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 \_99\_Final\_Game

{

public partial class Form1 : Form

{

int x1 = 536, xPerson = 32, xRock1 = 815, xRock2 = 722, xRock3 = 763, xRock4 = 910;

int y1 = 12, yPerson = 114, yRock1 = 103, yRock2 = 200, yRock3 = 297, yRock4 = 394;

int score = 0;

int round = 1;

int place;

bool upMove, downMove, leftMove, rightMove;

public bool quit = false;

public Form1()

{

InitializeComponent();

rock1.Location = new Point(xRock1, yRock1);

rock2.Location = new Point(xRock2, yRock2);

rock3.Location = new Point(xRock3, yRock3);

rock4.Location = new Point(xRock4, yRock4);

}

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

{

YouLoseForm loseForm = new YouLoseForm();

person.Location = new Point(xPerson, yPerson);

x1 -= (round + 1);

rock.Location = new Point(x1, y1);

xRock1 -= (round + 1);

rock1.Location = new Point(xRock1, yRock1);

xRock2 -= (round + 1);

rock2.Location = new Point(xRock2, yRock2);

xRock3 -= (round + 1);

rock3.Location = new Point(xRock3, yRock3);

xRock4 -= (round + 1);

rock4.Location = new Point(xRock4, yRock4);

Random spot = new Random();

place = spot.Next(536, 840);

if (x1 <= 0)

x1 = place;

place = spot.Next(536, 840);

if (xRock1 <= 0)

xRock1 = place;

place = spot.Next(536, 840);

if (xRock2 <= 0)

xRock2 = place;

place = spot.Next(536, 840);

if (xRock3 <= 0)

xRock3 = place;

place = spot.Next(536, 840);

if (xRock4 <= 0)

xRock4 = place;

//Checks if Person PictureBox is Touching a Rock

if (rock.Left <= person.Right && person.Left <= rock.Right &&

rock.Top <= person.Bottom && person.Top <= rock.Bottom)

{

timer1.Enabled = false;

timer2.Enabled = false;

loseForm.score = score;

loseForm.ShowDialog();

}

if (rock1.Left <= person.Right && person.Left <= rock1.Right &&

rock1.Top <= person.Bottom && person.Top <= rock1.Bottom)

{

timer1.Enabled = false;

timer2.Enabled = false;

loseForm.score = score;

loseForm.ShowDialog();

}

if (rock2.Left <= person.Right && person.Left <= rock2.Right &&

rock2.Top <= person.Bottom && person.Top <= rock2.Bottom)

{

timer1.Enabled = false;

timer2.Enabled = false;

loseForm.score = score;

loseForm.ShowDialog();

}

if (rock3.Left <= person.Right && person.Left <= rock3.Right &&

rock3.Top <= person.Bottom && person.Top <= rock3.Bottom)

{

timer1.Enabled = false;

timer2.Enabled = false;

loseForm.score = score;

loseForm.ShowDialog();

}

if (rock4.Left <= person.Right && person.Left <= rock4.Right &&

rock4.Top <= person.Bottom && person.Top <= rock4.Bottom)

{

timer1.Enabled = false;

timer2.Enabled = false;

loseForm.score = score;

loseForm.ShowDialog();

}

//Player Movement--person.\_\_\_\_ prevents player from leaving map

if (upMove && (person.Top >= 10))

yPerson -= (0 + (2 \* round));

if (downMove && (person.Bottom <= 475))

yPerson += (0 + (2 \* round));

if (leftMove && (person.Left >= 0))

xPerson -= (0 + (2 \* round));

if (rightMove && (person.Right <= 664))

xPerson += (0 + (2 \* round));

}

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

{

if(e.KeyCode == Keys.Up)

{

upMove = true;

}

if(e.KeyCode == Keys.Down)

{

downMove = true;

}

if (e.KeyCode == Keys.Left)

{

leftMove = true;

}

if (e.KeyCode == Keys.Right)

{

rightMove = true;

}

}

private void timer3\_Tick(object sender, EventArgs e)

{

}

private void timer2\_Tick(object sender, EventArgs e)

{

if(progressBar1.Value < progressBar1.Maximum)

{

score++;

progressBar1.Value++;

}

label1.Text = "Score: " + score.ToString();

if (progressBar1.Value == progressBar1.Maximum)

{

round++;

this.Text = "Round " + round.ToString();

progressBar1.Value = 0;

}

}

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

{

}

private void Form1\_KeyPress(object sender, KeyPressEventArgs e)

{

}

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

{

if (e.KeyCode == Keys.Up)

{

upMove = false;

}

if (e.KeyCode == Keys.Down)

{

downMove = false;

}

if (e.KeyCode == Keys.Left)

{

leftMove = false;

}

if (e.KeyCode == Keys.Right)

{

rightMove = false;

}

}

}

}