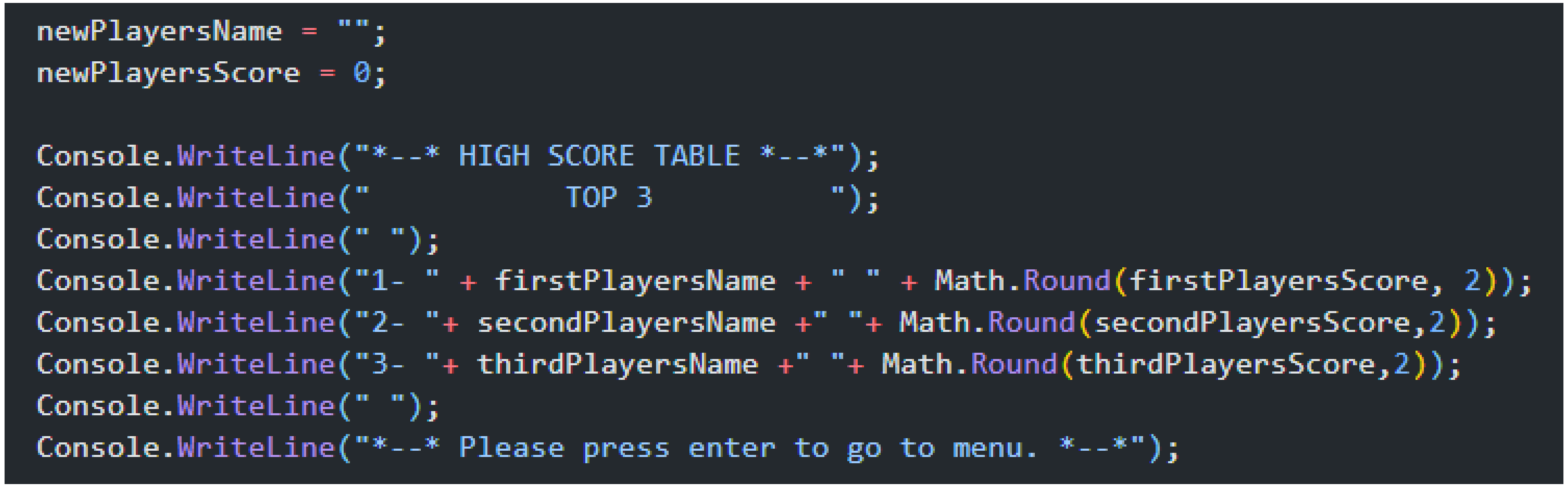
```
if (newPlayersScore > firstPlayersScore) // If the new player gets the highest score. (1.st)
{
    thirdPlayersName = secondPlayersName;
    thirdPlayersScore = secondPlayersScore;
    thirdPlayersPriority = secondPlayersPriority;

    secondPlayersName = firstPlayersName;
    secondPlayersScore = firstPlayersScore;
    secondPlayersPriority = firstPlayersPriority;

    firstPlayersScore = newPlayersScore;
    firstPlayersName = newPlayersName ;
    firstPlayersPriority = newPlayersPriority;
}

else if (newPlayersScore == firstPlayersScore && newPlayersPriority < firstPlayersPriority) // If the new player gets the same score as 1.st player. (1.st)
{
    thirdPlayersName = secondPlayersName;
    thirdPlayersScore = secondPlayersScore;
    thirdPlayersPriority = secondPlayersPriority;

    secondPlayersName = firstPlayersName;
    secondPlayersScore = firstPlayersScore;
    secondPlayersPriority = firstPlayersPriority;

    firstPlayersScore = newPlayersScore;
    firstPlayersName = newPlayersName;
    firstPlayersPriority = newPlayersPriority;
}
```



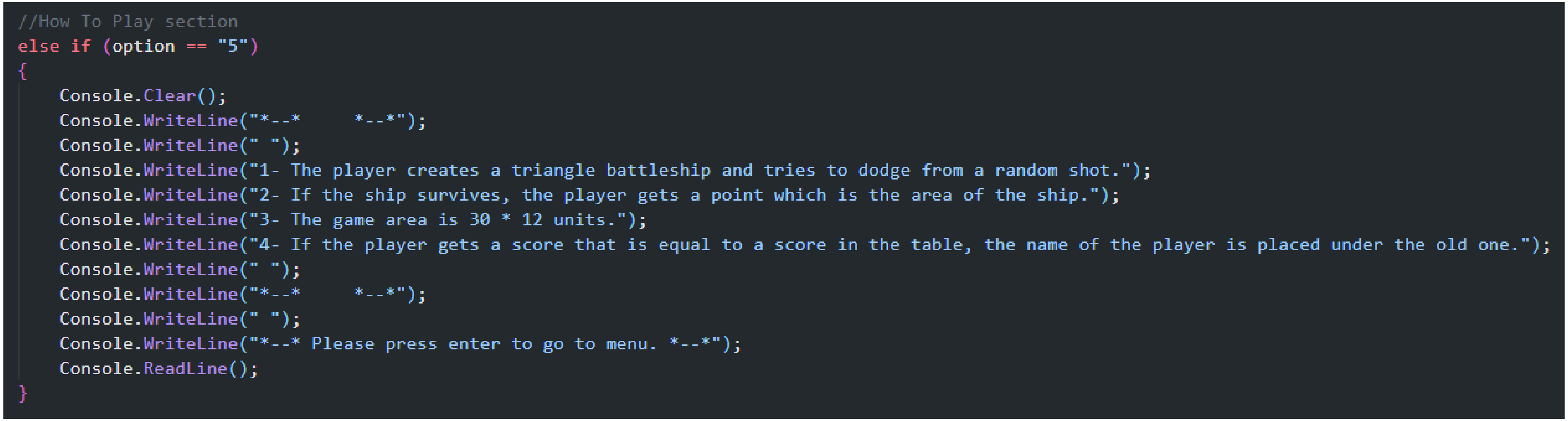
```
else if (newPlayersScore == firstPlayersScore && newPlayersPriority > firstPlayersPriority) // If the new player gets the same score as 1.st player. (2.nd)
{
    thirdPlayersName = secondPlayersName;
    thirdPlayersScore = secondPlayersScore;
    thirdPlayersPriority = secondPlayersPriority;

    secondPlayersScore = newPlayersScore;
    secondPlayersName = newPlayersName;
    secondPlayersPriority = newPlayersPriority;
}

else if (newPlayersScore > secondPlayersScore) // If the new player gets the second highest score. (2.nd)
{
    thirdPlayersName = secondPlayersName;
    thirdPlayersScore = secondPlayersScore;
    thirdPlayersPriority = secondPlayersPriority;

    secondPlayersScore = newPlayersScore;
    secondPlayersName = newPlayersName;
    secondPlayersPriority = newPlayersPriority;
}

else if (newPlayersScore == secondPlayersScore && newPlayersPriority < secondPlayersPriority)// If the new player gets the same score as 2.nd player. (2.nd)
{
    thirdPlayersName = secondPlayersName;
    thirdPlayersScore = secondPlayersScore;
    thirdPlayersPriority = secondPlayersPriority;

    secondPlayersScore = newPlayersScore;
    secondPlayersName = newPlayersName;
    secondPlayersPriority = newPlayersPriority;
}
```



```
else if (newPlayersScore == secondPlayersScore && newPlayersPriority > secondPlayersPriority)// If the new player gets the same score as 2.nd player. (3.rd)
{
    thirdPlayersScore = newPlayersScore;
    thirdPlayersName = newPlayersName;
    thirdPlayersPriority = newPlayersPriority;
}

else if (newPlayersScore > thirdPlayersScore)// If the new player gets the same score as 3.rd player. (3.rd)
{
    thirdPlayersScore = newPlayersScore;
    thirdPlayersName = newPlayersName;
    thirdPlayersPriority = newPlayersPriority;
}

else if (newPlayersScore == thirdPlayersScore && newPlayersPriority < thirdPlayersPriority)// If the new player gets the same score as 3.rd player. (3.rd)
{
    thirdPlayersScore = newPlayersScore;
    thirdPlayersName = newPlayersName;
    thirdPlayersPriority = newPlayersPriority;
}
```

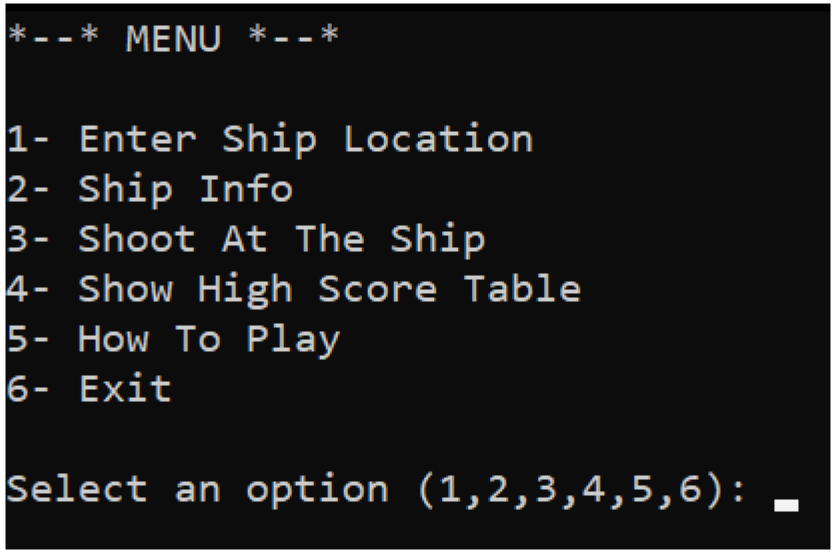


```
newPlayersName = "";
newPlayersScore = 0;

Console.WriteLine("*--* HIGH SCORE TABLE *--*");
Console.WriteLine("           TOP 3          ");
Console.WriteLine(" ");
Console.WriteLine("1- " + firstPlayersName + " " + Math.Round(firstPlayersScore, 2));
Console.WriteLine("2- "+ secondPlayersName +" "+ Math.Round(secondPlayersScore,2));
Console.WriteLine("3- "+ thirdPlayersName +" "+ Math.Round(thirdPlayersScore,2));
Console.WriteLine(" ");
Console.WriteLine("*--* Please press enter to go to menu. *--*");
```

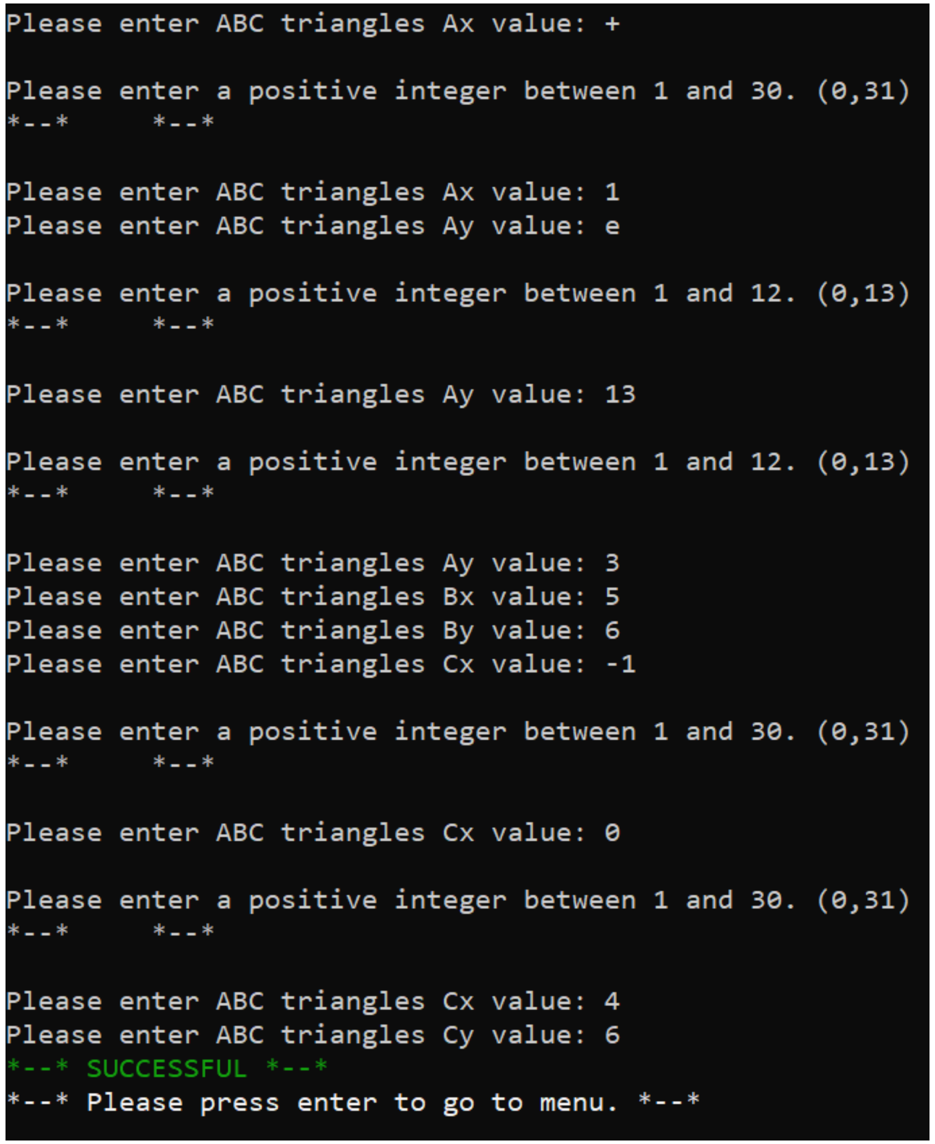


```
//How To Play section
else if (option == "5")
{
    Console.Clear();
    Console.WriteLine("*--*     *--*");
    Console.WriteLine(" ");
    Console.WriteLine("1- The player creates a triangle battleship and tries to dodge from a random shot.");
    Console.WriteLine("2- If the ship survives, the player gets a point which is the area of the ship.");
    Console.WriteLine("3- The game area is 30 * 12 units.");
    Console.WriteLine("4- If the player gets a score that is equal to a score in the table, the name of the player is placed under the old one.");
    Console.WriteLine(" ");
    Console.WriteLine("*--*     *--*");
    Console.WriteLine(" ");
    Console.WriteLine("*--* Please press enter to go to menu. *--*");
    Console.ReadLine();
}
```

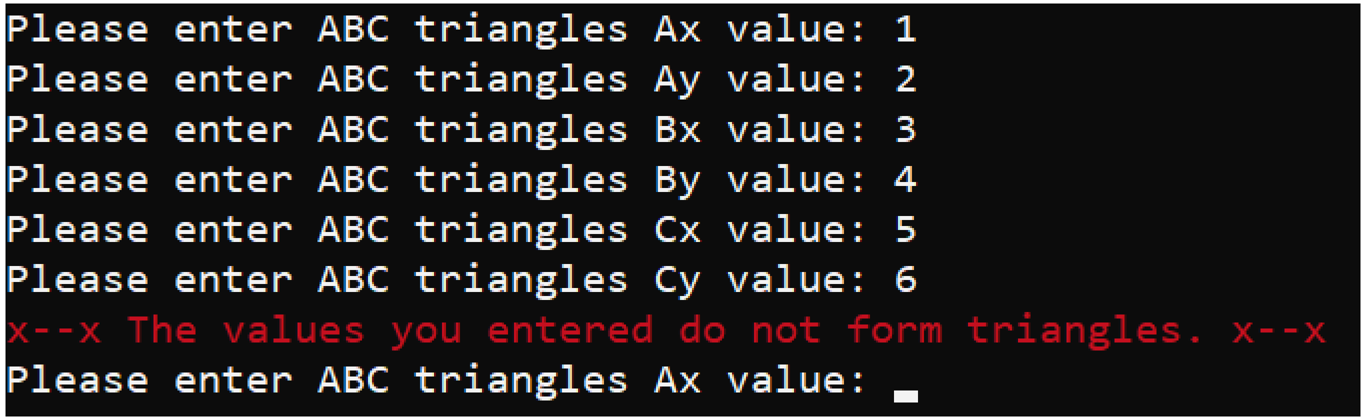
SCREENSHOTS



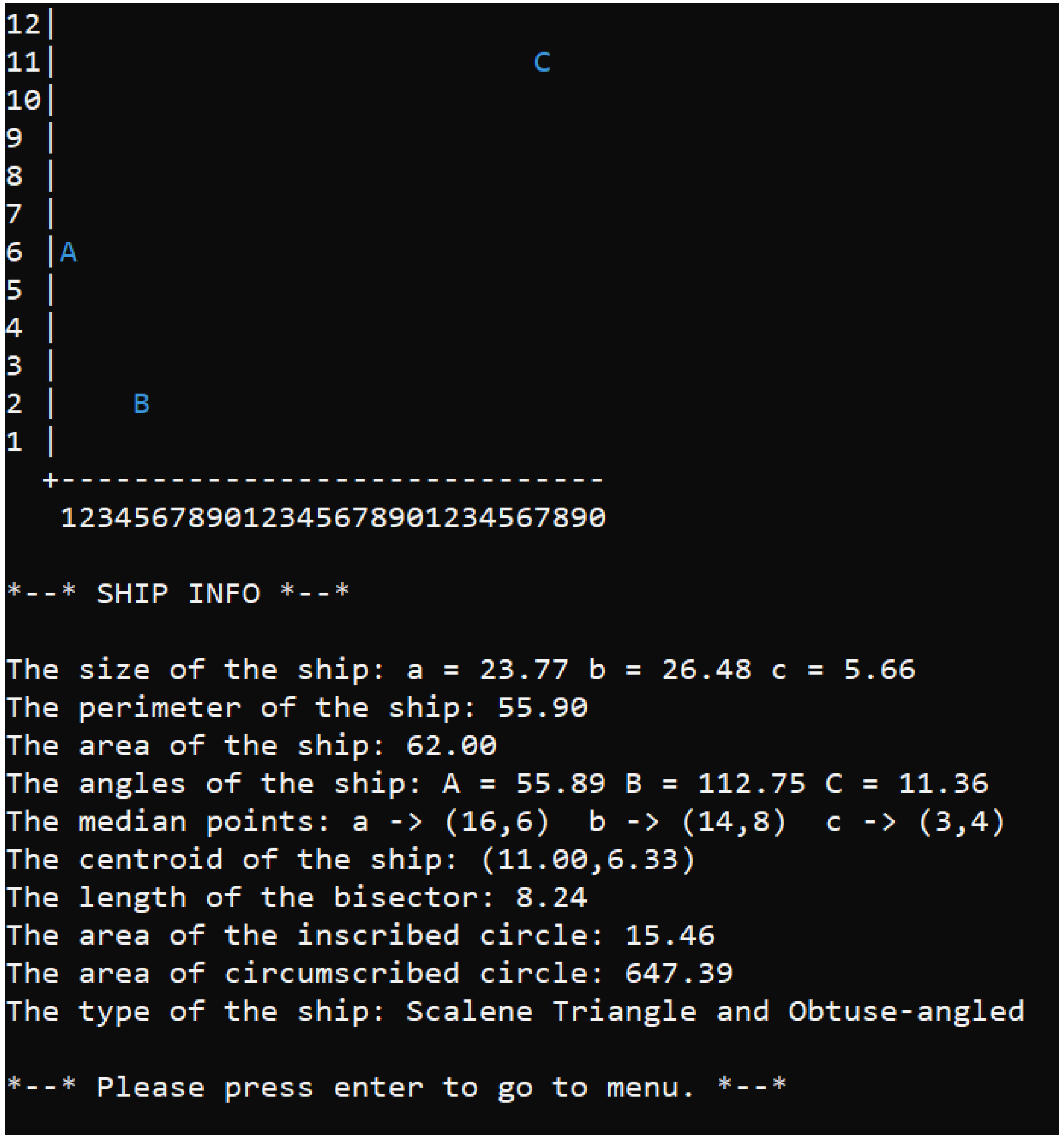
```
*--* MENU *--*

1- Enter Ship Location
2- Ship Info
3- Shoot At The Ship
4- Show High Score Table
5- How To Play
6- Exit

Select an option (1,2,3,4,5,6):
```



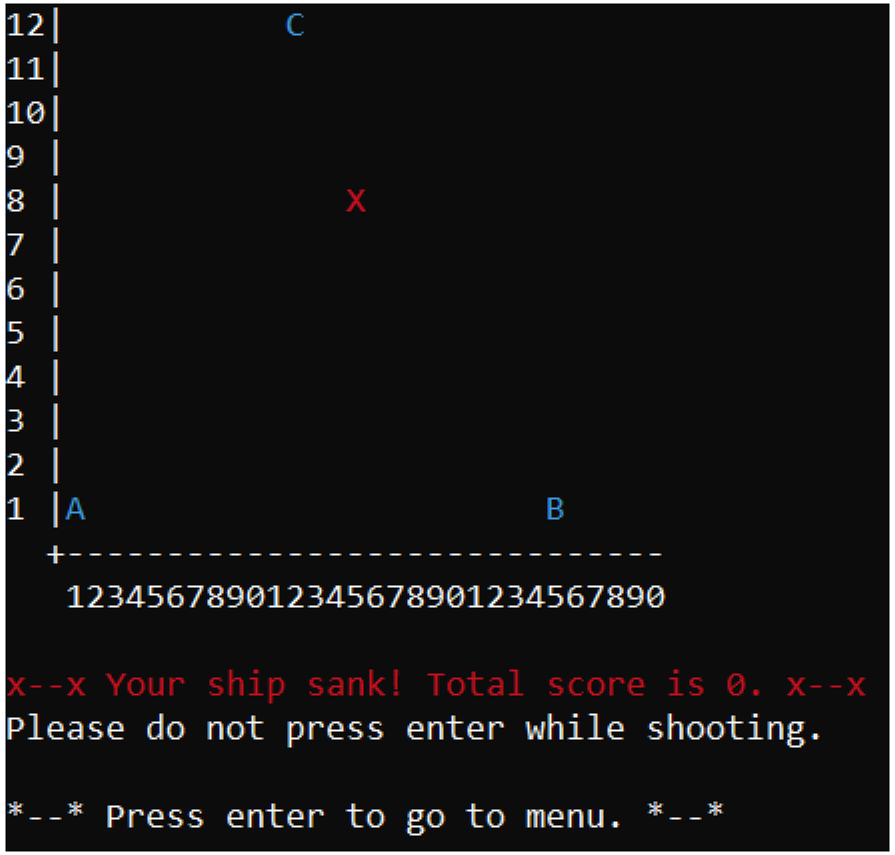
```
Please enter ABC triangles Ax value: +

Please enter a positive integer between 1 and 30. (0,31)
*--*     *--*

Please enter ABC triangles Ax value: 1
Please enter ABC triangles Ay value: e

Please enter a positive integer between 1 and 12. (0,13)
*--*     *--*

Please enter ABC triangles Ay value: 13

Please enter a positive integer between 1 and 12. (0,13)
*--*     *--*

Please enter ABC triangles Ay value: 3
Please enter ABC triangles Bx value: 5
Please enter ABC triangles By value: 6
Please enter ABC triangles Cx value: -1

Please enter a positive integer between 1 and 30. (0,31)
*--*     *--*

Please enter ABC triangles Cx value: 0

Please enter a positive integer between 1 and 30. (0,31)
*--*     *--*

Please enter ABC triangles Cx value: 4
Please enter ABC triangles Cy value: 6
*--* SUCCESSFUL *--*
*--* Please press enter to go to menu. *--*
```



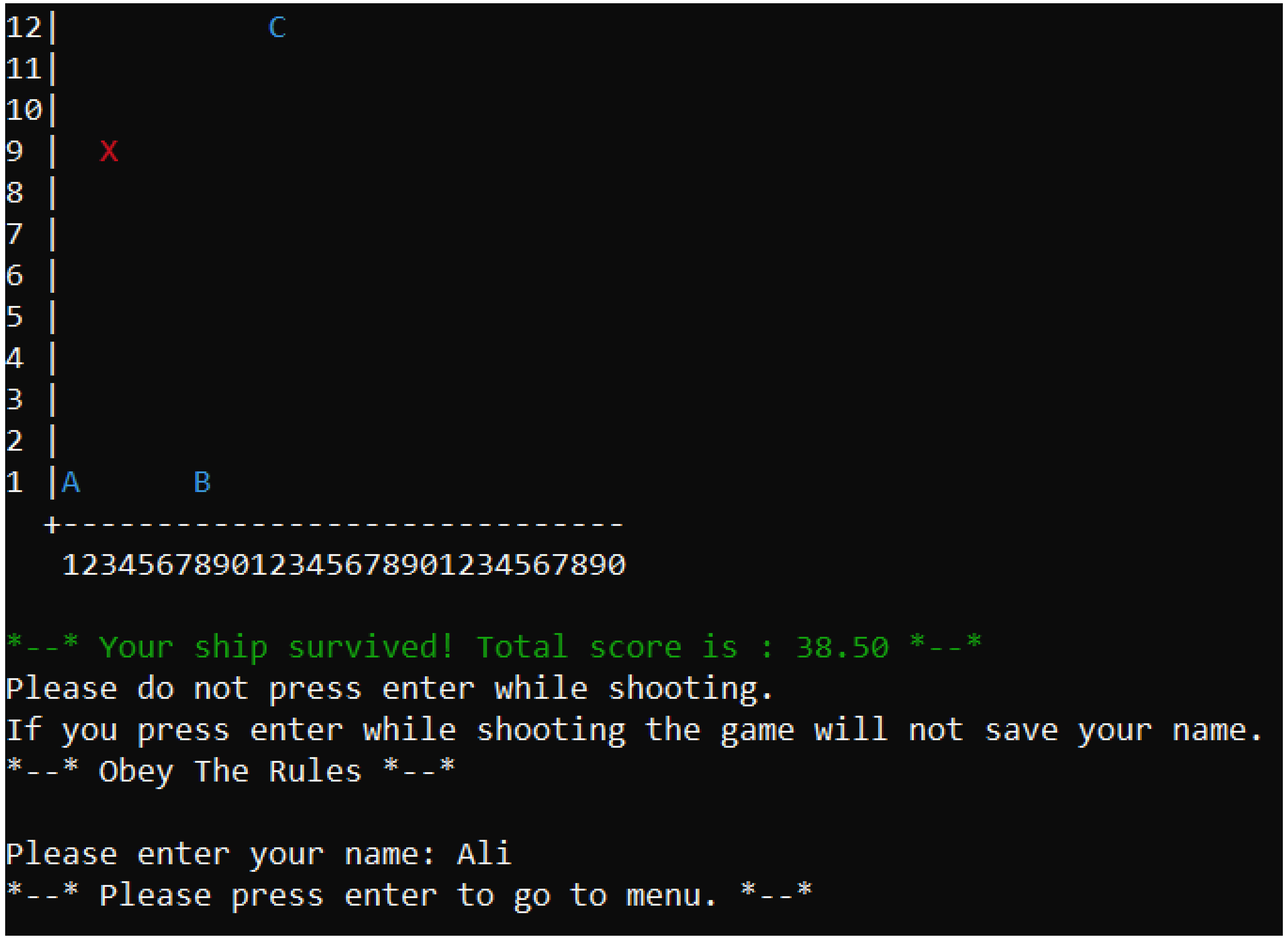
```
Please enter ABC triangles Ax value: 1
Please enter ABC triangles Ay value: 2
Please enter ABC triangles Bx value: 3
Please enter ABC triangles By value: 4
Please enter ABC triangles Cx value: 5
Please enter ABC triangles Cy value: 6
x--x The values you entered do not form triangles. x--x
Please enter ABC triangles Ax value: _
```



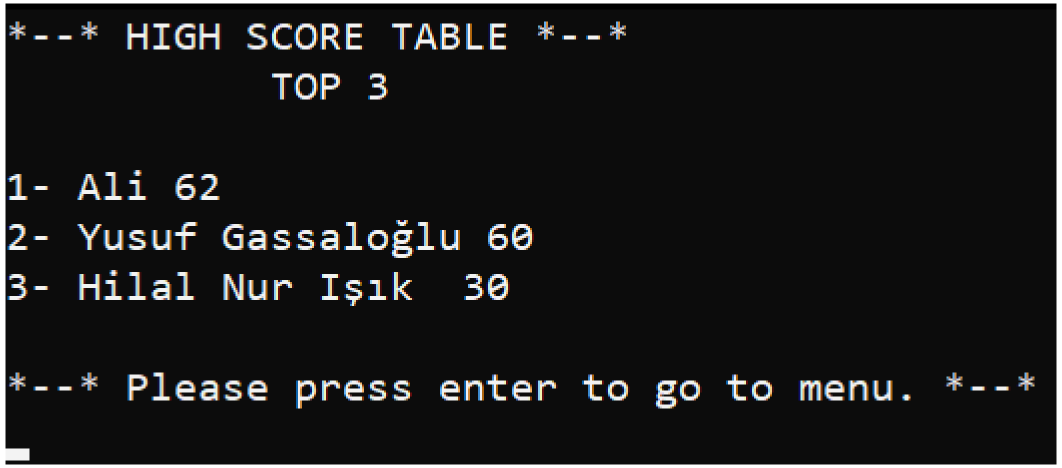
```
12|
11|                          C
10|
9 |
8 |
7 |
6 |A
5 |
4 |
3 |
2 |    B
1 |
  +------------------------------
   123456789012345678901234567890

*--* SHIP INFO *--*

The size of the ship: a = 23.77 b = 26.48 c = 5.66
The perimeter of the ship: 55.90
The area of the ship: 62.00
The angles of the ship: A = 55.89 B = 112.75 C = 11.36
The median points: a -> (16,6)  b -> (14,8)  c -> (3,4)
The centroid of the ship: (11.00,6.33)
The length of the bisector: 8.24
The area of the inscribed circle: 15.46
The area of circumscribed circle: 647.39
The type of the ship: Scalene Triangle and Obtuse-angled

*--* Please press enter to go to menu. *--*
```



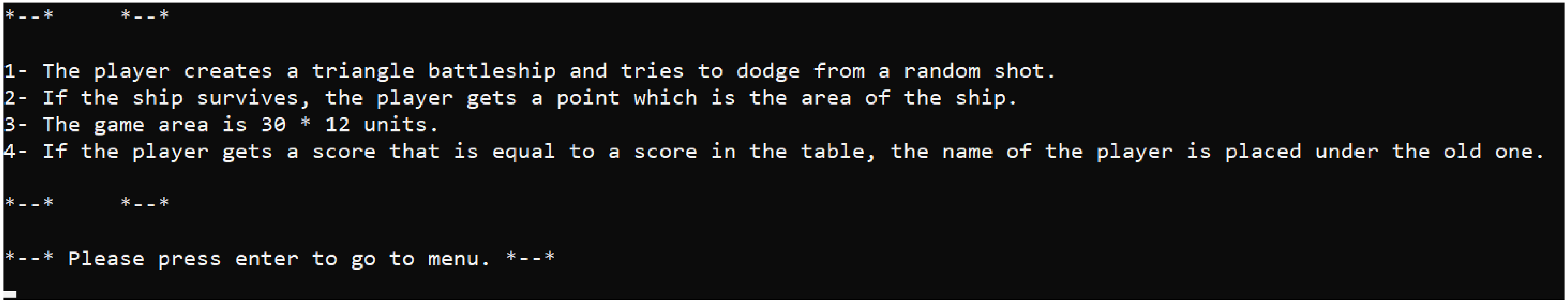
```
12|           C
11|
10|
9 |
8 |              X
7 |
6 |
5 |
4 |
3 |
2 |
1 |A                       B
  +------------------------------
   123456789012345678901234567890

x--x Your ship sank! Total score is 0. x--x
Please do not press enter while shooting.

*--* Press enter to go to menu. *--*
```



```
12|           C
11|
10|
9 |  X
8 |
7 |
6 |
5 |
4 |
3 |
2 |
1 |A      B
  +------------------------------
   123456789012345678901234567890

*--* Your ship survived! Total score is : 38.50 *--*
Please do not press enter while shooting.
If you press enter while shooting the game will not save your name.
*--* Obey The Rules *--*

Please enter your name: Ali
*--* Please press enter to go to menu. *--*
```



```
*--* HIGH SCORE TABLE *--*
           TOP 3

1- Ali 62
2- Yusuf Gassaloğlu 60
3- Hilal Nur Işık  30

*--* Please press enter to go to menu. *--*
```



```
*--*     *--*

1- The player creates a triangle battleship and tries to dodge from a random shot.
2- If the ship survives, the player gets a point which is the area of the ship.
3- The game area is 30 * 12 units.
4- If the player gets a score that is equal to a score in the table, the name of the player is placed under the old one.

*--*     *--*

*--* Please press enter to go to menu. *--*
```

REFERENCES

- https://blackpawn.com/texts/pointinpoly/
- https://www.w3schools.com/cs/index.php
- https://docs.microsoft.com/en-us/dotnet/csharp/