

**Lucrare de atestat la informatică**

# **FOX ADVENTURES**

**Profesor îndrumător:**

Dan Petrișor

**Autor:**

Porojan Mădălin

**- 2020 -**

**CUPRINS**

<b>I. Motivarea temei .....</b>	<b>2</b>
<b>II. Configurațiile necesare .....</b>	<b>3</b>
<b>III. Prezentarea teoretică.....</b>	<b>4</b>
<b>IV. Manual de utilizare.....</b>	<b>10</b>
<b>V. Programul sursă .....</b>	<b>15</b>
<b>VI. Bibliografie .....</b>	<b>28</b>

## I. Motivarea temei

Am ales această temă datorită pasiunii pentru dezvoltarea cunoștințelor în domeniul informaticii și al creării de jocuri, cât și datorită accesibilității programului **Unity**, acesta utilizând un limbaj de programare **C#** (C-Sharp), un limbaj accesibil, asemănător cu limbajul de programare **C++**.

Mediul de dezvoltare **Unity** oferă o mulțime de funcții și script-uri predefinite, care îți ușurează enorm efortul depus în stadiul de creare al unui joc, greșelile sau introducerea unei noi idei sau modificarea radicală a unor anumite aspecte ale jocului, fiind remediate, implementate cu ușurința prin interfața de care dispune acest engine.

Tipul de joc ales este **Platform, 2D**. Am optat pentru o dimensiune 2D, datorită simplității pe care o oferă platforma **Unity** în crearea tipului respectiv de jocuri, aplicația **Unity Asset Store**, cuprinzând diverse resurse, atât pentru dezvoltarea din punct de vedere al graficii cât și pentru unele animații ale unor obiecte sau caractere.

În ceea ce privește inspirația pentru genul de joc ales, seriile de jocuri, „*Jazz Jack Rabbit*”, „*Mario*”, „*LIMBO*”, „*Hollow Knight*”, „*Super Meat Boy*”, devenind exemple în realizarea temei.

## **II. Configurațiile necesare**

### **Cerințe minime de sistem:**

- **Sistem de operare:** Windows 7
- **Procesor:** 2 Ghz
- **Memorie RAM:** 2 GB RAM
- **Placă video:** 256 mb memorie video, model shader 3.0+
- **DirectX:** Versiunea 10
- **Spațiu stocare:** 500 MB spațiu disponibil

### **Cerințe recomandate de sistem:**

- **Sistem de operare:** Windows 10
- **Procesor:** Intel Core 2 Duo E4400 2.0GHz / AMD Athlon 64 X2 Dual Core 4200+
- **Placă video:** AMD Radeon X1900 GT / NVIDIA GeForce GT 340
- **System Memory:** 2 GB RAM
- **Spațiu stocare:** 500 MB spațiu disponibil
- **DirectX:** Versiunea 10

### III. Prezentarea teoretică



**Unity** ([www.unity.com](http://www.unity.com)) este un motor de joc **cross-platform** multiplă dezvoltat de **Unity Technologies**, anunțat și lansat pentru prima dată în iunie 2005, la Apple Inc. Worldwide Developers Conference, ca un motor de joc exclusiv pentru Mac OS X. Începând cu 2018, motorul fusese extins pentru a suporta mai mult de 25 de platforme. Motorul poate fi folosit pentru a crea jocuri de realitate tridimensională, bidimensională, VR, precum și simulări și alte experiențe. Motorul a fost adoptat de industrii din afara jocurilor video, precum film, automobile, arhitectură, inginerie și construcții. Motorul de joc este scris în C++ (pentru rulare) și C# (pentru dezvoltare, Unity Scripting API).

#### Istoric:

Motorul jocului **Unity** a fost lansat în 2005, având ca scop „democratizarea” dezvoltării de jocuri, realizând un mediu de dezvoltare accesibil mai multor dezvoltatori. Anul următor, Unity a fost desemnată una din cele mai bune aplicații în categoria „**Best Use of Mac OS X Graphics**” din **Apple Design Awards**. **Unity** a fost lansat inițial pentru Mac OS X, adăugând ulterior suport pentru Microsoft Windows și browsere Web.

**Unity 2.0** a fost lansat în 2007 cu aproximativ 50 de funcții noi. Versiunea a inclus un motor de teren optimizat pentru medii 3D detaliate, umbre dinamice în timp real, lumini direcționale, redare video și alte funcții. Versiunea a adăugat, de asemenea, funcții prin care dezvoltatorii ar putea colabora mai ușor. A inclus un strat de rețea pentru dezvoltatori pentru a crea jocuri multiplayer bazate pe User Datagram Protocol, oferind traducere de adrese de rețea și sincronizare și proceduri de accesare de la distanță.

Când Apple a lansat **App Store** în 2008, **Unity** a adăugat rapid asistență pentru iPhone. Timp de câțiva ani, motorul a fost necontestat pe iPhone și a devenit cunoscut cu dezvoltatorii de jocuri iOS.

**Unity 3.0** a fost lansat în septembrie 2010, ce introduce caracteristici care extind caracteristicile grafice ale motorului pentru computere desktop și console de jocuri video. Pe lângă suportul pentru Android, Unity 3 a oferit integrarea instrumentului *Beast Lightmap* de la *Illuminate Labs*, redarea amânată, un editor de arbori încorporat, redarea fontului autohton, maparea automată a UV și filtrele audio, printre altele.

În 2012, „*VentureBeat*” a scris: „*Puține companii au contribuit la fel de mult la fluxul de jocuri produse independent ca Unity Technologies. Mai mult de 1,3 milioane de dezvoltatori folosesc instrumentele sale pentru a crea grafici gee-whiz în iOS-ul lor, Android, consola, PC și jocuri bazate pe web. Unity vrea să fie motorul pentru jocuri cu mai multe platforme, punct.* ”. În noiembrie 2012, **Unity Technologies** a livrat Unity 4.0. Această versiune a adăugat *DirectX 11* și suport *Adobe Flash*, noi instrumente de animație numite Mecanim și acces la previzualizarea Linux.

Facebook a integrat un kit de dezvoltare de software pentru jocuri folosind motorul de jocuri Unity în 2013. Acesta a oferit instrumente care permiteau urmărirea campaniilor publicitare și legătura profundă, unde utilizatorii erau legați direct de la postările de pe social media la anumite porțiuni din jocuri și ușor în partajarea jocului de imagini. În 2016, Facebook a dezvoltat o nouă platformă de jocuri pentru PC cu Unity. Unity a oferit suport pentru platformele de jocuri Facebook, iar dezvoltatorii Unity ar putea exporta și publica mai rapid jocurile pe Facebook.

„*The Verge*” a spus despre lansarea Unity 5 din 2015: "*Unity a început cu scopul de a face dezvoltarea jocului universal accesibilă. Unity 5 este un pas mult așteptat către acel viitor.*" Cu Unity 5, motorul și-a îmbunătățit iluminarea și audio-ul. Prin **WebGL**, dezvoltatorii Unity și-ar putea adăuga jocurile în browserele web compatibile, fără a fi necesare pluginuri pentru jucători. Unity 5.0 a oferit iluminare globală în timp real, previzualizări de mapare a luminii, Unity Cloud, un nou sistem audio și motorul de fizică **Nvidia PhysX 3.3**. Cea de a cincea generație a motorului Unity a introdus de asemenea **Cinematic Image Effects** pentru a face ca jocurile Unity să pară mai puțin generice. Unity 5.6 a adăugat noi efecte de iluminare și particule, a actualizat performanțele generale ale motorului și a adăugat suport nativ pentru *Nintendo Switch*, *Facebook Gameroom*, *Google Daydream VR* și API grafică *Vulkan*. A introdus un player video 4K capabil să ruleze videoclipuri în 360° pentru realitate virtuală. Cu toate acestea, unii jucători au criticat

accesibilitatea Unity datorită volumului mare de jocuri produse rapid publicate pe platforma de distribuție Steam de către dezvoltatorii neexperimentați. CEO **John Riccitiello** a declarat într-un interviu că el consideră că acesta este un efect secundar al succesului Unity în democratizarea dezvoltării jocului: *„Dacă aș avea calea mea, aș dori să văd 50 de milioane de oameni care folosesc Unity - deși eu nu cred că vom ajunge acolo în viitorul apropiat. Aș dori să văd copii de liceu și colegiu care o folosesc, oameni din afara industriei de bază. Cred că este trist că majoritatea oamenilor sunt consumatori de tehnologie și nu creatori. Lumea este un loc mai bun când oamenii știu să creeze, nu doar consumă și asta încercăm să promovăm ”.*

În decembrie 2016, **Unity Technologies** a anunțat că va schimba sistemul de numerotație pentru Unity de la identificatorii pe bază de secvență, anul viitor, pentru a alinia versiunea cu cadența lor de lansare mai frecventă; Unity 5.6 a fost, așadar, urmată de Unity 2017. Instrumentele Unity 2017 au prezentat un motor de redare grafică în timp real, gradarea culorilor și construirea lumii, analiza operațiilor live și raportarea performanței. Unity 2017.2 a subliniat planurile Unity Technologies dincolo de jocurile video. Aceasta a inclus noi instrumente, cum ar fi Timeline, care le-a permis dezvoltatorilor să tragă și arunca animații în jocuri și Cinemachine, un sistem de camere inteligente în cadrul jocurilor. Unity 2017.2 a integrat, de asemenea, instrumentele *3DS Max* și *Maya* ale Autodesk în motorul Unity.

Unity 2018 a prezentat *Scriptable Render Pipeline* pentru ca dezvoltatorii să creeze grafică de înaltă performanță. Aceasta a inclus High-Definition Rendering Pipeline pentru experiențe de consolă și PC, și Lightweight Rendering Pipeline pentru mobil, realitate virtuală, realitate augmentată și realitate mixtă. Unity 2018 a inclus și instrumente de machine learning, cum ar fi *Imiting Learning*, prin care jocurile învață din obiceiurile reale ale jucătorilor, suport pentru Magic Leap și șabloane pentru noii dezvoltatori.

În 2019, a fost adăugată o nouă legătură la *Wolfram Language*, ceea ce face posibilă accesarea funcțiilor și operațiilor avansate ale limbajului Wolfram de către Unity.

## Prezentare generală:

Unity oferă utilizatorilor capacitatea de a crea jocuri și experiențe atât în **2D** cât și în **3D**, iar motorul oferă un API primar de scripturi în **C#** , atât pentru editorul Unity sub formă de pluginuri, cât și pentru jocuri în sine, precum și pentru funcționarea de drag and drop . Înainte ca limbajul **C#** să fie limbajul de programare primar folosit pentru motor, acesta a utilizat anterior **Boo** , care a fost eliminat odată cu lansarea Unity 5, și o versiune de *JavaScript* numită *UnityScript* , care a fost depreciată în august 2017, după lansarea Unity 2017.1, în favoarea **C#**.

În cadrul jocurilor **2D**, Unity permite importul de sprites-uri și un world renderer avansat **2D**. Pentru jocurile **3D**, Unity permite specificarea compresiei texturii , a mipmap-urilor și a setărilor de rezoluție pentru fiecare platformă pe care motorul de joc o susține și oferă suport pentru mapare cu *bump* , *mapare cu reflecție* , *mapare parallax* , *ocluzie spațială pe ecran (SSAO)*.

## Platforme suportate:

Unity este un motor ce suporta mai multe platforme. **Editorul Unity** este acceptat pe **Windows** și **macOS** , cu o versiune a editorului disponibilă pentru platforma **Linux** , deși într-o etapă experimentală. În timp ce motorul însuși acceptă în prezent jocuri pentru mai mult de 25 de platforme diferite, inclusiv mobil, desktop, console și realitate virtuală. Platformele includ *iOS, Android, Tizen, Windows, Universal Windows Platform, Mac, Linux, WebGL, PlayStation 4, PlayStation Vita, Xbox One, 3DS, Oculus Rift, Google Cardboard, Steam VR, PlayStation VR, Gear VR, Windows Mixed Reality, Daydream, Android TV, Samsung Smart TV, tvOS, Nintendo Switch, Fire OS, Facebook Gameroom, ARKit Apple, Google ARCore, Vuforia, și Magic Leap*.

Începând cu anul 2018 , Unity fusese utilizată pentru a crea aproximativ jumătate din jocurile mobile pe piață și 60% din conținutul de realitate augmentată și realitate virtuală, inclusiv aproximativ 90% pe platformele de realitate augmentată, precum Microsoft *HoloLens* și 90 la sută din conținutul *Samsung Gear VR*. Tehnologia Unity este baza pentru cele mai multe experiențe de realitate virtuală și realitate augmentată, iar „*Fortune*” a spus că Unity „domină afacerea cu realitatea virtuală”. Unity Machine Learning Agents sunt un software open-source prin care platforma Unity se conectează la programe de



machine learning, inclusiv *TensorFlow Google*. Software-ul este folosit pentru a dezvolta roboți și autovehicule auto.

Unity a acceptat anterior alte platforme, inclusiv propriul său player *Web Unity*, un plugin pentru browserul web. Cu toate acestea, a fost depășit în favoarea *WebGL*. Începând cu versiunea 5, Unity și-a oferit pachetul *WebGL* compilat în *JavaScript* folosind un traducător de limbă în două etape ( *C#* la *C++* și în cele din urmă în *JavaScript* ).

### **Model de licențiere:**

În primii zece ani ca produs, versiunile plătite ale Unity au fost vândute în mod direct; în 2016, corporația a trecut la un model de abonament. Unity are opțiuni de licență gratuite și plătite. Licența gratuită este destinată uzului personal sau companiilor mai mici care generează mai puțin de 100.000 USD anual, iar abonamentele se bazează pe veniturile generate de jocurile care folosesc Unity.

### **Unity Asset Store:**

Creatorii pot dezvolta și vinde resurse create de utilizatori, altor producători de jocuri prin intermediul magazinului *Unity Asset Store*. Aceasta include resurse și medii 3D și 2D pentru dezvoltatorii pentru a cumpăra sau a le vinde. Unity Asset Store a fost lansat în 2010. Până în 2018, au existat aproximativ 40 de milioane de descărcări prin intermediul magazinului digital.

### **Alte utilizări:**

În anul 2010, **Unity Technologies** și-a folosit motorul de joc pentru a trece la alte industrii folosind platforma 3D în timp real, inclusiv film și automobile. Unity a experimentat pentru prima dată filmarea cu Adam , un scurtmetraj despre un robot care a scăpat din închisoare. Mai târziu, Unity a asociat cu cineastul *Neill Blomkamp* , al cărui *Oats Studios* a folosit instrumentele motorului, inclusiv redarea în timp real și *Cinemachine*, pentru a crea două scurtmetraje generate de computer, „*Adam: The Mirror*” și „*Adam: The Profet*”. La conferința Unite Europe din Amsterdam din 2017, Unity s-a concentrat pe realizarea filmului cu noul instrument *Cinemachine* al Unity 2017.1. În 2018,

*Disney Television Animation* a lansat trei scurtmetraje, numite *Baymax Dreams*, care au fost create cu ajutorul motorului Unity.

Producătorii auto folosesc tehnologia Unity pentru a crea modele la scară completă de vehicule noi în realitate virtuală, pentru a construi linii de asamblare virtuale și pentru a instrui angajații. Motorul Unity este folosit de compania *DeepMind*, o companie *Alphabet Inc.*, pentru a antrena inteligența artificială. Alte utilizări urmărite de **Unity Technologies** includ arhitectură, inginerie și construcții.



**C#** este un limbaj de programare conceput de *Microsoft* la sfârșitul anilor 90'. A fost conceput ca un concurent pentru limbajul Java. Ca și acesta, **C#** este un derivat al limbajului de programare **C++**. A fost proiectat de *Anders Hejlsberg*, iar echipa sa de dezvoltare este în prezent condusă de Mads Torgersen. Versiunea cea mai recentă este 8.0, care a fost lansată în 2019 alături de *Visual Studio 2019* versiunea 16.3.

### Scopuri ale limbajului:

- Limbajul se dorește a fi un limbaj de programare simplu, modern, cu scop general, orientat pe obiecte.
- Limbajul și implementările ar trebui să ofere suport pentru principiile de inginerie software, cum ar fi verificarea puternică a tipului, verificarea limitelor de matrice, detectarea încercărilor de utilizare a variabilelor neinițializate și „*colectarea automată a gunoiului*”. Robustitatea, durabilitatea și productivitatea programatorului sunt importante.
- Deși aplicațiile **C#** sunt destinate să fie economice în ceea ce privește cerințele de memorie și puterea de procesare, limbajul nu a fost destinat să concureze direct la performanță și dimensiune cu C sau cu un limbaj de asamblare.
- Portabilitatea este foarte importantă pentru codul sursă și programatorii, în special cei deja familiarizați cu C și C++.






## IV. Manual de utilizare

### Obiectivul jocului:

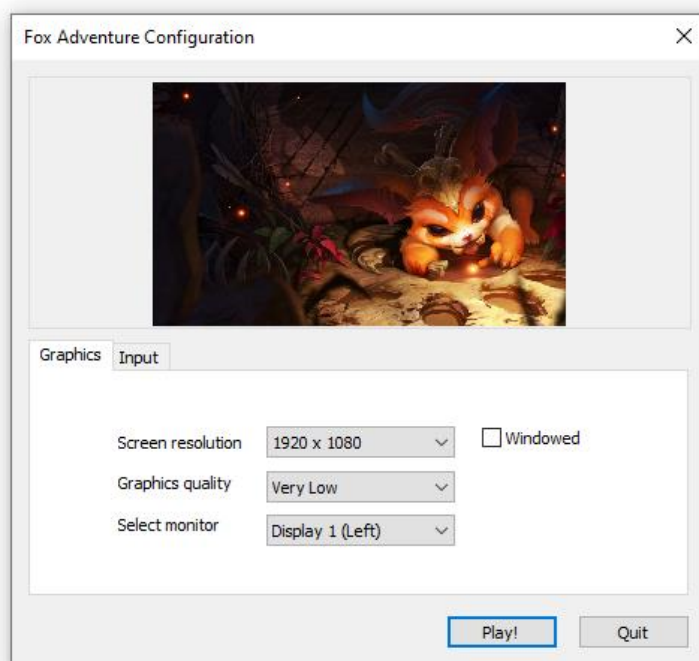
Fox Adventures este un joc clasic, platform, unde personajul controlat este o vulpe. Scopul principal al jocului este colectarea diamantelor și supraviețuirea într-o lume plină de surprize și inamici ce par inofensivi la prima vedere.

### Accesarea și configurarea jocului:

- După ce jocul s-a **instalat**, accesați directorul unde a-ți decis să îl stocați:

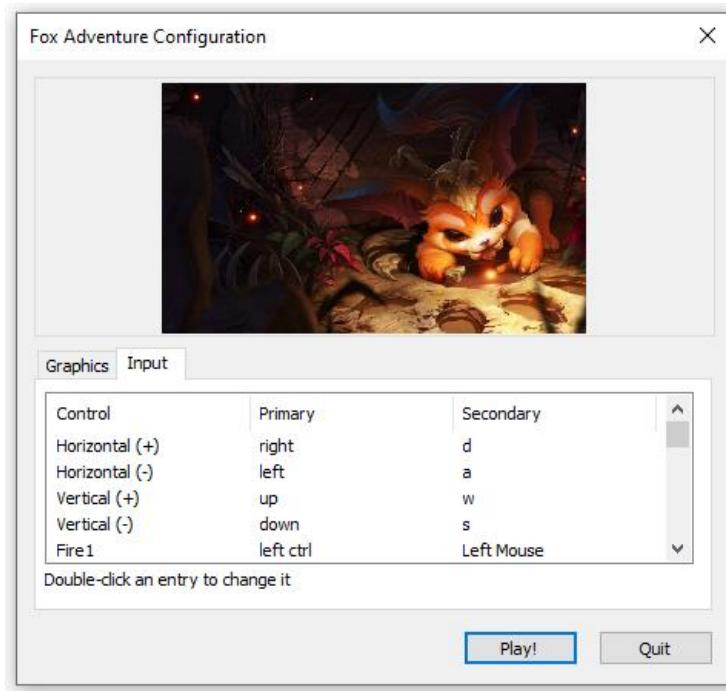
	Fox Adventure_Data	3/6/2020 9:14 PM	File folder	
	MonoBleedingEdge	3/6/2020 9:14 PM	File folder	
	Fox Adventure.exe	3/6/2020 9:14 PM	Application	636 KB
	UnityCrashHandler64.exe	7/9/2019 11:40 AM	Application	1,421 KB
	UnityPlayer.dll	7/9/2019 11:40 AM	Application exten...	23,405 KB

- Apăsați dublu click pe **Fox Adventure.exe**:



- După deschiderea executabilului, vă va apărea o fereastră nouă, pentru **configurarea video** a jocului:

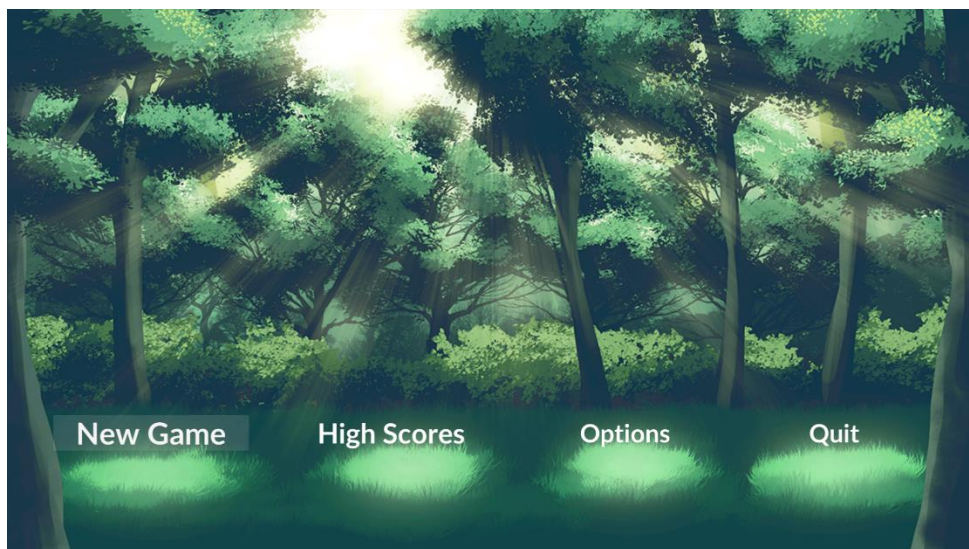
- Câmpul **Screen resolution** vă permite să schimbați rezoluția jocului, în funcție de rezoluția suportată de monitor.
- Câmpul **Graphics quality** vă permite să modificați setările grafice ale jocului, **very low, low, medium, high, ultra**.
- Câmpul **Select monitor** vă permite să schimbați afișarea jocului pe un alt monitor.
- Schimbarea implicită a tastelor se face prin accesarea meniului **Input**:



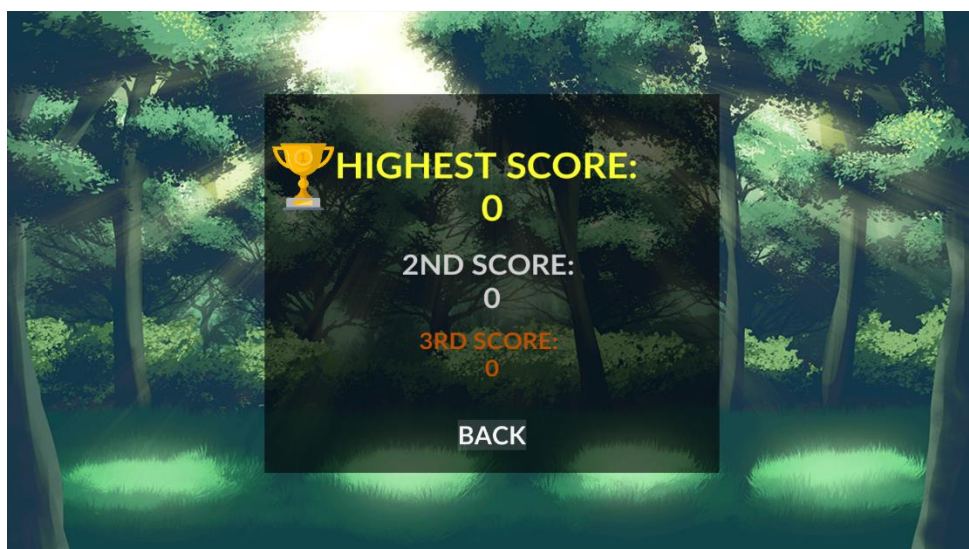
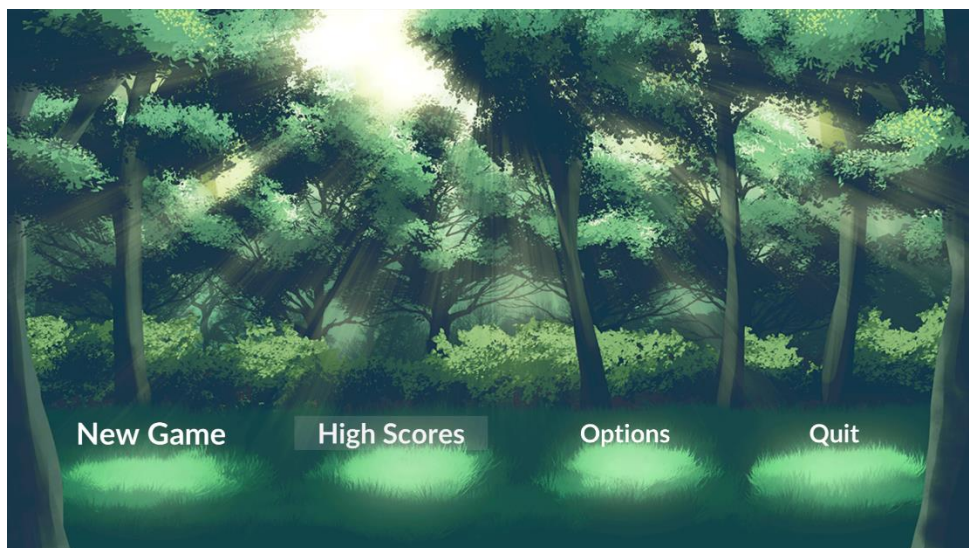
### Utilizarea meniului din joc:

- Meniul principal al jocului este împărțit în 3 submeniuri iar navigarea acestuia se face prin utilizarea tastelor, **Sageată stânga** sau **dreapta**, si tasta **ENTER**, pentru accesarea submeniurilor:

- **New Game**, începe jocul odată cu apăsarea tastei **ENTER**.



- **High Scores**, aici sunt afișate cele mai bune 3 scoruri:





- **Options**, aici se pot configura:
  - **Afișarea FPS-ului**;
  - **Nivelul de dificultate**, **Easy** (3 vieți), **Medium** (2 vieți), **Hard** (1 viață);
  - **FPS-ul maxim**, 60 sau 120;



- **Quit**, ieșirea din joc se face prin utilizarea aceluia.

### **Controlarea personajului și aspecte importante ale jocului:**

Controlarea vulpei se face prin utilizarea tastelor: **W A S D** sau a **sageților** sau prin intermediul unui **joystick**:

- **W: Sărit**



- **A: Stânga**

- **S: Târâș**



- **D: Dreapta**

Personajul are **3 vieți**, fiind situate în partea **stângă, de sus**, a ecranului. **Inamicii** scad **o viață** iar unele **capcane** pot scădea până la **3 vieți**:



**Capcane:**

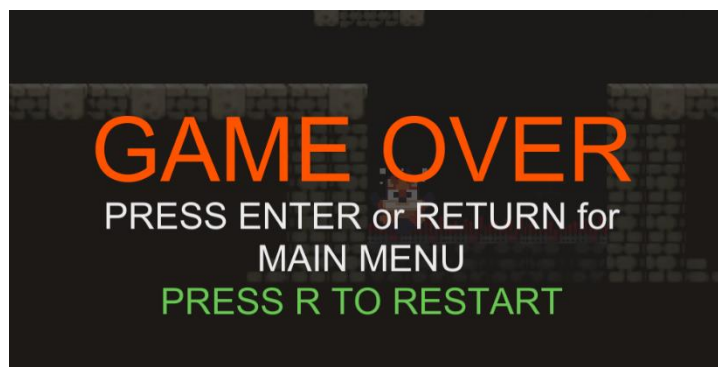



**Exemplu de inamic:**



**Inamicii** reprezintă problema principală a vulpei, aceștia fugărind vulpea în momentul în care se află mult prea aproape de ei.

În momentul pierderii a tuturor celor 3 vieți, jocul se termină iar scorul este resetat. Jucătorul având posibilitatea de a reseta nivelul (tasta **R**) sau de a accesa meniul (tasta **ENTER**):



**Scorul** se află în partea **stângă, de sus**, a ecranului. Acesta crește în momentul colectării unor obiecte, precum **diamante** (500 scor)  sau **cireșe** (50 scor + 1 viață):



## V. Programul sursă

### Script pentru sistemul de vieți:

```
using UnityEngine;
using UnityEngine.SceneManagement;
public class GameControlScript : MonoBehaviour
{
    public GameObject heart1, heart2, heart3;
    public static int health;
    public Animator animator;
    public GameObject popDeathMessage;
    bool fpsStatus=FpsCounter.isON;
    private bool isPlaying = false;
    void Start()
    {
        health = OptionsSelectButton.health;
        heart1.gameObject.SetActive(true);
        heart2.gameObject.SetActive(true);
        heart3.gameObject.SetActive(true);
    }
    void Update()
    {
        if (health > 3)
            health = 3;
        switch (health)
        {
            case 3:
                heart1.gameObject.SetActive(true);
```



```
heart2.gameObject.SetActive(true);  
heart3.gameObject.SetActive(true);  
break;
```

case 2:

```
heart1.gameObject.SetActive(true);  
heart2.gameObject.SetActive(true);  
heart3.gameObject.SetActive(false);  
break;
```

case 1:

```
heart1.gameObject.SetActive(true);  
heart2.gameObject.SetActive(false);  
heart3.gameObject.SetActive(false);  
break;
```

case 0:

```
heart1.gameObject.SetActive(false);  
heart2.gameObject.SetActive(false);  
heart3.gameObject.SetActive(false);  
animator.SetBool("isHurt", true);  
FpsCounter.isON = false;  
Time.timeScale = 0;  
GameObject.Find("MAINplayer").GetComponent<playerMovement>().enabled  
= false;  
break;
```

default:

```
heart1.gameObject.SetActive(false);  
heart2.gameObject.SetActive(false);  
heart3.gameObject.SetActive(false);
```

```

        animator.SetBool("isHurt", true);

        FpsCounter.isOn = false;

        Time.timeScale = 0;

        GameObject.Find("MAINplayer").GetComponent<playerMovement>().enabled
= false;

        break;
    }

    if (health <= 0)
    {

        if (!isPlaying)
        { PlaySoundsFX.PlaySound("gameoverFx"); }

        isPlaying = true;

        popDeathMessage.SetActive(true);

        if (Input.GetKeyDown(KeyCode.Return))
        {

            FpsCounter.isOn = fpsStatus;

            Time.timeScale = 1;

            SceneManager.LoadScene("MainMenu");

            Scorecounter.scoreValue = 0;

            GemCounter.gemValue = 0;

        }

        else if (Input.GetKeyDown(KeyCode.R))
        {

            FpsCounter.isOn = fpsStatus;

            Time.timeScale = 1;

            Scene scene = SceneManager.GetActiveScene();

            SceneManager.LoadScene(scene.name);

            Scorecounter.scoreValue = 0;

```

```

        GemCounter.gemValue = 0;
    }
}
}
}

```

### **Script pentru parole/trucuri:**

```

using UnityEngine;

public class Cheats : MonoBehaviour
{
    private int index00, index01, index02, index03;
    private string[] cheatCode00, cheatCode01, cheatCode02, cheatCode03;

    void Start()
    {
        cheatCode00 = new string[] { "k", "i", "l", "l", "m", "e" };
        cheatCode01 = new string[] { "r", "e", "s", "e", "t", "h", "s" };
        cheatCode02 = new string[] { "i", "m", "d", "y", "i", "n", "g" };
        cheatCode03 = new string[] { "g", "o", "l", "d", "f", "o", "x" };
        index00 = 0;
        index01 = 0;
        index02 = 0;
        index03 = 0;
    }

    void Update()
    {
        //Suicide.....KILLME

        if (Input.anyKeyDown)
        {

```

```

if (Input.GetKeyDown(cheatCode00[index00]))
{
    index00++;
}
else
{
    index00 = 0;
}
}
if (index00 == cheatCode00.Length)
{
    GameControlScript.health = 0;
    index00 = 0;
    PlaySoundsFX.PlaySound("cheatActivatedFx");
}
//Reset the High Scores.....RESETHS
if (Input.anyKeyDown)
{
    if (Input.GetKeyDown(cheatCode01[index01]))
    {
        index01++

    }
    else
{
        index01 = 0;
    }
}
if (index01 == cheatCode01.Length)

```

```

{
    PlayerPrefs.DeleteAll();

    index01 = 0;

    PlaySoundsFX.PlaySound("cheatActivatedFx");
}

//Gives 3 lives(1000 lives).....IMDYING
if (Input.anyKeyDown)
{
    if (Input.GetKeyDown(cheatCode02[index02]))
    {
        index02++;
    }
    else
    {
        index02 = 0;
    }
}

if (index02 == cheatCode02.Length)
{
    GameControlScript.health = 3;
    index02 = 0;
    PlaySoundsFX.PlaySound("cheatActivatedFx");
}

//Gives 50 diamonds and 2000 score.....GOLDFOX
if (Input.anyKeyDown)
{
    if (Input.GetKeyDown(cheatCode03[index03]))
    {

```

```

        index03++;
    }
    else
    {
        index03 = 0;
    }
}

if (index03 == cheatCode03.Length)

{
    Scorecounter.scoreValue += 2000;

    GemCounter.gemValue += 50;
    index03 = 0;
    PlaySoundsFX.PlaySound("cheatActivatedFx");
}

DontDestroyOnLoad(this);}}

```

### **Script pentru sunet:**

```

using UnityEngine;

public class PlaySoundsFX : MonoBehaviour

{
    public static AudioClip gemFx, cherryFx, jumpFx, hurtFx, runFx, gameoverFx,
    buttonFx, confirmFx, cancelFx, newgameFx, quitFx,cheatActivatedFx;

    static AudioSource audio;

    void Start()
    {
        gemFx = Resources.Load<AudioClip>("gemSounds");
        cherryFx = Resources.Load<AudioClip>("cherrySound");
    }
}

```

```

jumpFx = Resources.Load<AudioClip>("JumpFX");

runFx = Resources.Load<AudioClip>("RunFX");
hurtFx = Resources.Load<AudioClip>("HurtFX");
    gameOverFx = Resources.Load<AudioClip>("gameOverFX");
    buttonFx = Resources.Load<AudioClip>("buttonFX");
    confirmFx = Resources.Load<AudioClip>("confirmFX");
    cancelFx = Resources.Load<AudioClip>("cancelFX");
    newgameFx = Resources.Load<AudioClip>("newgameFX");
    quitFx = Resources.Load<AudioClip>("quitFX");
    cheatActivatedFx = Resources.Load<AudioClip>("cheatActivatedFX");
    audio = GetComponent<AudioSource>();
}

public static void PlaySound(string clip)
{
    switch (clip)
    {
        case "gemFx":
            audio.PlayOneShot(gemFx);
            break;
        case "cherryFx":
            audio.PlayOneShot(cherryFx);
            break;
        case "jumpFx":
            audio.PlayOneShot(jumpFx);
            break;
        case "runFx":
            audio.PlayOneShot(runFx);

```

```

break;

case "hurtFx":
    audio.PlayOneShot(hurtFx);
    break;

case "gameoverFx":
    audio.PlayOneShot(gameoverFx);
    break;

case "buttonFx":
    audio.PlayOneShot(buttonFx);
    break;

case "confirmFx":
    audio.PlayOneShot(confirmFx);
    break;

case "cancelFx":
    audio.PlayOneShot(cancelFx);
    break;

case "newgameFx":
    audio.PlayOneShot(newgameFx);
    break;

case "quitFx":
    audio.PlayOneShot(quitFx);
    break;

case "cheatActivatedFx": audio.PlayOneShot(cheatActivatedFx); break; } }

```

### **Script pentru meniul de opțiuni:**

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

```



```
using UnityEngine.UI;
```

```
public class OptionsSelectButton : MonoBehaviour
{
    [SerializeField] Animator animator;
    [SerializeField] MenuController muc;
    [SerializeField] int thisIndex;
    public bool isFpsLocked;
    public bool isFpsCounter;
    public GameObject fpsM;
    public GameObject fpsCounterSTATUS;
    Text mfps;
    Text fpsCounter;
    public Text difficulty;
    private int ind=0;
    public static int health=3;
    // Update is called once per frame
    void Start()
    {
        PlaySoundsFX.PlaySound("confirmFx");
        mfps = fpsM.GetComponent<Text>();
        fpsCounter = fpsCounterSTATUS.GetComponent<Text>();
        isFpsCounter = FpsCounter.isON;
        if (isFpsCounter)
        {
            fpsCounter.text = "FPS COUNTER: ON";
        }
        else
        {
```

```

        fpsCounter.text = "FPS COUNTER: OFF";
    }
    if (LOCKFPS.maxFPS == 60)
    {
        isFpsLocked = true;
        mfps.text = "MAX FPS: 60";
    }
    else if (LOCKFPS.maxFPS == 120)
    {
        isFpsLocked = false;
        mfps.text = "MAX FPS: 120";
    }
}

void Update()
{
    if (muc.index == thisIndex)
    {
        animator.SetBool("isSelected", true);
    }
    else
    {
        animator.SetBool("isSelected", false);
    }
    if (muc.index == 0 && Input.GetKeyDown(KeyCode.Return))
    {
        if (isFpsCounter)
        {
            isFpsCounter = false;
            fpsCounter.text = "FPS COUNTER: OFF";

```

```

        FpsCounter.isOn = false;

    }

else

    {

        isFpsCounter = true;

        fpsCounter.text = "FPS COUNTER: ON";

        FpsCounter.isOn = true;

    }

}if (muc.index == 1 && Input.GetKeyDown(KeyCode.Return))

{

    if (Input.GetKeyDown(KeyCode.Return))

    { ind++;

        if (ind > 2)

            ind = 0;

        if (ind == 0)

        {

            difficulty.text = "EASY";

            health = 3;

        }

        if (ind == 1)

        {

            difficulty.text = "MEDIUM";

            health = 2;

        }

    }

if (ind == 2)

    {

        health = 1;

        difficulty.text = "HARD";

```

```

}}}

if (muc.index == 2 && Input.GetKeyDown(KeyCode.Return))
{
    if (isFpsLocked)
    {
        LOCKFPS.maxFPS = 120;
        isFpsLocked = false;
        mfps.text = "MAX FPS: 120";
    }
    else
    {
        LOCKFPS.maxFPS = 60;
        isFpsLocked = true;
        mfps.text = "MAX FPS: 60";
    }
}

if (muc.index == 3 && Input.GetKeyDown(KeyCode.Return))
{
    PlaySoundsFX.PlaySound("cancelFx");
    SceneManager.LoadScene("MainMenu");}}}

```

## VI. Bibliografie

1. „*C# 7.0 in a Nutshell: The Definitive Reference*” de **Joseph Albahari**
2. „*Learning C# Programming with Unity 3D*” de **Alex Okita**
3. „*C# in Depth*” de **Jon Skeet**
4. [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
5. <https://unity.com/>
6. <https://assetstore.unity.com/>
7. [https://ro.wikipedia.org/wiki/C\\_sharp](https://ro.wikipedia.org/wiki/C_sharp)
8. <https://www.math.uaic.ro/~cgales/csharp/Curs1.pdf>
9. <https://tutoriale-pe.net/introducere-limbajul-de-programare-c/>