

Colegiul Național „Mihai Viteazul” Turda
Matematică-informatică, intensiv informatică
Atestat informatică

CAMPFIRE

Autor
Stancu Tudor-Andrei

Profesori îndrumători
prof. Miron Florin
prof. Pop Daniela

2020

Cuprins

Descriere Generala	3
Motivul alegerii temei	3
Resurse Hardware si Software	3
Realizarea aplicației	4
Construcția scenei jocului	4
Baza de date	6
Secvențe importante de cod	7
Manual de utilizare	10
Posibile îmbunătățiri	14
Bibliografie	15

Descriere Generală

Aplicația Campfire este un joc de tip 2D în care jucătorul controlează un mic personaj cu ajutorul tastaturii. Scopul jocului este de a menține focul aprins, punând lemne pe acesta. Lemnele necesare apar la întâmplare pe parcursul jocului. Odată cu acumularea punctelor, focul începe să se stingă din ce în ce mai repede, jucătorul fiind din ce în ce mai solicitat. Dacă focul se stinge, jocul se termină iar numărul de lemne plasate reprezintă scorul.

Motivul alegerii temei

Tema acestei lucrări este datorată pasiunii mele pentru jocuri, aceasta fiind puternic înrădăcinată încă din copilărie. Astfel, acest proiect mi-a oferit ocazia de a realiza un joc de unul singur.

Resurse Hardware si Software

1.Hardware

- Procesor 1.6 GHz sau mai bun
- 258 MB RAM
- 90 MB spatiu pe disk
- Unitatea optică cu DVD-ROM

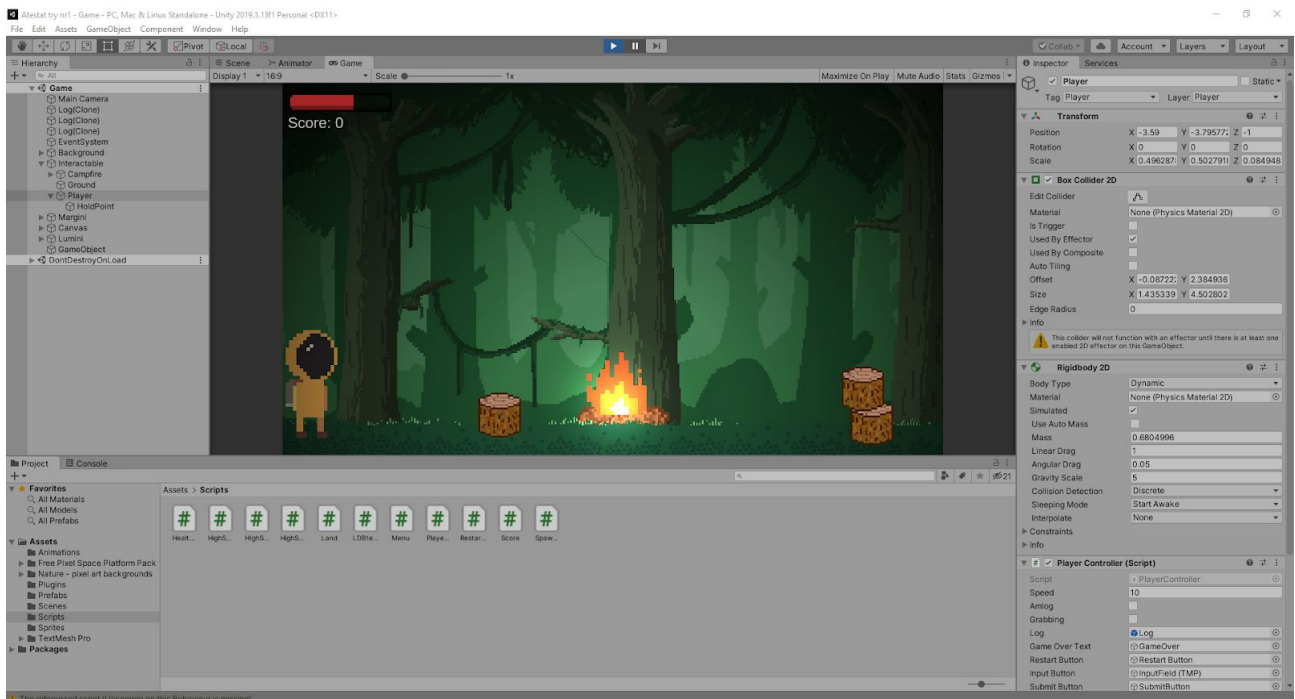
2.Software

- Windows XP(x86) Service Pack 3
- Windows 7(x86 sau x64)
- Windows 8/8.1(x86 sau x64)
- Windows 10(x86 sau x64)

Realizarea aplicației

Construcția scenei jocului

Aplicația conține 2 scene: scena meniului și scena jocului. Scena jocului conține mai multe obiecte de tip GameObject (jucătorul, focul, lemnele, platforma cu rol de podea) și obiecte de tip UI (bara de viață și scorul) care sunt amplasate pe un obiect de tip Canvas.

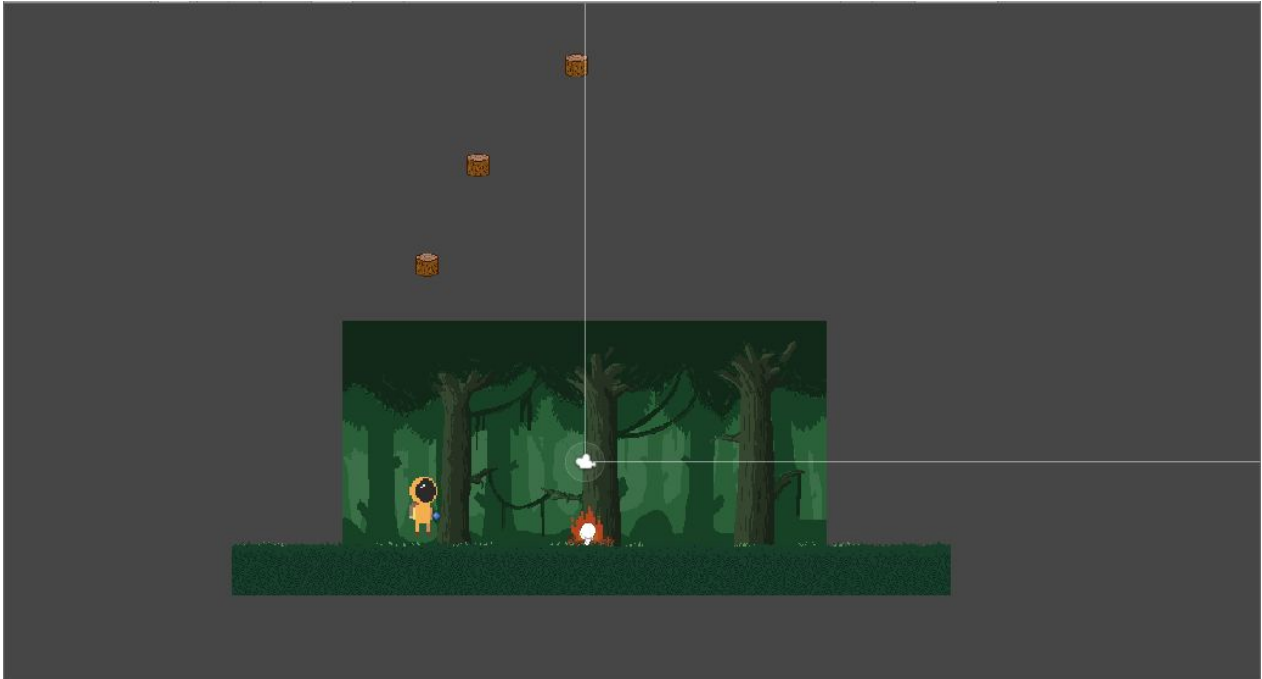


În partea stânga sus se află ierarhia jocului. Acolo se așează obiectele în scenă și sunt aplicate diferite relații între ele, de tipul părinte-copil.

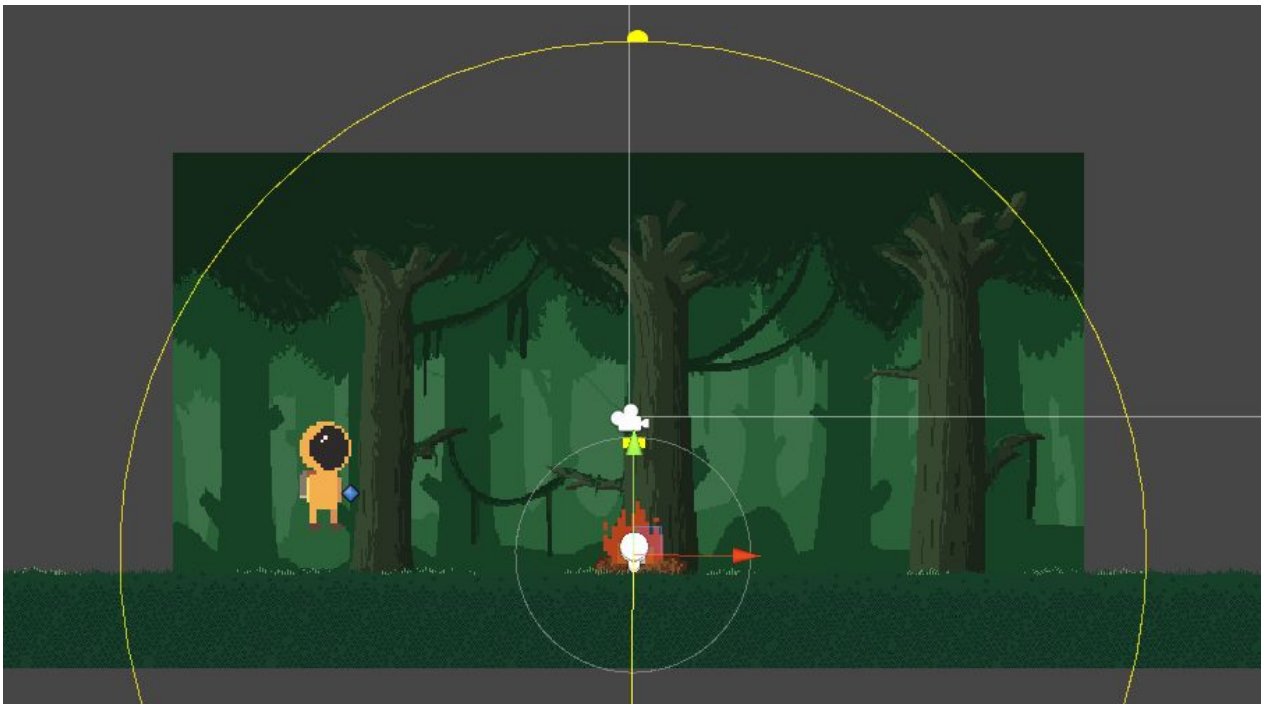
În partea dreaptă este inspectorul de obiecte. Aici pot fi schimbate proprietățile obiectelor aflate în scenă.

Obiectul numit Player este controlat de către jucător. Acesta are atașată o componentă de tip Rigidbody2D care se ocupă de partea de fizici ale obiectului, o componentă de tip Box Collider 2D care determină interacțiunile dintre corpuri și un script numit Player Controller.

La începerea jocului sunt instanțiate în afara cadrului 3 lemne, iar după este menținut un număr constant de 2 lemne.

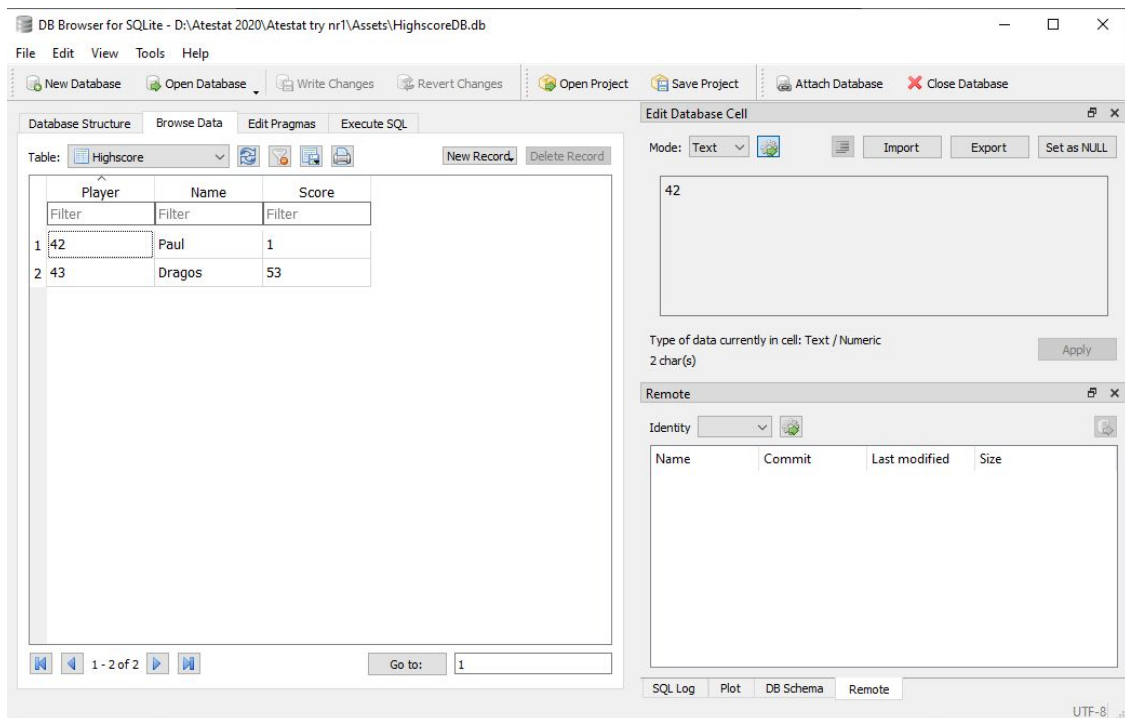


Deasupra focului se găsește un gameObject de tip Point Light 2D. Acesta este o sursă de lumină care devine din ce în ce mai slabă cu cât radiusul este mai mare.



Baze de date

Baza de date a fost realizată utilizând SQLite. SQLite este o mică bibliotecă C care implementează un motor de baze de date SQL încapsulat, oferă posibilitatea de a-l introduce în diverse sisteme și necesită zero-configurare. Baza de date folosită în aplicații conține un singur tabel numit Highscore care reține: doua câmpuri de tip INTEGER, unul pentru diferențiere iar unul pentru scor, și un camp de tip TEXT care reține numele asociat scorului. Pentru a crea și vizualiza baza de date a fost folosit un program extern numit DB Browser.



Secvențe importante de cod

Secvența următoare determină mișcarea jucătorului pe axa orizontală prin modificarea vitezei acestuia. Funcția `Input.GetAxisRaw` returnează -1 dacă jucătorul merge la stânga, 0 dacă stă pe loc sau 1 dacă merge la dreapta. Astfel putem face ca modelul jucătorului să se întoarcă înspre direcția în care merge folosind funcția `transform.eulerAngles`.

```
0 references
private void Update()
{
    moveInput = Input.GetAxisRaw("Horizontal");
    rb.velocity = new Vector2(moveInput * speed, rb.velocity.y);
    animator.SetFloat("Speed", Mathf.Abs(moveInput));

    if (moveInput == 1)
    {
        transform.eulerAngles = new Vector3(0, 0, 0);
    }

    if (moveInput == -1)
    {
        transform.eulerAngles = new Vector3(0, 180, 0);
    }
}
```

Prima dintre cele două verificări are rolul de a ridica un lemn, verificând dacă este apăsat butonul destinat acțiunii, dacă jucătorul nu are deja în posesie un lemn, și dacă face contact cu acesta. A doua verificare are rolul de a plasa pe foc un lemn urmărind dacă este apăsat butonul corect, când jucătorul are în posesie un lemn și dacă face contact cu focul. Verificarea scurtă dintre celelalte două se asigură ca după ce a fost ridicat un lemn, acesta stă într-o poziție prestabilită în raport cu jucătorul.

```
if (Input.GetKeyDown(KeyCode.K) && pickupAllowed == true && amlog == false)
{
    pickupAllowed = false;
    amlog = true;
    numar.lemnuri--;
    grabbing = true;
    Log.transform.position = holdpoint.position;
}

if (grabbing == true)
    Log.transform.position = holdpoint.position;

if (Input.GetKeyDown(KeyCode.L) && place == true && amlog == true)
{
    amlog = false;
    if (hpcampfire + 10 < 200)
        hpcampfire += 10;
    else
        hpcampfire = 200;

    PlayerPrefs.SetInt("Highscore", PlayerPrefs.GetInt("Highscore") + 1);

    if (PlayerPrefs.GetInt("Highscore") % 5 == 0 && PlayerPrefs.GetInt("Highscore") != 0)
        dificultate++;

    grabbing = false;
    Destroy(Log);
}
```

Funcția de mai de jos se asigură că în orice moment al jocului, în cadru se află două lemne disponibile iar funcția spawnLog() este apelată de fiecare dată când trebuie instanțiat un nou lemn. Poziția acestuia este luată aleatoriu din intervalul determinat de limitele laterale ale camerei.

```
0 references
public void Update()
{
    if (nrLogs < 2)
    {
        Debug.Log(nrLogs);
        spawnLog();
        nrLogs++;
    }
}

1 reference
private void spawnLog()
{
    GameObject a = Instantiate(logPrefab) as GameObject;
    a.transform.position = new Vector2(Random.Range(-screenBounds.x, screenBounds.x), screenBounds.y * 2);
}
```

Următoarea secvență se asigură că bara de viață se află în corelație cu viața jucătorului, acest lucru fiind realizat prin updatarea constantă a valorii sliderului.

```
1 reference
public class HealthBar : MonoBehaviour
{
    public Slider slider;
    1 reference
    public void SetMaxHealth(float maxHealth)
    {
        slider.maxValue = maxHealth;
        slider.value = maxHealth;
    }
    1 reference
    public void SetHealth(float Health)
    {
        slider.value = Health;
    }
}
```

Partea de baze de date este gestionată cu ajutorul următoarelor funcții. Acestea au rolul de a introduce date în tabelă, de a prelua toate rândurile din tabelă și de a completa primele 10 randuri de elemente dintr-un obiect UI de tip Grid Layout Group.


```

public void InsertScore(string name, int newScore)
{
    using (IDbConnection dbConnection = new SqliteConnection(connectionString))
    {
        dbConnection.Open();
        using (IDbCommand dbCmd = dbConnection.CreateCommand())
        {
            string sqlQuery = String.Format("INSERT INTO Highscore(name,score) VALUES(\"{0}\", \"{1}\")", name, newScore);
            dbCmd.CommandText = sqlQuery;
            dbCmd.ExecuteNonQuery();
            dbConnection.Close();
        }
    }
}

```

```

private void GetScores()
{
    highScores.Clear();
    using (IDbConnection dbConnection = new SqliteConnection(connectionString))
    {
        dbConnection.Open();
        using (IDbCommand dbCmd = dbConnection.CreateCommand())
        {
            string sqlQuery = "Select * FROM Highscore";

            dbCmd.CommandText = sqlQuery;
            using (IDataReader reader = dbCmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    highScores.Add(new HighScore(reader.GetInt32(0), reader.GetInt32(2), reader.GetString(1)));
                }
                reader.Close();
            }
            dbConnection.Close();
        }
    }
    highScores.Sort();
}

```

```

private void ShowScores()
{
    GetScores();
    for(int i=0; i < 10;i++)
    {
        if (i <= highScores.Count - 1)
        {
            GameObject tmpobj = Instantiate(scorePrefab);
            HighScore tmpscore = highScores[i];
            tmpobj.GetComponent<HighScoreScript>().SetScore(tmpscore.Name, tmpscore.Score.ToString(), "#" + (i + 1).ToString());
            tmpobj.transform.SetParent(scoreParent);
            tmpobj.GetComponent<RectTransform>().localScale = new Vector3(1, 1, 1);
        }
    }
}

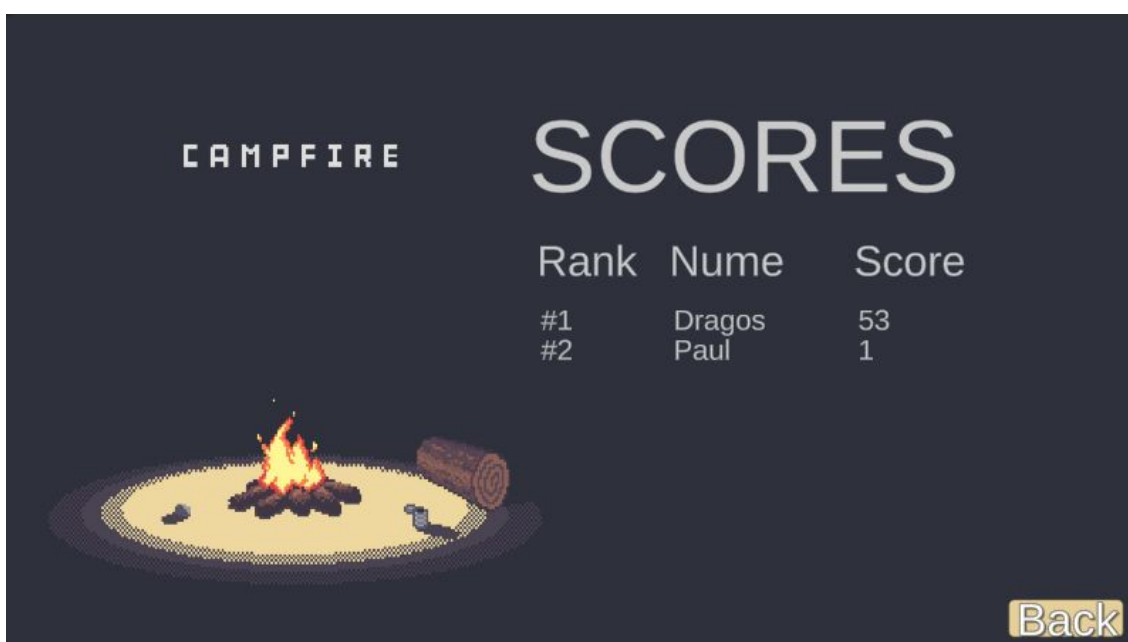
```

Manual de utilizare

După ce se lansează jocul prin dublu click pe executabil, aplicația deschide prima scenă, cea de Meniu. Acesta cuprinde patru butoane: “PLAY”, “SCORES”, “CONTROLS” și “EXIT”.



Butonul “SCORES” afișează 10 dintre cele mai mari scoruri în ordine crescătoare alături de numele și rankul jucătorului.



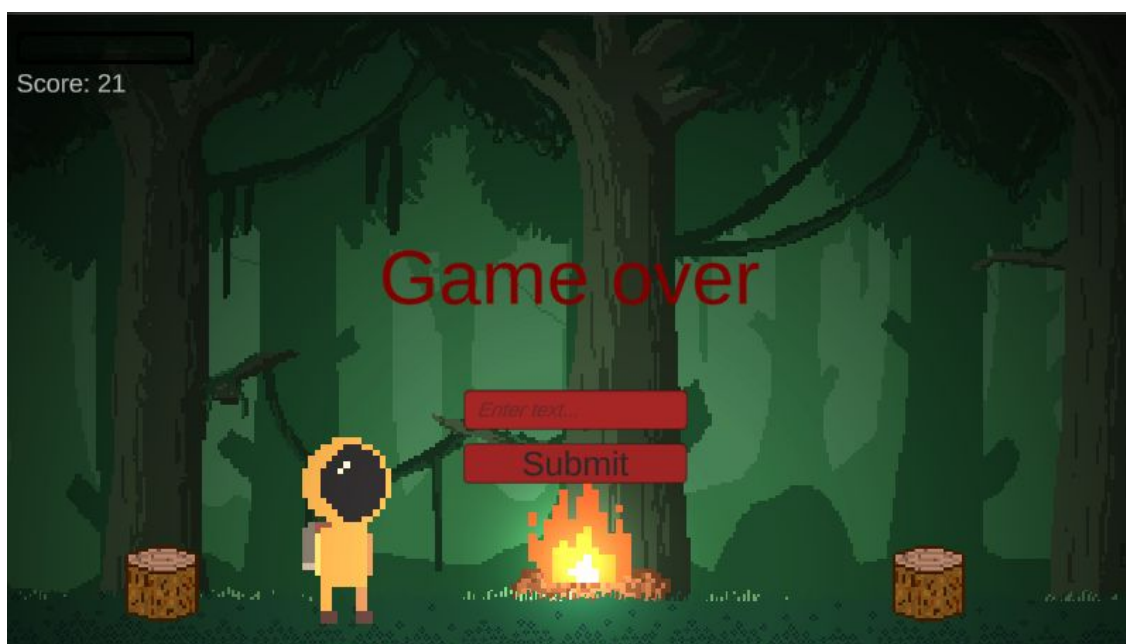
Butonul "CONTROLS" afișează un obiect text ce conține toate controalele necesare pentru a juca jocul.



La apăsarea butonului "PLAY", cea de-a doua scenă începe. La început, jucătorul se afla în partea din stanga a camerei, focul aflându-se în mijloc iar lemnele apărând din exteriorul cadrului. În orice moment din joc, în cadru se afla 2 lemne pe care jucătorul le poate ridica de pe jos și pune pe foc apasand tastele K respectiv L. Mișcarea este realizata apasand tastele A sau D. Pentru fiecare lemn pus, jucătorul primește un punct.



Dacă focul se stinge, adică bara de viață se epuizează, jocul se oprește iar o nouă interfață apare. Un nou gameobject de tip Input Field este disponibil unde jucătorul își introduce numele pentru a putea fi salvat în baza de date.



Apăsarea butonului "Submit" determina închiderea sesiunii jocului iar utilizatorul va fi trimis înapoi în meniul de start.

Posibile îmbunătățiri

- adaugarea unor noi tipuri de lemne care interacționează cu jucătorul
- adăugarea unor platforme
- îmbunătățirea fizicilor lemnurilor care apar
- noi texturi pentru elementele jocului
- elemente de UI noi

Bibliografie

<https://docs.unity3d.com/ScriptReference/>

<https://www.youtube.com/watch?v=nkgGyO9VG54>

<https://www.youtube.com/watch?v=wi-RL4sUayo>

<https://discord.com/invite/brackeys>

<https://discord.com/invite/unity>

<https://ro.wikipedia.org/wiki/SQL>

https://www.youtube.com/watch?v=laspFwXGprg&list=PLX-uZVK_OK_7NmsYfe2BTOk_lamWC2kU3

https://www.youtube.com/watch?v=BLfNP4Sc_iA