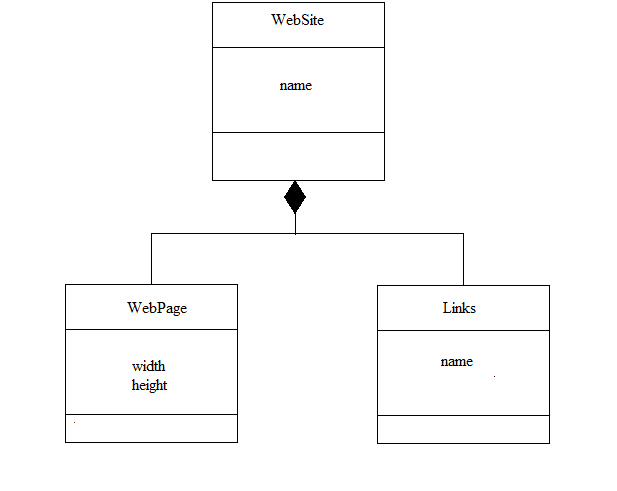
|  |  |  |
| --- | --- | --- |
|  | **NED University of Engineering and Technology** | **Object Oriented Programming(C++ and C#)** |
| **Department of Software Engineering** | **First Year(Section A and B)** |

**ASSIGNMENT # 2**

Q.1 **Problem Statement:**

We have the following part of class diagram showing composition relationship:



You are required to implement above class diagram (complete program) in C++ with all data members, constructors, member functions and concept (composition) given in the class diagram/table 1.

See the sample output to view the messages you need to print in constructors of all classes. In case of parameterize constructors, you can select any of data member(s) according to the type of constructor.

#include<iostream>

#include<string>

#include<string.h>

using namespace std;

class Website {

private:

string name = "";

public:

Website() {}

Website(string n):name(n){}

void setname(string n){

name = n;

}

string getname() { return name; }

};

class Webpage:public Website {

private:

float width;

float height;

public:

Webpage() {}

Webpage(float w,float h) {

width = w;

height = h;

}

void setwidth(float w) {

width = w;

}

float getwidth() {

return width;

}

void setheight(float h) {

height = h;

}

float getheight() {

return height;

}

void displaydata()

{

cout << "The name of website is "+getname()<<endl;

cout << "The width of website is " << getwidth() << endl;

cout << "The height of website is " << getheight() << endl;

}

};

class link :public Website {

private:

string name = "";

public:

link() {}

link(string n) {

name = n;

}

void setname(string n)

{

name = n;

}

string getname()

{

return name;

}

void displaydata()

{

cout << "The name of website is " << Website::getname()<<endl;

cout << "The link name attached to this website is " << getname() << endl;

}

};

int main()

{

Webpage w(50,50);

w.setname("www.google.com");

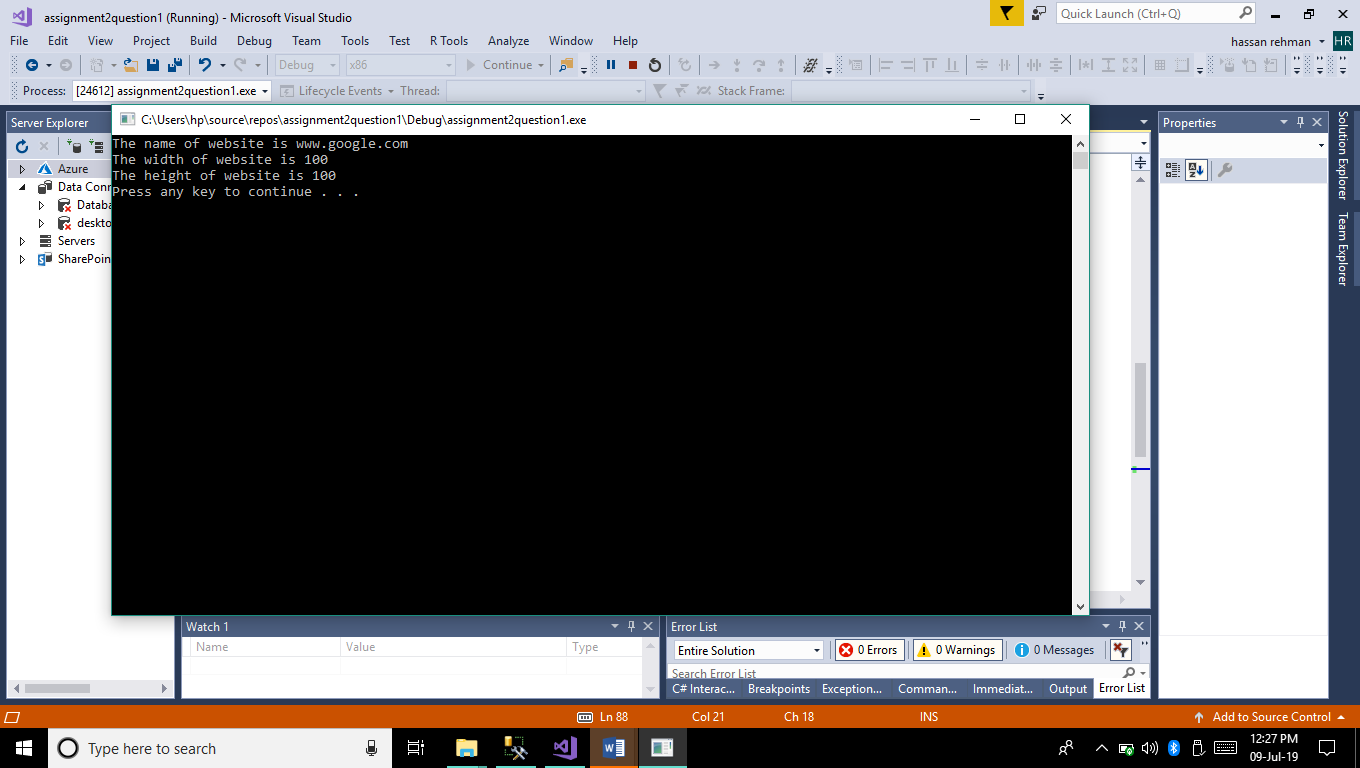
w.setheight(100);

w.setwidth(100);

w.displaydata();

return 0;

}



Q.2 Create an application in C# than can generate raining effect by using Labels or Picture Box and Timer Control.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Media;

namespace raining\_effect

{

public partial class form1 : Form

{

static int k = 40;

Rectangle[] l = new Rectangle[k];

Rectangle[] l1 = new Rectangle[k];

int y2 = 30;

int x1 = 10;

int y1 = 0;

// Rectangle r = new Rectangle(33, 55, 10, 2);

Pen p = new Pen(Color.Blue);

Brush b = new SolidBrush(Color.Blue);

public form1()

{

InitializeComponent();

}

private void form1\_Load(object sender, EventArgs e)

{

// g = CreateGraphics();

timer1.Enabled = true;

timer1.Start();

// g.DrawRectangle(p, r);

// g.FillRectangle(b, r);

for (int i = 0; i < k; i++)

{

l[i].X = x1;

l[i].Y = y1;

l[i].Width = 2;

l[i].Height = 8;

x1 += 25;

}

for (int i = 0; i < k; i++)

{

l1[i].X = x1;

l1[i].Y = y2;

l1[i].Width = 2;

l1[i].Height = 8;

x1 += 25;

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

y1 += 10;

y2 += 10;

// r.X+= 40;

for (int i = 0; i <k; i++)

{

//l[i].X = x1;

l[i].Y = y1;

l[i].Width = 2;

l[i].Height = 8;

l1[i].Y = y2;

l1[i].Width = 2;

l1[i].Height = 8;

}

if (y1 == 200)

{

y1 = 0;

}

if (y2 == 200)

{

y2 = 0;

}

Invalidate();

}

private void form1\_Paint(object sender, PaintEventArgs e)

{

for (int i = 0; i < k; i++)

{

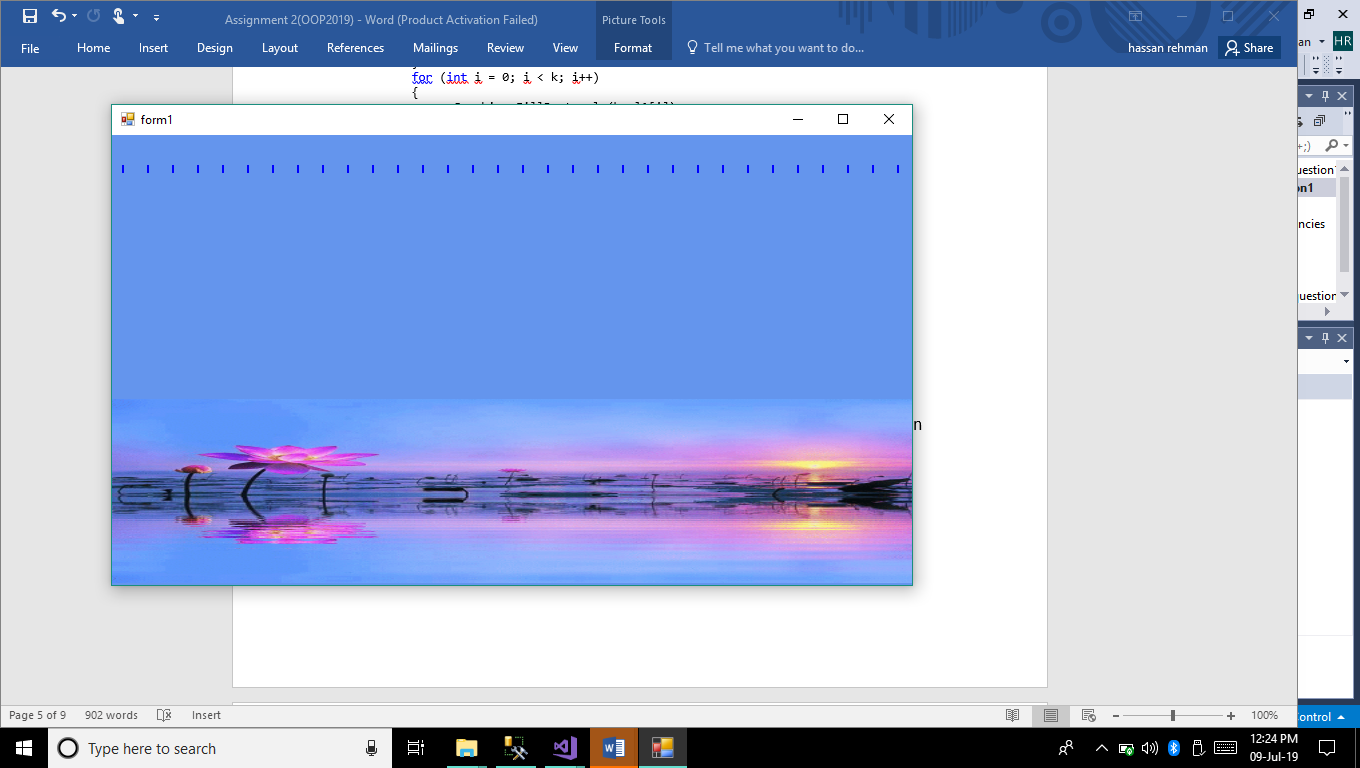
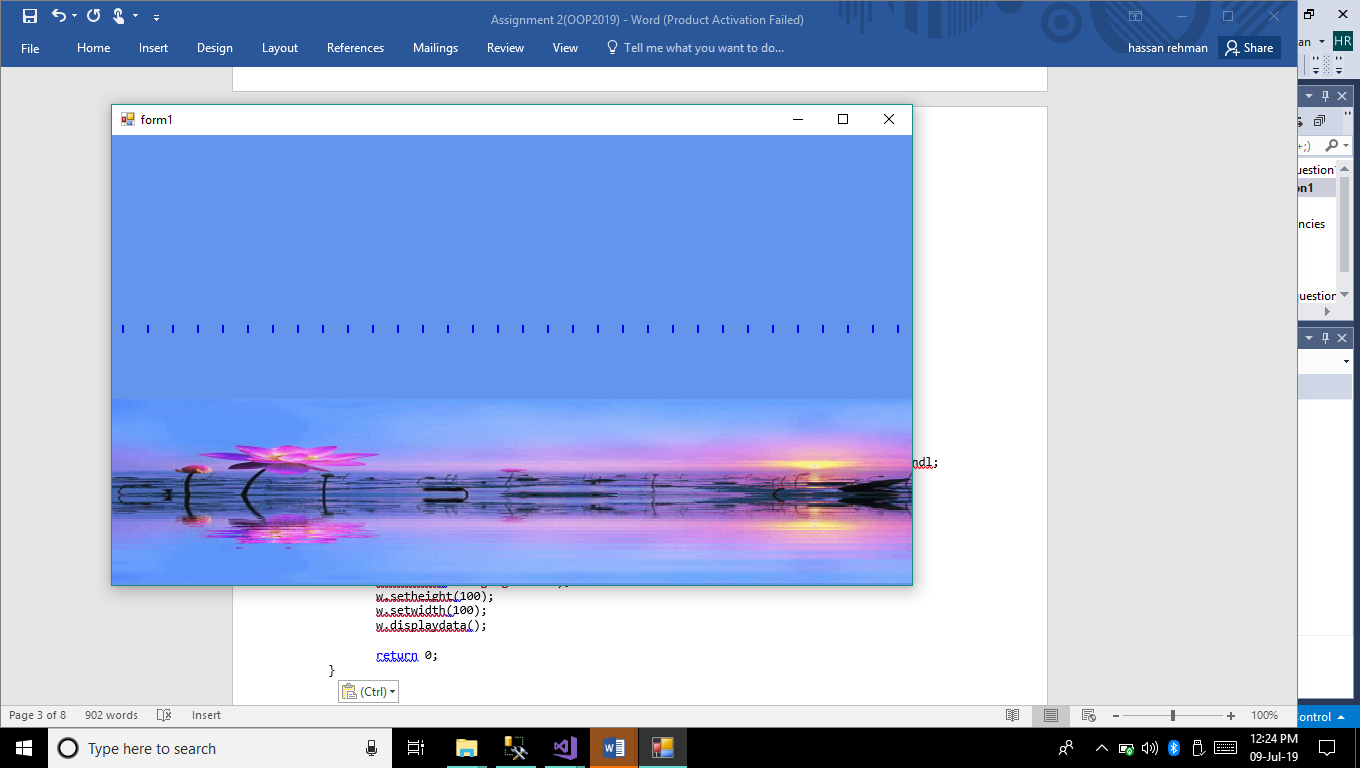
e.Graphics.FillRectangle(b, l[i]);

}

for (int i = 0; i < k; i++)

{

e.Graphics.FillRectangle(b, l1[i]);



}

}

}

}

Q.3 Create an application that can display moving car horizontally and vertically on track bar using C# code.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace racingtrack

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

pictureBox1.Image.RotateFlip(RotateFlipType.Rotate90FlipY);

}

private void Form1\_Load(object sender, EventArgs e)

{

timer1.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

// if (pictureBox1.intersa

}

int count = 0,count1=0;

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{if(count1==0||count1==4)

{

count1 = 0;

if (e.KeyCode == Keys.Left)

{

pictureBox1.Location = new Point(pictureBox1.Location.X - 5, pictureBox1.Location.Y);

}

else if (e.KeyCode == Keys.Right)

{

pictureBox1.Location = new Point(pictureBox1.Location.X + 5, pictureBox1.Location.Y);

}

}

if (pictureBox1.Location.X>=pictureBox5.Location.X-80&&count1==0)

{

pictureBox1.Image.RotateFlip(RotateFlipType.Rotate90FlipX);

count++;

count1++;

}

if (count1 == 1)

{

if (e.KeyCode == Keys.Up)

{

pictureBox1.Location = new Point(pictureBox1.Location.X, pictureBox1.Location.Y-5);

}

else if (e.KeyCode == Keys.Down)

{

pictureBox1.Location = new Point(pictureBox1.Location.X, pictureBox1.Location.Y+5);

}

}

if (pictureBox1.Location.Y <= pictureBox3.Location.Y + 37 && count1 == 1)

{

pictureBox1.Image.RotateFlip(RotateFlipType.Rotate90FlipX);

count++;

count1++;

}

if (count1==2)

{

if (e.KeyCode == Keys.Left)

{

pictureBox1.Location = new Point(pictureBox1.Location.X - 5, pictureBox1.Location.Y);

}

else if (e.KeyCode == Keys.Right)

{

pictureBox1.Location = new Point(pictureBox1.Location.X + 5, pictureBox1.Location.Y);

}

}

if (pictureBox1.Location.X <= pictureBox4.Location.X + 30 && count1 == 2)

{

pictureBox1.Image.RotateFlip(RotateFlipType.Rotate90FlipX);

count++;

count1++;

}

if (count1 == 3)

{

if (e.KeyCode == Keys.Up)

{

pictureBox1.Location = new Point(pictureBox1.Location.X, pictureBox1.Location.Y - 5);

}

else if (e.KeyCode == Keys.Down)

{

pictureBox1.Location = new Point(pictureBox1.Location.X, pictureBox1.Location.Y + 5);

}

}

if (pictureBox1.Location.Y >= pictureBox2.Location.Y - 45 && count1 == 3)

{

pictureBox1.Image.RotateFlip(RotateFlipType.Rotate90FlipX);

count++;

count1++;

}

}

private void pictureBox7\_Click(object sender, EventArgs e)

{

}

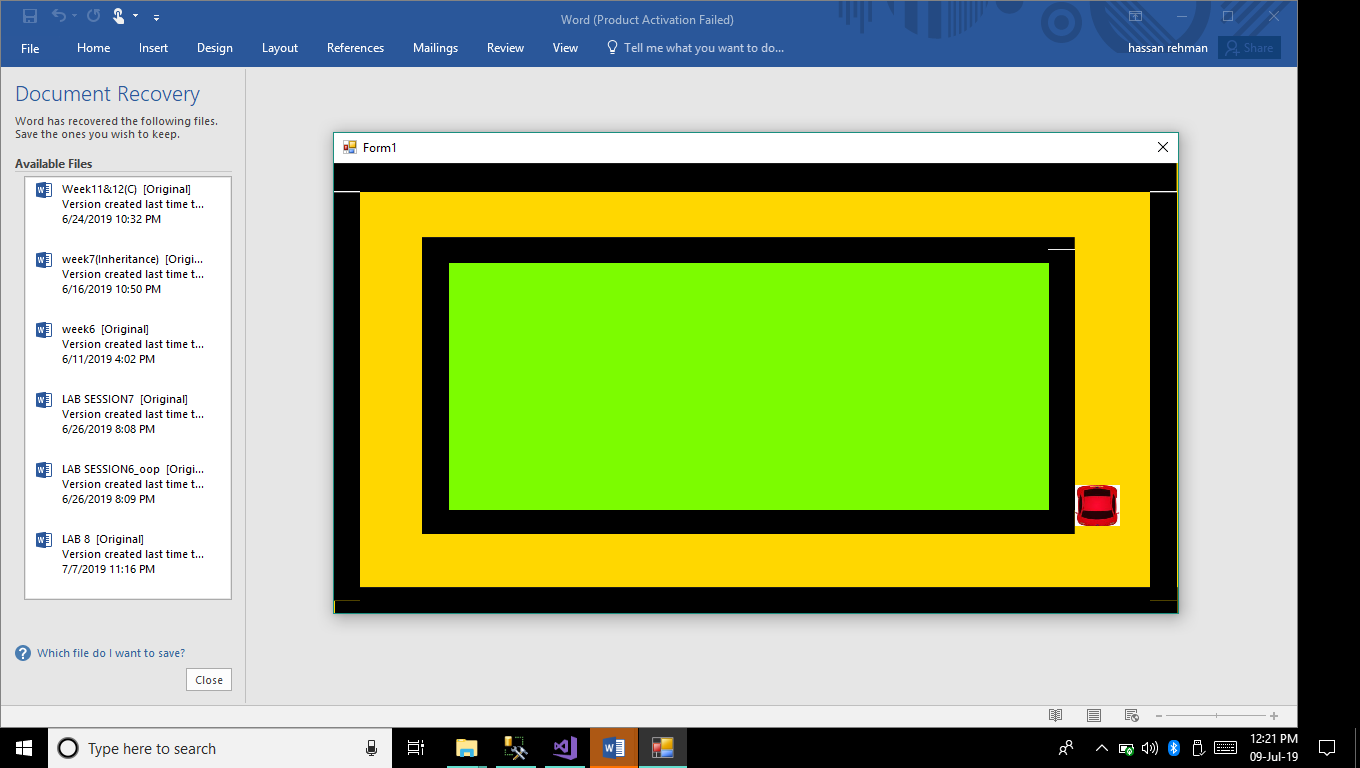
private void pictureBox2\_Click(object sender, EventArgs e)

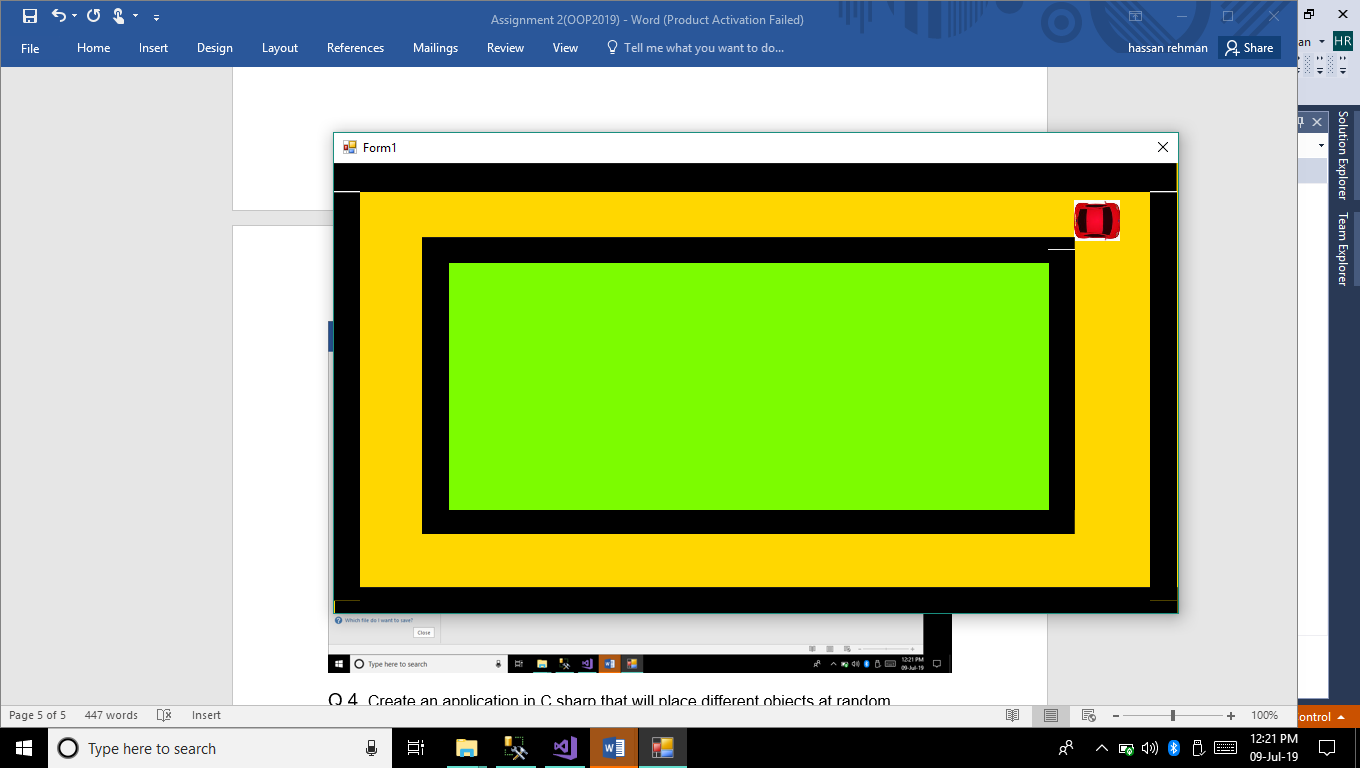
{

}

}

}





Q.4. Create an application in C sharp that will place different objects at random locations over the form and they are moving with random speed and would be killed by some other object. Application must be able to count the score that has been produced by the player.

int score = 0;

PictureBox[] gerry = new PictureBox[4];

PictureBox p2 = new PictureBox();

private void Form1\_Load(object sender, EventArgs e)

{

gerry[0] = pictureBox2;

gerry[1] = pictureBox3;

gerry[2] = pictureBox4;

gerry[0].Show();

gerry[1].Show();

gerry[2].Show();

this.Location = new Point(0, 0);

this.Size = Screen.PrimaryScreen.WorkingArea.Size;

}

private void timer1\_Tick(object sender, EventArgs e)

{

pictureBox1.Left -= 2;

for (int i = 0; i < 3; i++)

{

if (pictureBox1.Bounds.IntersectsWith(gerry[i].Bounds) && (gerry[i].Visible))

{

this.Controls.Add(p2);

p2.BringToFront();

p2.SizeMode = PictureBoxSizeMode.StretchImage;

p2.Size = new Size(100, 100);

p2.BackColor = Color.Transparent;

p2.Location = gerry[i].Location;

p2.Image = Image.FromFile(@"E:/fire.png");

gerry[i].Hide();

p2.Show();

pictureBox1.BringToFront();

score += 1;

}

}

label1.Text = "Score is " + "\t" + score;

}

**OUTPUT:**



