**Семинарска по предметот ВИЗУЕЛНО ПРОГРАМИРАЊЕ**

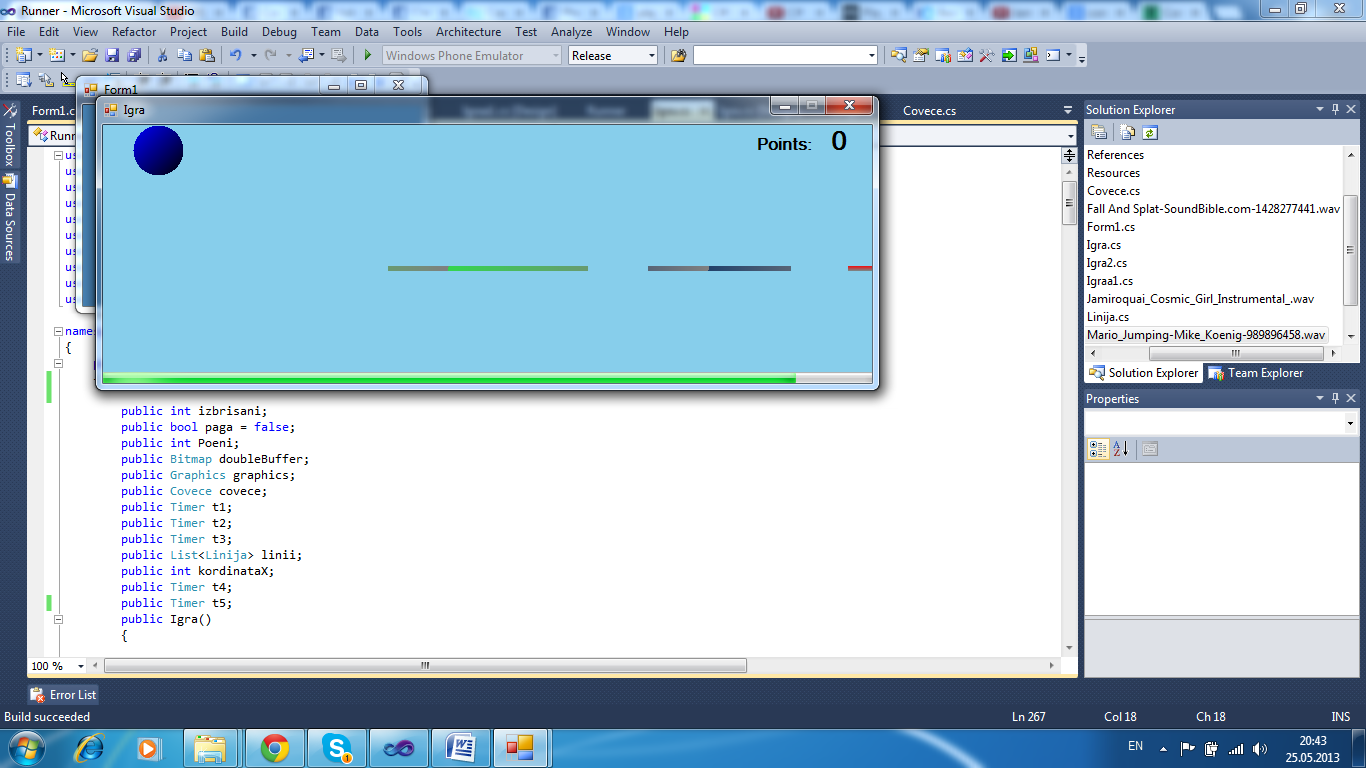
**Тема: Runner**

Професор: д-р Дејан Ѓорѓевиќ **Изработиле:**

Асистент: м-р Томче Делев **Александар Андонов 115004**

**Кристина Кочовска 115039**

**Краток опис на апликацијата:**

Runner е игра во која се појавуваат платформи на екранот, кои се всушност линии со различна должина и различна боја. Играчот преку една елипса скока од една на друга платформа и се обидува да не падне помеѓу платформите. На играчот му е дозволено единсвено притискање на копчето Space, со што елипсата скока. Играта се состои од 3 нивоа. Секое ниво трае по 10 секунди. Откако ќе истече времето на тајмерот,доколку играчот успеал да не падне од платформите се продолжува на следното ниво. При секое поминато ниво поените што ги освоил играчот на тоа ниво му се додаваат на неговите вкупни поени. Доколку топчето падне помеѓу платформите поените што ги освоил играчот во тоа ниво се рестартираат на 0 и повторно се започнува со играње на истото ниво доколку тоа го одбере играчот. Секој скок од една на друга платформа носи 3 поени. На почетокот на играта доколку играчот не притисне Space, топчето само ке падне по 5 секунди.

**Класна и податочна структура на апликацијата:**

Програмата е составена од три класи:

-Класа Player во која чуваме име и поени за секој играч.

-Класа Covece во која во конструкторот се чуваат кординатите x и y, visinata-променлива која ни покажува на која висина се движат линиите и до каде треба да падне топчето доколку падне на линија и променлива gore1 која ни ја означува висината до која треба да скокне топчето.

-Класа Linija во која во конструкторот се чиваат кординатите x и y, должината на линијата и специфичната боја на линијата.

При вклучување на играта потребно е играчот да внесе Nicname и да одбере ниво и при клик на Start се изминува целосната листа со играчи и се бара играчот кој се пријавил да игра. Доколку е пронајден неговите поени се ресетираат на 0, а доколку не е пронајден се креира нов играч и се додава во листата.

Секое ниво е претставено на посебен Windows Form.

Во тајмерот t1 се исцртуваат линиите и топчето. Тајмерот t2 овозможува скокање на топчето. Тајмерот t3 овозможува паѓање на топчето. Во тајмерот t4 се намалува вредноста на прогрес барот. Тајмерот t5 се повикува само кога топчето е во почетна положба и овозможува паѓање на топчето.

**Опис на целосна класа: Класата Covece**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Drawing;

using System.Drawing.Drawing2D;

namespace Runner

{

public class Covece

{

public int x { get; set; }

public int y { get; set; }

public bool dole = true;

public bool gore = false;

public int visina { get; set; }

public int gore1 { get; set; }

public Covece() { }

public Covece(int xx, int yy,int v,int g)

{

visina = v;

x = xx;

y = yy;

gore1 = g;

}

public void Draw(Graphics g)

{

LinearGradientBrush b = new LinearGradientBrush(new Point(x, y), new Point(x+50, y+50), Color.Blue, Color.Black);

g.FillEllipse(b, x, y, 50, 50);

}

public void Paga()

{

if (y != 380)

{

y += 10;

}

}

public void Skoka()

{

if (y == gore1)

{

gore = true;

dole = false;

}

if (y == visina)

{

dole = true;

gore = false;

}

if (dole == true)

{

y -= 10;

}

if (gore == true)

{

y += 10;

}

}

}

}

X,y се кординатите на елипсата со која игра играчот.

Променливите gore и dole ни означуваат дали топчето треба да се движи нагоре(да скока) или надоле(да паѓа).

Visina-променлива која ни означува на која висина се наогаат линиите и всушност до таму треба да стигне точето ако падне точно на линија.

Gore1-променлива која ни означува до која висина треба да скока топчето.

Во функцијата Paga():

Топчето ке се движи надоле се доде неговата y координата не е еднаква на 380.

Во функцијата Skoka():

Доколку топчето се наоѓа на највисоката можна точа(g1) тогаш gore=true и dole=false, односно топчето треба да се движи надоле. Доколку топчето се лизга на линијата, односно неговата y koordinata e еднаква на висина: dole=true и gore= false, односно топчето треба да се движи нагоре. Доколку dole==true го движиме топчето надоле со намалување на неговата y koordinata za 10, а пак во спротивен случај тоа се движи нагоре за вредност 10.

**Опис на функција:**

Следнава функција Dolzina() е употребена на последното ниво.

public void Dolzina()

{

if (povikano == false)

{

Random r = new Random();

int broj = r.Next(0, 2);

povikano = true;

if (broj == 0)

{

namaluvaj = true;

}

else

{

zgolemuvaj = true;

}

}

else

{

if (namaluvaj == true)

{

dolzina -= 3;

if (dolzina < 60)

{

namaluvaj = false;

}

}

if (namaluvaj == false)

{

dolzina += 3;

if (dolzina > 130)

{

namaluvaj = true;

}

}

}

}

Оваа функција овозможува визуелно зголемување или намалување на линиите додека се движат. Тоа е постигнато на следниов начин:

Имаме една public променлива povikano која на почетокот од програмата е false. Таа ни покажува дали за одредена линија е одредено намалување или зголемување. Доколку

povikano е false се бара рандом бројче од 0 до 1. Доколку бројката која ке ја добиеме рандом е 0 променливата namaluvaj која е public и на почеток false и поставуваме вредност true, а пак доколку рандом бројката е 1 на променливата зголемувај и поставуваме вредност true. Доколку namaluvaj има вредност true ја намалуваме должината на линијата се додека нејзината должина не е помала од 60, а кога таа ке ја достигне таа должина променливата namaluvaj добива вредност true со цел при следното повикување на функцијава сега таа да започне да се зголемува и да не ја зачува малата должина. Доколку пак namaluvaj е false ја зголемуваме должината на линијата се до 130 кога ја поставуваме namaluvaj true со цел следниот пат кога ке се повика функцијата таа линија да започне да се намалува.