

Colegiul National “Mihai Viteazul”, Turda
Sectia: matematica informatica – intensiv informatica

Atestat informatica

Jet Fighter

Autor	Profesor indrumator
Hoka David Stelian	Pop Daniela

Cuprins:

1. Introducere
2. Motivul alegerii temei
3. "Manualul utilizatorului"
4. "Manualul programatorului"
5. Resurse hardware si software necesare
6. Bibliografie

1. Introducere

a) Prezentare joc:

Jet Fighter este o conceptie de joc video in care jucatorul este reprezentat printr-un avion. Misiunea jucatorului este de a impusca avioanele inamice fara a le permite sa treaca mai departe.

b) Motivul alegerii temei:

Am ales aceasta tema din trei motive, si anume: pasiunea personala in raport cu avioanele si jocurile video si fiindca a fost un exercitiu foarte util pentru programare, astfel am avut de asemenea si sansa de a invata ceva nou.

2.Structura proiectului:

Programul contine :

- Un meniu principal ce contine:
 - Optiune pentru a incepe jocul, in cazul in care aveti deja cont si va puteti loga
 - Optiune de inregistrare utilizator, logare apoi inceperea jocului
 - Optiune pentru a vedea lista jucatorilor
 - Buton de iesire din aplicatie
- Doua ferestre principale, meniul si fereastra in care se desfasoara jocul

3.Manualul utilizatorului:

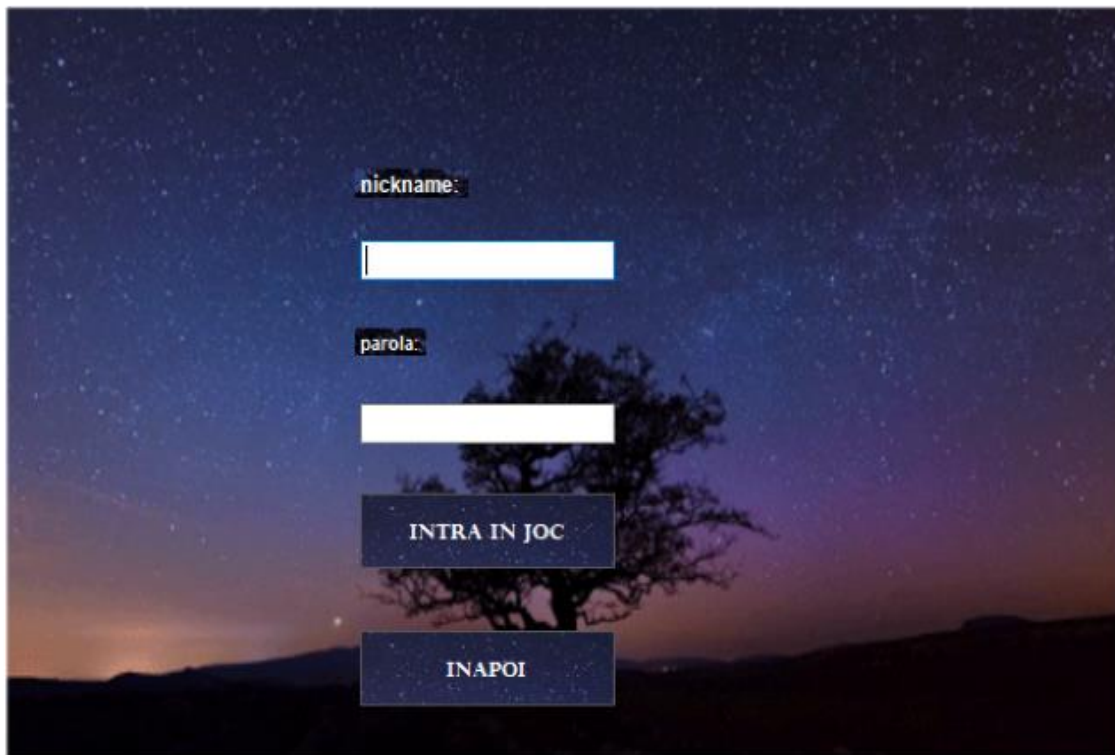
In aceasta parte este prezentat modul in care poate fi utilizata aplicatia. In meniul principal utilizatorul are posibilitatea de a apasa pe unul din 4 butoane.



- Butonul “INTRA IN JOC” va duce la fereastra de logare a utilizatorului, dupa care,daca va logati correct, veti putea intra in joc. De

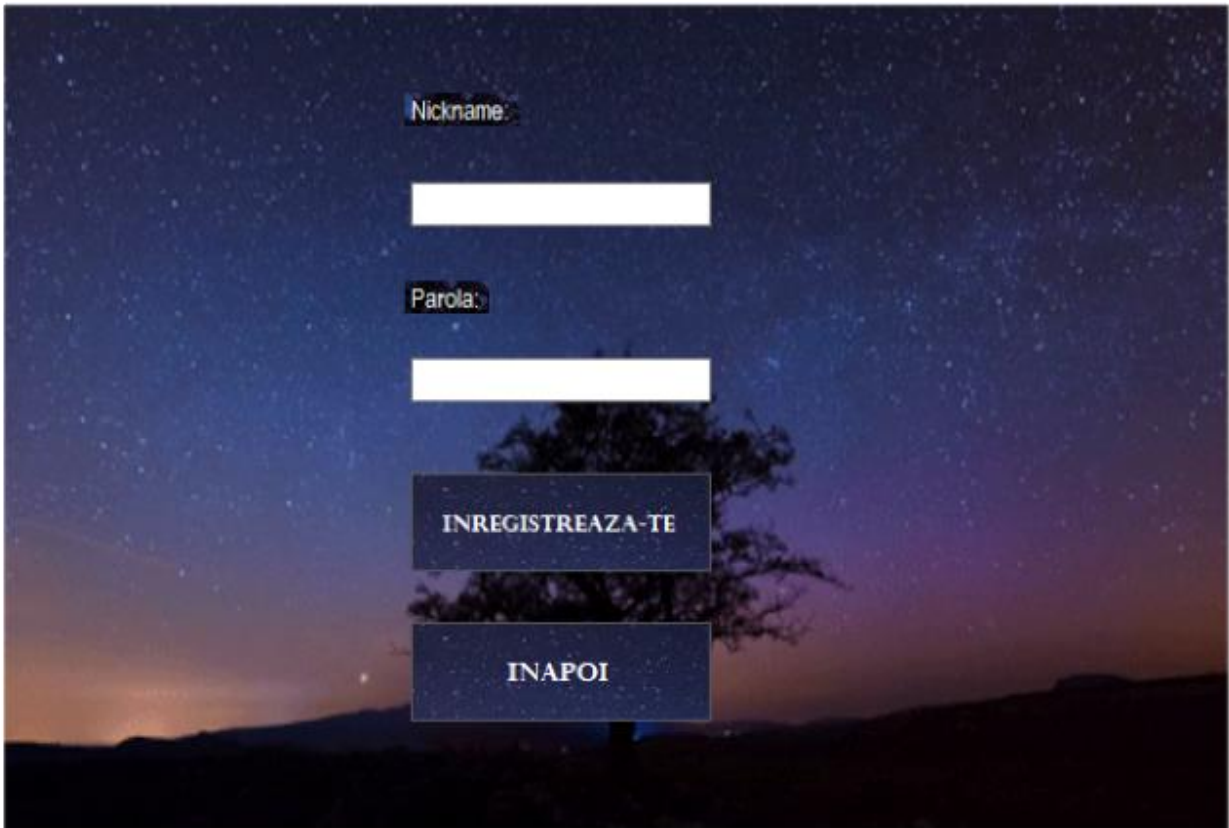
asemenea, exista butonul de
“Inapoi” din fereastra LogIn care va
adduce inapoi la meniul principal.





- Butonul “INREGISTREAZA-TE” va duce la fereastra de inregistrare utilizator nou, de asemenea exista si aici un buton care va poate aduce inapoi la meniu.





- Butonul “LISTA JUCATORILOR” va permite sa vedeti lista jucatorilor inregistrati.



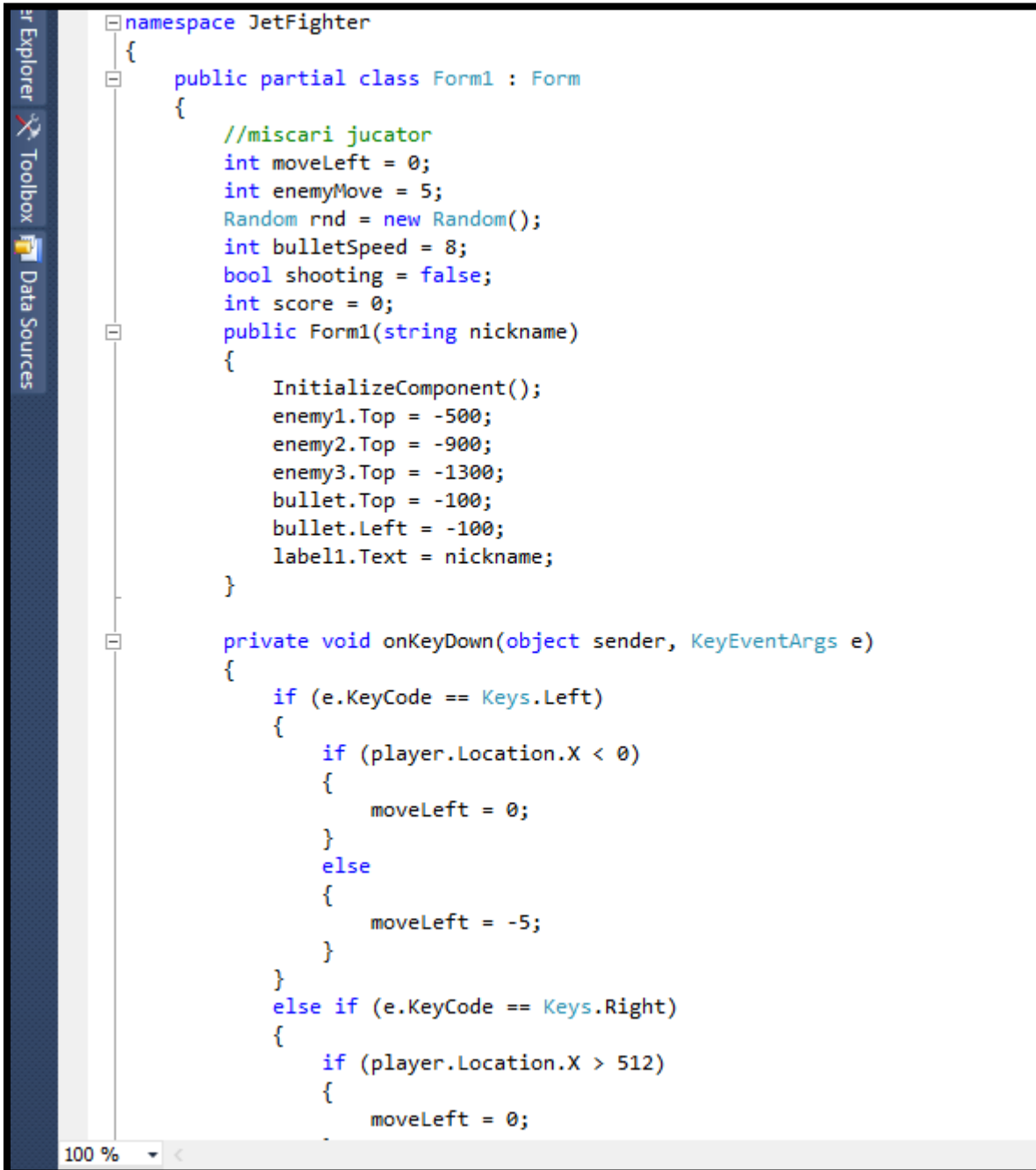
	id	nickname	scor
▶	1	david	12
*			

INAPOI

- Butonul “IESIRE” va permite sa iesiti din joc.



4. Manualul programatorului:



```
namespace JetFighter
{
    public partial class Form1 : Form
    {
        //miscari jucator
        int moveLeft = 0;
        int enemyMove = 5;
        Random rnd = new Random();
        int bulletSpeed = 8;
        bool shooting = false;
        int score = 0;
        public Form1(string nickname)
        {
            InitializeComponent();
            enemy1.Top = -500;
            enemy2.Top = -900;
            enemy3.Top = -1300;
            bullet.Top = -100;
            bullet.Left = -100;
            label1.Text = nickname;
        }

        private void onKeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Left)
            {
                if (player.Location.X < 0)
                {
                    moveLeft = 0;
                }
                else
                {
                    moveLeft = -5;
                }
            }
            else if (e.KeyCode == Keys.Right)
            {
                if (player.Location.X > 512)
                {
                    moveLeft = 0;
                }
            }
        }
    }
}
```

```

        {
            moveLeft = -5;
        }
    }
    else if (e.KeyCode == Keys.Right)
    {
        if (player.Location.X > 512)
        {
            moveLeft = 0;
        }
        else
        {
            moveLeft = 5;
        }
    }
    else if (e.KeyCode == Keys.Space)
    {
        if (shooting == false)
        {
            bulletSpeed = 8;
            bullet.Left = player.Left + 50;
            bullet.Top = player.Top;
            shooting = true;
        }
    }
}

private void onKeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Left)
    {
        moveLeft = 0;
    }
    else if (e.KeyCode == Keys.Right)
    {
        moveLeft = 0;
    }
}

```

100 %

```

private void playTimer_Tick(object sender, EventArgs e)
{
    explosion1.Visible = false;
    explosion2.Visible = false;
    explosion3.Visible = false;
    player.Left += moveLeft;
    bullet.Top -= bulletSpeed;
    enemy1.Top += enemyMove;
    enemy2.Top += enemyMove;
    enemy3.Top += enemyMove;
    scoreText.Text = "" + score;
    if (enemy1.Top == 660 || enemy2.Top == 660 || enemy3.Top == 660)
    {
        gameOver();
    }
    if (shooting && bullet.Top < 0)
    {
        shooting = false;
        bulletSpeed = 0;
        bullet.Top = -100;
        bullet.Left = -100;
    }
    enemyHit();
    playerHitEnemy();
}

```

In functia *playTimer_Tick* se petrece toata actiunea jocului, cu alte cuvinte, aceasta este functia “main”.

```
private void enemyHit()
{
    //cod lovire inamic
    if (bullet.Bounds.IntersectsWith(enemy1.Bounds))
    {
        score += 1;
        scoreText.Text = "Scor : " + score;
        enemy1.Top = -500;
        int ranP = rnd.Next(1, 300);
        enemy1.Left = ranP;
        shooting = false;
        bulletSpeed = 0;
        bullet.Top = -100;
        bullet.Left = -100;
    }
    else if (bullet.Bounds.IntersectsWith(enemy2.Bounds))
    {
        score += 1;
        scoreText.Text = "Scor : " + score;
        enemy2.Top = -900;
        int ranP = rnd.Next(1, 400);
        enemy2.Left = ranP;
        shooting = false;
        bulletSpeed = 0;
        bullet.Top = -100;
        bullet.Left = -100;
    }
    else if (bullet.Bounds.IntersectsWith(enemy3.Bounds))
    {
        score += 1;
        scoreText.Text = "Scor : " + score;
        enemy3.Top = -1300;
        int ranP = rnd.Next(1, 500);
        enemy3.Left = ranP;
        shooting = false;
        bulletSpeed = 0;
        bullet.Top = -100;
        bullet.Left = -100;
    }
}
```

In enemyHit este reprezentat deznodamantul actiunii glontului daca acesta tinteste avionul inamic.

```
private void playerHitEnemy()
{
    if (player.Bounds.IntersectsWith(enemy1.Bounds))
    {
        gameOver();
    }
    else if (player.Bounds.IntersectsWith(enemy2.Bounds))
    {
        gameOver();
    }
    else if (player.Bounds.IntersectsWith(enemy3.Bounds))
    {
        gameOver();
    }
}
```

playerHitEnemy reprezinta functia ce verifica daca jucatorul se ciocneste cu unul din inamicii acestuia. In caz afirmativ, aceasta functie ne va conduce spre functia *gameOver*.


```

private void gameOver()
{
    Form2 f2 = new Form2(label1.Text, score);
    f2.scor = score;
    player.Hide();
    enemy1.Hide();
    enemy2.Hide();
    enemy3.Hide();
    bullet.Hide();
    playTimer.Enabled = false;
    explosion1.Left = player.Left;
    explosion1.Visible = true;
    timer1.Enabled = true;
    DialogResult rez = MessageBox.Show("Scorul tau: " + score + " Vrei sa joci din nou?", "Game over", MessageBoxButtons.YesNoCancel);
    if (rez == DialogResult.Yes)
    {
        score = 0;
        scoreText.Text = "0";
        enemy1.Top = -500;
        enemy2.Top = -900;
        enemy3.Top = -1300;
        bullet.Top = -100;
        bullet.Left = -100;
        player.Show();
        enemy1.Show();
        enemy2.Show();
        enemy3.Show();
        bullet.Show();
        timer1.Enabled = false;
        playTimer.Enabled = true;
    }
    else if (rez == DialogResult.No)
    {
        timer1.Enabled = false;
        playTimer.Enabled = false;
        //Form2 f2 = new Form2();
        f2.Show();
        this.Hide();
    }
    else if (rez == DialogResult.Cancel)
    {
        DialogResult r1 = MessageBox.Show("Esti sigur ca vrei sa iesi din joc?", "Out game", MessageBoxButtons.YesNo);
        if (r1 == DialogResult.Yes)
        {
            Application.Exit();
        }
    }
}

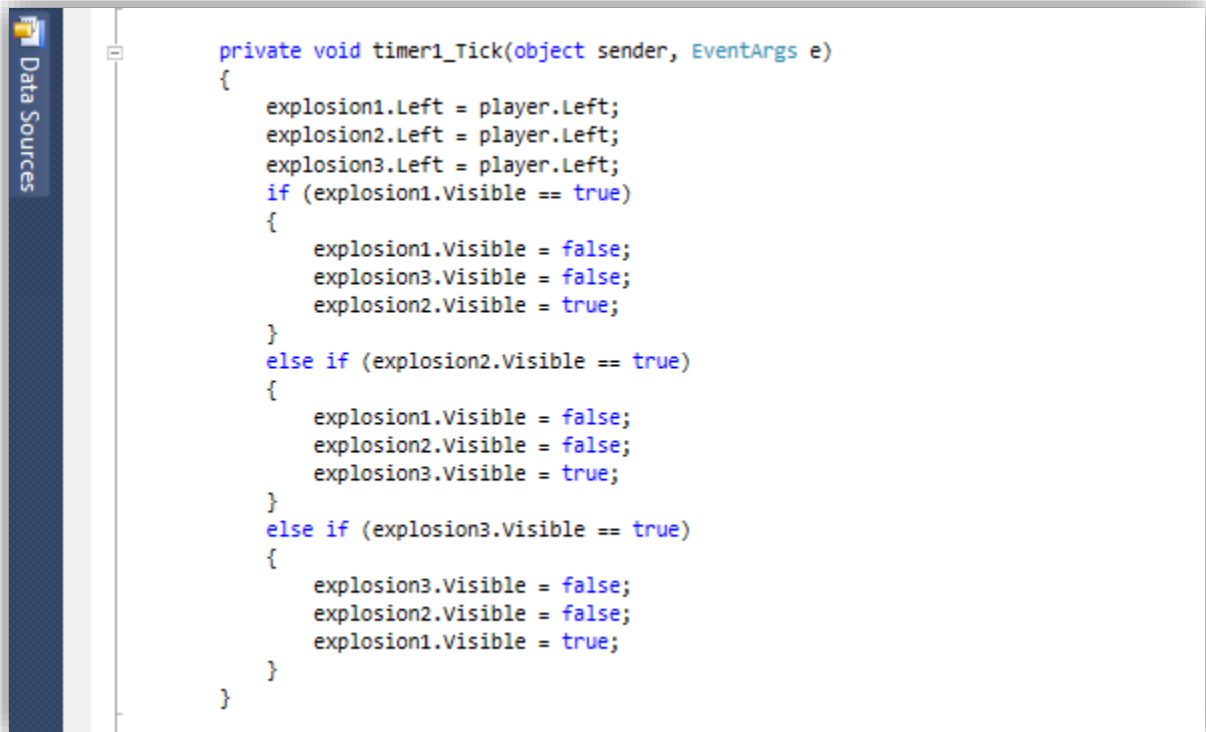
```

```

        {
            Application.Exit();
        }
        else if (r1 == DialogResult.No)
        {
            gameOver();
        }
    }
}

```

Functia *gameOver* are rolul de a incheia runda curenta a jocului si de a ne permite sa alegem daca mai vrem sa jucam inca o runda sau dca vrem sa ne intoarcem al meniu.



```
private void timer1_Tick(object sender, EventArgs e)
{
    explosion1.Left = player.Left;
    explosion2.Left = player.Left;
    explosion3.Left = player.Left;
    if (explosion1.Visible == true)
    {
        explosion1.Visible = false;
        explosion3.Visible = false;
        explosion2.Visible = true;
    }
    else if (explosion2.Visible == true)
    {
        explosion1.Visible = false;
        explosion2.Visible = false;
        explosion3.Visible = true;
    }
    else if (explosion3.Visible == true)
    {
        explosion3.Visible = false;
        explosion2.Visible = false;
        explosion1.Visible = true;
    }
}
```

In *timer1_Tick* am ales sa fac o “animatie” pentru sfarsitul fiecarei runde, astfel avionul jucatorului, explodeaza cand runda se sfarseste.

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    playTimer.Enabled = false;
    DialogResult r = MessageBox.Show("Esti sigur ca vrei sa iesi din joc?", "Iesi din joc", MessageBoxButtons.YesNo);
    if (r == DialogResult.Yes)
    {
        Application.Exit();
    }
    else
    {
        playTimer.Enabled = true;
    }
}
```

In *Form1_FormClosing* am ales sa pun jocul in pauza, in timp ce utilizatorul poate alege daca vrea sa iasa din joc sau sa se intoarca in runda curenta.

5. Resurse hardware si software necesare:

Resurse hardware:

- Procesor 1.44 GHz sau mai rapid.
- 1024 MB RAM.

Resurse software:

- Windows 7/Vista sau mai nou (x86 sau x64)

6. Bibliografie:

- <https://www.mooict.com/c-tutorial-create-a-fighter-jet-shooting-game/>
- <https://www.youtube.com/watch?v=RAAxqTK1W-k>