COLEGIUL NAŢIONAL „ALEXANDRU IOAN CUZA” CORABIA

Calatorie in timp si spatiu - joc

LUCRARE PENTRU ATESTAREA

COMPETENŢELOR PROFESIONALE

Elev: Craciun Alin-Ionut

Clasa: XII A

Profesor coordonator: Stanescu Elida

2015

**Motivatia alegerii temei:**

Am ales acest subiect pentu divertisment dar si pentru ca jocurile fac parte din viaţa tinerilor, fiind un domeniu in plina dezvoltare.

Consider ca trebuie sa facem de la inceput diferenta intre categoriile de jocuri pe care le gasim pe piata. Exista jocuri educative, care dezvolta puterea de gandire a copiilor dar si jocuri care ii consuma atat de tare nervos pe copii, adolescenti si tineri, incat acestia devin mult mai irascibili, sunt greu de induplecat si sunt foarte nerabdatori.

Pentru realizarea jocului am ales Visual Basic (VB), un limbaj de programare produs de Microsoft, care a fost derivat din limbajul Basic. VB este popular datorită interfeței grafice pe care o folosește, interfață relativ simplă față de cea a altor limbaje.

**Cerinte hardware si software:**

**Hardware:**

Minimum:

-1.6 GHz processor

-1024 MB RAM

-5400 RPM hard-disk drive

-DirectX 9

**Software:**

-Microsoft Visual C# 2010 Express Edition

**Scurt Istoric al limbajului C#**

In contextul societatii actuale, caracterizata printr-o explozie informationala fara precedent in istoria omenirii, sistemele informatice reprezinta unul din elementele fundamentale care genereaza si controleaza fluxurile informationale la nivel micro si macroeconomic.

Limbajul C# fost dezvoltat de o echipă restrânsă de ingineri de la Microsoft, echipă din care s-a evidenţiat Anders Hejlsberg (autorul limbajului Turbo Pascal şi membru al echipei care a proiectat Borland Delphi). C# este un limbaj simplu, cu circa 80 de cuvinte cheie, şi 12 tipuri de date predefinite. El permite programarea structurată, modulară şi orientată obiectual, conform perceptelor moderne ale programării profesioniste. Principiile de bază ale programării pe obiecte (INCAPSULARE, MOSTENIRE, POLIMORFISM) sunt elemente fundamentale ale programării C#. În mare, limbajul moşteneşte sintaxa şi principiile de programare din C++. Sunt o serie de tipuri noi de date sau funcţiuni diferite ale datelor din C++, iar în spiritul realizării unor secvenţe de cod sigure (safe), unele funcţiuni au fost adăugate (de exemplu, interfeţe şi delegări), diversificate (tipul struct), modificate (tipul string) sau chiar eliminate (moştenirea multiplă şi pointerii către funcţii). Unele funcţiuni (cum ar fi accesul direct la memorie folosind pointeri) au fost păstrate, dar secvenţele de cod corespunzătoare se consideră ”nesigure”.

**Mediul de dezvoltare Visual C#**

Componenta centrala a limbajului Visual C# este mediul Developer Studio, adica mediul de dezvoltare integrata (IDE). Mediul Studio Developer este folosit pentru integrarea instrumentelor de dezvoltare impreuna cu compilatorul Visual C#.

Mediul de dezvoltare Microsoft Visual C# dispune de instrumente specializate de proiectare, ceea ce permite crearea aplicaţiilor în mod interactiv, rapid şi uşor. Pentru a construi o aplicaţie Windows **(File New Project)** se selectează ca template **Windows Application**. O aplicaţie Windows conţine cel puţin o fereastră (**Form**) în care se poate crea o interfaţă cu utilizatorul aplicaţiei. Componentele vizuale ale aplicaţiei pot fi prelucrate în modul **Designer (Shift+F7)** pentru a plasa noi obiecte, a le stabili proprietăţile etc.

O fereastră poate fi activată cu **form.Show()** sau cu **form.ShowDialog()**, metoda a doua permiţând ca revenirea în fereastra din care a fost activat noul formular să se facă numai după ce noul formular a fost inchis (spunem că formularul nou este deschis modal). Un propietar este o fereastră care contribuie la comportarea formularului deţinut. Activarea propietarului unui formular deschis modal va determina activarea formularului deschis modal. Când un nou formular este activat folosind **form.Show()** nu va avea nici un deţinător, acesta stabilindu-se direct :

**public Form Owner { get; set; }**

**F\_nou form=new F\_nou(); form.Owner = this; form.Show();**

Formularul deschis modal va avea un proprietar setat pe null. Deţinătorul se poate stabili setând proprietarul înainte să apelăm **Form.ShowDialog()** sau apelând **From.ShowDialog()** cu proprietarul ca argument.

**F\_nou form = new F\_nou();form.ShowDialog(this);**

Vizibilitatea unui formular poate fi setată folosind metodele **Hide** sau **Show**. Pentru a ascunde un formular putem folosi :

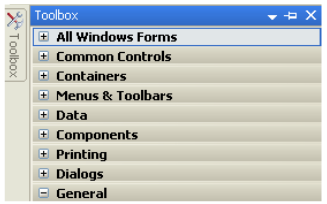
**this.Hide();** // setarea propietatii Visible indirect

sau

**this.Visible = false;** // setarea propietatii Visible direct

Codul ”din spatele” unei componente vizuale este accesibil în modul **Code (F7)**.

În fereastra **Solution Explorer** sunt afişate toate fişierele pe care Visual Studio.NET le-a inclus în proiect. **Form1.cs** este formularul creat implicit de Visual Studio.NET ca parte a proiectului. Fereastra **Properties** este utilizată pentru a vizualiza şi eventual schimba proprietăţile obiectelor. **Toolbox** conţine controale standard **drag-and-drop** şi componente utilizate în crearea aplicaţiei Windows. Controalele sunt grupate în categoriile logice din imaginea alăturată. **Designer, Code, Solution Explorer** şi celelalte se află grupate în meniul **View**. La crearea unei noi aplicaţii vizuale, Visual Studio.NET generează un spaţiu de nume ce conţine clasa statică **Program**, cu metoda statică ce constituie punctul de intrare (de lansare) a aplicaţiei:



Definirea unei funcţii de tratare a unui eveniment asociat controlului se realizează prin selectarea grupului **Events** din ferestra **Properties** a controlului respectiv şi alegerea evenimentului dorit. Dacă nu scriem nici un nume pentru funcţia de tratare, ci efectuăm dublu click în căsuţa respectivă, se generează automat un nume pentru această funcţie, ţinând cont de numele controlului şi de numele evenimentului (de exemplu button1\_Click).

Dacă în **Designer** efectuăm dublu click pe un control, se va genera automat o funcţie de tratare pentru evenimentul implicit asociat controlului (pentru un buton evenimentul implicit este **Click**, pentru **TextBox** este **TextChanged**, pentru un formular **Load** etc.).

Printre evenimentele cele mai des utilizate, se numără :

•**Load**  apare când formularul este pentru prima data încărcat în memorie.

•**FormClosed** apare când formularul este închis.

•**FormClosing** apare când formularul se va inchide ca rezultat al acţiunii utilizatorului asupra butonului **Close** (Dacă se setează **CancelEventArgs.Cancel =True** atunci se va opri închiderea formularului).

•**Activated** apare pentru formularul activ.

•**Deactivate** apare atunci când utilizatorul va da click pe alt formular al aplicatiei.

**Controale**

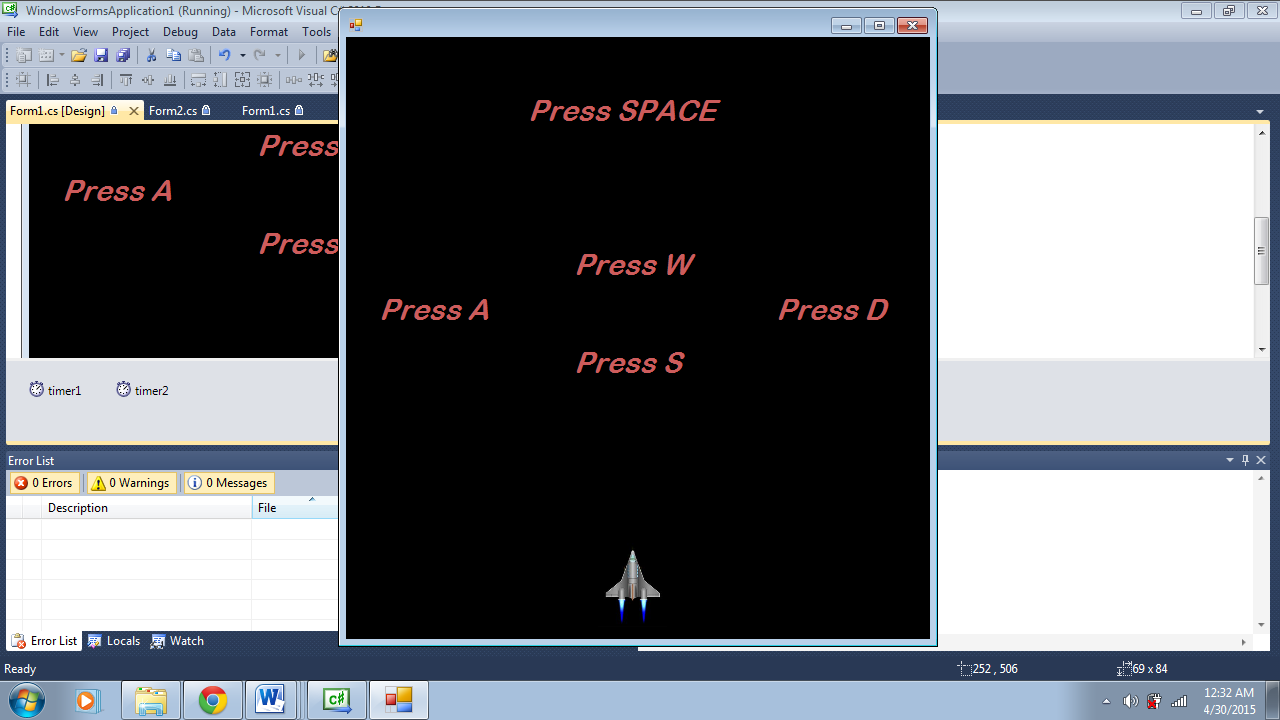
Unitatea de bază a unei interfeţe Windows o reprezintă un control. Acesta poate fi „găzduit” de un container ce poate fi un formular sau un alt control. Un control este o instanţă a unei clase derivate din System.Windows.Forms şi este reponsabil cu desenarea unei părţi din container. Visual Studio .NET vine cu o serie de controale standard, disponibile în Toolbox. Aceste controale pot fi grupate astfel:

•**Controale de actiune** (de exemplu **button**) care, atunci când sunt acţionate, se poate executa o prelucrare. De exemplu, cel mai important eveniment pentru **Button** este **Click** (desemnând acţiunea click stânga pe buton).

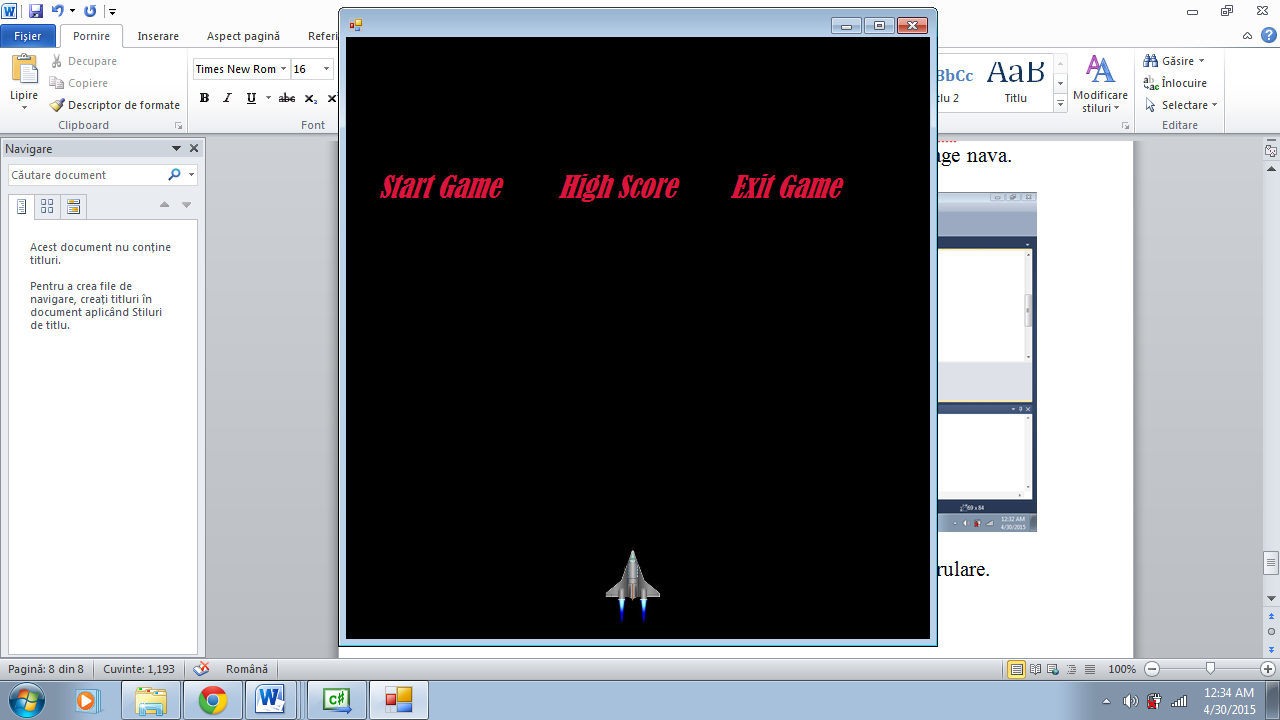
**Detalii tehnice de implementare:**

„Calatorie in timp si spatiu” este un joc de tip endurace. Acesta este cu o nava pe care o poti misca dupa tastele „A”, ”S”, ”D” si „W” si poti trage cu tasta „Space”. Este un singur tip de nivel care devine din ce in ce mai dificil, navele inamice tragand din ce in ce mai repede.

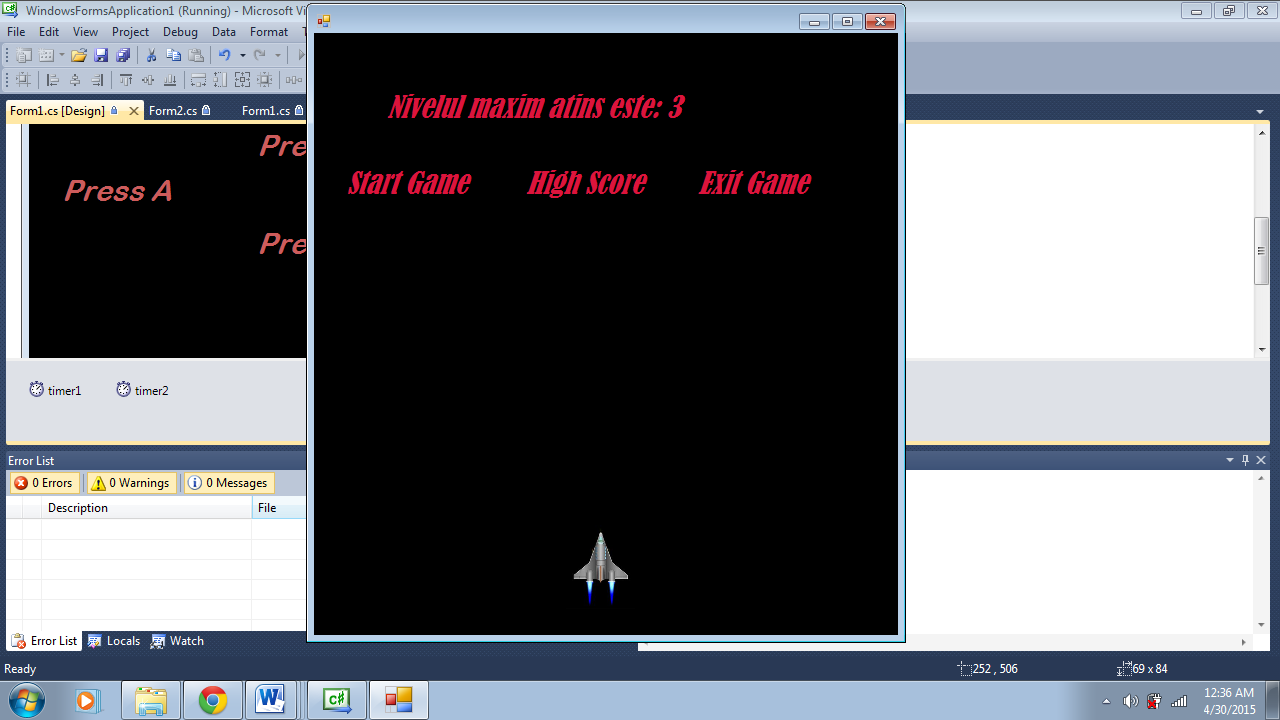
Pentru inceput am facut o fereastra cu un mini tutorial care dupa ce este completat va pune la dispozitie optiunile de incepere , iesire sau de vizualizare al celui mai mare scor de pana atunci.Pentru asta am avut nevoie initial de 5 label-uri pentru tutorial urmate de alte 4 pentru optiunile de mai sus, de un picture box , de un button declarat in codul de program care este defapt acel „laser” cu care trage nava si de 2 timers.



Acesta este primul lucru care ne apare pe ecran dupa rulare. Dupa ce apasam tastele mentionate o sa apara asta:



Daca tragem asupra label-ului cu textul Exit Game aplicatia va fi inchisa, daca tragem asupra celui cu High Score va mai aparea inca un label astfel:



Iar pentru Start Game se va deschide o noua fereastra cu jocul propriu-zis.

Pentru a reusi lansa o actiune cu acel „laser” am avut nevoie de a verifica in timp real locatia butonului iar eu am facut asta cu urmatorul cod introdus in timer2 care are intervalul setat la 1:

if (label7.Visible == true && nr > 1)//nr retine numarul tragerilor navei

{

if (fire.Location.Y < label7.Location.Y + label7.Height && fire.Location.Y > label7.Location.Y && fire.Location.X < label7.Location.X + label7.Width && label7.Location.X < fire.Location.X)

{

this.Close();

}

if (fire.Location.Y < label6.Location.Y + label6.Height && fire.Location.Y > label6.Location.Y && fire.Location.X < label6.Location.X + label6.Width && label6.Location.X < fire.Location.X)

{

this.Hide();

f.Show();

}

if (fire.Location.Y < label8.Location.Y + label8.Height && fire.Location.Y > label8.Location.Y && fire.Location.X < label8.Location.X + label8.Width && label8.Location.X < fire.Location.X)

{

string scor;

scor = System.IO.File.ReadAllText(@"C:\atestat\scor.txt");

label9.Visible = true;

label9.Text = "Nivelul maxim atins este: " + scor;

}

}

Dupa cum puteti vedea am folosit un fisier pentru a retine cel mai mare scor astfel incat acesta sa ramana acelasi indeferent daca parasim aplicatia.

Pentru a pune „laser-ul” in miscare m-am folosit de timer1 cu intervalul setat la 15 astfel:

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

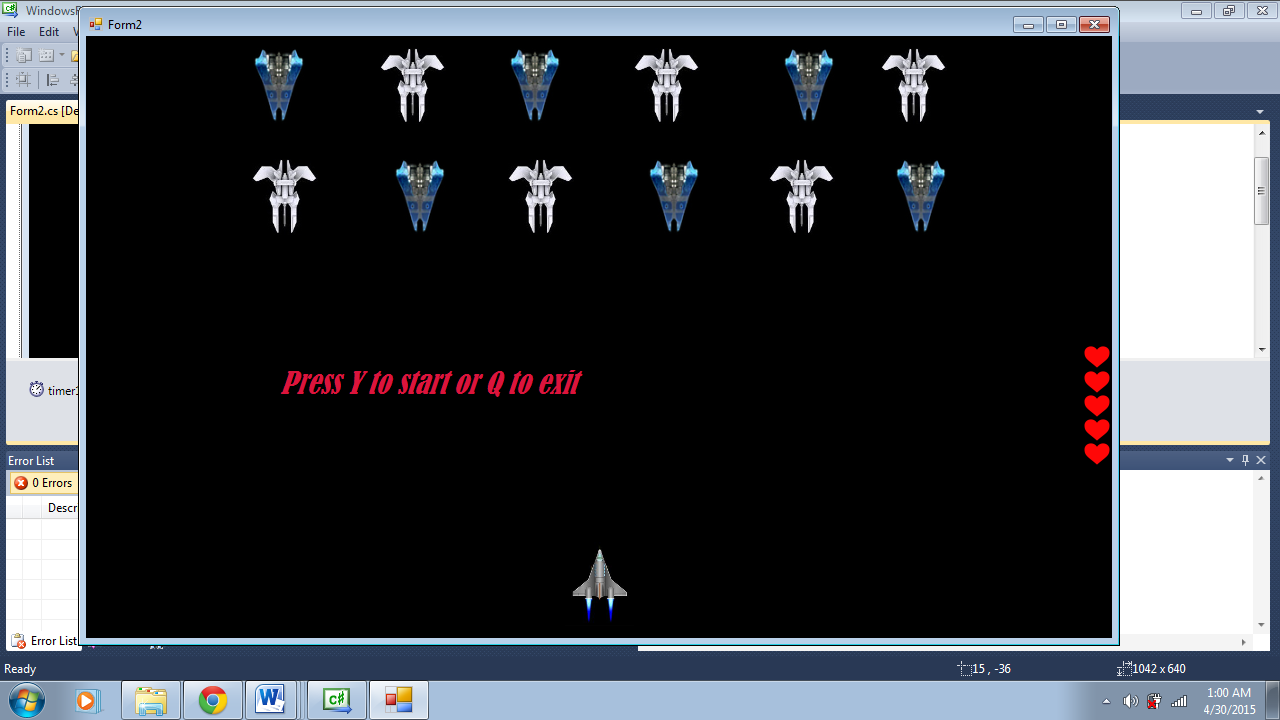
{

fire.Location = new Point(fire.Location.X, fire.Location.Y - 15);

}

//Fire este butonul care reprezinta laser-ul

Pentru jocul propriu-zis, care se declanseaza la tragerea in eticheta Start Game, m-am folosit de 19 picturebox-uri, 6 timers, un label si de 13 butoane declarate in codul de program.



De timer1 ma folosesc pentru a verifica daca mai sunt in „viata” navele inamice si daca sunt si le nimeresc cu laser-ul sa „moara”:

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

{

fire.Location = new Point(fire.Location.X, fire.Location.Y - 15);

if ( fire.Location.Y < pictureBox6.Location.Y + pictureBox6.Height && fire.Location.Y > pictureBox6.Location.Y && fire.Location.X < pictureBox6.Location.X + pictureBox6.Width && pictureBox6.Location.X < fire.Location.X)

{ if(pictureBox6.Visible==true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox6.Visible = false; }

if (fire.Location.Y < pictureBox7.Location.Y + pictureBox7.Height && fire.Location.Y > pictureBox7.Location.Y && fire.Location.X < pictureBox7.Location.X + pictureBox7.Width && pictureBox7.Location.X < fire.Location.X)

{if(pictureBox7.Visible==true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox7.Visible = false; }

if (fire.Location.Y < pictureBox8.Location.Y + pictureBox8.Height && fire.Location.Y > pictureBox8.Location.Y && fire.Location.X < pictureBox8.Location.X + pictureBox8.Width && pictureBox8.Location.X < fire.Location.X)

{

if (pictureBox8.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox8.Visible = false; }

if (fire.Location.Y < pictureBox9.Location.Y + pictureBox9.Height && fire.Location.Y > pictureBox9.Location.Y && fire.Location.X < pictureBox9.Location.X + pictureBox9.Width && pictureBox9.Location.X < fire.Location.X)

{

if (pictureBox9.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox9.Visible = false; }

if (fire.Location.Y < pictureBox10.Location.Y + pictureBox10.Height && fire.Location.Y > pictureBox10.Location.Y && fire.Location.X < pictureBox10.Location.X + pictureBox10.Width && pictureBox10.Location.X < fire.Location.X)

{

if (pictureBox10.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox10.Visible = false; }

if (fire.Location.Y < pictureBox11.Location.Y + pictureBox11.Height && fire.Location.Y > pictureBox11.Location.Y && fire.Location.X < pictureBox11.Location.X + pictureBox11.Width && pictureBox11.Location.X < fire.Location.X)

{

if (pictureBox11.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox11.Visible = false; }

if (fire.Location.Y < pictureBox12.Location.Y + pictureBox12.Height && fire.Location.Y > pictureBox12.Location.Y && fire.Location.X < pictureBox12.Location.X + pictureBox12.Width && pictureBox12.Location.X < fire.Location.X)

{

if (pictureBox12.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox12.Visible = false; }

if (fire.Location.Y < pictureBox13.Location.Y + pictureBox13.Height && fire.Location.Y > pictureBox13.Location.Y && fire.Location.X < pictureBox13.Location.X + pictureBox13.Width && pictureBox13.Location.X < fire.Location.X)

{

if (pictureBox13.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox13.Visible = false; }

if (fire.Location.Y < pictureBox14.Location.Y + pictureBox14.Height && fire.Location.Y > pictureBox14.Location.Y && fire.Location.X < pictureBox14.Location.X + pictureBox14.Width && pictureBox14.Location.X < fire.Location.X)

{

if (pictureBox14.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox14.Visible = false; }

if (fire.Location.Y < pictureBox15.Location.Y + pictureBox15.Height && fire.Location.Y > pictureBox15.Location.Y && fire.Location.X < pictureBox15.Location.X + pictureBox15.Width && pictureBox15.Location.X < fire.Location.X)

{

if (pictureBox15.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox15.Visible = false; }

if (fire.Location.Y < pictureBox16.Location.Y + pictureBox16.Height && fire.Location.Y > pictureBox16.Location.Y && fire.Location.X < pictureBox16.Location.X + pictureBox16.Width && pictureBox16.Location.X < fire.Location.X)

{

if (pictureBox16.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox16.Visible = false; }

if (fire.Location.Y < pictureBox17.Location.Y + pictureBox17.Height && fire.Location.Y > pictureBox17.Location.Y && fire.Location.X < pictureBox17.Location.X + pictureBox17.Width && pictureBox17.Location.X < fire.Location.X)

{

if (pictureBox17.Visible == true)

fire.Location = new Point(fire.Location.X, -150);

pictureBox17.Visible = false; }

}

Cu timer2 misc randul de sus de nave de-a lungul form-ului incepand sa plece spre partea stanga:

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

{

if (pictureBox6.Location.X > 0 && nr%2==0)

{

if (pictureBox6.Location.X - 5 <= 0)

nr = 3;

pictureBox6.Location = new Point(pictureBox6.Location.X - 5, pictureBox6.Location.Y);

pictureBox10.Location = new Point(pictureBox10.Location.X - 5, pictureBox10.Location.Y);

pictureBox8.Location = new Point(pictureBox8.Location.X - 5, pictureBox8.Location.Y);

pictureBox7.Location = new Point(pictureBox7.Location.X - 5, pictureBox7.Location.Y);

pictureBox14.Location = new Point(pictureBox14.Location.X - 5, pictureBox14.Location.Y);

pictureBox17.Location = new Point(pictureBox17.Location.X - 5, pictureBox17.Location.Y);

}

if (pictureBox7.Location.X + pictureBox7.Width < ClientSize.Width && nr%2!=0)

{

if (pictureBox7.Location.X + 5+pictureBox7.Width >= ClientSize.Width)

nr=2;

pictureBox6.Location = new Point(pictureBox6.Location.X + 5, pictureBox6.Location.Y);

pictureBox10.Location = new Point(pictureBox10.Location.X + 5, pictureBox10.Location.Y);

pictureBox8.Location = new Point(pictureBox8.Location.X + 5, pictureBox8.Location.Y);

pictureBox7.Location = new Point(pictureBox7.Location.X + 5, pictureBox7.Location.Y);

pictureBox14.Location = new Point(pictureBox14.Location.X + 5, pictureBox14.Location.Y);

pictureBox17.Location = new Point(pictureBox17.Location.X + 5, pictureBox17.Location.Y);

}

}

Iar de timer3 pentru a misca randul de jos de nave incepand spre partea dreapta:

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

{

if (pictureBox16.Location.X + pictureBox16.Width < ClientSize.Width && nr1 % 2 != 0)

{

if (pictureBox16.Location.X + 5 + pictureBox16.Width >=ClientSize.Width)

nr1 = 2;

pictureBox16.Location = new Point(pictureBox16.Location.X + 5, pictureBox16.Location.Y);

pictureBox15.Location = new Point(pictureBox15.Location.X + 5, pictureBox15.Location.Y);

pictureBox11.Location = new Point(pictureBox11.Location.X + 5, pictureBox11.Location.Y);

pictureBox9.Location = new Point(pictureBox9.Location.X + 5, pictureBox9.Location.Y);

pictureBox12.Location = new Point(pictureBox12.Location.X + 5, pictureBox12.Location.Y);

pictureBox13.Location = new Point(pictureBox13.Location.X + 5, pictureBox13.Location.Y);

}

if (pictureBox12.Location.X>0 && nr1 % 2 == 0)

{

if (pictureBox12.Location.X -5 <= 0)

nr1 = 3;

pictureBox16.Location = new Point(pictureBox16.Location.X - 5, pictureBox16.Location.Y);

pictureBox15.Location = new Point(pictureBox15.Location.X - 5, pictureBox15.Location.Y);

pictureBox11.Location = new Point(pictureBox11.Location.X - 5, pictureBox11.Location.Y);

pictureBox9.Location = new Point(pictureBox9.Location.X - 5, pictureBox9.Location.Y);

pictureBox12.Location = new Point(pictureBox12.Location.X - 5, pictureBox12.Location.Y);

pictureBox13.Location = new Point(pictureBox13.Location.X - 5, pictureBox13.Location.Y);

}

}

Cu timer4 fac navele care mai sunt in viata sa traga:

private void timer4\_Tick(object sender, EventArgs e)

{

if ((b6.Location.Y > ClientSize.Height && pictureBox6.Visible == true) || (b7.Location.Y > ClientSize.Height && pictureBox7.Visible == true) || (b8.Location.Y > ClientSize.Height && pictureBox8.Visible == true) || (b10.Location.Y > ClientSize.Height && pictureBox10.Visible == true) || (b14.Location.Y > ClientSize.Height && pictureBox14.Visible == true) || (b17.Location.Y > ClientSize.Height && pictureBox17.Visible == true))

{

if (pictureBox6.Visible == true)

b6.Location = new Point(pictureBox6.Location.X + pictureBox6.Width / 2, pictureBox6.Location.Y + 22 + pictureBox6.Height);

if (pictureBox7.Visible == true)

b7.Location = new Point(pictureBox7.Location.X + pictureBox7.Width / 2, pictureBox7.Location.Y + 22 + pictureBox6.Height);

if (pictureBox8.Visible == true)

b8.Location = new Point(pictureBox8.Location.X + pictureBox8.Width / 2, pictureBox8.Location.Y + 22 + pictureBox6.Height);

if (pictureBox10.Visible == true)

b10.Location = new Point(pictureBox10.Location.X + pictureBox10.Width / 2, pictureBox10.Location.Y + 22 + pictureBox6.Height);

if (pictureBox14.Visible == true)

b14.Location = new Point(pictureBox14.Location.X + pictureBox14.Width / 2, pictureBox14.Location.Y + 22 + pictureBox6.Height);

if (pictureBox17.Visible == true)

b17.Location = new Point(pictureBox17.Location.X + pictureBox17.Width / 2, pictureBox17.Location.Y + 22 + pictureBox6.Height);

Controls.Add(b6); Controls.Add(b7); Controls.Add(b8); Controls.Add(b10); Controls.Add(b14); Controls.Add(b17);

}

b6.Location = new Point(b6.Location.X, b6.Location.Y + viteza);

if (b6.Location.Y < nava.Location.Y + nava.Height && b6.Location.Y > nava.Location.Y && b6.Location.X < nava.Location.X + nava.Width && nava.Location.X < b6.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b6.Location = new Point(-20, b6.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b6.Location = new Point(-20, b6.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b6.Location = new Point(-20, b6.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b6.Location = new Point(-20, b6.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b6.Location = new Point(-20, b6.Location.Y); }

b7.Location = new Point(b7.Location.X, b7.Location.Y + viteza);

if (b7.Location.Y < nava.Location.Y + nava.Height && b7.Location.Y > nava.Location.Y && b7.Location.X < nava.Location.X + nava.Width && nava.Location.X < b7.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b7.Location = new Point(-20, b7.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b7.Location = new Point(-20, b7.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b7.Location = new Point(-20, b7.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b7.Location = new Point(-20, b7.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b7.Location = new Point(-20, b7.Location.Y); }

b8.Location = new Point(b8.Location.X, b8.Location.Y + viteza);

if (b8.Location.Y < nava.Location.Y + nava.Height && b8.Location.Y > nava.Location.Y && b8.Location.X < nava.Location.X + nava.Width && nava.Location.X < b8.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b8.Location = new Point(-20, b8.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b8.Location = new Point(-20, b8.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b8.Location = new Point(-20, b8.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b8.Location = new Point(-20, b8.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b8.Location = new Point(-20, b8.Location.Y); }

b10.Location = new Point(b10.Location.X, b10.Location.Y + viteza);

if (b10.Location.Y < nava.Location.Y + nava.Height && b10.Location.Y > nava.Location.Y && b10.Location.X < nava.Location.X + nava.Width && nava.Location.X < b10.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b10.Location = new Point(-20, b10.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b10.Location = new Point(-20, b10.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b10.Location = new Point(-20, b10.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b10.Location = new Point(-20, b10.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b10.Location = new Point(-20, b10.Location.Y); }

b14.Location = new Point(b14.Location.X, b14.Location.Y + viteza);

if (b14.Location.Y < nava.Location.Y + nava.Height && b14.Location.Y > nava.Location.Y && b14.Location.X < nava.Location.X + nava.Width && nava.Location.X < b14.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b14.Location = new Point(-20, b14.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b14.Location = new Point(-20, b14.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b14.Location = new Point(-20, b14.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b14.Location = new Point(-20, b14.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b14.Location = new Point(-20, b14.Location.Y); }

b17.Location = new Point(b17.Location.X, b17.Location.Y + viteza);

if (b17.Location.Y < nava.Location.Y + nava.Height && b17.Location.Y > nava.Location.Y && b17.Location.X < nava.Location.X + nava.Width && nava.Location.X < b17.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b17.Location = new Point(-20, b17.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b17.Location = new Point(-20, b17.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b17.Location = new Point(-20, b17.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b17.Location = new Point(-20, b17.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b17.Location = new Point(-20, b17.Location.Y); }

if ((b9.Location.Y > ClientSize.Height && pictureBox9.Visible == true) || (b11.Location.Y > ClientSize.Height && pictureBox11.Visible == true) || (b12.Location.Y > ClientSize.Height && pictureBox12.Visible == true) || (b13.Location.Y > ClientSize.Height && pictureBox13.Visible == true) || (b15.Location.Y > ClientSize.Height && pictureBox15.Visible == true) || (b16.Location.Y > ClientSize.Height && pictureBox16.Visible == true))

{

if (pictureBox9.Visible == true)

b9.Location = new Point(pictureBox9.Location.X + pictureBox9.Width / 2, pictureBox9.Location.Y + 22 + pictureBox9.Height);

if (pictureBox11.Visible == true)

b11.Location = new Point(pictureBox11.Location.X + pictureBox11.Width / 2, pictureBox11.Location.Y + 22 + pictureBox11.Height);

if (pictureBox12.Visible == true)

b12.Location = new Point(pictureBox12.Location.X + pictureBox12.Width / 2, pictureBox12.Location.Y + 22 + pictureBox12.Height);

if (pictureBox13.Visible == true)

b13.Location = new Point(pictureBox13.Location.X + pictureBox13.Width / 2, pictureBox13.Location.Y + 22 + pictureBox13.Height);

if (pictureBox15.Visible == true)

b15.Location = new Point(pictureBox15.Location.X + pictureBox15.Width / 2, pictureBox15.Location.Y + 22 + pictureBox15.Height);

if (pictureBox16.Visible == true)

b16.Location = new Point(pictureBox16.Location.X + pictureBox16.Width / 2, pictureBox16.Location.Y + 22 + pictureBox16.Height);

Controls.Add(b16); Controls.Add(b15); Controls.Add(b13); Controls.Add(b11); Controls.Add(b12); Controls.Add(b9);

}

b9.Location = new Point(b9.Location.X, b9.Location.Y + viteza);

if (b9.Location.Y < nava.Location.Y + nava.Height && b9.Location.Y > nava.Location.Y && b9.Location.X < nava.Location.X + nava.Width && nava.Location.X < b9.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b9.Location = new Point(-20, b9.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b9.Location = new Point(-20, b9.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b9.Location = new Point(-20, b9.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b9.Location = new Point(-20, b9.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b9.Location = new Point(-20, b9.Location.Y); }

b11.Location = new Point(b11.Location.X, b11.Location.Y + viteza);

if (b11.Location.Y < nava.Location.Y + nava.Height && b11.Location.Y > nava.Location.Y && b11.Location.X < nava.Location.X + nava.Width && nava.Location.X < b11.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b11.Location = new Point(-20, b11.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b11.Location = new Point(-20, b11.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b11.Location = new Point(-20, b11.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b11.Location = new Point(-20, b11.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b11.Location = new Point(-20, b11.Location.Y); }

b12.Location = new Point(b12.Location.X, b12.Location.Y + viteza);

if (b12.Location.Y < nava.Location.Y + nava.Height && b12.Location.Y > nava.Location.Y && b12.Location.X < nava.Location.X + nava.Width && nava.Location.X < b12.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b12.Location = new Point(-20, b12.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b12.Location = new Point(-20, b12.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b12.Location = new Point(-20, b12.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b12.Location = new Point(-20, b12.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b12.Location = new Point(-20, b12.Location.Y); }

b13.Location = new Point(b13.Location.X, b13.Location.Y + viteza);

if (b13.Location.Y < nava.Location.Y + nava.Height && b13.Location.Y > nava.Location.Y && b13.Location.X < nava.Location.X + nava.Width && nava.Location.X < b13.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b13.Location = new Point(-20, b13.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b13.Location = new Point(-20, b13.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b13.Location = new Point(-20, b13.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b13.Location = new Point(-20, b13.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b13.Location = new Point(-20, b13.Location.Y); }

b15.Location = new Point(b15.Location.X, b15.Location.Y + viteza);

if (b15.Location.Y < nava.Location.Y + nava.Height && b15.Location.Y > nava.Location.Y && b15.Location.X < nava.Location.X + nava.Width && nava.Location.X < b15.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b15.Location = new Point(-20, b15.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b15.Location = new Point(-20, b15.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b15.Location = new Point(-20, b15.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b15.Location = new Point(-20, b15.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b15.Location = new Point(-20, b15.Location.Y); }

b16.Location = new Point(b16.Location.X, b16.Location.Y + viteza);

if (b16.Location.Y < nava.Location.Y + nava.Height && b16.Location.Y > nava.Location.Y && b16.Location.X < nava.Location.X + nava.Width && nava.Location.X < b16.Location.X)

if (pictureBox5.Visible == true)

{ pictureBox5.Visible = false; b16.Location = new Point(-20, b16.Location.Y); }

else if (pictureBox4.Visible == true)

{ pictureBox4.Visible = false; b16.Location = new Point(-20, b16.Location.Y); }

else if (pictureBox3.Visible == true)

{ pictureBox3.Visible = false; b16.Location = new Point(-20, b16.Location.Y); }

else if (pictureBox2.Visible == true)

{ pictureBox2.Visible = false; b16.Location = new Point(-20, b16.Location.Y); }

else if (pictureBox1.Visible == true)

{ pictureBox1.Visible = false; b16.Location = new Point(-20, b16.Location.Y); }

}

Cu timer 5 verific daca au fost distruse toate navele inamice sau daca a fost distrus eroul. Daca au fost distruse toate navele reinitializez jocul si le maresc viteza de tragere navelor inamice, iar daca eroul a fost distrus verific daca este un nou record si daca este sa-l salveze in fisier apoi inchid aplicatia.

private void timer5\_Tick(object sender, EventArgs e)

{

if (pictureBox1.Visible == false && pictureBox2.Visible == false && pictureBox3.Visible == false && pictureBox4.Visible == false && pictureBox5.Visible == false)

{

timer5.Enabled = false;

System.IO.File.WriteAllText(@"C:\atestat\scor.txt", (viteza - 4).ToString());

MessageBox.Show("Game Over! Ati murit la nivelul "+(viteza-3).ToString(),"Game Over",MessageBoxButtons.OK,MessageBoxIcon.Asterisk);

Application.Exit();

}

if (pictureBox6.Visible == false && pictureBox7.Visible == false && pictureBox8.Visible == false && pictureBox9.Visible == false && pictureBox10.Visible == false && pictureBox11.Visible == false && pictureBox12.Visible == false && pictureBox13.Visible == false && pictureBox14.Visible == false && pictureBox15.Visible == false && pictureBox16.Visible == false && pictureBox17.Visible == false)

{

timer5.Enabled = false;

label6.Text = "Felicitari ati terminat nivelul " + (viteza-3).ToString()+ " press Y to continue or q to exit";

label6.Visible = true;

viteza++;

pictureBox6.Visible = true;

pictureBox7.Visible = true;

pictureBox8.Visible = true;

pictureBox9.Visible = true;

pictureBox10.Visible = true;

pictureBox11.Visible = true;

pictureBox12.Visible = true;

pictureBox13.Visible = true;

pictureBox14.Visible = true;

pictureBox15.Visible = true;

pictureBox16.Visible = true;

pictureBox17.Visible = true;

timer2.Enabled = false;

timer3.Enabled = false;

timer4.Enabled = false;

}

}

Iar cu timer6 tin focus-ul doar pe nava principala:

private void timer6\_Tick(object sender, EventArgs e)

{

this.Focus();

}

Acesta este programul in mare parte dar mai sunt multe lucruri de spus.

**Modalitati de dezvoltare:**

Aplicatia constituie un instrument de divertisment, si poate fi imbunatatita in mai multe feluri. Putem realiza o interfata grafica mult mai atractiva, putem implementa teme (diferite culori).

Se pot adauga sunete, inamici noi, nivele noi. Poate fi facut mai optimal.

Sa mai poate adauga o nava de tip erou si sa devina jocul multyplayer.

Sa faca posibila autentificarea ca un jucator sa ca un administrator.

**Aplicabilitate practica**

Aplicaţia poate fi utilizată de oricine doreşte să-şi petreacă timpul ȋntr-un mod plăcut, el fiind si o provocare pentru realizarea unui punctaj ridicat.

Jocul este unul simplu, usor de inteles . Scopul acestuia este de a-ti testa atentia si rapiditatea , incercand sa realizezi un scor cat mai mare.

**Bibliografie**

-Cartea „Curs de C# Programare in Visual C# 2008 Express Edition” de Constantin Galatan si Susana Galatan

-https://msdn.microsoft.com/

**Cuprins:**

Coperta pagina ……………………………………………………….1

Motivatia alegerii temei …………………………………...................2

Cerinte hardware si software………………………………………….2

Scurt istoric al limbajului………………………………………...……3-4

Mediul de dezvoltare VisualC#..............................................................4-7

Detalii tehnice de implementare……………………………………….8-22

Modalitati de dezvoltare……………………………………………….22

Aplicabilitate practica............................................................................23

Bibliografie…………………………………………………………....23