**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM**

**CSÍKSZEREDAI KAR**

**GAZDASÁGI INFORMATIKA SZAK**

**DIPLOMADOLGOZAT**

**Nagy bolond futások**

**Végzős hallgató:**

**Marosi Szilárd**

**Témavezető:**

**Bíró Piroka, egyetemi docens**

**2023**

**Román kivonat**

**Cím románul**

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus. Morbi et sem orci. Praesent accumsan odio in ante ullamcorper id pellentesque mauris rhoncus. Duis vitae neque dolor.

Duis sed purus at eros bibendum cursus nec a nulla. Donec turpis quam, ultricies id pretium sit amet, gravida eget leo.

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus. Morbi et sem orci. Praesent accumsan odio in ante ullamcorper id pellentesque mauris rhoncus. Duis vitae neque dolor.

Duis sed purus at eros bibendum cursus nec a nulla. Donec turpis quam, ultricies id pretium sit amet, gravida eget leo.

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus.

**Angol kivonat**

**Big runs for your life**

I could have chosen from many things, but in the end, I decided on a computer game because I had never encountered anything like it before, and it presented a great challenge for me. In this game, we control a character who must avoid obstacles in its path, and the objective is to collect as many points as possible. I have also placed coins in the game that the player can pick up to purchase different characters. The more coins they collect, the more diverse characters they can buy.

I utilized various technologies during the development of the game, including Unity and C#. The game interface was created using Unity, while the server-side was implemented in C#. Additionally, I incorporated MySQL to create the leaderboard for the game.

**Tartalomjegyzék**

[**1.** **Bevezető** 5](#_Toc136255600)

[**2.** **Használt technológiák** 6](#_Toc136255601)

[**2.1.** **Unity** 6](#_Toc136255602)

[**2.2.** **C#** 8](#_Toc136255603)

[**2.3.** **Mixamo** 9](#_Toc136255604)

[**2.4.** **GitHub** 9](#_Toc136255605)

[**2.5.** **MySQL** 10](#_Toc136255606)

[**3.** **Játék bemutatása** 10](#_Toc136255607)

[**3.1.** **Programozási környezet** 12](#_Toc136255608)

[**3.2.** **Adatbázis** 12](#_Toc136255609)

[**3.3.** **Az alkalmazás fontosabb funkciói** 13](#_Toc136255610)

[**3.3.1.** **Felhasználói felület** 16](#_Toc136255611)

[**3.3.2.** **Ranglista** 16](#_Toc136255612)

[**4.** **Következtetések** 17](#_Toc136255613)

[**Irodalomjegyzék** 18](#_Toc136255614)

# **Bevezető**

Az államvizsga egy nagy mérföldkő volt előttem és rengeteg ötlet cikázott a fejemben már az egyetem elkezdésekor. Lassan ahogy a vége fele közeledek a tanulmányaimnak, egyre kevesebb időm maradt gondolkodni azon, hogy milyen területet célozzak meg az államvizsgám elkészítésével.   
Egy játék lett a végső döntésem, mert ezt nem láttam még sehol az elmúlt években, vagy ha volt is, akkor elkerülte a figyelmemet. Ez a játék, amit készítettem egy elsősorban számítógépre tervezett játék, de majd szeretném a jövőben mobiltelefonra is implementálni.

Nehéz egy olyan játékot létrehozni, ami már nincs a piacon, ezért kell arra törekedni, hogy valamiben különböző legyen a többi játéktól. Az én játékomban próbáltam minél letisztultabb, minél egyszerűbb formákat és színeket használni, hogy az átlagos felhasználó ne egy színözönt kelljen folyamatosan nézzen a képernyőjén, mert egyrészt megterhelő a szemnek és hamar unalmassá válik. A játék készítése során rengeteg akadályba ütköztem, amit kisebb nagyobb sikerekkel le is tudtam küzdeni, néhány igazán megizzasztott, de a legvégén mindig megtudtam oldani valahogy a problémákat. A nehézségek közé tartozott az is, hogy egy olyan pályát hozzak létre, amelyik végtelennek tűnik, mégis változatosan nézzen ki és érdekes maradjon a felhasználó számára. A karakter vezérlése egy olyan pont volt a játék fejlesztése során, ami megkövetelte, hogy mélyebben értelmezzem a programomat, mivel figyelembe kellett vegyem azokat a tényezőket is, amelyek befolyásolták, hogy a karakter ne tudjon kimenni a pályáról az oldalakon. A karakter animálása volt még egy bonyolultabb folyamat, különböző oldalakról gyűjtöttem az információkat hozzá és sajátítottam el az animálások módját.

A játék könnyű irányítása és könnyen megérthető játékmenete révén széles körben népszerű a számítógépes játékok között.

Végkifejletben pedig sikerrel könyvelhetem el a játékom részleges megvalósítását. A jövőben is szeretnék majd még játékokat készíteni, mert elég izgalmas és egyben bonyolult folyamat is. Tervezek még egy versenyzős játékot is megvalósítani az év végére és remélem, hogy sikerrel tudom zárni.

# **Használt technológiák**

Az államvizsgám készítése során több technológiát és eszközöket használtam, ami jelentős mértékben megkönnyítette a játékom elkészítését. Az alkalmazás fejlesztéséhez a Unity Game Engine biztosította az alapot, amely egy rugalmas játékfejlesztő motor, lehetővé teszi 3D és 2D játékok készítését.

A játékban használt karakterek implementálásához és animálásához a Mixamo nevű szolgáltatást vettem igénybe. A játék működéséhez a C# programozási nyelvet használtam. A Unity támogatja a C# használatát a játéklogika implementálásához és az események kezelésére. A C# segítségével készítettem a játékmenetet, a karakter irányítását, az akadályok kezelését és az eredmények megjelenítését.

## **Unity**

A Unity egy népszerű játékfelesztő motor és integrált fejlesztői környezet (IDE). A Unityben minden munka egy projektbe történik, egy projekt tartalmazza az összes szükséges fájlt, forráskódot és beállítást a fejlesztéshez. A projektek kezelése Unity Hub segítségével történik, amely lehetővé teszi új projektek létrehozását és meglévő projektek folytatását és kezelését.



2.1.Ábra. A Unity Editor megjelenése és funkciói

(forrás: <https://docs.unity3d.com/Manual/UsingTheEditor.html>)

A Unity Editor egy nagyon kifinomult és felhasználóbarát felületének használata egy kevés tanulással könnyen elsajátítható és máris különféle játékokat tudunk készíteni, már csak a képzeletünk szabhat határt nekünk. A mellékelt (2.1) ábrán megtalálhatóak különféle szerkesztőablakok, legfontosabb közülük az Eszköztár (A), amely segítségével hozzáférhetünk a felhasználói fiókunkhoz, a Unity szolgáltatásokhoz, megtalálható az Indítás, Stop és Szünet gombok is egyaránt amelyek a játék mód elindításához vagy megállításához szükségesek. A Hierarchia ablak (B), ahol minden játék objektumot megtalálunk amit létrehoztunk és megmutatja a struktúráját különböző játék objektumoknak, hogy hogy helyezkednek el egymás alatt és hogyan kapcsolódnak egymáshoz. A játék nézet (C), ahol hogyha elindítjuk a játékot, láthatjuk hogy hogyan fog kinézni a végső, renderelt verzió. A Jelenet (D) nézet, ahol módosítani, mozgatni, átméretezni tudjuk a játékban szereplő objektumokat. A Megfigyelő (F) ablakban a kiválasztott objektum különböző paramétereit tudjuk kedvünk szerint változtatni. Különböző játékobjektumoknak különféle paraméterei vannak, amiket szükség esetén össze is tudunk hangolni. A megfigyelő ablak minden alkalommal változik amikor egy más játék objektumot választunk ki. A Projekt (G) ablak megjeleníti az összes tartalmat(„asset”), eszközt amik rendelkezésre állnak a projektünkben, hogyha új tartalmat adunk a projektünkhöz, az is itt fog megjelenni.

A Unity Asset Store egy könyvtár amelyben ingyenes és kereskedelmi célú tartalmak találhatók, amelyeket a Unity Technologies és a közösség tagjai hoznak létre. Számos különböző eszköz elérhető, beleértve textúrákat, modelleket, animációkat, teljes projekt példákat, oktatóanyagokat és a Unity Editor kiterjesztéseit. 2020-tól már az Asset Store dedikált ablaka nem található meg a Unity Editoron belül, azonban továbbra is elérhetőek maradnak a tartalmak az Asset Store webhelyén, ahonnan a megszerzett tartalmaidat importálhatod és letöltheted őket közvetlenül a Package Manager ablakban.

## **C#**

A C# egy modern objektumorientált programozási nyelv, a C# lehetővé teszi a fejlesztőknek hogy biztonságos és megbízható alkalmazásokat készítsenek a .NET keretrendszerben. A C# a C programozási nyelvcsaládhoz tartozik, így azok számára is könnyen érthető, akik már ismerik a C, C++, Java vagy JavaScript nyelvek egyikét.

A C# objektumorientált, ami azt jelenti, hogy az alkalmazásokat objektumok és osztályok segítségével építjük fel. Az objektumokat az osztályok példányosításával hozzuk létre. Az osztályok meghatározzák az objektumok tulajdonságait és viselkedését.

A C# típusbiztos, ami azt jelenti, hogy erősen típusos nyelv, amely lehetővé teszi a fordítási időben és futási időben történő típusellenőrzést. A C#-ban két fajta típus létezik: értéktípusok és referenciatípusok. Az értéktípusok változói közvetlenül tartalmazzák az adatukat. A referenciatípusok változói referenciát tárolnak az adatukra, objektumként ismertek. A C#-ban az automatikus memóriakezelés(garbage collection) gondoskodik a memória felszabadításáról és a memóriahasználat optimalizálásáról.

A Unity integrálódik a Visual Studio-val a Visual Studio Code Editor Package segítségével. Ez a csomag automatikusan telepítésre kerül, amikor a Unity-t telepítjük, hogyha a Visual Studio telepítve van a számítógépünkre a Unity telepítésekor, akkor a Unity alapértelmezetten a Visual Studio-t használja a kódok megnyitására és szerkesztésére.

## **Mixamo**

A Mixamo egy ingyenes online szolgáltatás, amelyet az Adobe készített, mindenkinek hozzáférhető egy Adobe ID-vel és nem szükséges semmiféle előfizetés a szolgáltatás használatához. Lehetővé teszi a karakterek és animációk importálását és személyre szabását. A Mixamo egy tökéletes szolgáltatás mindazoknak, akik szeretnének karaktereket, vagy animációkat használni a videójátékuk készítése során, emellett még azoknak is hasznos lehet, akik 3D nyomtatáshoz szeretnének karaktereket alkotni vagy keresni és kisfilmek készítésére is jó választás lehet. Minden szolgáltatás és tartalom ingyenes, személyes felhasználásra.

A Mixamo támogatja a legtöbb közismert karaktermodellező szoftvertől származó karaktermodelleket, mint például a Blender, Autodesk Maya, szükséges viszont, hogy az importált modellnek tartalmaznia kell a megfelelő hierarchiát.

Nagyon egyszerű a karakterek importációja is Unity-ben, A Mixamo automatikusan érzékeli a karakter modelljének rögzített pontjait amelyek alapján az animációkat illeszti az adott karakterre.

Rengeteg animáció és karakter található a Mixamo weboldalán, mindenki meg tudja találni a magának megfelelőt, hogyha nem lenne megfelelő az időtartam, a sebesség vagy a karakterspecifikus paraméterek, a Mixamo testreszabási lehetőséget is kínál ezen problémák orvosolására.

A Mixamo jelentős mértékben megkönnyítette a játék elkészítését, mivel nem rendelkezek jelenleg a megfelelő tudással, hogy saját karaktereket és animációkat készítsek, persze a jövőben, hogyha fogok még játékot készíteni, akkor már megpróbálkozok saját karakterek és animációk elkészítésével.

## **GitHub**

A github egy webes alapú verziókezelő platform és verziókezelő rendszer, amelyet a fejlesztők használnak a kódváltoztatások nyomon követésére és veziókezelésre. Rengeteg előnye van a GitHub-nak, mint például a távoli hozzáférés az adattárakhoz(„repository”) és a kollaboráció más fejlesztőkkel, vagy személyekkel, akiknek hozzáférést adunk az adattárainkhoz.

A GitHub egy könnyen átlátható és megérthető rendszeren alapszik, könnyű használhatósága miatt kiemelkedő szerepet tölt be nagyobb projektek készítése során. Pár alapvető parancsot kell a felhasználó megtanuljon, vagy megkeressen, amivel már ő is fel tud tölteni adatokat, kódrészleteket a saját adattárjába. A weboldalukon részletetsen le van írva, hogy hogyan lehet létrehozni, klónozni egy adattárat, különböző ágak(„branch”) elkülönítését, használtát. Emellett még van néhány parancs, amire nem fogok nagyon kitérni, mint például a „commit”, ami a kódváltoztatások rögzítésére szolgál, „add”, „pull” „push” parancsok, amik szükségeltetnek egy sikeres adattár létrehozásához és naprakészen tartásához.

Rengeteg pozitív oldala van, amiért megéri használni a GitHub-ot, többek között könnyen átlátható, megérthető, többen tudnak dolgozni ugyanazon a projekten, különböző ágak segítségével külön fejlesztési vonalakat hozhatnak létre, hogy új funkciókon, vagy hibajavításon dolgozzanak, majd lésőbb ezeket az ágakat összevonják a fő ággal.

## **MySQL**

A MySQL használata is hasznos volt a játék felhasználói felülete megalkotása során. Vannak más különböző módszerek amikkel meg lehet valósítani a kívánt felületet, de ezt tanultuk az egyetemi éveink alatt is és közelebb is áll hozzám, azért esett a választásom erre.

A MySQL egy nyílt forráskódú adatbázis-kezelő rendszer, széles körben használnak a webfejlesztés és alkalmazásfejlesztés terén. Rengeteg fejlesztői funkciót kínál és egy hatékony, megbzízható adatbázismotorral rendelkezik.

A MySQL kjváló választás lehet olyan alkalmazásokhoz, amelyeken szükség van adatbázis-kezelésre, lehetővé teszi a fejlesztők számára az adatok hatékony tárolását, lekérdezését és módosítását. Rendelkezik különböző funkciókkal, mint például tranzakciókezelés, indexelés, tárolt eljárások és funkciók, valamint támogatja az adatbázisok replikációját és másolását.

Széles körben használják az iparban, nagy tejlesítményű adatbázisok kezelésére alkalmas. Számos nagyvállalat, szoftverfejlsztő támaszkodik a MySQL-re adatok tárolása és kezelése szempontjából.

A dokumentáció bemutatja, hogyan hozhatunk létre és kezelhetünk adatbázis-kapcsolatot a Visual Studio fejlesztői környezetben. A lépések közé tartozik a MySQL adatbázis szerver adatainak megadása, például a szerver címe, a portszám, a felhasználónév és a jelszó. Ezután megtudjuk, hogy hogyan ellenőrizzük és teszteljük a kapcsolatot, hogy megbizonyosodjunk arról, hogy mnden sikeres volt és sikeresen csatlakozunk-e az adatbázishoz.

# **Játék bemutatása**

A játéknak amit készítettem egyszerű szabályai vannak, könnyen felismerhető mindenki számára, sokan találkozhattak vele már életük során. A játékban a karakter folyamatosan szalad előre és a feladat az, hogy úgy irányítsd, hogy kerülje ki az akadályokat. Az akadályok változatosak lehetnek, például ugrásra, vagy oldal mozgásra lesz szükség kikerülésükhöz, rönköket, fákat, köveket és más veszélyeket kell elkerülj. Az irányítás rendkívül könnyen kezelhető, a WASD gombokkal és a nyilakkal történik. A játék során minél több pontszámot kell összegyűjteni, amit úgy lehet elérni, hogy minél tovább életben maradsz, vagyis minél több akadályt kikerülsz. A játék végtelenül játszható, ami azt jelenti, hogy nincs konkrét végcélja, hanem a cél az, hogy minél több pontszámot összegyűjts, a játék viszont izgalmas marad, hiszen mindig póbálod megdönteni a korábbi rekordodat.

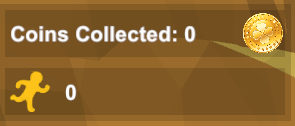
A mellékelt ábrákon látható pár részlet, amit kiragadtam a játékból, a 3.1. Ábrán található néhány egyszerű menüpont, a legelső, ahol el tudod indítani a játékot, a másodikkal pedig karaktereket tudsz választani, a harmadik menüonttal meg tudod nézni a ranglistát, azt hogy ki tartja az első helyet, a negyedikkel pedig a fejlesztő szavait tudod elolvasni és az utolsó menüponttal pedig ki tudsz lépni a játékból, hogyha úgy gondolod hogy eleget játszodtál vele.

A 3.2. Ábrán pedig az eredményjelzőt találjuk, ami a játék menete közben mutatja, hogy aktuálisan mennyi a pontszámunk és hogy mennyi érmét gyűjtöttünk össze.



3.1 Ábra. A menü megjelenése

Az eredményjelzőt fontosnak találtam megjeleníteni a játékban, mert folyamatos visszajelzést ad a játékosnak, hogy mennyi pontja és érméje van az adott körben és motiválja arra, hogy minél tovább tudjon haladni és minél több pontot és érmét gyűjtsön. Próbáltam egy olyan eredményjelzőt készíteni a játékban, ami nem csak két számot mutat, hanem a játékosok tudják kötni valamihez, mint például az érme vagy a kis szaladó emberke az eredményjelző bal alsó sarkában.

. 

3.2 Ábra. Az eredményjelző a játékban

A játék véget érésekor gondoltam rá, hogy nem csak egy elsötétülő képernyő fogadja majd a játékosokat, hanem egy kissé színesebb, információval gazdagabb oldal, a 3.3. Ábrán látható az, hogy milyen módon jelenik meg a játékosok számára. Láthatják az hogy milyen messzire jutottak el a játékban, az összpontszámukat és azt hogy mennyi érmét gyűjtöttek össze a játék során.



3.3 Ábra. Végső eredményjelző



3.4 Ábra karakterválasztó menüpont

A 3.4. Ábrán pedig a másik karakter látható, amit hozzáadtam a játékhoz, egyelőre teszt fázisban van, mivel még vannak olyan részei a játéknak, amit a jövő során szeretnék kifejleszteni. A karakterválasztó menüben két karakter közül fogunk tudni választani, az egyik egy kislány, akit Mirabelle-nek hívnak és a másik egy lovag, aki az Andew nevet kapta. A karakterválasztás nem szolgál teljesítménybeli előnyöket, nem különbözik semmi a két karakter irányíthatósága és mozgáskészségei között, egyedül optikai differenciát szolgál, hogy ne legyen a játék unalmas és

## **Programozási környezet**

A C# programozási környezet a Unity Engine-re épül, amely egy népszerű játékfejlesztő motor és fejlesztői környezet, lehetővé teszi a játékfejlesztést és a játékok több platformra történő exportálását.

A C# nyelvet használva a Unity-ben kódolhatjuk a játéklogikát, kezelhetjük az adatbázis-kommunikációt és irányíthatjuk a karakter mozgását és animációját.

Az adatbázis tervezésével kezdődik a ranglista megvalósítása. Első lépésként meg kell tervezni az adatbázis strukturáját, definiálni kell az adatbázis táblákat,amelyek tartalmazzák a játékosok nevét és pontszámát. Az adatbázis kapcsolatot meg kell valósítani a játék és az adatbázis között. Ehhez használtam a MySQL adatbázis-kezelő rendszert és az azt támogató C# adatbázis-kezelő könyvtárat. Implementálni kell a szükséges adatbázis műveleteket, mint például az adatok lekérdezését a ranglista megjelenítéséhez, az adatok frissítését új pontszámok hozzáadásával és az adatok beszúrását új játékosok hozzáadásával.

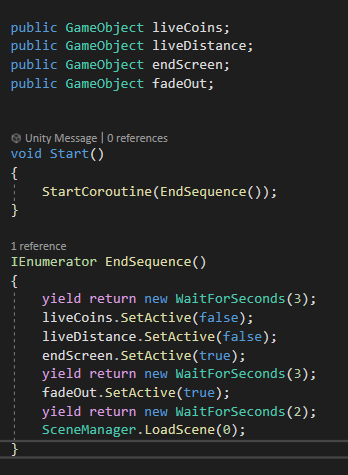
A karaktermozgás és játéklogika terén kezdtem a játékobjektumok létrehozásával, először is a karakter létrehozásával, amely a játékost képviseli az Everlasting Escape játékban. Ezután definiáltam az akadályokat és más elemeket, amelyekkel a karakter interakcióba léphet. A karaktermozgás logikáját kellett megértenem az elején, hogy a karakter automatikusan előre mozogjon, és nekiütközzön az akadályoknak. Szükségem volt ütközésdetekcióra is hogy azonosítsa az ütközéseket az akadályokkal, vagy más elemekkel.

A bemenetkezeléshez is a C# nyelvet használtam, hogy lehetőséget biztosítsak a játékosnak a karakter irányítására például az oldalra történő mozgásra és az ugrásra, ehhez kellett figyelnem a játékos bemeneteire, például a billentyűleütésekre.

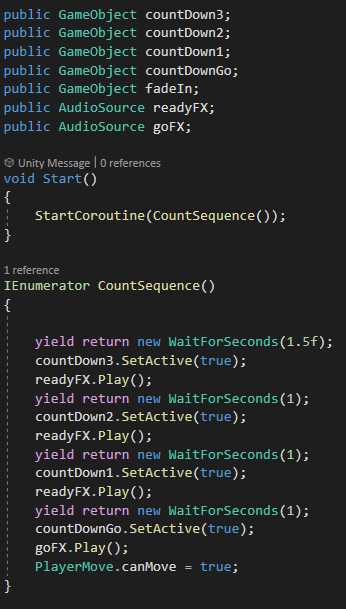
Az animációk és karakterek importálásához és implementálásához a Mixamo-t használtam. Az animációkat megfelelően kellett hozzárendeljem a karakter modelljéhet a Unity animációs rendszerével. Különböző hosszúságú animációkra volt szükségem minden animációnál, mert egyik sem ugyanakkora időtartamú mint a másik. Az ugráshoz más intervallumú animáció szükséges és a mozgásokhoz is egy különböző intervallumú animáció tartozik.

A Unity számos beépített funkciót és eszközt biztosít a játékfejlesztéshez. A játéklogikát C# nyelven valósítottam meg a Unity projektemben. Ehhez létre kellett hozzak különböző kódrészleteket a Unity Editorban(“scriptek”), amelyek kezelik a karaktermozgást, az akadályokat az adatbáziskapcsolatot, a karakter-adatbázist. A Unity komponensek tették lehetővé a könnyű hozzáférést a játékobjektumokhoz, a grafikához és más funkciókhoz.

A 3.3.1. Ábrán látható kódrészlettel készítettem el a játék véget érésekor megjelenő képernyőt. Egy Coroutine-t használtam, amiben megadtam, hogy egy bizonyos idő után megjelenjenek a játékos által elért pontok és az összgyűtött érmék száma.



3.1.1. Ábra. Játékvégi kijelző



3.1.2.Ábra. A játék kezdetén lévő hangok és visszaszámláló



3.1.3.Ábra. Ütközéskor bekövetkező animáció és eseménykezelés

## **Adatbázis**

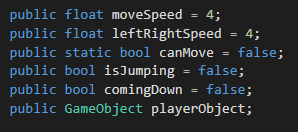
Az adatbázisomhoz MySQL-t használtam, PhpMyAdmin felületen készítettem a táblákat és helyi rendszeren tárolom az adatokat, amíg nem lesz egy nagy felhasználóigényű játék, addig tökéletesen megfelel a helyi adattárolás, nem szükséges egy fizetős szolgáltatás igénybevétele, de majd a jövő során erre is sor kerülhet, ami remélem hogy majd be is fog következni. Az adatbázis tartalmazza a mellékelt ábrán látható táblákat, ahol a játékos felhasználóneve, az elért pontszáma és egy ID fog tárolódni, ezekkel az adatokkal fogom feltölteni a ranglistán megtalálható eredményjelzőt.



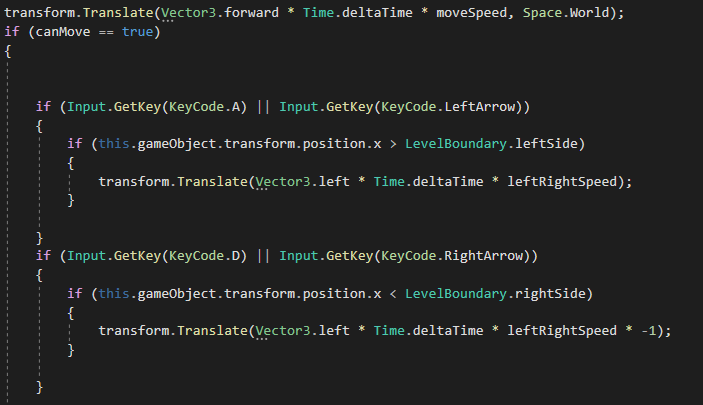
3.1.Ábra. Az alkalmazás adatbázis sémája

## **Az alkalmazás fontosabb funkciói**

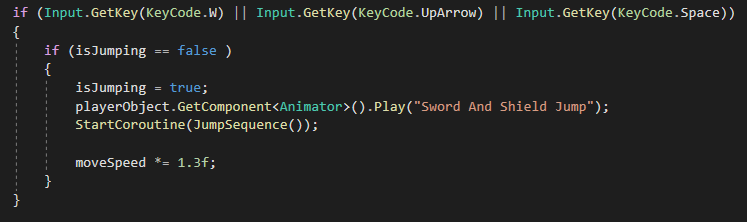
Leirni ide, hogy miket tud a játék, hogy megy a mozgás, stb.



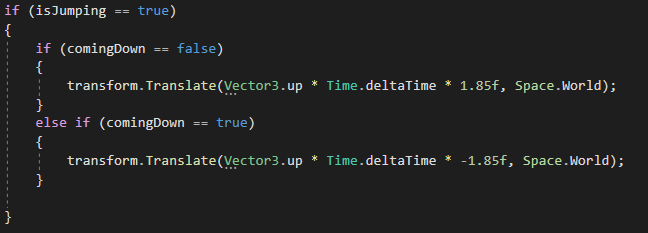
3.5. Ábra. Mozgás paraméterek



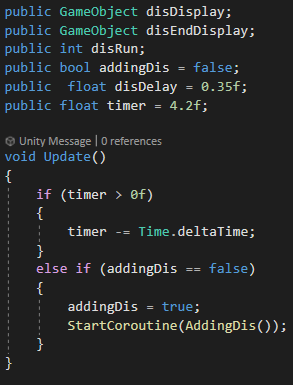
3.6. Ábra. Mozgás kódolása



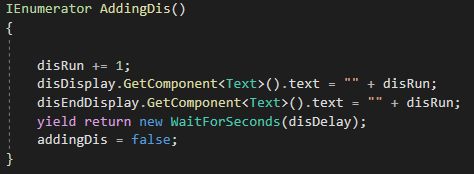
3.7. Ábra. Ugrás animáció társítása



3.8. Ábra. Ugrás kódolása



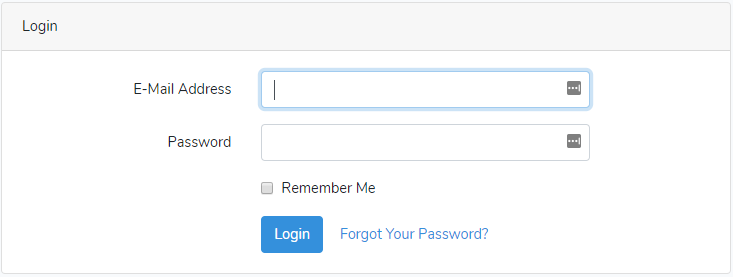
3.9. Ábra. Eredményjelző



3.10. Eredményjelző

### **Felhasználói felület**

A játék elején lesz egy login, ahova beirja a játékos a játékosnevét. Le van kezelve, hogy egy játékos egy eszközről csak egy felhasználóval tudjon játszani, hogy elkerüljük a duplikálásokat és hogy más játékosok is sikerélménnyel tudják játszani a játékot.



3.2.Ábra. Login ablak

### **Ranglista**

A ranglistában megtekinthető az első x játékos legjobb pontszáma, a

3.3.Ábra. Felhasználó hozzáadása ranglistához

# **Következtetések**

Ennek a játéknak az elkészítése rengeteg tapasztalatot és bátorságot adott nekem, hogy majd belekezdjek egy jobban kidolgozott, modernebb játéba a közeljövőben. A játék készítés egy elég bonyolult, de annál kifinomultabb folyamat, rengeteg türelem és odaadás kell hozzá, nagyon sok mindent le kell tesztelni, ki kell próbálni a funkciókat, javítani a hibákat és megtenni mindent, hogy egy olyan játékot tudjunk elkészíteni, ami megragadja a játékost, nem unalmas és leköti a figyelmet.

Az eddigi államvizsga dolgozatokban nem véltem felfedezni hasonló játékot, ami egy kissé megrémisztett az elején, de végül még is belevágtam, amit nem bántam meg, mert máshonnan ezt a tudást nem tudtam volna megszerezni.

# **Irodalomjegyzék**

1. <https://docs.unity3d.com/Manual/Toolbar.html>.
2. <https://docs.unity3d.com/Manual/UsingTheEditor.htmle>.
3. <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
4. <https://docs.unity3d.com/Manual/VisualStudioIntegration.html>
5. <https://helpx.adobe.com/creative-cloud/faq/mixamo-faq.html>
6. <https://docs.github.com/en>
7. <https://dev.mysql.com/doc/visual-studio/en/visual-studio-intro.html>