

**COLEGIUL NAȚIONAL DE INFORMATICĂ TUDOR VIANU**

# **LUCRARE DE ATESTAT**

**- Joc Snake -**

**Profesor coordonator,  
Livia Țoca**

**Autor,  
Ștefan Butură  
Clasa a XII-a H**

**2014**

## **CUPRINS**

<b>Introducere.....</b>	<b>1</b>
<b>Instalare.....</b>	<b>2</b>
<b>Prezentarea aplicației.....</b>	<b>3</b>
<b>Utilizarea aplicației.....</b>	<b>4</b>
<b>Opțiuni.....</b>	<b>5</b>
<b>Formulare secundare.....</b>	<b>7</b>
<b>Cod sursă.....</b>	<b>8</b>

# INTRODUCERE

Snake este un joc video apărut la sfârșitul anilor 70 dar care a devenit faimos în 1998, lucru datorat telefoanelor Nokia pe care acesta putea fi găsit. Motivul pentru care am ales să reproduc acest joc este pasiunea mea pentru jocurile simple care pot avea un succes urias iar Snake este unul dintre ele. Versiunea mea vine cu câteva opțiuni care îl vor face și mai captivant.

Aplicația poate fi folosită atât pentru a vă relaxa cât și pentru a dovedi îndemânarea și a vă testa reflexele.

Programul este realizat cu ajutorul mediului de programare Microsoft Visual Studio 2008, în limbajul C#.

Visual Studio 2008 are numele de cod Orcas și este prima versiune care permite dezvoltatorilor să folosească versiuni de .NET Framework. Visual Studio este bazat pe platforma de design .NET Framework 3.5, setul de instrumente .NET 3.0 a introdus versiuni actualizate ale ASP.NET, ADO.NET, Visual Basic, C# și CLR. Pentru interoperabilitatea și gestionarea codului mașină, Visual C++ introduce STL/CLR, care este un port de C++ Standard Template Library (STL) conține containere și algoritmi pentru gestionarea codului.

# Instalare

Cerințe de sistem:

- Sistem de operare: Windows XP/Vista/7
- .Net Framework 3.5 sau mai nou
- CD-ROM
- Spațiu liber pe disk: 50 Kb

Aplicația vine într-un folder ce conține executabilul snake.exe si fisierul scores.sna .

Pentru a rula aplicația faceți dublu click pe snake.exe .

## Prezentarea Aplicației

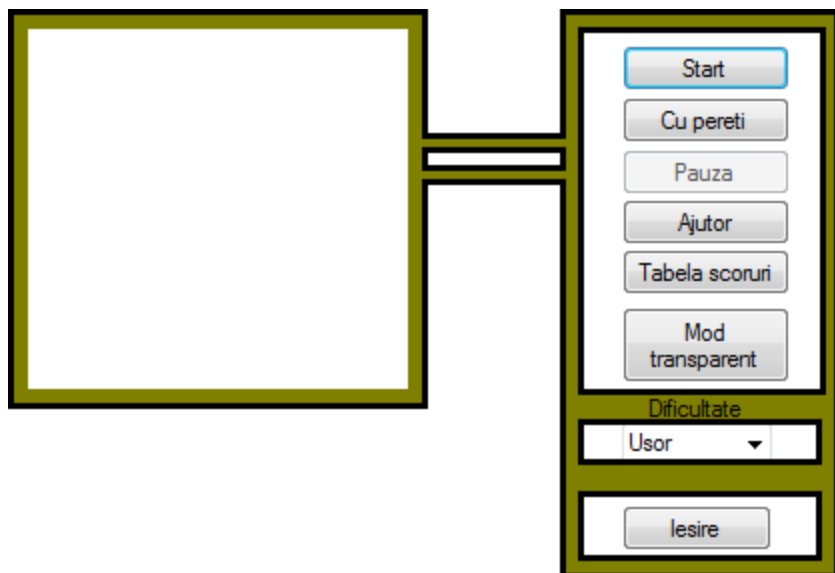
Microsoft Visual Studio este o aplicație Windows, deci folosește elemente de interfață specifice acestui mediu de lucru: lucru cu ferestre, meniuri, butoane, obiecte de control.

Scopul jocului este controlarea șarpelui și adunarea merelor de pe terenul de joc. Capul șarpelui nu poate atinge propriul corp sau cei patru pereți. În caz contrar jocul va fi terminat.

Aplicația este construită pe 10 formulare:

- Formularul „Meniu” - aici este prezent meniul cu opțiuni și jocul propriu-zis.
- Formularul „ScorNou” - aici utilizatorul este rugat să își introducă numele în scopul stocării scorului realizat de acesta într-o bază de date, în cazul în care este reușit un nou scor maxim.
- Formular „Tabela” - aici sunt afișate scorurile maxime însoțite de numele utilizatorilor aferenți.
- Formularul „Ajutor” - aici se găsesc instrucțiuni necesare folosirii aplicației

## Utilizarea aplicației



Acesta este formularul principal al aplicației și este format din două secțiuni:

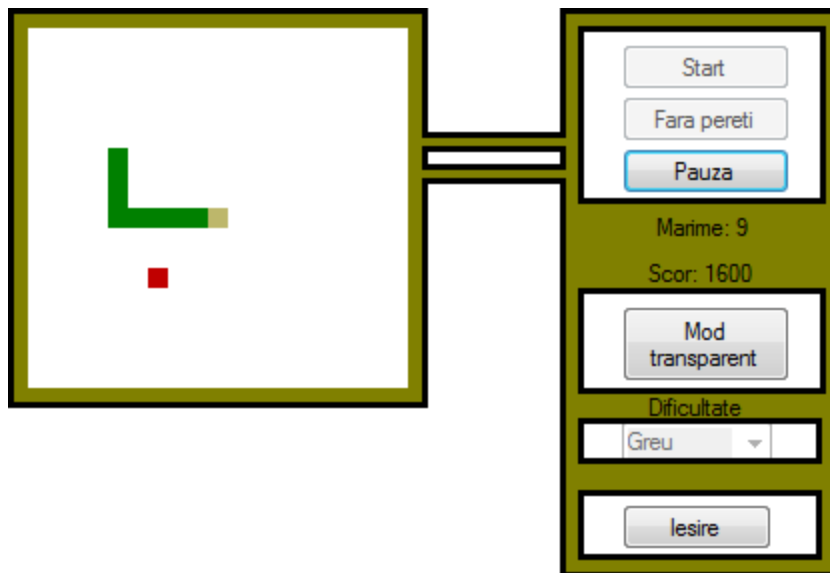
- Terenul de joc: patratul din partea stângă a formularului este destinat jocului propriu-zis.
- Meniul principal: dreptunghiul din partea stângă a formularului unde utilizatorul are la dispoziție o serie de opțiuni.

# Opțiuni

În meniul principal al aplicației se regăsesc următoarele opțiuni:

- Start
- Cu pereti/ Fara pereti
- Pauza
- Ajutor
- Tabela scoruri
- Mod transparent/opac
- Dificultate
- Iesire

Butonul Start este folosit pentru a porni jocul. De asemenea acesta modifică meniul astfel încât utilizatorul are la dispoziție doar opțiuni relevante și afișează scorul curent.



Butonul „Fara pereti” schimbă mecanica din spatele jocului. Această opțiune îi permite șarpelui să treacă prin pereți și să reapară pe partea opusă a ecranului dar punctajul obținut nu va fi contorizat în eventualitatea unui nou scor maxim. În cazul în care această opțiune este deja activată, butonul o va dezactiva.

Butonul Pauza suspendă temporar jocul până la reapăsarea lui.

Butonul Ajutor deschide formularul ce conține informații despre utilizarea aplicației.

Butonul Tabela scoruri deschide formularul în care sunt listate scorurile maxime.

Butonul Mod transparent face ca secțiunea de culoare albă a formularului să devină transparentă. În cazul în care opțiunea este activată deja, ea va fi dezactivată.

Comutatorul de dificultate îi permite utilizatorului să modifice viteza de joc punându-i la dispoziție opțiunile Usor, Mediu, Greu, Spartan. Prin selectarea unei dificultăți mai mari, punctajul va fi multiplicat cu un coeficient specific.

Butonul Iesire inchide aplicația.

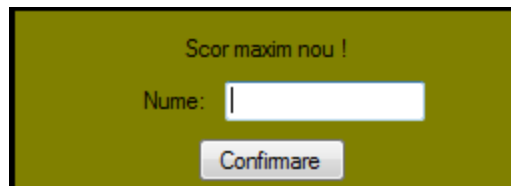
De asemenea, formularul acceptă comenzi de la tastatură în timpul jocului care pot fi regăsite pe formularul Ajutor:

- Tasta R oprește complet jocul
- Tasta F pauzează jocul
- Tastele W,S,A,D folosite pentru a controla șarpele



## Formulare secundare

Formularul „ScorNou” cu opțiunea Confirmare folosită pentru a adauga numele si scorul utilizatorului în baza de date.



Scor maxim nou !

Nume:

Confirmare

Formularul „Tabela” cu opțiunile Resetare, folosită pentru a șterge toate scorurile maxime existente în baza de date și Inchidere, folosită pentru a închide formularul.

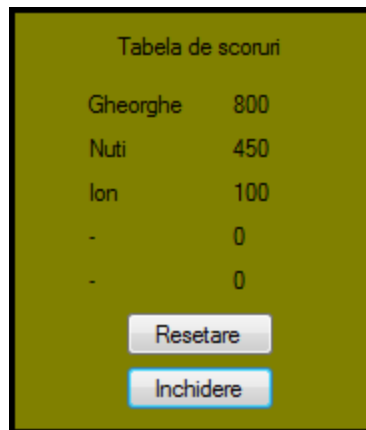


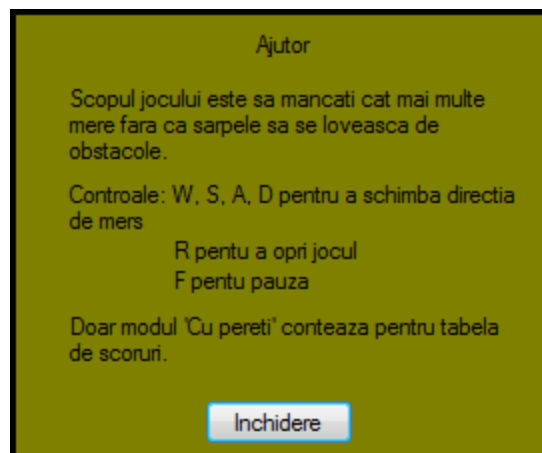
Tabela de scoruri

Gheorghe	800
Nuti	450
Ion	100
-	0
-	0

Resetare

Inchidere

Formularul „Ajutor” cu opțiunea Inchidere.



Ajutor

Scopul jocului este sa mancati cat mai multe mere fara ca sarpele sa se loveasca de obstacole.

Controale: W, S, A, D pentru a schimba directia de mers

R pentu a opri jocul

F pentu pauza

Doar modul 'Cu pereti' conteaza pentru tabela de scoruri.

Inchidere

## Codul sursă

La baza acestui joc stă următorul algoritm: la fiecare moment de timp, capul șarpelui își va modifica poziția în funcție de variabila **dir**, care controlează direcția de mișcare. Această variabilă se modifică în funcție de ultima tastă apasată (W, A, S, D) . Corpul șarpelui este format din pătrate stocate într-un vector. De la coada spre cap, fiecare bucată de corp se va muta pe poziția bucății următoare de corp.

```
while (i > 0)
{
    snake[i].Location = snake[i - 1].Location;
    i--;
}

switch (dir)
{
    case 0: xcap += 10; break;
    case 1: ycap += 10; break;
    case 2: xcap -= 10; break;
    case 3: ycap -= 10; break;
}

if (e.KeyCode == Keys.S && dir != 3 && able==1)
{
    able = 0;
    dir = 1;
}
else if (e.KeyCode == Keys.D && dir != 2 && able == 1)
{
    dir = 0;
    able = 0;
}
```

Merele apărute pe hartă sunt generate cu o poziție aleatoare. Când poziția capului coincide cu poziția unui măr, un nou pătrat este adăugat la corp, la următorul moment de timp, pe poziția anterioara ultimei bucăți din corp.

```
void marnou()
{
    Random rnd = new Random();
    marx=rnd.Next(0,18)*10+10;
    mary=rnd.Next(0,18)*10+10;
    mar.Location = new Point(marx, mary);
}

if (ad == 1)
{
    lung++;
    this.Controls.Add(snake[lung]);
    snake[lung].Visible = true;
    snake[lung].Location = snake[lung - 1].Location;
}
```

Timer-ul se va opri și variabilele utilizate vor fi resetate în cazul în care poziția capului coincide cu poziția unui perete sau a unei bucăți din corp.

La pornirea aplicației, scorurile și numele utilizatorilor sunt preluate din fișierul scores.sna. Pe parcursul folosirii aplicației, fișierul este actualizat în cazul în care acest lucru este necesar.

```
StreamWriter sw = new StreamWriter(Application.StartupPath + "\\scores.sna");
for (i = 0; i <=4 ; i++)
{
    sw.WriteLine(nume[i]);
    sw.WriteLine(scoruri[i]);
}
sw.Close();
```

```

if (e.KeyCode == Keys.S && dir != 3 && able==1)
{
    able = 0;
    dir = 1;
}
else if (e.KeyCode == Keys.D && dir != 2 && able == 1)
{
    dir = 0;
    able = 0;
}

StreamReader sr=new StreamReader(Application.StartupPath + "\\scores.sna");
while (!sr.EndOfStream)
{
    if (k % 2 == 0)
        nume[k / 2] = sr.ReadLine();
    else scoruri[k / 2] =Convert.ToInt32(sr.ReadLine());
    k++;
}
sr.Close();

```

