

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI  
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

**Joc 3D pe platforma Unity**

propusă de

***Alexandru-Dănuț Zaharia***

**Sesiunea:** *Februarie, 2020*

Coordonator științific

**Lector. Dr. Alex Moruz**

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ

# Joc 3D pe platforma Unity

***Alexandru-Dănuț Zaharia***

**Sesiunea:** *Februarie, 2020*

Coordonator științific

***Lector. Dr. Alex Moruz***

Avizat,  
Îndrumător Lucrare de Licență  
Titlul, Numele și prenumele

Data \_\_\_\_\_  
Semnătura \_\_\_\_\_

### **DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul(a) .....

domiciliul în .....

născut(ă) la data de ....., identificat prin CNP .....,  
absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de  
..... specializarea ....., promoția  
....., declar pe propria răspundere, cunoscând consecințele falsului în  
declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale  
nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_elaborată sub îndrumarea dl. / d-na  
\_\_\_\_\_, pe care urmează să o susțină în fața  
comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată  
prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la  
introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări  
științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei  
lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie

răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi, .....

Semnătură student .....

## DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Joc 3D pe platforma Unity*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, *data*

Absolvent *Alexandru-Dănuț Zaharia*

---

(semnătura în original)

## **Cuprins:**

<b>INTRODUCERE ȘI MOTIVAȚIE</b>	7
<b>CAPITOLUL I. PLATFORMA DE DEZVOLTARE</b>	8
<b>I.1. Jocuri Horror</b>	9
<b>I.1.1. Scurt istoric</b>	9
<b>I.1.2. Jocuri populare pe platforma Unity</b>	9
<b>I.2. Tehnologii folosite</b>	11
<b>I.2.1 Limbajul C#</b>	11
<b>I.2.2 Platforma .NET</b>	12
<b>I.2.4 Unity</b>	12
<b>I.2.5 Assets</b>	14
<b>I.2.6 Sunet</b>	15
<b>CAPITOLUL II. PREZENTAREA APLICAȚIEI</b>	16
<b>II.1 Specificații funcționale</b>	16
<b>II.2 Arhitectura Aplicației</b>	17
<b>II.3 Harta</b>	19
<b>II.4 Zombie</b>	20
<b>II.5 Meniu și Intro</b>	23
<b>II.6 Obiecte utilizabile</b>	24
<b>CONCLUZIILE LUCRĂRII</b>	27
<b>1. Concluzii și contribuții personale</b>	27
<b>2. Posibilitatea extinderii pe viitor</b>	28
<b>BIBLIOGRAFIE ȘI SITOGRAFIE</b>	29

## Introducere și motivație

Un joc video este un joc electronic care implică interacțiune cu o interfață de utilizator pentru a genera feedback vizual pe un dispozitiv de afișare video în două sau trei dimensiuni, cum ar fi un ecran tactil, căști de realitate virtuală sau monitor / televizor. Începând cu anii 80, jocurile video au devenit o parte din ce în ce mai importantă a industriei divertismentului.

Motivul pentru care lumea este atrasă de jocurile video este acela că evidențiază din ce în ce mai multe beneficii. Mark D. Griffiths este un psiholog englez care se concentrează în domeniul dependențelor de comportament, respectiv tulburarea jocurilor de noroc, dependența de jocuri, dependența de internet și dependența de muncă. El susține faptul că jocurile video dețin un potențial instructiv din cel puțin patru motive:

- Atrag participarea oamenilor, indiferent de vârstă, cultură sau statutul acestora;
- Ajută copiii în stabilirea unor obiective;
- Oferă un feedback constant și susțin jucătorii în îndeplinirea sarcinilor;
- Sunt distractive și antrenante;<sup>1</sup>

În această lucrare, voi prezenta cum am dezvoltat un joc 3D în Unity, care are ca scop controlarea unui personaj de tip “first person” (acțiunea este văzută din ochii personajului). Acest personaj trebuie să afle despre poveștile dintr-un anumit oraș și dintr-o pădure, fiind în același timp urmărit de anumiți zombie care au intenția de a-l omorî, iar el trebuie să supraviețuiască. Spațiul în care se desfășoară acțiunea este o hartă modelată după imaginația mea.

Am creat jocul cu ajutorul anumitor tutoriale despre unity de pe **YouTube**, al unor scripturi în **limbajul de programare C#**, al unor **asset-uri gratuite** de pe platforma unity și a unor clipuri audio luate de pe internet aplicate obiectelor de pe platforma unde a rezultat jocul prezentat mai sus.

---

<sup>1</sup> Meredith Watkins, *Video Game Addiction Symptoms and Treatment*, <https://americanaddictioncenters.org/video-gaming-addiction>.

## Capitolul I. Platforma de Dezvoltare

Unul dintre lucrurile fascinante despre a fi un dezvoltator de jocuri în ziua de astăzi este numărul mare de resurse disponibile de pe internet. În momentul de față, piața platformelor de dezvoltare a jocurilor ofera o gamă imensă de opțiuni. Acestea fiind accesibile, oricine ar avea o idee ar putea începe dezvoltarea unui joc. Platformele cele mai populare au ajuns într-un stadiu atât de avansat, încât pentru crearea unui joc nu sunt necesare foarte multe cunoștințe de programare. Fiecare platformă dispune, pe lângă documentație, și de câte o comunitate imensă formată din persoane care au mai lucrat pe platforma respectivă, așadar, indiferent de nivelul de experiență se poate oferi asistență pentru anumite nivele de dificultate în crearea jocului.

O să enumăr câteva platforme de dezvoltare renumite :

### Unity

Multă vreme, mulți dezvoltatori au subestimat Unity, dar asta nu se mai întâmplă acum. Unity este un instrument de creare a jocului pe scară largă, care poate fi folosit atât pentru realiza jocuri 2D , cât și 3D pe aproape orice platforma de dezvoltare.

### RPG Maker

RPG-urile sunt unul dintre cele mai populare genuri din industrie. Datorita RPG Maker, multi dezvoltatori au avut șansa de a-și face propriile aventuri epice cu acest motor de dezvoltare a jocului.

### Unreal Engine 4

În 2015, Epic Games a anunțat că Unreal Engine 4<sup>2</sup> va putea fi descărcat și utilizat pentru toată lumea. Acest renumit set de instrumente de dezvoltare a jocurilor a fost folosit pentru succesul pe aproape orice platforma .

---

<sup>2</sup> <https://www.unrealengine.com/what-is-unreal-engine-4>.



## I.1 Jocuri Horror

În acest capitol se vor găsi descrieri teoretice ale unor concepte care au influențat procesul de dezvoltare al aplicației prezentate în această lucrare.

### I.1.1 Scurt istoric

Atât filmele, cât și jocurile se pot împărți în mai multe categorii, printre care și cea horror (care este și printre cele mai ușor de observat). Un aspect interesant în privința acestui gen este faptul că este definit de modul în care intenționează să afecteze consumatorul. În timp ce unele genuri precum science fiction și western sunt definite de către acțiunile descrise și timpul și spațiul în care acestea au loc, altele, precum comedie și horror sunt definite pe baza unor anumite emoții provocate la nivelul consumatorului.

Cunoscutul vlogger de pe Youtube PewDiePie a popularizat în mare parte ca urmare a unei vânzări de jocuri de groază<sup>3</sup> (de exemplu, primul videoclip de la **Outlast** avea peste 16 milioane de vizualizări). Aceste tipuri de jocuri sunt interesante pentru că îți pot oferi un sentiment de control, știind că poți ocoli pericolul.

### I.1.2. Jocuri populare pe platforma Unity

#### De ce folosim Unity?

Este gratuit, de înaltă calitate, are suport excelent pentru platformă și are o comunitate uriașă și activă de profesioniști și pasionați care contribuie la baza de cunoștințe și de resurse de învățare, precum și la magazinul „asset store”. „Asset store” poate scurta zeci de mii de dolari de investit în cercetare și dezvoltare dacă știți ce fel de lucruri să căutați. Sistemul unity pentru programarea *asset*-urilor și instrumentelor noi este printre cele mai flexibile și mai

---

<sup>3</sup> Patricia Hernandez, *What People Get Wrong About PewDiePie, YouTube's Biggest Star*, <https://kotaku.com/what-people-get-wrong-about-pewdiepie-youtubes-biggest-1673109786>.

extensibile disponibile, care permit jocuri în 2D și 3D, care permite crearea unui flux de lucru propriu.

### **Ce se poate face cu Unity? <sup>4</sup>**

Datorită complexității sale, este aproape imposibil să avem o imagine completă a întregului conținut realizat cu motorul Unity. Iată o serie de câteva jocuri făcute în Unity care au făcut diferența prin vizuale uimitoare, precum și prin narațiuni și sunete grozave.

### **Hearthstone**

La baza sa, Hearthstone este un joc de cărți de colecție care se învârtă în jurul meciurilor online bazate pe rând, între doi adversari. Jucătorii câștigă un set de cărți de bază și pot obține carduri „avansate” achiziționând pachete de cărți cu aur câștigat în joc sau cu bani reali. La crearea acestui joc, echipa de la Blizzard nu a avut atât timp să parcurgă ciclurile sale tradiționale de iterație și s-a bazat foarte mult pe Unity. Motorul le-a permis să identifice erori și să facă modificări corecte și, de asemenea, a ajutat echipa să învețe mai rapid despre îmbunătățirea jocului lor.

### **Assassin's Creed: Identity**

Assassin's Creed nu are nevoie de o introducere sau o descriere specială. Identity este cel mai recent din seria AC, actualmente în versiune beta și setat pentru o lansare la nivel mondial la începutul anului 2015. Assassin's Creed Identity este primul joc din universul Assassin's Creed, construit pentru iPad și iPhone, și este un RPG 3D pentru o a treia persoană. Permite jucătorilor să își creeze, să dezvolte și să își personalizeze propriii asasini și îi permite să joace în cele mai cunoscute medii AC - Renașterea italiană. Este construit pentru iOS 7, iPad3 sau o versiune ulterioară sau pentru iPhone5 sau o versiune ulterioară, iar versiunile mai vechi nu sunt acceptate.

---

<sup>4</sup> <http://unity3d.com/>

## **Praey for the Gods<sup>5</sup>**

Praey for the Gods este o aventură extremă într-un univers totemic, unde jucătorii sunt trimiși într-o insulă înghețată pustie, pentru a descoperi misterul din spatele unei ierni fără sfârșit și pentru a învinge fiarele enorme care apără teritoriul. Pentru a fi lansat pe Windows prin Steam, apoi pe PlayStation 4 și Xbox One, este un joc neliniar în care trebuie să alegeți cu cine luptați și cum, să traversați terenul cu zăpadă dinamică, să explorați temple misterioase care conțin secrete și comori și să încercați să supraviețuiți. Tot în acest capitol o sa prezint cateva jocuri horror care nu sunt facute in unity , dar care m-au ajutat si motivat pentru a crea acest joc:

- Amnesia: The Dark Descent (PC)
- Outlast (PC, Xbox One, and PS4)
- Resident Evil HD remaster (PC, Xbox One and PS4)
- Alien: Isolation (PC, Xbox One and PS4)

## **I.2. Tehnologii folosite**

În acest capitol voi prezenta platforma Unity, limbajul de programare folosit, precum și framework-ul folosit.

### **I.2.1. Limbajul C#**

C # este un limbaj de programare puternic orientat pe obiecte. C # este open source, simplu, modern, flexibil și versatil. În acest subcapitol, să învățăm ce este C #, ce poate face C # și cum diferă C # decât C ++ și alte limbaje de programare. Un limbaj de programare pe informatica este un limbaj care este folosit pentru a scrie programe software.

---

<sup>5</sup> <https://www.praeyforthegods.com/>

C # este un limbaj de programare dezvoltat și lansat de Microsoft în 2001. Este un limbaj orientat pe obiecte, care oferă dezvoltatorilor moderni flexibilitate și funcții pentru a construi software care nu va funcționa doar astăzi, dar va fi aplicabil ani de zile în viitor.

Caracteristicile cheie ale limbajului C # includ:

1. Modern și ușor
2. Sursă rapidă și deschisă

### **I.2.2. Platforma .NET**

Cadrul .Net este o platformă de dezvoltare software dezvoltată de Microsoft. Cadrul a fost menit să creeze aplicații, care să ruleze pe platforma Windows. Prima versiune a cadrului .Net a fost lansată în anul 2002. Versiunea a fost numită .Net framework 1.0. Cadrul .Net a parcurs un drum lung de atunci, iar versiunea actuală este 4.7.1. Cadrul .Net poate fi utilizat pentru a crea atât aplicații bazate pe formular, cât și bazate pe Web. Serviciile web pot fi, de asemenea, dezvoltate folosind cadrul .Net. Cadrul acceptă, de asemenea, diferite limbaje de programare, cum ar fi Visual Basic și C #. Astfel, dezvoltatorii pot alege și selecta limba pentru a dezvolta aplicația dorită. În acest capitol, veți afla câteva elemente de bază ale cadrului .Net.

### **I.2.3. Unity**

În funcție de scopul produsului finit, există posibilitatea alegerii între mai multe platforme, printre care și cele enumerate mai sus. În cazul de față, știam de la bun început că doresc să dezvolt un joc a cărui acțiune se va desfășura într-un spațiu tridimensional și implicit, aria de platforme dintre care puteam alege s-a restrâns. Am luat în considerare doar platformele gratuite. De asemenea, doream să lucrez cu cele mai noi tehnologii disponibile, și astfel am ajuns pus în fața unei alegeri între două platforme: **Unity** și **Unreal Engine 4**.

Pentru a lua o decizie cât mai potrivită, am luat în calcul mai mulți factori. Înainte de aceasta, am citit câteva articole de pe diverse bloguri tematice care comparau cele două platforme<sup>6</sup>.

Precizez faptul că nu am mai avut șansa să lucrez pe nicio astfel de platforma de dezvoltare. Așadar, aveam o foarte vagă idee despre cum ar putea decurge dezvoltarea unui joc. Deși aveam o ușoară preferință pentru Unity, am fost foarte tentat și de alegerea Unreal Engine, știind că acesta din urmă a fost baza succesului comercial al multor titluri. În alte cuvinte, Unity era o opțiune atrăgătoare datorită facilității, iar Unreal Engine era o opțiune atrăgătoare datorită calității aproape inegalabile a produsului final. De asemenea, merită menționat faptul că Unreal Engine a ajuns la îndemâna publicului mult mai recent decât Unity (care avea deja un avantaj de câțiva ani în creșterea comunității)<sup>7</sup>. Acestea fiind spuse, am început să mă documentez în legătură cu procesul de dezvoltare și cu opțiunile disponibile în scopul realizării ideii mele atât în Unity, cât și în Unreal Engine și am ales în final Unity.

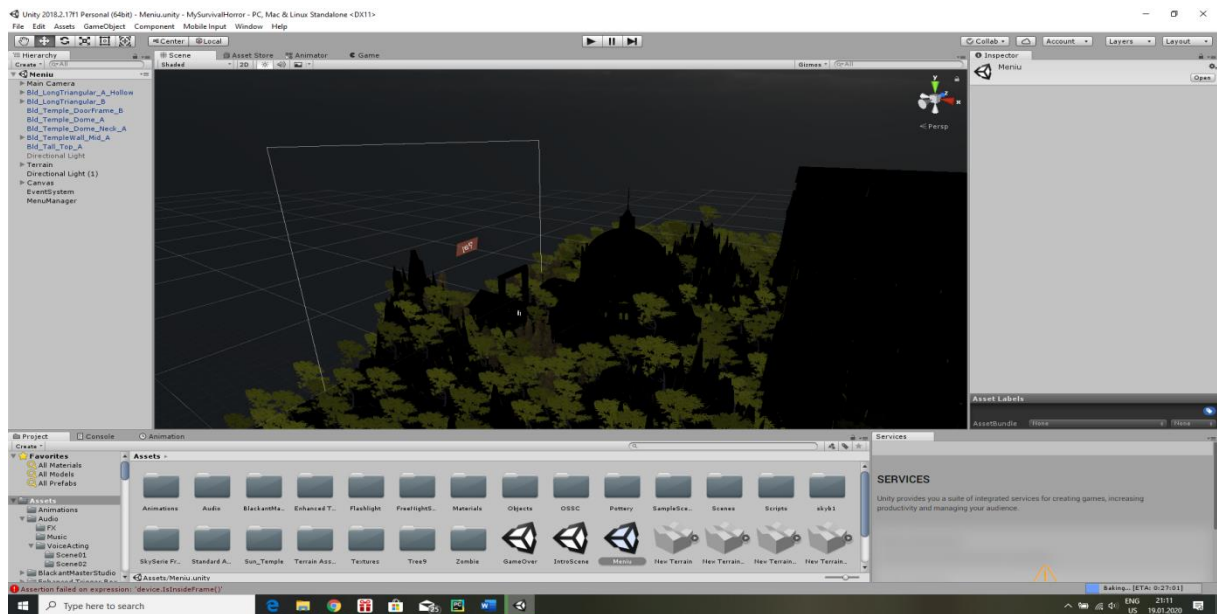
Unity este un gameengine scris în C și C++ lansat inițial în 2005 (versiunea 1.0) al cărui ultim release stabil a avut loc acum câteva luni (martie 2016, versiunea 5.3.4). Unity se bucură de popularitatea pe care o are mulțumită mai multor factori: oferă soluții pentru dezvoltarea aplicațiilor atât 2D cât și 3D ce pot fi rulate pe un număr mare de platforme (Linux, Mac, Windows, Android, iOS, WebGL, BlackBerry, Xbox, Play Station, Nintendo WiiU & 3DS, etc.). Portabilitatea nu se limitează doar la mediile în care aplicațiile Unity pot fi rulate. Astfel, dezvoltatorii nu sunt forțați să opteze pentru o anumită platformă în vederea dezvoltării, și au posibilitatea de a utiliza atât Windows și MacOS, cât și Linux (Ubuntu încă în beta).

Unul dintre aspectele care s-au dovedit a fi foarte utile în procesul de dezvoltare al aplicației mele este posibilitatea de a scrie și utiliza scripturi, cele mai utilizate limbaje fiind C# și Javascript. Personal, am ales să scriu în C#, principalele motive fiind ușurința aplicării unei gândiri POO și faptul că marea majoritate a comunității încurajează utilizarea acestor limbaje pentru a ușura rezolvarea a potențialelor probleme ce pot apărea de-a lungul procesului de dezvoltare. Majoritatea tutorialurilor despre unity folosesc scripturi în C#.

---

<sup>6</sup> *Unreal Engine vs Unity: Which is Better?*, <https://www.gamedesigning.org/engines/unity-vs-unreal/>.

<sup>7</sup> <https://www.unrealengine.com/what-is-unreal-engine-4>.



**Figura 1.** Imagine preluată din scena Meniu.

#### I.2.4. Assets <sup>8</sup>

Asset Store este o platformă de cumpărare a unor assets create în majoritatea timpului de către comunitate. Un asset poate fi un grup de modele, sunete, scripturi, texturi, etc ce formează un tot unitar ce poate fi importat cu ușurință într-o aplicație Unity. Pentru o mai ușoară exemplificare a noțiunii de asset, poate fi observată importanța asseturilor în proiectul de față. În această aplicație sunt folosite următoarele asseturi: FPSController și tot ce ține de aspectul jocului (camere, clădiri, pădure, obiecte).

FPSController este un asset care crează un personaj controlabil de către utilizator cu o perspectivă “firstperson” (persoana I: utilizatorul observă universul jocului prin ochii personajului). Acest asset este un pachet care include scripturi pentru detectarea și tratarea inputului utilizatorului, sunete pentru pași, un Collider (un obiect cu proprietăți fizice, folosit în cazul de față pentru a detecta coliziuni cu suprafața podelei) și o cameră atașată acestui Collider (camera reprezintă perspectiva din care poate privi un utilizator universul aplicației). Majoritatea asseturilor, precum acesta, conțin și unul sau mai multe “prefabs” care reprezintă un obiect al jocului prefabricat ce include toate caracteristicile menționate mai sus și le configurează pentru a lucra împreună ca un tot unitar.

<sup>8</sup> <http://unity3d.com/>

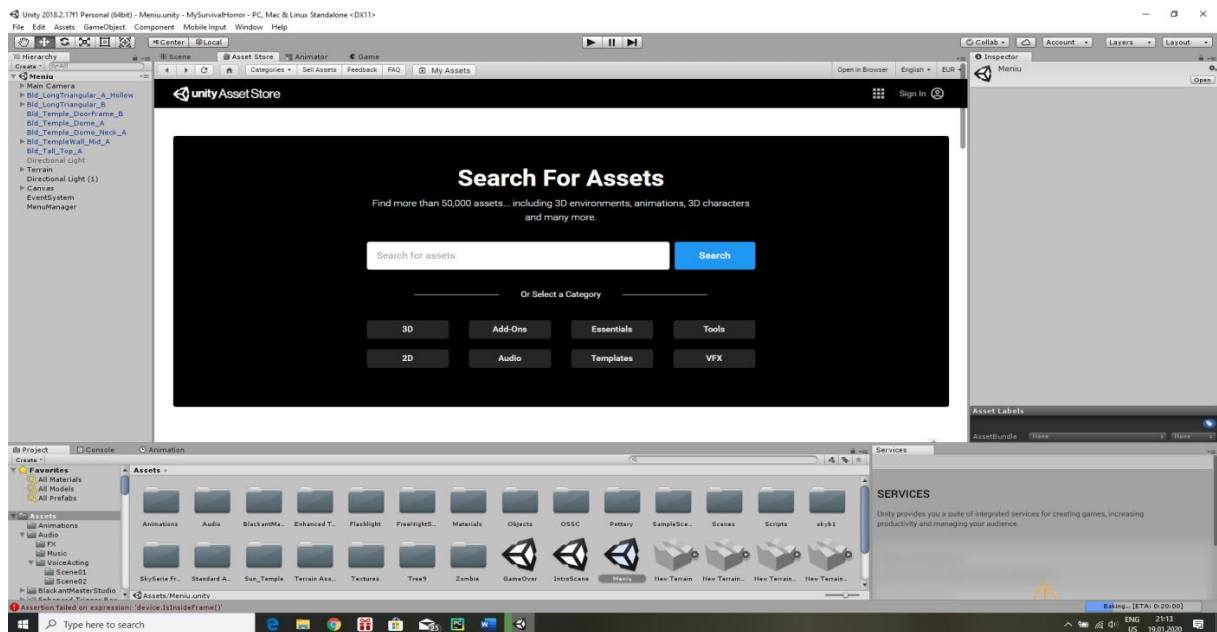


Figura 2. Imagine cu Asset Store din Unity.

## I.2.5. Sunet

Pentru crearea sunetelor folosite în aplicație am folosit **voicechanger.io**. Am ales să folosesc voicechanger.io deoarece este un tool foarte versatil. Am creat cu ajutorul unor sintetizatoare niște sunete ambientale și am modificat sunete înregistrate de mine pentru a-i oferi entității negative din universul jocului o voce.

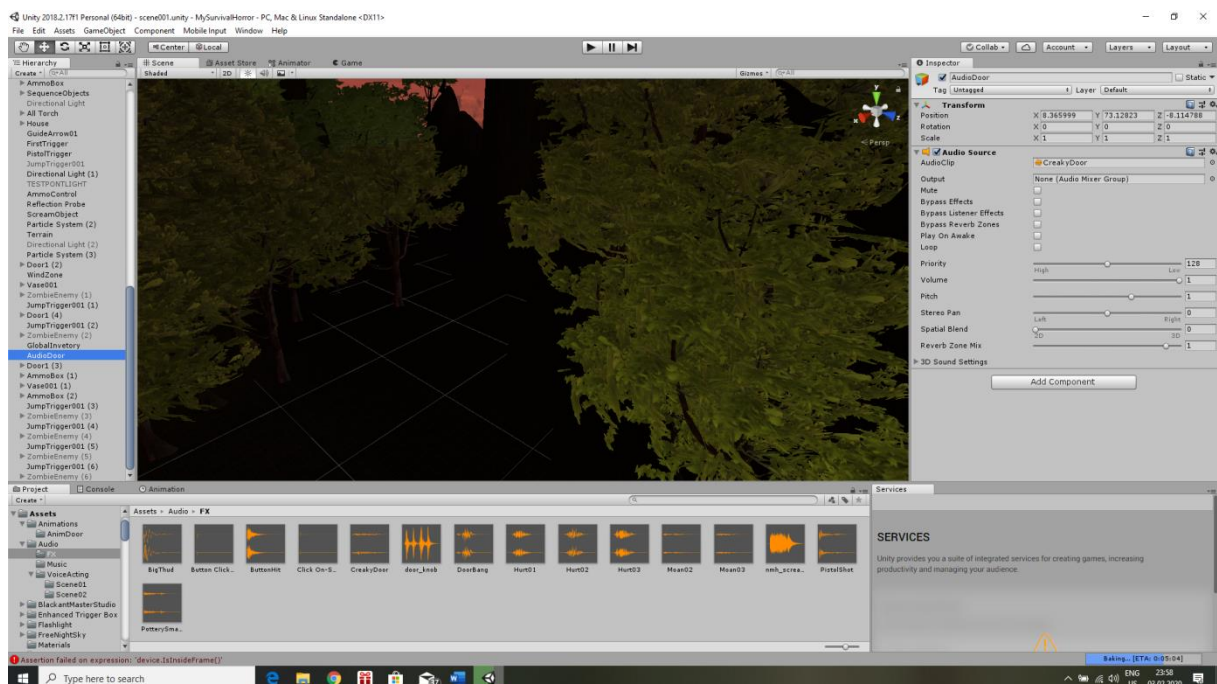


Figura 3. Imagine din scena001

## Capitolul II. Prezentarea aplicației

În acest capitol voi prezentat în detaliu întreaga aplicație, problemele întâmpinate precum și rezolvarea lor, dar mai ales cum am creat meniul, intro-ul jocului, harta, personajul negativ, jumpscare-urile, obiectele utilizabile și luminile.

### II.1 Specificații funcționale

Aplicația constă într-un joc 3D definit prin doua aspecte concrete:

- Un spațiu necunoscut de personaj, cu o atmosferă menită să țină utilizatorul în suspans;
- Un personaj negativ de care utilizatorul trebuie să se ferească pentru a putea supraviețui.

Așadar, utilizatorul va prelua controlul unui personaj cu o anumită poveste, a unui fi polițist care dorește să afle adevărul despre un anumit oraș dintr-o pădure. Acesta se va trezi la un moment dat într-o casă în care el trebuie să supraviețuiască. În tot acest timp, el trebuie să exploreze toate încăperile, pentru a putea găsi o cale de ieșire din oraș și pentru a nu fi omorât de personajul negativ.

Întreaga atmosfera a jocului se bazează ca pe restul jocurilor horror, în sensul că jucatorul este mereu sub o presiune psihologică în timpul jocului. Factorii cei mai importanți din acest joc sunt:

- Senzația personajului de singurătate;
- Efectele sonore din fundalul jocului;
- Mediul înfiorător, cu aspect sumbru;
- Căutarea unor chei sau gloanțe pentru a se putea apăra;
- Personajul negativ (sunete, aspect înfricoșător, perseverența de a te omorî).



Personajul negativ va fi rezultatul aplicării unui script ce îi va oferi următoarele caracteristici:

- Va urmări personajul și îl poate ucide în funcție de anumite declanșatoare (exista și declanșatoare care îl pot doar panica pe jucător);
- Pentru primul declanșator, acela când jucătorul poate fi omorât, viteza personajului negativ (zombie) este una relativ egală cu cea a personajului;
- Al doilea declanșator este menit să întrețină intensitatea jocului, atunci când jucătorul poate intra doar în panică, fără a putea fi rănit.

## **II.2 Arhitectura aplicației**

Această aplicație este secționată în mai multe scene care oferă o continuitate jocului. Voi descrie succint fiecare scenă și rolul ei:

- **Meniul**

În această scenă regăsim o imagine de asamblu a jocului și un buton “Play” care îl va conduce pe utilizator spre următoarea scenă a jocului.

- **Intro**

Este practic povestea jocului, unde cu ajutorul unor animații setate pe camera principală din scenă, pentru un interval de timp, personajul se plimbă printr-o pădure găsim la un moment dat o casă, loc care va conduce spre alta scenă: începerea jocului. Tot aici, avem și un buton de „SKIP”, pe care îl putem utiliza atunci când dorim să nu vizualizăm intro-ul.

- **Primul nivel**

Utilizatorul va descoperi primele obiecte de care are nevoie pentru a putea avansa în joc și de a supraviețui.

- **GameOver**

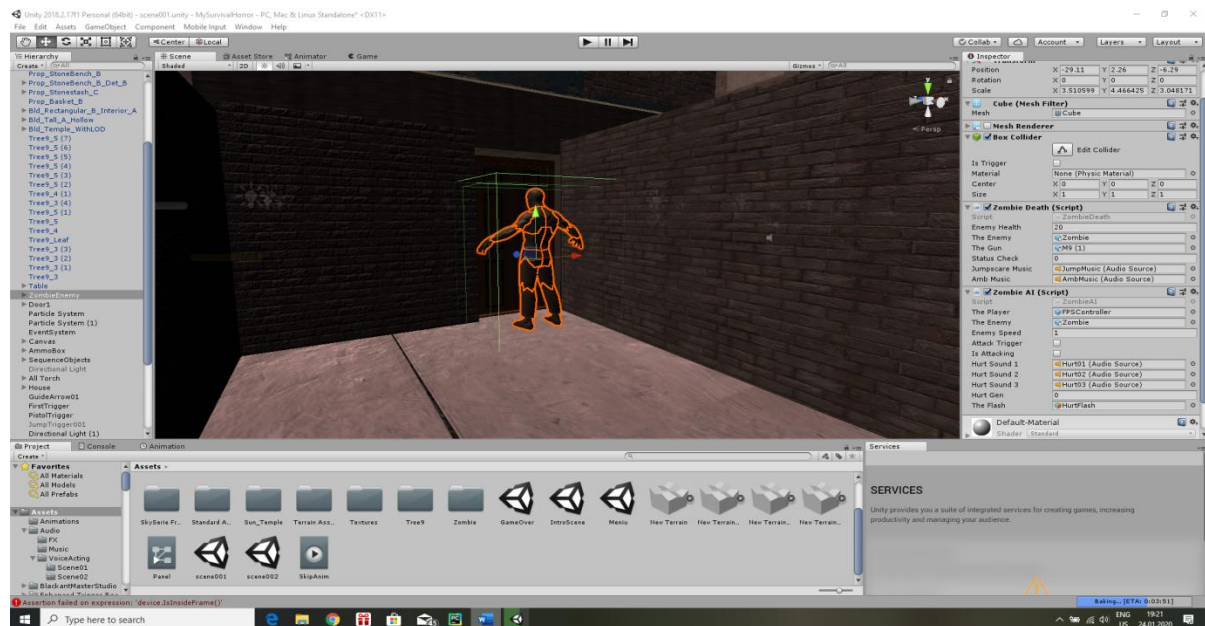
În această scenă se ajunge atunci când am pierdut jocul, iar utilizatorul este forțat să înceapă jocul de la început, el fiind redirecționat la scena “primul nivel”.

- **Ultimul nivel**

În această scenă , după mai multe încercări de a supraviețui ,jucătorul nu va gasi cale de scăpare și jocul se va sfârși câștigând personajul negativ .

Într-o reprezentare a arhitecturii unui proiect în Unity,trebuie precizat ca aceste aplicații sunt bazate pe componente,iar aici ma refer la conceptul de GameObject.

GameObject este cel mai important tip de obiect din Unity ,orice element din această aplicație este un GameObject,acesta nu funcționează de unul singur , are nevoie de anumite proprietăți pentru a putea face ce ne dorim, iar aceste proprietăți sunt in componente.Așadar, un GameObject poate fi privit ca un un loc unde punem mai multe obiecte,în care putem adăuga texturi,sunete,animații,scriptu-uri sau lumini.



**Figura 4.** Imagine preluată din scene scene001(Primul Nivel).

## II.3 Hartă

În acest capitol, voi prezenta cum am dezvoltat harta, ce texturi și materiale am folosit și de unde le-am importat.

Primul GameObject creat a fost un cube 3D, căruia i-am adăugat o textură sub forma unor cărămizi. Odată aplicată această textură, se va crea automat un obiect într-un fișier numit Materials ce va reprezenta noul nostru obiect cu textura aplicată deoarece în unity texturile nu aparțin obiectelor, doar materiale aparțin obiectelor. Toate texturile pe care le-am aplicat acestor obiecte sunt preluate gratuit de pe Asset store-ul prezentat anterior. Cu aceste texturi aplicate unor obiecte, am reușit să dezvolt prima încăpere din joc, căruia în timp i-am adăugat câteva obiecte: scaune, uși, mese, cărți (luate tot din Asset store) pentru a putea popula camera, ca aceasta să pară cât mai reală.

Odată realizată camera, am adăugat un **Firstpersoncontroller**. Acest standard asset va reprezenta în jocul meu personajul principal ce va face posibil interacțiunea cu personajul negativ și modul în care trebuie să găsească anumite obiecte pentru a putea supraviețui. Pentru un mediu mai intens, am adăugat ceață în timpul jocului pe anumite porțiuni: **GameObject->Effects->Particle System**. Având acest obiect la dispoziție și pentru anumite setări am reușit să adaug o atmosferă mai intensă. Tot aici, am creat un obiect Torch căruia i-am atribuit un particle system asemănător cu cel folosit pentru a crea ceața: o animație aplicată unui Point Light pentru a controla intensitatea luminii, aceasta având un sample setat pentru 60 de frameruri pe secundă și un script care are ca scop cum vor fi activate acele animații pentru obiectul respectiv. Toate acestea au ca scop de a reda o torță cu flăcări ca în viața reală.

Voi prezenta și partea de UI (user interface). Cum creăm o ușă pe care o putem deschide cu ajutorul **UI->Text**. Ușa folosită în joc este luată din asset store, iar primul pas pentru a putea deschide ușa este crearea unui obiect cube 3, denumit de mine DoorHinge. Acest Gameobject va fi poziționat între ușă și balamaua ușii, având ca scop deschiderea ușii, acest fapt va fi realizat cu ajutorul unei animații aplicate acestui GameObject și al unui script ce va face posibilă deschiderea ușii la apăsarea tastei E. Acest script va conține 3 GameObject acestea fiind ActionDisplay ce reprezintă tasta ce trebuie apăsată, ActionText ce reprezintă un mesaj, în cazul nostru „Open de door”, TheDoor ce va conține obiectul Hinge, ce are atașată animația explicată mai sus, AudioSource ce va conține sunetul la deschiderea ușii, **ExtraCross** ce va conține un **RawImage**, acesta reprezintă un patrat ce va apărea doar atunci

când vrem să deschidem ușa și în final o variabilă de tip float ce va reține distanța minimă la care va apărea mesajul pe ecran la apropierea de ușă. Pentru a putea afla această distanță, am făcut un alt script numit **PlayerCasting**, unde avem o variabilă statică ce reține această distanță.

Odată ieșit din această încăpere, jucătorul va putea observa harta populată cu clădiri medievale, toate fiind preluate de pe Asset store.

După anumite încercări de a scăpa, personajul principal se va trezi într-o padure, pe care am creat-o după imaginația mea. În Unity, avem conceptul de Terrain, care este de asemenea un 3D object, care îți oferă posibilitatea de a crea propria ta lume.

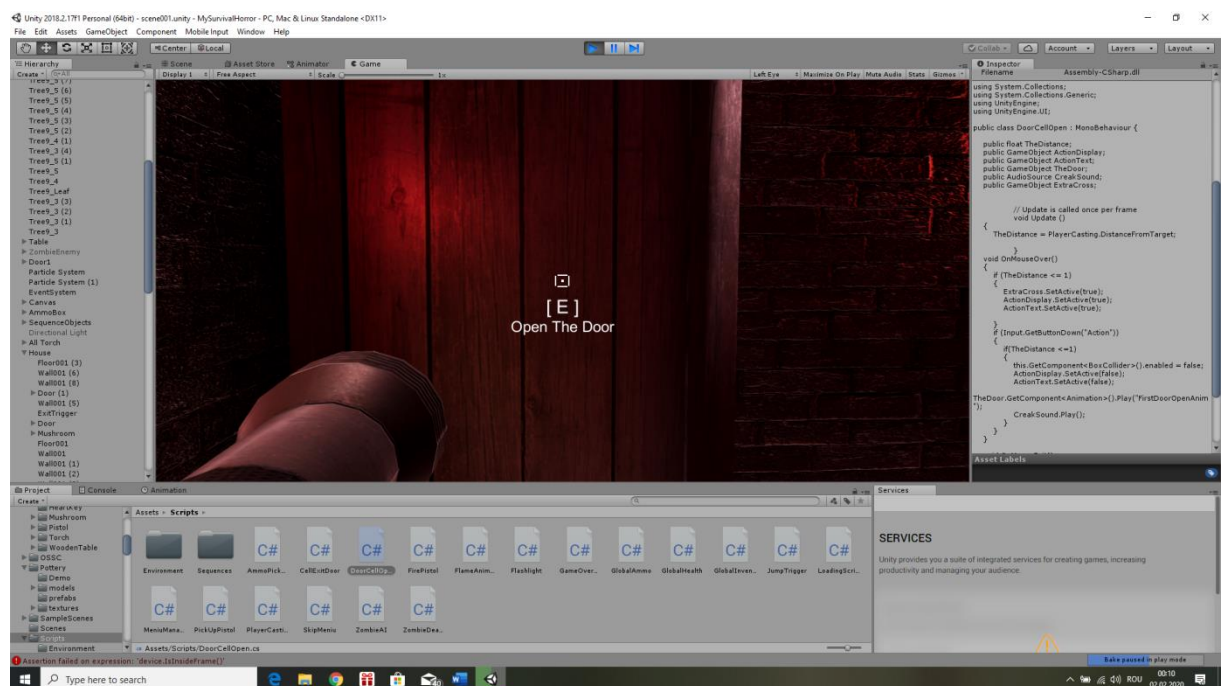


Figura 5. Imagine preluată din scene001, primul contact cu ușă.

## II.4 Zombie

Voi descrie cum a fost creat acest personaj negativ, tot ce ține de modul cum vei fi urmărit , speriat și cum poate fi omorât. Prima interacțiune cu personajul negativ este atunci când vom lua arma de pe masă. Ca acest lucru să fie posibil, m-am folosit de obiectul creat

anterior,uşa, pentru a crea un jumpscare, atunci cand jucatorul se va apropia de uşă ,uşa se va deschide singură şi va aparea personajul negativ .

Voi enumera paşii pe care i-am urmat pentru primul *jumpscare* din joc:

- **Uşa**

Am importat în joc un alt asset diferit faţă de prima uşă caruia i-am aplicat o animaţie pentru a se deschide uşa singură când vom trece prin trigger-ul setat de mine.

- **Declanşatorul<sup>9</sup>**

Aceste reprezinta un obiect 3D în care vom avea dezactivat **Mesh Renderer** pentru a nu putea vedea obiectul în timpul jocului, acesta are rol de declanşa uşa să se deschidă atunci când vom trece prin acest obiect.Atunci când personajul principal va fi atacat am adăugat la Firstpersoncontroller un obiect de tip Cylinder ca atunci cand va fi atacat , personajul negativ trebuie să se afle în aria acestui obiect ,tot aici trebuie să avem dezactivat Mesh Renderer.Pe lângă aceste tipuri de declanşatoare , am mai creat doua declanşatoare la începutul primului nivel cu scop informativ,acesta il vor bloca pe utilizator pentru un moment afişând un mesaj atât audio cât şi vizual cu ce are de făcut.

- **Scripturi<sup>10</sup>**

În fişierul **BZJumpTrigger** ne vom folosi de 4 obiecte, acestea fiind un AudioSource atunci cand uşa se va deschide , un AudioSource pentru muzica de fundal atunci cand am declanşat personajul negativ să ne atace, un GameObject ce va reprezenta personajul negativ şi un GameObject ce va reprezenta uşa.

În fişierul **ZombieDeath** vom avea un AudioSource pentru muzica din fundal, doua variabile, una pentru viaţa personajului negativ şi un status ce ne va indica atunci când personajul negativ este omorât.

---

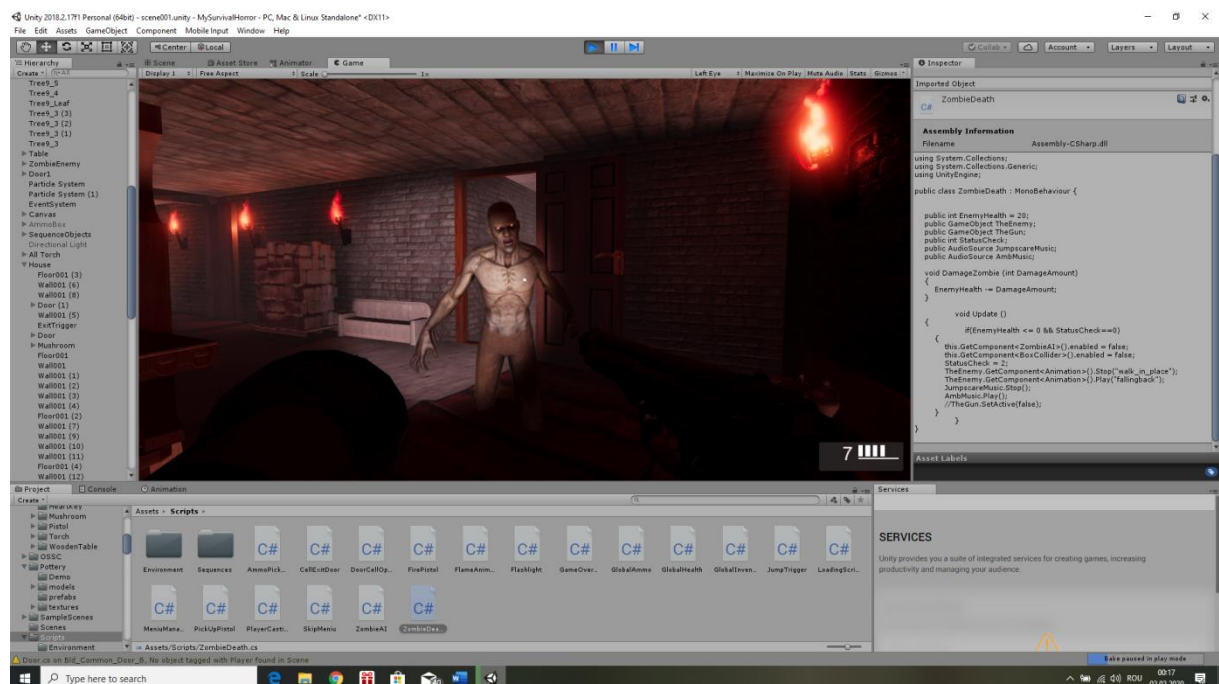
<sup>9</sup> SamsTeach Yourself – Unity Game Development in 24 hours  
<http://docshare02.docshare.tips/files/28607/286070481.pdf>

<sup>10</sup> <https://jvunity.weebly.com/>

În fișierul **ZombieAI** vom avea modul în care personajul negativ va încerca să ne urmărească și în același timp să ne și omoare. Am declarat două **GameObject**, unul în care se va afla player-ul și unul personajul negativ, o variabilă de tip **float** ce va reprezenta viteza cu care se va mișca personajul negativ și în final două variabile de tip **bool** în care vom seta când va porni animația de atac al personajului negativ la o distanță apropiată de player și când o vom dezactiva.

În fișierul **GlobalHealth** vom avea modul în care gestionăm viața personajului principal, aici am definit o variabilă statică ce va conține numărul maxim de vieți și o altă variabilă ce va reține viața rămasă în timpul jocului. De la începutul jocului personajul principal pornește cu patru vieți, odată ce aceste vieți sunt folosite, utilizatorul pierde jocul și va fi trimis în scena **GameOver**.

În fișierul **JumpTrigger** se va regăsi ultimul *jumpscare* din joc, care are rolul doar de a-l speria pe jucător, el neputând fi rănit. Acest jumpscare a fost creat punând obiectul **Zombie** în fața camerei, acesta va apărea doar atunci când vom intra printr-un alt obiect 3D precizat mai sus unde va activa personajul negativ cu o anumită animație, scriptul atașat acestui obiect va conține un **AudioSource** ce reprezintă urletul personajului negativ și trei **GameObjecturi** ce reprezintă player-ul, personajul negativ din fața camerei și un efect atunci când va fi activat declanșatorul.



**Figura 6.** Imagine preluată din scena001, primul contact cu personajul negativ.

## II.5 Meniu și Intro

Meniul reprezintă prima scenă a jocului, am facut-o cât mai simplă, iar aici mă refer că am creat doar un Terrain unde l-am populat cu mulți copaci și câteva clădiri după care am adăugat în centrul camerei jocului un buton Play. Acest buton a fost creat utilizând **UI > Button** și cu ajutorul unui script care la apăsarea lui ne va redirecționa către următoarea scenă a jocului, aceea fiind Intro.

Intro reprezintă a doua scenă din joc și are ca scop să ofere utilizatorului povestea jocului, care este rolul personajului principal și ce are de făcut. În această scenă am creat un Terrain unde personajul principal se află într-o pădure, acesta se va mișca prin această pădure până când va observa o casă și se va trezi deodată în ea. Acest lucru a fost posibil realizat cu ajutorul unor animații aplicate camerei, a unor voci ce explică expediția personajului principal în pădure și al unui script ce face posibil aceste lucruri, acest script va activa animațiile la o anumită perioadă de timp și vocea personajului principal. Tot în această scenă am implementat pe baza primei scene și un buton Skip, dacă utilizatorul nu dorește să vadă povestea jocului până la capăt, poate trece la următoarea scenă a jocului. Pentru calitatea imaginii am folosit PostProcessing, acesta poate îmbunătăți drastic imaginea aplicației cu un timp mic de configurare, este utilizat în cele mai multe cazuri pentru a stimula proprietățile fizice ale camerei și ale filmului.

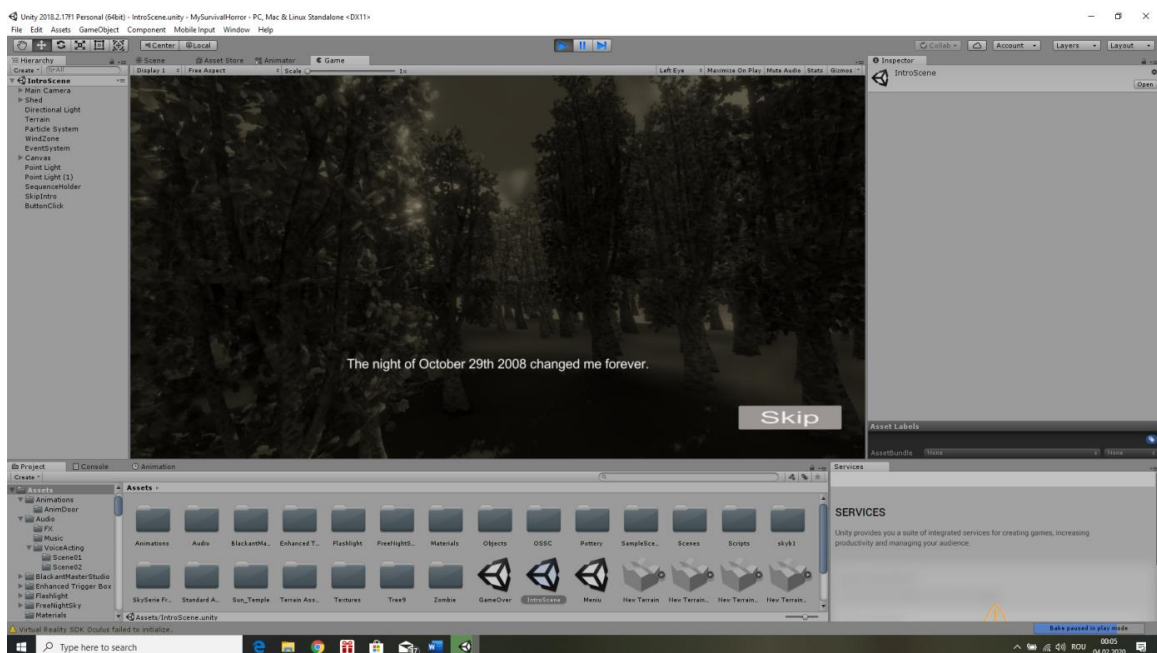


Figura 7. Imagine preluată din IntroScene, prezentarea poveștii jocului.



## II.6 Obiecte utilizabile

În acest capitol, voi prezenta toate obiectele pe care le poate folosi personajul principal și cum funcționează.

- **Lanterna**

Datorită unui mediu foarte întunecat , am atașat obiectului FirstPersonController o lanternă, aceasta fiind luată de pe asset store, ce va avea ca scop să lumineze anumite porțiuni pentru a putea găsi anumite obiecte de care are nevoie utilizatorul, cum ar fi chei, arme sau gloanțe. Am aplicat lanternei un **Collider** pentru a nu putea intra cu ea în alte obiecte și un script ce face posibilă deschiderea lanternei la apăsarea click dreapta.

- **Pistol si muniție**

Pentru a putea supraviețui în acest joc, am oferit utilizatorului șansa de a putea lua o armă, doar că nu va avea gloanțe permanente , așa ca el va trebui sa caute mereu gloanțe pentru a putea supraviețui unui atac. Ca acest obiect să poată fi utilizat am refolosit scriptul cu care deschid ușa prezentat mai sus, doar că am mai adăugat obiectul FakePistol, acesta fiind pistolul de pe masă ce urmează a fi luat, obiectul RealPistol, pistol ce se află atașat de FirstPersonController și urmează să fie activat ,iar în final un obiect GuideArrow ce are ca scop să ajute utilizatorul de a găsi pistolul.

În continuare am sa prezint mecanismul de tragere al pistolului. Primul pas a fost sa adaug un nou **GameObject->Plane**, unde am atașat o imagine cu o flacără, iar aceasta va apărea doar cand vom apasa click stânga datorită unei animații aplicată acestui obiect. La acest obiect vom adăuga și un script FirePistol unde mă voi folosi de următoarele obiecte: un GameObject ce va reprezenta pistolul, un GameObject ce va reprezenta imaginea cu o flacără ce are ca scop de a oferi utilizatorului o situație reală de a trage cu arma, un AudioSource atunci când se trage cu arma.

Pentru muniție am folosit un obiect de pe asset store. În primul rând am creat un **UI->Panel** unde va fi afișat în permanență câte gloanțe mai avem la dispoziție în timpul jocului, odată ce nu mai avem gloanțe nu mai putem trage cu arma, animația este oprită. Acestui obiect i-am atașat un script ce va conține muniția si panel-ul unde ne este



afișată muniția având ca scop să afișeze panel-ul și să dezactiveze obiectul din joc oferind utilizatorului impresia că a colectat muniția, pentru fiecare cutie de muniție luată, jucatorul primește șapte gloanțe.

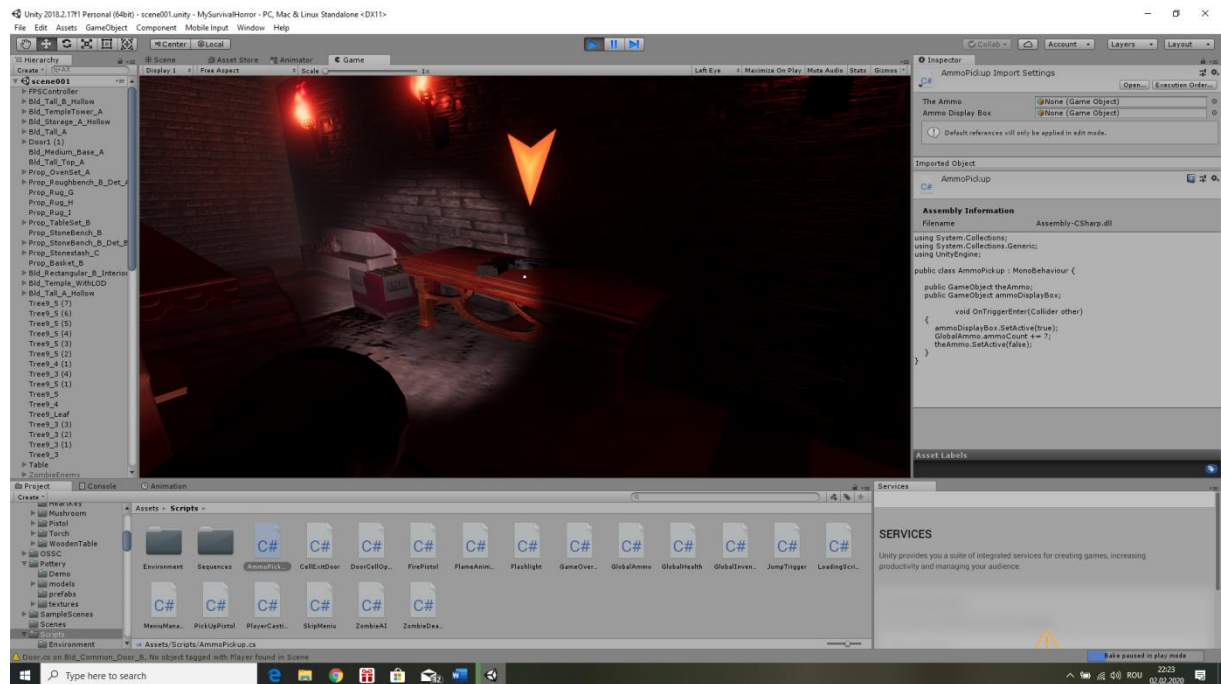


Figura 7. Imagine preluată din scena001, primul obiect utilizabil.

- Cheie pentru uși

Probabil cea mai utilizată secvență dintr-un joc horror, pentru realizarea ei am folosit aceeași ușă utilizată la declanșarea personajului negativ, am atașat un Cube 3d cu un Collider pentru a nu putea trece prin ușă decât doar atunci când ușa este deschisă. Ca și script aplicat acestui Cube 3d m-am refolosit de scriptul PickUpPistol, unde am adăugat un AudioSource atunci când se încearcă ușa și am adăugat un **Coroutine**. Execuția unei coroutine poate fi întreruptă în orice moment folosind declarația yield. Când se folosește o declarație yield, coroutine întrerupe execuția și reia automat la cadrul următor.

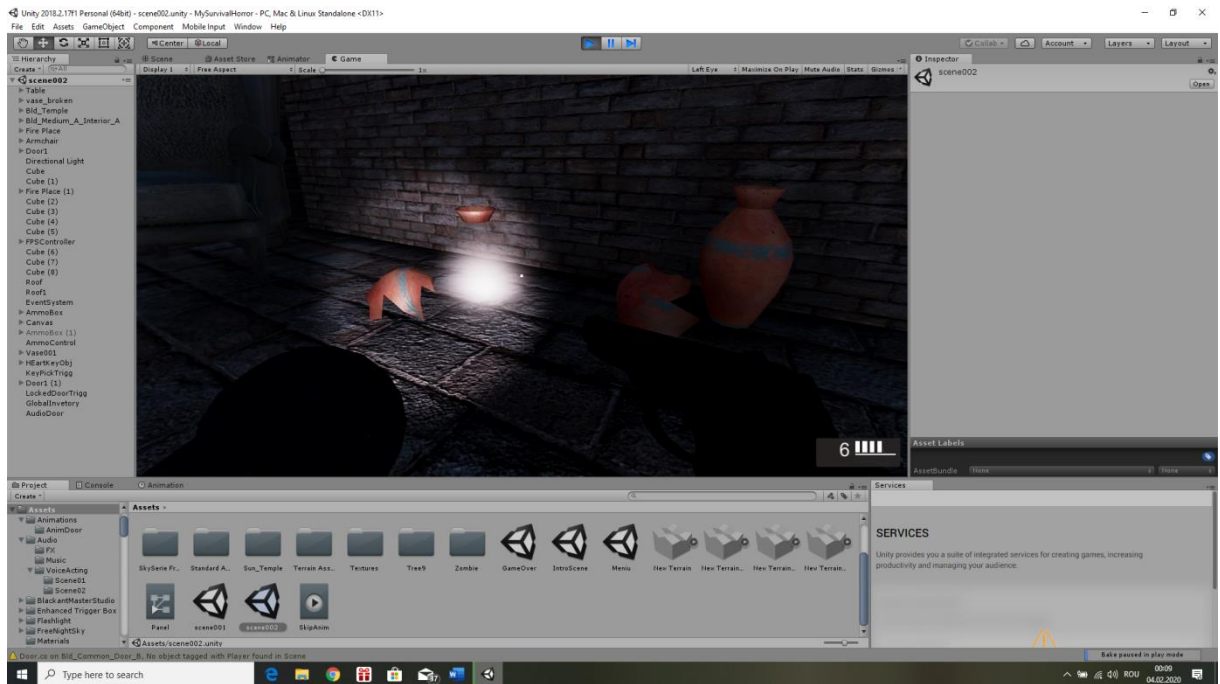
Pentru cheie am adăugat un script ce va conține distanța dintre obiect și personaj, un GameObject ce reprezintă tasta ce trebuie apăsată pentru a lua cheia, un GameObject ce reprezintă declanșatorul de cheie, acesta va fi dezactivat odată ce cheia va fi preluată de personaj, un GameObject pentru afișarea unui mesaj, un GameObject pentru chenarul din mijloc al acțiunii (aceste 3 obiecte sunt asemănătoare cu cele folosite la deschiderea primei uși din joc) și în final ușa care trebuie deschisă. Tot pentru cheie pentru a atrage atenția

utilizatorului am aplicat asupra cheii un Particle System , ce va apărea ca un efect strălucitor de a sesiza că acel obiect este utilizabil.

- **Obiecte destructibile**

Pentru a nu putea fi gasită prea repede cheia din joc m-am gândit că utilizatorul să încerce să tragă în anumite obiecte pentru a putea gasi o cheie ce îl poate duce la nivelul următor.În primul rând am importat de pe asset store o vază, aceasta având o animație să se spargă în mai multe bucăți.Tot aici am aflat și de componenta **Rigidbody**,aceasta activează GameObjects pentru a acționa sub controlul fizicii.Rigidbody poate primi forțe și cuplu pentru a face obiectele să se miște într-un mod realist.Orice GameObject trebuie să conțină o perosoană rigidă care să fie influențată de gravitate,să acționeze sub forțe adăugate prin intermediul scriptului sau să interacționeze cu alte obiecte prin intermediul sistemului NVIDIA PhysX.

În realizarea acestui obiect am folosit un Cube 3d și o sferă ce are ca scop distrugerea vazei (aceste obiecte având dezactivate Mesh Render),în aceste obiecte am inserat vaza întreagă și vaza care se va sparge odată ce utilizatorul va trage cu arma.Pentru a da impresia ca împuşcam vaza , mă voi folosi de funcția folosită atunci când scădem din viața personajului negativ,odată ce vom trage spre vaza , va fi ca și cum am trage spre vază și s-ar sparge în bucăți din ochii utilizatorului.Pe lângă această funcție vom crea un Coroutine ce se va folosi de 3 obiecte pentru a face acest lucru posibil, un GameObject ce reprezintă vaza în întregime, un GameObject ce reprezintă vaza ce se va distruge și un GameObject ce va face ca această vază să se distrugă.



**Figura 8.** Imagine preluată din scena002, obiect destructibil.

## CONCLUZIILE LUCRĂRII

### Concluzii și contribuții personale

Lucrarea “Joc 3D pe platforma Unity” surprinde elementele esențiale din viața unui joc horror și își propune dezvoltarea unui joc care să conțină toate caracteristicile unei aplicații reușite având la bază scripturi ușor de înțeles și de modificat, conținând doar asseturi gratuite, toate acestea să ofere utilizatorului o experiență de joc cât mai dinamică și intensă.

Am reușit prin utilizarea platformei Unity să creez o aplicație ce oferă o experiență intuitivă și plăcută prin îmbinarea unor elemente vizuale și audio cu o implementare specifică.

Experiența jocului este completată prin utilizarea unor obiecte în timpul jocului și găsirea lor, supraviețuirea într-un mediu necunoscut, un personaj negativ (Zombie) ce are ca scop tot timpul să te urmărească, trecerea de la un nivel la altul și modelarea unei hărți după propria imaginație.

Probleme întâmpinate în elaborarea lucrării:

- Dacă codul nu compilează, jocul nu poate fi rulat.
- Nefolosirea Box Collider pe obiecte , acest fapt făcând jucătorul să poată trece prin obiecte în timpul jocului.
- Dezactivarea unor obiecte după folosirea lor, de exemplu atunci când vom utiliza declanșatorul, dorim să fie utilizat doar o dată , acest lucru fiind posibil printr-o comandă într-un script.
- Animațiile obiectelor,când cream o animație trebuie avut în vedere că aceasta să nu fie repetată în mod repetitiv,ea fiind activată printr-un script.
- Folosirea unor funcții din unity:  
transform.LookAt(),Vector3(),Raycast(),StartCoroutine(IEnumerator name).

### **Posibilitatea extinderii pe viitor:**

Pe viitor există mai multe direcții în care această aplicație poate fi dezvoltată:

- Introducerea de noi obiecte , cât și al unui inventar pentru aceste obiecte;
- Introducerea unor puncte de salvare al jocului, pentru a nu fi nevoie să începi mereu jocul de la început;
- Dezvoltarea mai multor nivele cât și declanșatoare pentru personajul negativ;
- Calculator performant pentru adăugarea unor asseturi calitative;

## Bibliografie și sitografie

1. Meredith Watkins, *Video Game Addiction Symptoms and Treatment*, <https://americanaddictioncenters.org/video-gaming-addiction>.
2. Patricia Hernandez, *What People Get Wrong About PewDiePie, YouTube's Biggest Star*, <https://kotaku.com/what-people-get-wrong-about-pewdiepie-youtubes-biggest-1673109786>.
3. <https://www.unrealengine.com/what-is-unreal-engine-4>.
4. <https://americanaddictioncenters.org/video-gaming-addiction>
5. <http://www.rpgmakerweb.com/>
6. <http://unity3d.com/>
7. <https://jvunity.weebly.com/>
8. *Unreal Engine vs Unity: Which is Better?*, <https://www.gamedesigning.org/engines/unity-vs-unreal/>.
9. <https://www.praeyforthe gods.com/>
10. SamsTeach Yourself – Unity Game Development in 24 hours  
<http://docshare02.docshare.tips/files/28607/286070481.pdf>
11. Jeremy Gibson -Introduction to Game Design, Prototyping and Development  
<http://ptgmedia.pearsoncmg.com/images/9780321933164/samplepages/9780321933164.pdf>