

פרויקטים שהוכנו על-ידי משתתפי קורס מורים מובילים תש"ע

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

משחק סולמות וחבילים

גרסת סי שרפ

פיתוח הפרויקט :

יהודה אבני: גרסת סי שרפ

אביטל (Evi) גרינולד: גרסת ג'אוה

דורית כהן: גרסת סי שרפ

הפרויקט פותח בסביבת WindowsFormsApplication EXPRES C#

תיעוד מחלקות API

המחלקה `GameBoard` יוצרת `Board` בהתאם לתיעוד הבא

תכונות:

```
private int size = 101;
```

```
private int[] board;
```

פעולות:

כותרת השיטה	תיעוד
<code>public GameBoard()</code>	שיטה בונה יוצרת עצם חדש מטיפוס <code>GameBoard</code> מאתחלת את אברי הלוח
<code>public int GetValue(int place)</code>	השיטה מקבלת מקום בלוח ומחזירה את ערך האיבר במקום
<code>public int GetSize()</code>	מחזירה את ערך הגודל של הלוח
<code>public override string ToString()</code>	מחזירה מחרוזת המכילה את ערכי האברים בלוח

המחלקה `Die` מדמה זריקת קובייה

תכונות:

```
private int top;
```

פעולות:

כותרת השיטה	תיעוד
<code>public Die()</code>	שיטה בונה יוצרת עצם חדש מטיפוס <code>Die</code> מאתחלת את ערך הפאה העליונה של הקובייה.
<code>public int ThrowDie()</code>	השיטה מחזירה את ערך הקובייה.
<code>public int GetSize()</code>	השיטה מעדכנת את ערך הפאה העליונה של הקובייה, ומחזירה את ערכה.
<code>public override string ToString()</code>	מחזירה מחרוזת המכילה את ערך הקובייה.

המחלקה **Player** מאתחלת שחקן ומנהלת את נתוניו.

תכונות:

```
private string name;
private int points;
private int place;
private string color;
```

פעולות:

כותרת השיטה	תיעוד
<code>public Player(string name, string color)</code>	שיטה בונה מאתחלת את נתוני השחקן על פי הנתונים שקבלה, כולל איפוס נקודות, ואת מיקומו בתחילת הלוח.
<code>public string GetName()</code>	השיטה מחזירה את שם השחקן כמחרוזת.
<code>public string GetColor()</code>	השיטה מחזירה את צבע השחקן.
<code>public int GetPlace()</code>	השיטה מחזירה את מיקומו של השחקן בלוח.
<code>public int GetPoints()</code>	השיטה מחזירה את מספר הנקודות של השחקן.
<code>public void SetPlace(int place)</code>	השיטה מקבלת מקום בלוח, ומעדכנת את מיקומו של השחקן על פי המיקום שהתקבל.
<code>public void SetPoints(int add)</code>	השיטה מקבלת ערך כניקוד, ומוסיפה למספר הנקודות של השחקן.
<code>public override string ToString()</code>	השיטה מחזירה מחרוזת המכילה את תכונות השחקן.

המחלקה **Ladder** בונה סולם או חבל בהתאם לדרישה

תכונות:

```
private int top;
```

פעולות:

כותרת השיטה	תיעוד
<code>public Ladder ()</code>	שיטה בונה יוצרת עצם חדש מטיפוס Ladder מאתחלת את ערך הפאה העליונה של הקוביה.
<code>fromNumToPoint(int cellNum)</code>	השיטה מקבלת מספר תא בלוח ומחזירה נקודה תואמת על הלוח הגרפי.
<code>draw(Graphics g)</code>	השיטה מקבלת אובייקט גרפי ומציירת סולם/חבל על הלוח הגרפי.
<code>SetUpPoint(Point up)</code>	מעדכנת את מיקום ראש הסולם/חבל.
<code>SetDownPoint(Point dp)</code>	מעדכנת את מיקום זנב הסולם/חבל.

המחלקות המרכיבות את המשחק

מחלקת הניהול Form1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading;

namespace LaddersAndRopes
{
    public partial class Form1 : Form
    {
        Player p1 = new Player();//1 יצירת מופע של שחקן
        Player p2 = new Player();//2 יצירת מופע של שחקן
        GameBoard gb = new GameBoard();//קמת משחק של לוח
        Ladder l = new Ladder();
        Board b=new Board();
        Board1 b1 = new Board1();
        int mp = 0;
        int tr;
        System.Windows.Forms.PaintEventArgs f;

        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            Graphics g = panel1.CreateGraphics();
        }

        private void restartButton_Click(object sender, EventArgs e)
        {
            Graphics g = panel1.CreateGraphics();
            SolidBrush erase = new SolidBrush(Color.Snow);
            g.FillRectangle(erase, 0, 0, 405, 405);
            gb.DrawBord(g);
        }

        //throw die button
        private void throwButton_Click(object sender, EventArgs e)
        {
            Die d=new Die();
            this.tr = d.ThrowDie();//d.Next(1, 7);
        }
    }
}
```

```

if (tr == 1)
    pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._1;
if (tr == 2)
    pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._2;
if (tr == 3)
    pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._3;
if (tr == 4)
    pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._4;
if (tr == 5)
    pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._5;
if (tr == 6)
    pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._6;
pictureBox1.Visible = true;

if (mp % 2 == 0)
{
    moveButton.Visible = true;
    movePlayer2Button.Visible = false;
}
else
{
    moveButton.Visible = false;
    movePlayer2Button.Visible = true;
}
throwButton.Visible = false;
}

// הזזת שחקן בלחיצת עכבר
private void pictureBox1_Click(object sender, EventArgs e)
{
    Random r = new Random();
    Die d = new Die();
    int tr = d.ThrowDie();
    Point p = new Point();

    if (tr == 1)
        pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._1;
    if (tr == 2)
        pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._2;
    if (tr == 3)
        pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._3;
    if (tr == 4)
        pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._4;
    if (tr == 5)
        pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._5;
    if (tr == 6)
        pictureBox1.Image = global::LaddersAndRopes.Properties.Resources._6;
}

// הוספת סולמות וחבלים
private void button1_Click(object sender, EventArgs e)
{

```

```

Ladder l = new Ladder();
Point p = new Point();
Graphics g = panel1.CreateGraphics();
l.draw(g);
Point p1 = new Point();
Point p2 = new Point();
Random r = new Random();
for (int i = 0; i < 10000000; i++) ;
Random r2 = new Random();
p1.X = r.Next(10);
p1.Y = r2.Next(10);

gb.SetPosition(p1);
label1.Text = gb.GetPosition(p1) + ":" + p1.X.ToString() + "," + p1.Y.ToString();
}

private void drawNew_Click(object sender, EventArgs e)
{
    Graphics g = panel1.CreateGraphics();

    b.DrawBord(g);

}
//draw the BordGame
private void panel1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = panel1.CreateGraphics();
    b.DrawBord(g);
}
//move the player position
private void panel1_MouseClick(object sender, MouseEventArgs e)
{
    if (mp % 2 == 0)
    {
        Point position = e.Location;
        pictureBox2.Location = position;
        //drawbord();
    }
    else
    {
        Point position = e.Location;
        pictureBox3.Location = position;
    }
    mp++;
}
//move player 1
public void MovePlayer1(int steps)
{
    int x = 0;

```

```

int y = 360;
int row = 0;
int col = 0;
int lable = 0;
Point p = new Point();

lable = p1.GetPlace();
row = lable / 10;
col = lable % 10;
Point cp = new Point(row, col);
if ((b.GetCellValue(cp).X != 0) || (b.GetCellValue(cp).Y != 0))
{
    label2.Text = "STOP";
    row = b.GetCellValue(cp).Y;
    col = b.GetCellValue(cp).X;
    p1.SetPlace(row*10+col);
}
if ((col == 0) && (row % 2 == 1))
{
    y -= (row - 1) * 40;
    x += (9 - col) * 40;
}
else
if ((col == 0) && (row % 2 == 0))
{
    y -= (row - 1) * 40;
    x = 0;
}
else
{
    if (row % 2 == 1)
    {
        col = 10 - col;
        x += col * 40;
        y -= row * 40;
    }
    else
    {
        col--;
        x += col * 40;
        y -= row * 40;
    }
}
p.X = x;
p.Y = y;
pictureBox2.Location = p;
label2.Text = row.ToString() + " " + col.ToString();

if (p1.GetPlace() >= 100)
{

```

```

label12.Text = "Player 1 win";
label12.Visible = true;
MessageBox.Show("Player 1 win");
//p2.SetPlace(row * 10 + col);
}
}
//move player 2
public void MovePlayer2(int steps)
{
    int x = 0;
    int y = 360;
    int row = 0;
    int col = 0;
    int lable = 0;
    Point p = new Point();

    lable = p2.GetPlace();
    row = lable / 10;
    col = lable % 10;
    Point cp = new Point(row, col);
    if ((b.GetCellValue(cp).X != 0) || (b.GetCellValue(cp).Y != 0))
    {
        label2.Text = "STOP";
        row = b.GetCellValue(cp).Y;
        col = b.GetCellValue(cp).X;
        p2.SetPlace(row * 10 + col);
    }
    if ((col == 0) && (row % 2 == 1))
    {
        y -= (row - 1) * 40;
        x += (9 - col) * 40;
    }
    else
    if ((col == 0) && (row % 2 == 0))
    {
        y -= (row - 1) * 40;
        x = 0;
    }
    else
    {
        if (row % 2 == 1)
        {
            col = 10 - col;
            x += col * 40;
            y -= row * 40;
        }
        else
        {
            col--;
            x += col * 40;
        }
    }
}

```

```
        y -= row * 40;
    }
}

p.X = x+20;
p.Y = y+20;
pictureBox3.Location = p;
label2.Text = row.ToString() + " " + col.ToString();

if (p2.GetPlace() >= 100)
{
    label12.Text = "Player 2 win";
    label12.Visible = true;
    MessageBox.Show("Player 2 win");
}
}
//from cell number to cell coordinat translation

public Point fromNumToPoint(int cellNum)
{

    Point p = new Point();
    int x = 0;
    int y = 360;
    int row = 0;
    int col = 0;

    row = cellNum / 10;
    col = cellNum % 10;
    if ((col == 0) && (row % 2 == 1))
    {
        y -= (row - 1) * 40;
        x += (9 - col) * 40;
    }
    else
    if ((col == 0) && (row % 2 == 0))
    {
        y -= (row - 1) * 40;
        x = 0;
    }
    else
    {
        if (row % 2 == 1)
        {
            col = 10 - col;
            x += col * 40;
            y -= row * 40;
        }
        else
        {
```

```

        col--;
        x += col * 40;
        y -= row * 40;
    }
}

p.X = x;
p.Y = y;
pictureBox2.Location = p;
label2.Text = x.ToString() + " " + y.ToString();

return p;
}

//from cell number to cell coordinat
private void button1_Click_1(object sender, EventArgs e)
{
    int x, y;

    x = pictureBox2.Location.X + 20;

    y = pictureBox2.Location.Y + 20;

    label2.Text = x.ToString() + " " + y.ToString();
    int sx, sy, lx, ly;
    Graphics g = panel1.CreateGraphics();
    sx = pictureBox2.Location.X ;
    sy = pictureBox2.Location.Y ;
    Point pu = new Point();
    pu.X = sx;
    pu.Y = sy;

    Point pd = new Point();
    pd = l.fromNumToPoint(int.Parse(textBox1.Text));
    pd.X += 20;
    pd.Y += 20;
    l.SetupPoint(pu);
    l.SetDownPoint(pd);
    l.draw(g);
}
//control the visibility of controls
private void moveButton_Click(object sender, EventArgs e)
{
    //this.tr = 1;
    Point p = new Point();
    pictureBox1.Visible = true;
    p1.SetPoints(tr);
    p1.SetPlace(p1.GetPlace() + tr);
    label7.Text = "player 1";
}

```

```

label9.Text = p1.GetPlace().ToString();
MovePlayer1(tr);
label9.Text = p1.GetPlace().ToString();

p.X = pictureBox2.Location.X;
p.Y = pictureBox2.Location.Y;
p.X = p1.GetPosition().X * 40 + 15;
p.Y = p1.GetPosition().Y * 40 + 15;
//pictureBox2.Location = p;
label3.Text = p1.GetPoints().ToString();
label6.Text = p2.GetPoints().ToString();

mp++;
throwButton.Visible = true;
moveButton.Visible = false;
}
//player 2 button
private void movePlayer2Button_Click(object sender, EventArgs e)
{
    //this.tr = 1;
    Point p = new Point();

    pictureBox1.Visible = true;
    p2.SetPoints(tr);
    p2.SetPlace(p2.GetPlace() + tr);
    label7.Text = "player 2";
    label11.Text = p2.GetPlace().ToString();
    MovePlayer2(tr);
    label11.Text = p2.GetPlace().ToString();

    p.X = pictureBox2.Location.X;
    p.Y = pictureBox2.Location.Y;
    p.X = p2.GetPosition().X * 40 + 15;
    p.Y = p2.GetPosition().Y * 40 + 15;

    label3.Text = p1.GetPoints().ToString();
    label6.Text = p2.GetPoints().ToString();

    mp++;
    throwButton.Visible = true;
    movePlayer2Button.Visible = false;
}
}
}

```

המחלקה GameBoard יוצרת לוח משחק

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;

namespace LaddersAndRopes
{
    class GameBoard
    {
        private int size = 101;
        private int[] board;
        Point[,] D2Board=new Point[10,10];
        int[,] bArray = new int[10, 10];
        Random rnd = new Random();

        public GameBoard()
        {
            this.board = new int[this.size];
            int i, num;
            Point p=new Point();
            for (i = 1; i < this.size-1; i++)
            {
                num = rnd.Next(-50, 50);
                while (num < 0 && ((i + num) <= 0))
                {
                    num = rnd.Next(-50, 50);
                }
                while (num > 0 && (i + num)>this.size-1)
                {
                    num = rnd.Next(-50, 50);
                }
                if (num != 0) this.board[i] = num;
            }

            for (int k = 0; k < D2Board.GetLength(0); k++)
            {
                for (int j = 0; j < D2Board.GetLength(1); j++)
                {
                    p.X=k; p.Y=j;
                    D2Board[k,j]=p;
                    bArray[k, j] = 0;
                }
            }
        }

        public int GetValue(int place)
        {
            return this.board[place];
        }
    }
}
```

```

public int GetSize()
{
    return this.size;
}

public void DrawBord(Graphics g)
{
    int x = 0, y = 0;
    Pen p=new Pen(Color.Green);
    Pen e = new Pen(Color.Snow);
    //SolidBrush e = new SolidBrush(Color.Snow);
    SolidBrush b = new SolidBrush(Color.Blue);
    SolidBrush bg = new SolidBrush(Color.Green);
    Font f = new Font("Comic Sans MS", 8);
    g.DrawRectangle(e, 0, 0, 405, 405);

    for (int i = 0; i < 10; i++)//draw the bord by the point array position
    {
        for (int j = 0; j < 10; j++)
        {
            g.DrawRectangle(p, D2Board[i, j].X+x-i, D2Board[i, j].Y+y-j, 40, 40);
            g.DrawString(i.ToString() + ",", f, b, D2Board[i, j].X + x - i, D2Board[i, j].Y + y - j);
            g.DrawString(j.ToString(), f, b, D2Board[i, j].X + x - i + 10, D2Board[i, j].Y + y - j);
            x += 40;//delta x
        }
        x = 0;
        y += 40;//delta y
    }
}

public void SetPosition(Point p)
{
    this.bArray[p.X, p.Y] = 99;
}

public string GetPosition(Point p)
{
    return bArray[p.X, p.Y].ToString();
}

public override string ToString()
{
    string stBoard = "";
    for (int i=1; i<this.size; i++)
        stBoard+=i+" "+(this.board[i])+"\n";
    return stBoard;
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;

namespace LaddersAndRopes
{
    class Board
    {
        Point[,] p;//the board array
        int[,] intArray;
        Point[,] tp;//array for
        Point ep=new Point();
        int size=10;
        int delta = 40;
        public Board()
        {
            Random ri,rj;
            p = new Point[size, size];
            intArray = new int[size, size];
            tp = new Point[size, size];
            //init the board array & referenc of ladders ropes
            //for (int i = size-1; i >= 0; i--)
            //{

            //    rj = new Random();
            //    for (int j = 0; j < size; j++)
            //    {
            //        for (int k = 0; k < 10000000; k++) ;

            //        ri = new Random();
            //        if (ri.Next(12) == 1)
            //        {
            //            p[i, j].X = rj.Next(10);
            //            p[i, j].Y = rj.Next(10);

            //        }
            //        else
            //        {
            //            //p[i, j].X = 0;
            //            //p[i, j].Y = 0;
            //        }
            //    }
            //}

            //set position for arow
            p[0, 6].X = 5;
            p[0, 6].Y = 4;
        }
    }
}

```

```

p[3, 4].X = 1;
p[3, 4].Y = 4;
p[6, 2].X = 9;
p[6, 2].Y = 2;
p[7, 6].X = 4;
p[7, 6].Y = 8;
intArray[0, 6] = 55;

intArray[3, 4] = 20;

intArray[6, 2] = 33;
intArray[6,4] = 93;
intArray[7, 6] = 32;
}

public void DrawBord(Graphics g)
{
    int x = 0, y = 360;//set the first point for the first rectangle
    int ox=0, oy=0;
    Point p = new Point();
    Pen pp = new Pen(Color.Green);
    Pen e = new Pen(Color.Snow);
    int counter;
    //int counter2 = 1;
    //int rowCounter = 1;

    SolidBrush b = new SolidBrush(Color.Blue);
    SolidBrush r = new SolidBrush(Color.Red);
    SolidBrush bg = new SolidBrush(Color.Black);
    SolidBrush bs = new SolidBrush(Color.Snow);
    Font f = new Font("Comic Sans MS", 8);
    Font end = new Font("Comic Sans MS", 12);
    g.DrawString("end",end,bg,5,5);
    g.DrawString("start", end, bg, 1, 365);
    Point up = new Point();
    Point dp = new Point();
    for (int i = 0; i < this.intArray.GetLength(0); i++)//draw the bord by the point array
position
    {
        for (int j = 0; j < this.intArray.GetLength(1); j++)
        {
            counter = RectangleNum(i,j);
            g.DrawRectangle(pp, x , y , 40, 40);

            tp[i, j].X = i;
            tp[i, j].Y = j;

            //if (this.p[i, j].X != 0 || this.p[i, j].Y != 0)
            //{
            //    ox = this.p[i, j].X;

```

```

// oy = this.p[i, j].Y;
// g.DrawLine(pp, x + 20, y + 20, oy * 40 + 20, 360 - (ox * 40 - 20));
// g.FillEllipse(r, x + 15, y + 15, 10, 10);
// g.DrawString(this.p[i, j].X.ToString() + ", " + this.p[i, j].Y.ToString(), f, bg, x, y+20 );
// g.FillEllipse(b, oy * 40 + 15, 360 - (ox * 40 - 15),10,10 );
//}
g.DrawString(counter.ToString(), f, bg, x, y);
if (this.intArray[i, j] != 0)
{
    up = this.fromNumToPoint(counter);
    up.X -= 20;
    up.Y += 20;
    if (intArray[i, j] < 0)
        intArray[i, j] = counter + intArray[i, j];
    dp = this.fromNumToPoint(intArray[i, j]);
    dp.X += 20;
    dp.Y += 20;

    g.DrawLine(pp, up, dp);

    g.FillEllipse(r, up.X - 5, up.Y - 5, 10, 10);

    g.FillEllipse(b, dp.X-5,dp.Y-5, 10, 10 ) ;

    g.DrawString("goto"+intArray[i, j].ToString(), f, bg, x-40, y+10);
}
//if (rowCounter % 2 == 0)
//    counter1 = 1;
//g.DrawString(counter.ToString(), f, bg, x, y);
//counter1++;
//if (rowCounter % 2 == 0)
//    counter1 = 1;
x += this.delta;//delta x
}
//rowCounter++;
x = 0;
y -= this.delta;//delta y
}
}

//from cell number to cell coordinat
public Point fromNumToPoint(int cellNum)
{
    Point p = new Point();
    int x = 0;
    int y = 360;
    int row = 0;
    int col = 0;

```

```

row = cellNum / 10;
col = cellNum % 10;
if ((col == 0) && (row % 2 == 1))
{
    y -= (row - 1) * 40;
    x += (9 - col) * 40;
}
else
if ((col == 0) && (row % 2 == 0))
{
    y -= (row - 1) * 40;
    x = 0;
}
else
{
    if (row % 2 == 1)
    {
        col = 10 - col;
        x += col * 40;
        y -= row * 40;
    }
    else
    {
        col--;
        x += col * 40;
        y -= row * 40;
    }
}

p.X = x;
p.Y = y;
//pictureBox2.Location = p;
//label2.Text = x.ToString() + " " + y.ToString();

return p;
}

public int RectangleNum(int rowNum,int colNum)
{

    int numL = 10;
    int num =1;
    int numR = 10;
    if(rowNum%2==0)
        num = (rowNum) * 10 + colNum + 1;
    if (rowNum % 2 == 1)
    {
        num = (rowNum) * 10 + (10-colNum);
    }
}

```

```
    return num;
}

public void MovePlayer1(int steps, Graphics g)
{
    int x;
    int y;
}

public int GetSize()
{
    return this.size;
}

public Point GetPoint()
{
    return this.ep;
}
//return the value of the wanted cell
public Point GetCellValue(Point op)
{
    Point IP = new Point();
    if (op.X < 10 && op.Y < 10)
    {
        IP.X = this.p[op.X, op.Y].X;
        IP.Y = this.p[op.X, op.Y].Y;
    }
    return IP;
}
}
}
```

המחלקה DIE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace LaddersAndRopes
{
    public class Die
    {
        private int top;

        Random rnd = new Random();

        public Die()
        {
            Random rnd = new Random();
            this.top = rnd.Next(1, 7);
        }

        public int GetTop()
        {
            return this.top;
        }

        public int ThrowDie()
        {
            Random rnd = new Random();
            this.top = rnd.Next(1, 7);
            //this.top++;
            return this.top;
        }

        public override string ToString()
        {
            return " " + this.top;
        }
    }
}
```

המחלקה Player

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;

namespace LaddersAndRopes
{
    public class Player
    {
        private string name;
        private int points;
        private int place;//num of cell
        private string color;
        private Point pr, pc;//picture location
        private Point position;//num of cell

        public Player()
        {
            this.name = "default";
            this.color = "default";
            this.points = 0;
            this.place = 1;
            pr = new Point();
            pc = new Point();
            position = new Point(0, 9);
            pr.X = 315; pr.Y = 370;
            pc.X = 0; pc.Y = 0;
        }

        public Player(string name, string color)
        {
            this.name = name;
            this.color = color;
            this.points = 0;
            this.place = 0;
        }

        public string GetName()
        {
            return this.name;
        }

        public string GetColor()
        {
            return this.color;
        }
    }
}
```

```
public int GetPlace()
{
    return this.place;
}

public int GetPoints()
{
    return this.points;
}

public void SetPlace(int place)
{
    this.place = place;
}
//set new player position
public void SetPosition(int diePoints)
{
    int temp = this.position.X + diePoints;
    if (this.position.Y % 2 == 1)
    {
        if (temp <= 9)
        {
            this.position.X += diePoints;
        }
        else
        {
            this.position.Y--;
            //this.position.X = 9 - diePoints;
            //this.position.X -= diePoints;
        }
    }

    else
        if (temp > 9)
        {
            this.position.X -= diePoints;
            if(this.position.X <=0)
                this.position.Y--;
        }
    }
}
//set position cordinat after throw die
public void SetPositionX(int x)
{
    this.position.X += x;
}
public void SetPositionY(int y)
{
    this.position.Y += y;
}
```

```
//return player position
public Point GetPosition()
{
    return this.position;
}
public void SetPoints(int add)
{
    this.points += add;
}

public void PlayerPosition(int tr, System.Windows.Forms.PictureBox pictureBox2,
System.Windows.Forms.Panel panel1)
{
    //pictureBox2.Visible = false;
    //Point p = new Point(300,30);
    //pictureBox2.Location = p;
    //pictureBox2.Visible = true;
}
public override string ToString()
{
    return this.name + " who's color is: " + this.color + " is in place " + this.place + " and
has " + this.points + " points.";
}
}
}
```

המחלקה Ladder יוצרת סולם או חבל חדש

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;

namespace LaddersAndRopes
{
    class Ladder
    {
        Point p1 = new Point(50, 100);
        Point p2 = new Point(45, 110);
        Point p3 = new Point(55, 110);
        Point p4 = new Point(55, 110);
        Point[] pArray = new Point[15];
        int x=0, y=0;
        int ePos;
        int upPos;
        int downPos;
        Point pu;
        Point pd;
        public Ladder()
        {
            Random ur = new Random();
            pu = new Point();
            pd = new Point();

            for (int i = 0; i < 15; i+=3)
            {
                this.ePos = ur.Next(1, 100);
                this.pArray[i].X = (ePos % 10) * 40 + x+15;// +15;
                this.pArray[i].Y = (ePos / 10) * 40+15;// +65;

                this.upPos = ur.Next(1, 100);
                this.pArray[i + 1].X = (upPos % 10) * 40 + x + 15;// +20;
                this.pArray[i + 1].Y = (upPos / 10) * 40 + 15;// +70;

                this.downPos = ur.Next(1, 100);
                this.pArray[i + 2].X = (downPos % 10) * 40 + x + 15;// +20;
                this.pArray[i + 2].Y = (downPos / 10) * 40 + 15;// +70;
            }
        }

        //from cell number to cell coordinat
        public Point fromNumToPoint(int cellNum)
        {
            Point p = new Point();
            int x = 0;    int y = 360;
            int row = 0;    int col = 0;

            row = cellNum / 10;

```

```

col = cellNum % 10;
if ((col == 0) && (row % 2 == 1))
{
    y -= (row - 1) * 40;
    x += (9 - col) * 40;
}
else
if ((col == 0) && (row % 2 == 0))
{
    y -= (row - 1) * 40;
    x = 0;
}
else
{
    if (row % 2 == 1)
    {
        col = 10 - col;
        x += col * 40;
        y -= row * 40;
    }
    else
    {
        col--;
        x += col * 40;
        y -= row * 40;
    }
}

p.X = x;
p.Y = y;
return p;
}

public void SetUpPoint(Point up)
{
    this.pu = up;
}

public void SetDownPoint(Point dp)
{
    this.pd = dp;
}

public void draw(Graphics g)
{
    Pen p = new Pen(Color.Red,4);
    Pen pb = new Pen(Color.Blue,4);
    SolidBrush b = new SolidBrush(Color.Red);
    g.DrawLine(p, this.pu, this.pd);
}
}
}

```