

Colegiul Național „Moise Nicoară”

LUCRARE DE ATESTAT  
DISCIPLINA INFORMATICĂ

Candidat,

Giosu Robert-Michael

Coordonator,

Prof. Daniela Cameniță

2024

## QUIZTOPIA: SALVAREA REGATULUI

# Cuprins

<b>I.</b>	<b>Introducere</b>	
1.	Justificarea temei.....	2
2.	Necesitățile software-hardware.....	2
3.	Programele utilizate.....	2
<b>II.</b>	<b>Partea teoretică</b>	
1.	Limbajul C#.....	3
2.	Programarea în Unity.....	3
3.	Microsoft Visual Studio.....	3
<b>III.</b>	<b>Partea practică</b>	
1.	Interfața jocului.....	4
2.	Mișcarea jucătorului.....	5
3.	Mișcarea camerei.....	5
4.	Checkpoint-uri.....	6
5.	Quiz.....	6
<b>IV.</b>	<b>Concluzii.....</b>	<b>8</b>
<b>V.</b>	<b>Bibliografie.....</b>	<b>9</b>

## **I. Introducere**

### **1. Justificarea temei**

„Quiztopia: Salvarea Regatului” este un joc conceput pe baza șansei oferite de atestatul la informatică din clasa a XII-a de a crea propriul proiect.

Întrucât în ziua de azi jocurile 3D sunt predominante, jucătorii au pierdut din vedere esența, bazele de unde au plecat jocurile, și anume, clasicele jocuri 2D.

Prin acest proiect, am dorit să readuc farmecul jocurilor de odinioară, iar în același timp, să testez și să îmbogățesc cunoștințele de cultură generală a utilizatorilor. Acest fapt l-am realizat prin crearea unui joc de tip quiz, în care avem un împărat aflat într-o mare încurcătură. Acesta este domnitorul și conducătorul întregului continent, însă locul său pe tron este amenințat de pierderea unui obiect esențial acestuia: sabia magică, care s-a transmis fiecărui urmaș la tron. Însă, împăratul care a fost într-o călătorie plină de peripeții, în care s-a bătut cu zmeii, a fost neatent, iar sabia magică a fost pierdută. Jucătorul trebuie să-l ajute pe împărat să recapete puterea asupra regatului prin trecerea unor probe care testează cunoștințele, astfel încât sabia să poată fi recuperată.

Un alt motiv pentru care am ales această temă este pasiunea mea pentru jocuri și posibilitatea de a învăța un nou mod de a folosi limbajul C# pentru crearea unei aplicații practice.

### **2. Necesitățile software-hardware**

- Procesor: Intel Core i3 generația 2 sau mai recent
- Memorie RAM: minim 2 GB
- Placă video: Intel HD Graphics (integrată) sau orice placă dedicată
- Spațiu de stocare liber: aprox. 2 GB
- Sistem de operare: minim Windows 7

### **3. Programe utilizate**

- Unity
- Microsoft Visual Studio (pentru partea de cod C#)

## II. Partea teoretică

### 1. Limbajul C#

Lansat oficial în anul 2002, C# este un limbaj de programare care îmbină tehnici testate de-a lungul timpului cu inovațiile momentului. A fost creat de Microsoft, mai precis de echipa condusă de Anders Hejlsberg. Deși a fost conceput inițial pentru Windows, există compilatoare și pentru alte sisteme precum Linux sau Macintosh. Întrucât a fost creat ca un instrument de dezvoltare pentru arhitectura .NET, C# facilitează scrierea programelor pentru sistemul de operare Windows sau internet. În esență, C# derivă din alte două limbaje de programare, renumitele C și C++, având elemente în comun și cu Java.

### 2. Programarea în Unity

Unity este una dintre cele mai populare platforme de dezvoltare de jocuri disponibile în prezent. Ea oferă un mediu de lucru flexibil și puternic pentru dezvoltatorii de jocuri, indiferent de nivelul lor de experiență. Avantajele acestei platforme sunt numeroase:

- Este cross-platform, permițând dezvoltatorilor să creeze jocuri care rulează pe o gamă largă de platforme, inclusiv PC, console, dispozitive mobile și chiar VR/AR.
- Oferă un motor grafic puternic, care permite dezvoltatorilor să creeze jocuri cu grafică impresionantă și efecte vizuale captivante.
- Suportă mai multe limbaje de programare, precum C# și JavaScript, permițând utilizarea acestora de către o gamă largă de programatori cu diferite competențe.
- Comunitatea este vastă și activă, fiind accesibile resurse, sfaturi și suport tehnic. În plus, există numeroase forumuri, tutoriale și evenimente dedicate dezvoltatorilor Unity.

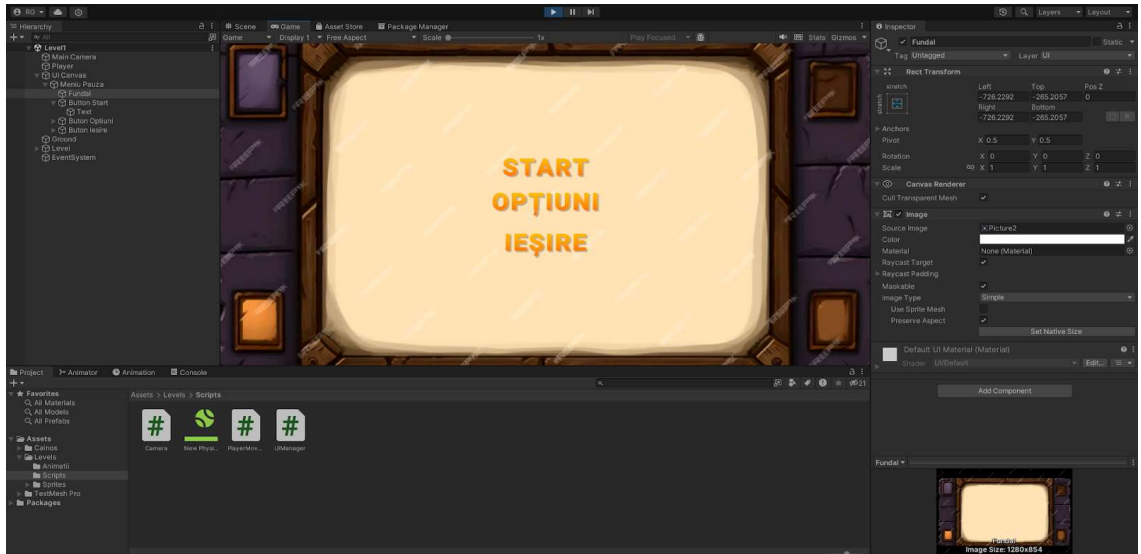
### 3. Microsoft Visual Studio

Este un mediu integrat de dezvoltare (IDE), utilizat pentru crearea aplicațiilor desktop, web și mobile. Printre principalele avantaje, se numără suportul pentru multiple limbaje de programare, funcționalități avansate de debug și integrarea armonioasă cu celelalte aplicații Microsoft, precum Azure, Team Foundation Server și GitHub.

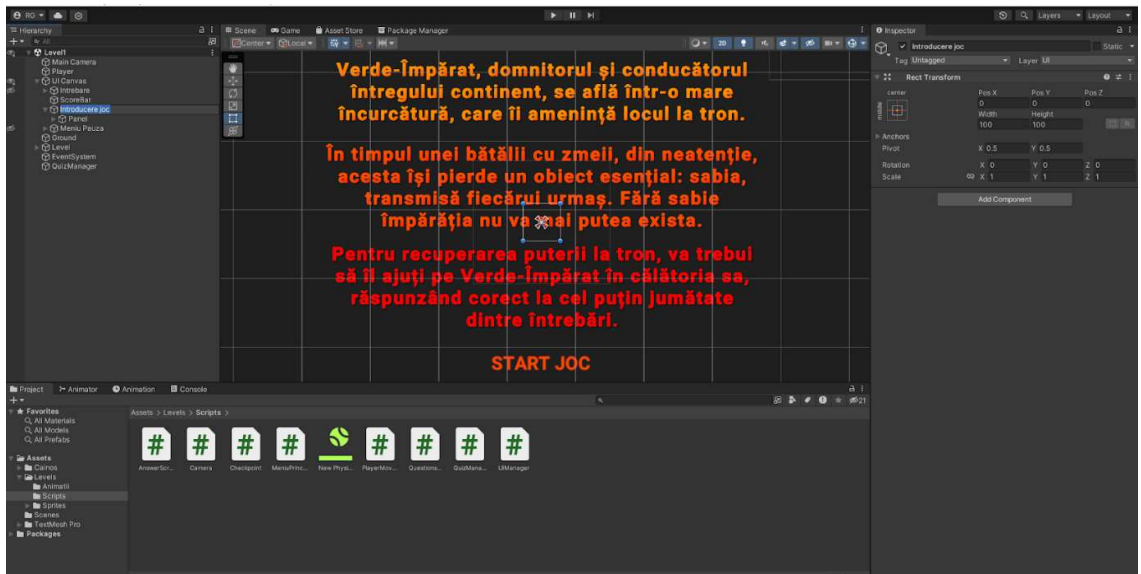
### III. Partea practică

#### 1. Interfața jocului

Când se pornește jocul, utilizatorul are un meniu inițial în care poate face anumite setări, precum volumul sunetului jocului, poate porni jocul sau să iasă complet din joc.



După ce se pornește jocul, va fi făcută o scurtă introducere în care este prezentată situația împăratului, apoi jucătorul va porni în călătorie, apăsând butonul de “start joc”.



## 2. Mișcarea jucătorului

Se bazează pe funcția `Input.GetAxis("horizontal")`, care returnează valori între -1 și 1, -1 când jucătorul merge spre stânga, și 1 spre dreapta. Vectorul de poziție al jucătorului se actualizează corespunzător, pe axa Ox scăzându-se sau mărindu-se, în funcție de caz, valorile. Dacă se detectează apăsarea tastei space, vectorul de poziții își va modifica valorile pe axa Oy.

```
private void Update()
{
    float horizontalInput = Input.GetAxis("Horizontal");//returneaza valori între -1 si 1 in functie de tasta apasata A/D
    body.velocity = new Vector2(horizontalInput * speed, body.velocity.y);

    //stabilim in ce directie merge caracterul si stabilim directia in care se uita
    if (horizontalInput > 0.01f)//merge la stanga
        transform.localScale = new Vector3(1.5f, 1.5f, 1.5f);
    else if (horizontalInput < -0.01f)//merge la dreapta
        transform.localScale = new Vector3(-1.5f, 1.5f, 1.5f);

    //saritura
    if (Input.GetKey(KeyCode.Space) && isGrounded())
        Jump();
}
```

## 3. Mișcarea camerei

- Linia `transform.position = new Vector3(player.position.x + LookAhead, transform.position.y, transform.position.z)`; actualizează poziția camerei pe axa X, astfel încât camera să fie mereu la o anumită distanță în fața jucătorului. `LookAhead` este adăugat la poziția X a jucătorului pentru a obține poziția camerei.
- Linia `LookAhead = Mathf.Lerp(LookAhead, (AheadDistance * player.localScale.x), Time.deltaTime * CameraSpeed)`; ajustează cât de mult în față trebuie să se uite camera în funcție de dimensiunea jucătorului și de viteza camerei.

```
4
5 public class Camera : MonoBehaviour
6 {
7     [SerializeField] private float speed;
8     private float pozX;
9     private Vector3 viteza = Vector3.zero;
10    [SerializeField] private Transform player;
11    [SerializeField] private float AheadDistance;
12    [SerializeField] private float CameraSpeed;
13    private float LookAhead;
14    void Update()
15    {
16        //transform.position = Vector3.SmoothDamp(transform.position, new Vector3(pozX, transform.position.y, transform.position.z), ref viteza, speed * Time.deltaTime);
17        transform.position = new Vector3(player.position.x + LookAhead, transform.position.y, transform.position.z);
18        LookAhead = Mathf.Lerp(LookAhead, (AheadDistance * player.localScale.x), Time.deltaTime * CameraSpeed);
19    }
20
21 }
```

#### 4. Checkpoint-uri

Când jucătorul atinge checkpointul, se deschide quiz-ul, jucătorul având 20 secunde în care să răspundă la mai multe întrebări. După ce expiră cele 20 secunde, se închide pagina de quiz, iar scorul este actualizat.

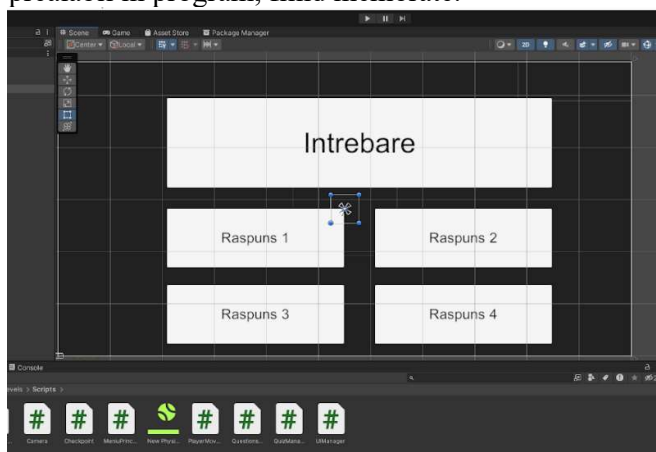
```
public class Checkpoint : MonoBehaviour
{
    [SerializeField] GameObject Quiz;
    private float timeRemaining = 20f; // 20 seconds
    private bool isQuestionVisible = false;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if(collision.tag == "Player")
        {
            Debug.Log("Generate question");
            Quiz.SetActive(true);
            isQuestionVisible = true;
            timeRemaining = 10f;
        }
    }

    void Update()
    {
        if (isQuestionVisible)
        {
            if (timeRemaining > 0)
            {
                timeRemaining -= Time.deltaTime;
            }
            else
            {
                Debug.Log("Timer expired!");
                GetComponent<BoxCollider2D>().enabled = false;
                Quiz.SetActive(false);
                isQuestionVisible = false;
            }
        }
    }
}
```

#### 5. Quiz

Am început prin crearea ecranului de quiz care va apărea atunci când un checkpoint este atins. Acesta conține o casetă de text în care se generează aleatoriu (prin funcția Random.Range), o întrebare care nu a mai fost răspunsă deja. Întrebările sunt adăugate în prealabil în program, fiind memorate.





```

public class QuizManager : MonoBehaviour
{
    public List<QuestionsAndAnswers> QnA;
    public GameObject[] options;
    public int currentQuestion;
    public Text QuestionText;
    public Text ScoreText;
    public int score = 0;

    private void Start()
    {
        ScoreText.text = "SCOR: 0";
        generateQuestion();
    }

    public void correct()
    {
        score = score + 10;
        ScoreText.text = "SCOR:" + score;
        QnA.RemoveAt(currentQuestion);
        generateQuestion();
    }

    public void wrong()
    {
        QnA.RemoveAt(currentQuestion);
        generateQuestion();
    }

    void setAnswers()
    {
        for(int i=0; i<options.Length; i++)
        {
            options[i].GetComponent<AnswerScript>().isCorrect = false;
            options[i].transform.GetChild(0).GetComponent<Text>().text = QnA[currentQuestion].Answers[i];
            if (QnA[currentQuestion].CorrectAnswer == i+1)
            {
                options[i].GetComponent<AnswerScript>().isCorrect = true;
            }
        }
    }

    void generateQuestion()
    {
        if (QnA.Count > 0)
        {
            currentQuestion = Random.Range(0, QnA.Count);
            QuestionText.text = QnA[currentQuestion].Question;
            setAnswers();
        }
    }
}

```

- a. public List<QuestionsAndAnswers> QnA; - Declarația unei liste care va conține întrebările și răspunsurile pentru quiz.
- b. public GameObject[] options; - Declarația unui tablou care va conține opțiunile de răspuns.
- c. public int currentQuestion; - Variabila pentru a ține evidența întrebării curente.
- d. public Text QuestionText; - Referință către obiectul de text care va afișa întrebarea.
- e. public Text ScoreText; - Referință către obiectul de text care va afișa scorul.
- f. private void Start() {...} - Funcția de inițializare a quiz-ului.
- g. generateQuestion(); - Generarea unei întrebări aleatorii.
- h. currentQuestion = Random.Range(0, QnA.Count); - Alege o întrebare aleatorie din lista de întrebări disponibile.
- i. QuestionText.text=QnA[currentQuestion].Question; - Afișează întrebarea curentă.
- j. setAnswers(); - Setează opțiunile de răspuns pentru întrebarea curentă.

#### **IV. Concluzii**

Jocul „Quiztopia: Salvarea Regatului” a fost conceput în scopul lucrării de atestat la informatică. Consider că prin crearea acestui joc 2D, voi reuși să îi fac pe jucători să experimenteze tipul de joc care a stat la bazele propulsării industriei jocurilor. Mai mult, utilizatorii de vârstă noastră vor putea să își readucă aminte de jocurile și perioada nostalgică a copilăriei și să își testeze și să își îmbunătățească cunoștințele de cultură generală.

Printre avantajele aplicației, se numără interfața user-friendly, care permite și celor mai neexperimentați să pătrundă în lumea virtuală. În plus, cerințele reduse de sistem înseamnă că jocul poate fi rulat pe aproape orice dispozitiv. Un alt avantaj este caracterul educativ, astfel că jocul îmbogățește cunoștințele de cultură generală.

Un punct slab este reprezentat de lipsa diversității. Pe viitor, aplicația poate fi dezvoltată, astfel încât să prezinte mai multe nivele la care dificultatea să crească progresiv, în funcție de cunoștințele jucătorului. Un alt aspect este popularitatea jocurilor 2D, care a scăzut drastic odată cu apariția jocurilor 3D. Însă, există încă jucători dornici să experimenteze noile jocuri 2D, fiind atrași de acest „univers” mai simplist care să le aducă, probabil, aminte de copilărie.

## **V. Bibliografie**

<https://www.youtube.com/watch?v=TcranVQUQ5U>

[https://www.youtube.com/watch?v=zc8ac\\_qUXQY](https://www.youtube.com/watch?v=zc8ac_qUXQY)

<https://www.youtube.com/watch?v=G9QDFB2RQGA>

<https://www.youtube.com/watch?v=POUemIGCyr0&t=454s>

<https://docs.unity3d.com/Manual/Unity2D.html>

<https://www.math.uaic.ro/~cgales/csharp/Curs1.pdf>