**ZÁRÓDOLGOZAT**

**SZABÓ KEVIN MÁRK**

**SZOFTVERFEJLESZTŐ**

**BAJAI SZC TÜRR ISTVÁN TECHNIKUM**

**SZOFTVERFEJLESZTŐ**

**ZÁRÓDOLGOZAT**

CSAK ELŐRE!

SZABÓ KEVIN MÁRK  
2024

**NYILATKOZAT A ZÁRÓDOLGOZATRÓL**

Alulírott Szabó Kevin Márk (név) tanuló

**kijelentem, hogy**

Csak Előre!című záródolgozatomat (nyomtatott és elektronikus formában) a Bajai SZC Türr István Technikum pedagógusai és tanulói:

* + felhasználhatják (pl. hivatkozás alapjául, olvasótermi használatra) későbbi munkájukhoz a szerzői jogok tiszteletben tartása mellett).
  + nem használhatják fel (titoktartási nyilatkozat csatolása mellett).

Ugyanakkor kijelentem, hogy a záródolgozat ***saját munkám eredménye*.**

Baja,  
20*............................*

*.........................................*

aláírás

Tartalom

[1.Bevezetés 6](#_Toc165533599)

[1.1 A záródolgozat témája 6](#_Toc165533600)

[1.2 Témaválasztás indoklása 6](#_Toc165533601)

[1.3 A dolgozat célkitűzései 8](#_Toc165533602)

[1.4 A dolgozat felépítése 8](#_Toc165533603)

[2. Felhasználói dokumentáció 8](#_Toc165533604)

[2.1 Általános specifikáció 8](#_Toc165533605)

[2.1.1 Az alkalmazás használata 9](#_Toc165533606)

[2.1.2 Rendszerigény 9](#_Toc165533607)

[2.2 Indítás és játékmenet 9](#_Toc165533608)

[2.2.1 Felhasználói felület 12](#_Toc165533609)

[2.2.2 Játékmenet funkciók 18](#_Toc165533610)

[3. Fejlesztői dokumentáció 19](#_Toc165533611)

[3.1 Felhasznált technológiák 19](#_Toc165533612)

[3.1.1 Unity 2023.2.6f1 19](#_Toc165533613)

[3.1.2 Visual Studio 2022 - C# 19](#_Toc165533614)

[3.1.3 XAMPP– MySql 20](#_Toc165533615)

[3.1.4 Blender 21](#_Toc165533616)

[3.1.5 Unity Assets Store 22](#_Toc165533617)

[3.2 Verziókövetés 22](#_Toc165533618)

[3.2.1 Git 22](#_Toc165533619)

[3.2.2 GitHub 23](#_Toc165533620)

[3.2.3 GitHub Desktop 23](#_Toc165533621)

[3.3 Projektmenedzsment 23](#_Toc165533622)

[3.3.1 Atlassian Jira 24](#_Toc165533623)

[3.4 Adatbázis 24](#_Toc165533624)

[3.5 Forráskódok 26](#_Toc165533625)

[3.5.1 Write.cs 26](#_Toc165533626)

[3.5.2 MenuControl.cs 27](#_Toc165533627)

[3.5.3 Control.cs 28](#_Toc165533628)

[3.5.4 MusicControl.cs 29](#_Toc165533629)

[3.5.5 EredmenyRead.cs 30](#_Toc165533630)

[3.6 Fájlstruktúra 32](#_Toc165533631)

[4. Összegzés 35](#_Toc165533632)

[4.1 A terv és az elkészült projekt különbségei 35](#_Toc165533633)

[4.2 Személyes tapasztalatok és problémák 36](#_Toc165533634)

[4.3 Tesztelés 37](#_Toc165533635)

[4.4 Fejlesztési lehetőségek 37](#_Toc165533636)

[5. Források, mellékletek 38](#_Toc165533637)

[5.1 Források 38](#_Toc165533638)

[5.2 Mellékletek 43](#_Toc165533639)

# 1.Bevezetés

Ajáték GitHub repository hivatkozása:   
<https://github.com/K3WiiiN/CsakElore.git>

A buildelt játék pedig itt érhető el:  
<https://mega.nz/file/lb9SCY5a#3SizfgPN6EW9aGuR_7N540xILYwWoGc82UivHIP5o2k>

## A záródolgozat témája

A záródolgozatom címe: Csak Előre!, témája egy 3D-s egyszemélyes platform játékszoftver.

A játék lényege, hogy a felhasználó (továbbiakban játékos) a játékkarakterrel pályákat teljesítsen. A játékban 5 pálya elérhető, amelyek kialakítása különböző. A pályák mindegyike áll a talajból; akadályokból amelyek a talajon, vagy levegőben is lehetnek; gyorsítókból amelyet ha játékos elér, gyorsabban tud mozogni; célvonalból, amelyet ha elér, a pálya teljesítve lesz, és mentésre kerül az idő, mivel a pályákat időre kell teljesíteni.

## 1.2 Témaválasztás indoklása

A szakdolgozatom témájának választásakor több szempontot fontosnak tartottam figyelembe venni. Az akadálypálya játék egy olyan terület, amely ötvözi a szórakoztató játékelemeket és a kihívást jelentő problémamegoldást. Az alábbiakban bemutatom, miért döntöttem úgy, hogy éppen ezt a témát választom, és miért tartom fontosnak a szakdolgozatom megírásában.

1. Érdeklődésem a játékok iránt: Már gyermekkorom óta szeretem a játékokat, és mindig is érdekelt, hogyan lehet őket tervezni, fejleszteni. Az akadálypálya játékok különösen izgalmasak számomra, mert sokféle kreatív és technikai elemet ötvöznek.

2. Játékok és tanulás kapcsolata: A játékoknak hatalmas szerepük van a tanulásban és a készségfejlesztésben. Az akadálypálya játékok kiválóan alkalmasak olyan készségek fejlesztésére, mint a problémamegoldás, a tervezés és a stratégiaalkotás. Személy szerint a játékok sokat segítettek az angol nyelv tanulásában, bár a záródolgozatom játéka magyar nyelvű.

3. Informatikai vonatkozások: A szakdolgozatom során lehetőségem nyílik mélyebben megismerni az informatikai alkalmazásokat és technológiákat, amelyeket a játékfejlesztéshez lehet alkalmazni. Ez a gyakorlati tapasztalat nagyban hozzájárulhat a szakmai fejlődésemhez.

4. Kreativitás kibontakoztatása: Az akadálypálya játék tervezése és fejlesztése során kreativitásomnak csak a képzelet szabhat határt. Ezáltal lehetőségem van új és egyedi ötletek kipróbálására, amelyeket később más projektekben is hasznosíthatok.

5. Kihívás és fejlődés: A szakdolgozatom során számos kihívással kell szembenéznem, például a programozási kódok optimalizálásával, adatbázis összekapcsolással vagy a játékmenet tervezésével kapcsolatban. Ezek a kihívások lehetőséget adnak a szakmai fejlődésre és a problémamegoldó képességek gyakorlására.

6. Felhasználói élmény javítása: Az akadálypálya játékok tervezése során fontos szempont a felhasználói élmény optimalizálása. A játékosoknak kellemes és izgalmas élményt kell nyújtanom, amihez részletes tervezésre és tesztelésre van szükség.

7. Közösségi érték: A játékok mindig is képesek voltak embereket összehozni, legyen szó barátokról, családtagokról vagy akár teljesen ismeretlen emberekről. Az akadálypálya játékok szórakoztató és közösségépítő tevékenységeket kínálnak, amelyeknek pozitív hatása lehet a társadalomra. Annak ellenére, hogy a záródolgozatom egyjátékos játék, az előbb leírtak is fontosak lehetnek későbbi programok fejlesztése során.

Összességében tehát az akadálypálya játék témáját választottam szakdolgozatomnak, mert hiszek abban, hogy ez a téma lehetőséget nyújt számomra a szórakoztató és innovatív projektek tervezésére, valamint a szakmai és személyes fejlődésem elősegítésére is.

## 1.3 A dolgozat célkitűzései

Záródolgozatként készült szoftver célkitűzése:

-Felhasználók kezelése: regisztráció, bejelentkezés lehetőség.

-Felhasználó játékban elért eredményeinek megjelenítése.

-A felhasználói felületek egyszerűsége, letisztultsága.

-Játék karakter, világ, objektumok a lehetőségeimhez képest élethűek legyenek.

## 1.4 A dolgozat felépítése

1. rész - Bevezetés

2. rész - Felhasználói dokumentáció

3. rész - Fejlesztői dokumentáció

4. rész - Összefoglalás

5. rész - Források, mellékletek

# Felhasználói dokumentáció

Ez a rész a felhasználó számára mutatja be az alkalmazás használatát, a futtatáshoz szükséges rendszerkövetelményt, az indítás menetét, továbbá ismerteti a játékmenetet, folyamatokat és a felhasználói felületek kezelését.

## 2.1 Általános specifikáció

Az alkalmazás azoknak a felhasználóknak ajánlott, akik 3D egyjátékos platformjátékkal szeretnének játszani.

## 2.1.1 Az alkalmazás használata

Az alkalmazást használók (továbbiakban játékos) általános felhasználói jogokkal rendelkeznek. A játékosnak lehetősége van és szükséges is, hogy felhasználót regisztráljon felhasználónévvel, e-mail címmel és jelszóval. Csak regisztrált felhasználóval lehet bejelentkezni, a felhasználónév és jelszó megadásával. A bejelentkezés azért fontos, mert csak így érhetőek el az alkalmazás fő elemei, például a játszani kívánt pálya választása, és így a pálya teljesítése. A pálya teljesítésének eredménye (teljesítésre szánt idő) a bejelentkezett felhasználónak eredményeként kerül mentésre, tehát az eredmény későbbi megtekintéséhez is szükséges a felhasználó kezelése az alkalmazásban.

## 2.1.2 Rendszerigény

A rendszerigény részletesen nem ismert!

- RAM: 4GB vagy több

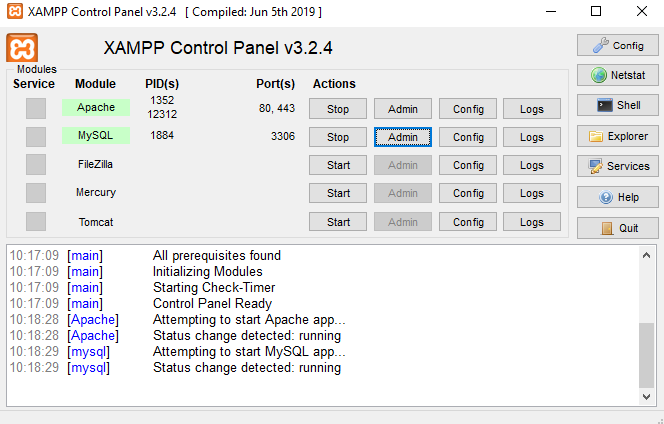
- Operációs rendszer: Windows 10

- Minimum 5GB szabad tárhely

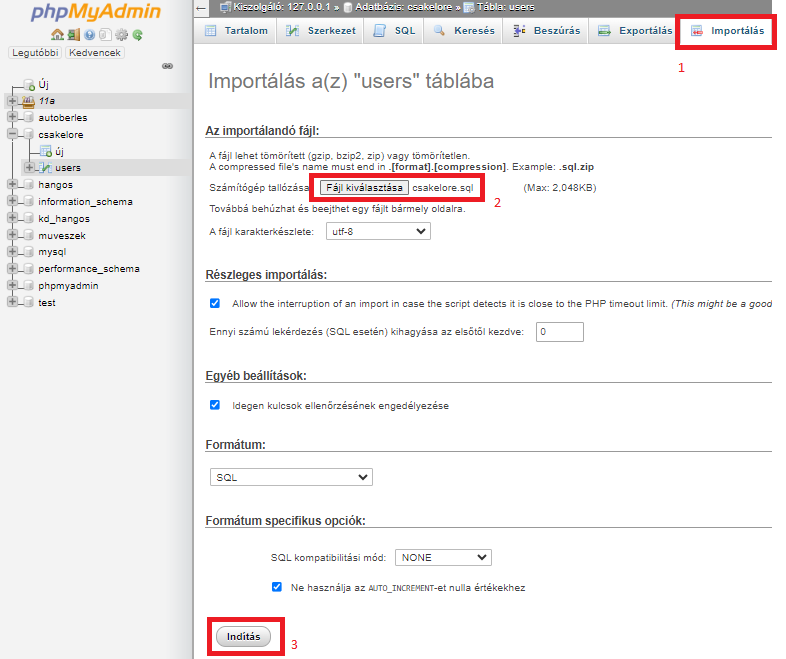
- Szélessávú internetkapcsolat

## 2.2 Indítás és játékmenet

Mivel a játékban a felhasználókezelés adatbázisban történik, a legfontosabb elindítani az adatbázist. Ezt az XAMPP Control Panelben kell megtenni, Apache és MySql szolgáltatásoknál a Start gombra kattintani, majd a MySql szolgáltatásnál az Admin gombra, amely a böngészőben nyitja meg a phpMyAdmin nevű felületet, majd amennyiben az első indítás történik, be kell importálni a *csakelore.sql* fájlt. Ennek eredményeként létrejön a *csakelore* nevű adatbázis, és gond nélkül futtathatóvá válik a játék.

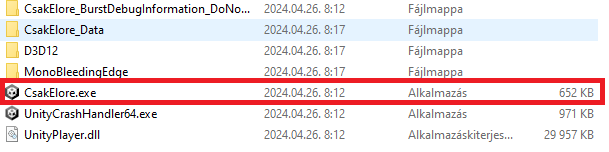


1. ábra XAMPP Control Panel



2. ábra phpMyAdmin felülten csakelore.sql fájl beimportálása

A játék elindítása után (CsakElore.exe) a Belépés menü jelenik meg, a játék címével, majd a “Belépés” és “Kilépés” gomb.



3. ábra CsakElore.exe a játék buildelt mappájában



4. ábra Belépés menü

## 2.2.1 Felhasználói felület

A “Belépés” gomb megnyitja a Bejelentkezés menüt, amelyben meg kell adnunk az előre regisztrált adatainkat (felhasználónév, jelszó), a “Kilépés” gomb bezárja a játékot. Ha még nem regisztráltunk, a regisztráció részen tehetjük meg (szintén a Bejelentkezés menüben), a felhasználói adatok adatbázisban vannak kezelve. A regisztrációhoz szükséges megadni felhasználónevet (minimum 4 karakter; (illetve az adatbázis 30 karaktert ment)), e-mail címet (tartalmaznia kell @; . karaktert, különben helytelen a formátum), jelszót (minimum 8 karakter), jelszó megerősítést (egyeznie kell a jelszóval). A kilépés gomb visszairányít a Belépés menübe. Sikertelen bejelentkezés vagy regisztráció esetén a felületen hibaüzenetek jelennek meg.

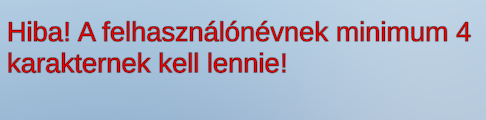


5. ábra Bejelentkezés menü

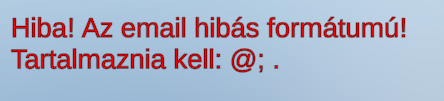
Hibaüzenetek:



6. ábra Bejelentkezési hiba: üresen hagyott mező vagy még nem létező felhasználó esetén



7. ábra Regisztrációs hiba: Felhasználónév rövidebb 4 karakternél



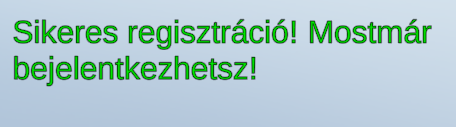
8. ábra Regisztrációs hiba: E-mail cím helytelen formátumú



9. ábra Regisztrációs hiba: Jelszó rövidebb 8 karakternél



10. ábra Regisztrációs hiba: Jelszó és ellenőrző jelszónem egyezik



11. ábra Sikeres regisztráció visszajelzés

Sikeres bejelentkezés után Főmenü nyílik meg, játék címe alatt: “Pálya választás” gomb; “Eredmények” gomb; “Beállítások” gomb; “Kilépés” gomb, bal alsó sarokban a bejelentkezett felhasználó neve.



12. ábra Kép a főmenüről

A gombok menükbe vezetnek:

“Pálya választás” gomb - Pálya választás menü

“Eredmények” gomb - A pályákhoz tartozó eredmények menüje

“Beállítások” gomb - Irányítás leírásához tartozó menü

“Kilépés” gomb - Kilépés a főmenüből

Pálya választás menü:

A pályák sorszámozva vannak, az azonosítás érdekében

A pálya kiválasztásához, elindításához kattintani kell a pálya gombjára.

A gombok alatt segédlet, a kupák szerzéséhez tartozó idők leírásával.

Kilépés gomb a főmenübe lépéshez.



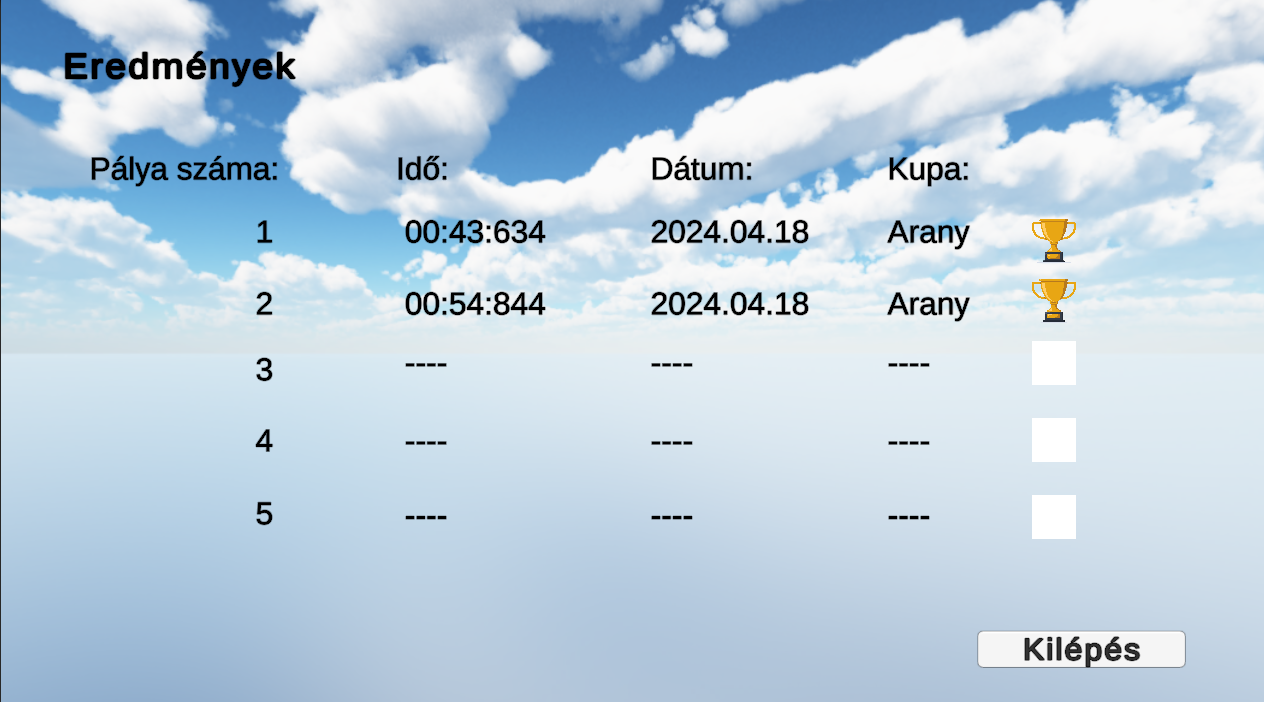
13. ábra Kép a Pályaválasztás menüről

Eredmények menü:

Az eredményeket minden pálya teljesítése során fájlba menti a program. A fájlban megtalálható a teljesítésre szánt idő, és a mentés dátuma.

Táblázatban jelenik meg a pályák sorszáma, mellette a legutóbbi teljesítésre szánt idő, az idő alapján kapott kupa típusa és képe.

Kilépés gomb a főmenübe lépéshez.



14. ábra Kép az eredmények menüről

Beállítások menü:

Felsorolva jelennek meg a következő információk:

Billentyűzet és egérrel:

W - előre, A – balra, S - hátra, D – jobbra, Szóköz - ugrás, Shift (bal) - sprint, ESC - kilépés, Egér mozgatás - kamera forgatása a játékos körül.

Kilépés gomb a főmenübe lépéshez.



15. ábra Kép a Beállítások menüről

## 2.2.2 Játékmenet funkciók

A pálya kiválasztása majd betöltése után visszaszámlálás: 3 másodperc - leteltével a játékos irányíthatja a karakterét.

A pálya teljesítésére szánt idő mérése a visszaszámlálás letelésével indul a képernyő jobb felső sarkában kiírva, a célba jutáskor megáll, mentésre kerül. A játékos a teljesítése ideje alapján 3 típusú kupa közül kaphat egyet, minden pályánál egyedileg meghatározottan. Legjobb idő - Arany kupa; Közepes idő - Ezüst kupa; Legrosszabb idő - Bronz kupa.

A pálya áll a talajból; akadályokból amelyek a talajon, vagy levegőben is lehetnek; gyorsítókból amelyet ha játékos elér, gyorsabban tud mozogni; célvonalból, amelyet ha elér, a pálya teljesítve lesz, és mentésre kerül az idő.

A játékfolyamat bármikor megállítható az ESC megnyomásával, ilyenkor a mozgás és az idő megáll, ”Beállítások” menü jelenik meg.

“Folytatás gomb” - Folytatódik a játék megtartva a játékos pozícióját és idejét.

“Kilépés gomb” - Kilépés a pályaválasztó menübe

# 3. Fejlesztői dokumentáció

Ez a rész ismerteti a projekt készítéséhez használt technológiákat, programokat, és a fejlesztés menetét, illetve bemutatom egyes kódrészletek működését.

## 3.1 Felhasznált technológiák

Mivel a projekt elkészítéshez nem szándékoztam fizetős technológiákat használni, így a következő programokat és rendszereket vettem igénybe, ezek mind ingyenesen elérhetőek:

## 3.1.1 Unity 2023.2.6f1

A Unity egy olyan játékfejlesztő platform, amely lehetővé teszi fejlesztők számára, hogy egyszerűen és hatékonyan hozzanak létre különböző típusú játékokat. A Unity széles körű eszközöket és funkciókat kínál, amelyek segítségével a fejlesztők könnyedén tervezhetnek és fejleszthetnek játékokat különböző platformokra, mint például mobil eszközök, számítógépek vagy konzolok. Összességében a Unity egy nagyon népszerű és sokoldalú játékfejlesztő platform, amely lehetővé teszi a fejlesztőknek, hogy magas minőségű játékokat hozzanak létre gyorsan és hatékonyan.

A Unity-n belüli játékfeljesztést a középiskola 5. év elején ismertem meg, a Unity Create With Code kurzusai segítettek az alapok elsajátításában. Munkámat megkönnyíti a program használatának egyszerűsége, és az interneten széles választékban megtalálható segédanyagok.

## 3.1.2 Visual Studio 2022 - C#

A Visual Studio 2022 a Microsoft integrált fejlesztői környezete (IDE) webes, mobil, asztali és felhőplatformokra való alkalmazások építéséhez. Számos eszközt és funkciót tartalmaz, amely segíti a fejlesztőket abban, hogy hatékonyabban írják, hibákat keressenek és teszteljék a kódjukat. A Visual Studio 2022 javított teljesítményt, kibővített együttműködési funkciókat és támogatást kínál a legújabb technológiákhoz és keretrendszerekhez. Elérhető Windowsra és Mac operációs rendszerekre.

A C# egy népszerű, objektumorientált programozási nyelv, amit a Microsoft fejlesztett ki. Széles körben használják különféle alkalmazások fejlesztésére, beleértve az asztali, webes és mobil alkalmazásokat is. A C#-ot a szimplicitása, sokoldalúsága és skálázhatósága miatt ismerik, amelyek miatt a fejlesztők körében kedvelt a hatékony szoftvermegoldások létrehozására.

Számomra a Visual Studio és a C# programozási nyelv középiskolai tanulmányaim kezdetekor vált ismertté, azóta használom, és az évek alatt elmondhatom, hogy szeretek benne dolgozni. A Visual Studio és a Unity remekül működik együtt, így ez nagy segítség a munkám elkészítéséhez. A C# programozási nyelv számomra viszonylag könnyen tanulható, és az egyik legkedveltebb programozási nyelv, amit használni szoktam, ezért választottam a záródolgozatomhoz is.

## 3.1.3 XAMPP– MySql

Az XAMPP egy ingyenes és nyílt forráskódú, platformfüggetlen webszerver megoldáscsomag, amit az Apache Friends fejlesztett ki. Főleg az Apache HTTP Server-t, MySQL adatbázist és az PHP és Perl programozási nyelvek interpretálóit tartalmazza. Az XAMPP-ot weboldalak és alkalmazások helyi fejlesztéshez és teszteléséhez használják személyi számítógépen.

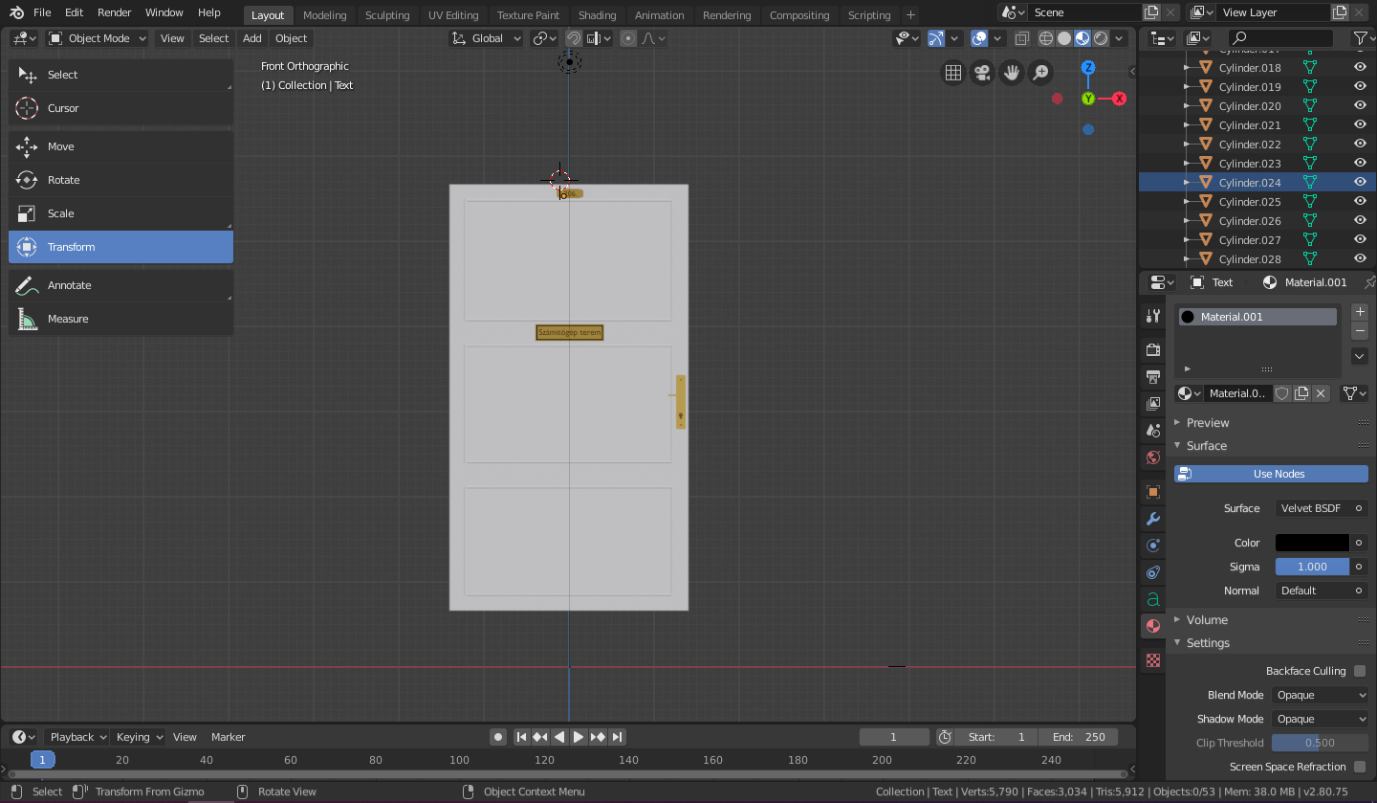
MySQL egy nyílt forráskódú relációs adatbázis-kezelő rendszer, amely gyakran használatos webalkalmazásokhoz. Ismert a megbízhatóságáról, skálázhatóságáról és könnyű használatáról. MySQL használ egy SQL (Struktúrált Lekérdezési Nyelv) nevű nyelvet az adatbázisokkal való interakcióhoz és olyan műveletek elvégzéséhez, mint a lekérdezés, frissítés és adatkezelés. Széles körben használják különböző típusú alkalmazásokban, és kompatibilis különböző operációs rendszerekkel.

Az SQL nyelvet az informatikai ismeretek érettségire való felkészülés során ismertem meg, és gyorsan tanulhatósága, egyszerűsége miatt rövid időn belül meg tudtam tanulni. Adatbázisok kezeléséhez az XAMPP control panelben, PHP Myadmin felületen szeretek dolgozni, és szívesen használom az SQL-t is.

## 3.1.4 Blender

A Blender egy sokoldalú és erőteljes 3D modellező szoftver, amelyet művészek, tervezők és animátorok használnak lenyűgöző vizuális effektek, animációk és grafikák létrehozására. Ingyenes és nyílt forráskódú, így széles körű felhasználók számára elérhető különböző szintű képességekkel.

Mivel a Blender modellek viszonylag egyszerűen megjeleníthetőek Unity-ben, ezért a projektemben megjelenő saját modelljeim elkészítéséhez használtam a programot. Bár a program tanulása sok időt vett igénybe, sikerült egy-két modellt kivitelezni.



16. ábra Kép egy saját modell készítéséről Blenderben

## 3.1.5 Unity Assets Store

A Unity Assets Store egy online piactér, ahol fejlesztők tudnak vásárolni és eladni eszközöket és erőforrásokat (továbbiakban Asset) Unity játékfejlesztéshez. Ide tartoznak 3D modellek, textúrák, hangfájlok, animációk, scriptek és még sok más. A fejlesztők böngészhetnek és vásárolhatnak eszközöket, hogy felgyorsítsák játékfejlesztésüket és javítsák projektjeik minőségét.

Az Assets Storeban nagyon sok ingyenes Asset található, én is ezekből válogattam a projektem készítéséhez.

## 3.2 Verziókövetés

A verziókövetés egy olyan folyamat, amely során egy adott szoftverfejlesztési projekt minden egyes változatát nyomon követjük és dokumentáljuk. Ennek célja, hogy könnyen vissza lehessen térni egy korábbi változatra, ha szükség van rá, valamint hogy nyomon lehessen követni a fejlesztési folyamatot. A verziókövetésnek számos előnye van, többek között a jobb eszközhasználat, a könnyebb hibajavítás, a jobb együttműködés és a nagyobb átláthatóság a projekt fejlesztése során. Éppen ezért a verziókövetés egy fontos és nélkülözhetetlen része a modern szoftverfejlesztési folyamatnak.

A verziókövetést középiskola 11.évfolyamban ismertem meg, és azóta használom a nagyobb projektek fejlesztése során.

Eddigi és jelenlegi projektemben is az egyik legnépszerűbb verziókövetési rendszert használtam, a Git-et.

## 3.2.1 Git

Git egy ingyenes és nyílt forráskódú verziókezelő rendszer. Lehetővé teszi több fejlesztő együttműködését egy projekten és nyomon követi a kódbázison végzett változtatásokat. A Git segít különböző verziók kezelésében, konfliktusok feloldásában és visszatérésben korábbi verziókhoz szükség esetén. Széles körben használják a szoftverfejlesztés iparában, és az verziókezelésre szánt standard eszközként tekintenek rá.

## 3.2.2 GitHub

A GitHub egy webes platform, amely lehetővé teszi fejlesztőknek, hogy együttműködjenek projekteken, nyomon kövessék a kód változásait, kezeljék és tárolják a kód tárolókat, valamint részt vegyenek nyílt forráskódú projektekben. Széles körben használják fejlesztők és szervezetek a verziókezelés és kódkezelés céljából. A GitHub továbbá olyan funkciókat kínál, mint a probléma nyomon követése, projektkezelési eszközök és kód-ellenőrzési lehetőségek.

## 3.2.3 GitHub Desktop

A GitHub Desktop egy asztali alkalmazás, amely lehetővé teszi a felhasználók számára, hogy könnyen kezeljék GitHub tárolóikat anélkül, hogy használniuk kellene a parancssoros felületet. Vizuális felületet biztosít olyan feladatokhoz, mint például tárolók klónozása, helyi változtatások végrehajtása és változtatások feltöltése a GitHub-ra. A GitHub Desktop elérhető mind Windows, mind Mac operációs rendszerekhez, és arra tervezték, hogy egyszerűsítse a projekt együttműködését másokkal a GitHub-on.

Projektemben azért használtam ezt a megoldást, mivel egyszerűen összekapcsolható a Unity projektekkel.

## 3.3 Projektmenedzsment

A szoftverfejlesztés projektmenedzsmentje nagyon fontos az irányított és hatékony munkavégzés érdekében. A projektmenedzser felelős a projekt teljes körű irányításáért és koordinálásáért, a szükséges erőforrások biztosításáért, a határidők betartásáért és a költségek megfelelő kezeléséért. A folyamatos projektmonitoring fontos a projekt sikere és a szoftver sikeres befejezése érdekében. A projektmenedzsmentnek hatékonyan kell irányítania a projektet, hogy minden résztvevő és érdekelt fél elégedett legyen a végeredménnyel.

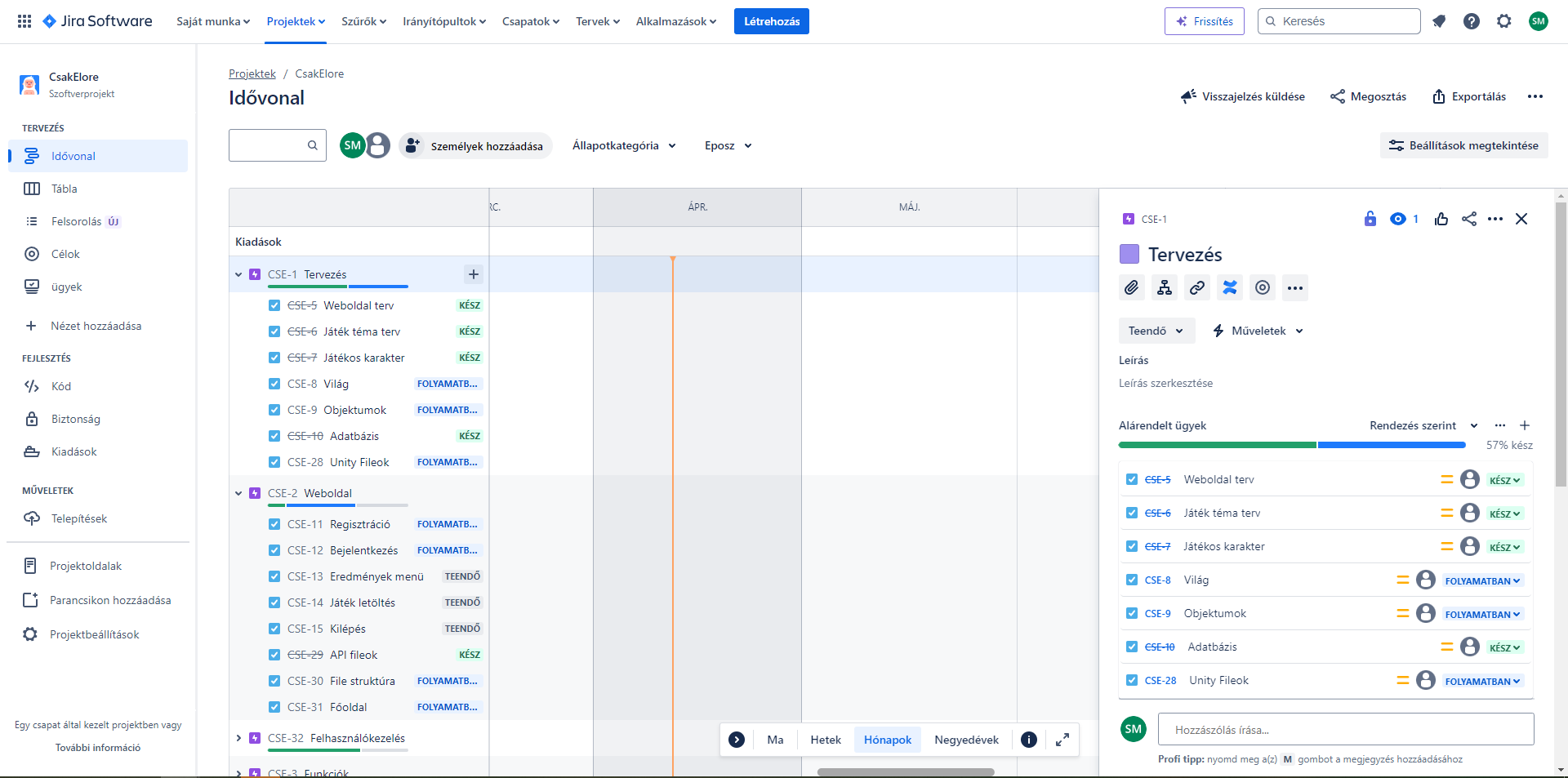
Bár én a projektemet egyedül fejlesztettem, úgy gondolom, minden fejlesztő számára fontos a projektmenedzsment, az önálló munkákhoz is.

Projektem menedzseléséhez az Atlassian Jira-t használtam.

## 3.3.1 Atlassian Jira

A Jira egy projektmenedzsment eszköz, amelyet az Atlassian fejlesztett ki, és amelyet hibakövetésre, problémakövetésre és projektmenedzsmentre használnak. Lehetővé teszi a csapatoknak, hogy feladatokat hozzanak létre, kövessék a határidőket és figyeljék az előrehaladást. Széles körben használják szoftverfejlesztő csapatokban, valamint különböző iparágak csapataiban a projektek és feladatok hatékony kezelésére.

Számomra nagyon hasznos volt az alkalmazás, mivel GitHub-bal összekötve nagyon jól működik, és így a verziókövetési változások Jira-hoz kapcsolásával, illetve a különböző csatolmányok feltöltésével tudtam végig követni a folyamatokat a projektmenedzselés közben.



17. ábra Kép a Jira idővonalról a projekt készítése közben.

## 3.4 Adatbázis

A projekthez kapcsolódó adatbázis kezeli a felhasználókat, illetve azok eredményeit.

Az adatbázist az XAMPP control panel MySql phpMyAdmin felületén készítettem.

Az adatbázisom neve *csakelore*

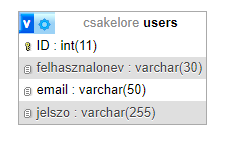
UTF-8 karakterkódolással, utf8\_hungarian\_ci karakterkészlettel.

Az adatbázisban egyetlen tábla van, mivel csak a felhasználókezelés történik adatbázis alapján, tehát a regisztráció és bejelentkezés.

Tábla:

users

* ID - int típusú; AutoIncrement, azaz minden sikeres regisztrált felhasználónak más az ID-ja
* felhasznalonev – varchar típusú, 30 karakter hosszú
* email – varchar típusú, 50 karakter hosszú
* jelszo – varchar típusú, 255 karakter hosszú, mivel a jelszó titkosítva (SHA256 HASH) kerül elmentésre



18. ábra Kép az adatbázis tábláról

## 3.5 Forráskódok

A játékban futó forráskódok (továbbiakban scriptek) programozási nyelve a C#.

A scriptek működtetik a játékban a

* Felhasználókezelést - Write.cs
* Felhasználói felületek vezérlését - MenuControl.cs
* Játékfolyamatokat - Control.cs
* Kamera vezérlést - CameraControl.cs
* Játék háttérzenéjét - MusicControl.cs
* Az eredmények fájljainak beolvasását - EredmenyRead.cs

## 3.5.1 Write.cs

Ebben a scriptben történik a felhasználó kezelése, azaz a regisztráció és bejelentkezés.

A kódban adatbázis kapcsolatot hozok létre a *csakelore* adatbázissal.

A sikeres kapcsolat létesítése után a játékban a Bejelentkezés menü felhasználói felületén keresztül a regisztrációs adatok feltöltésre kerülnek az adatbázis *users* táblájába.

A regisztrációnál ellenőrzésre kerül, hogy létezik-e már felhasználó azzal a felhasználónévvel, amelyet regisztrálni szeretne a felhasználó. Amennyiben létezik, a regisztrációs folyamat meghiúsul és hibaüzenet jelenik meg.

A bejelentkezésnél ellenőrzésre kerül hogy létezik-e regisztrált felhasználó a megadott felhasználónévvel. Amennyiben létezik, a bejelentkezési folyamat meghiúsul és hibaüzenet jelenik meg.

public void Registration(){  
Connection();  
// Már létező felhasználó  
MySqlCommand checkUserCommand = new MySqlCommand($"SELECT COUNT(\*) FROM users WHERE felhasznalonev = '{regusername.text}'", MS\_Connection);  
int userCount = Convert.ToInt32(checkUserCommand.ExecuteScalar());  
...  
 string hashedPassword = HashHelper.CalculateSHA256(regpassword.text);  
//Regisztráció  
 query = $"insert into users(felhasznalonev , email , jelszo) values('{regusername.text}','{regemail.text}','{hashedPassword}');";  
 MS\_Command = new MySqlCommand(query, MS\_Connection);  
 MS\_Command.ExecuteNonQuery();  
 MS\_Connection.Close();  
...  
}

A script egyik legfontosabb része pedig az adatbázis kapcsolat kialakítása.

//Adatbázis kapcsolódás  
public void Connection()  
{  
 connectionString = "Server = localhost ; Database = csakelore ; User = root ; Password= ; Charset = utf8 ;";  
 MS\_Connection = new MySqlConnection(connectionString);  
 MS\_Connection.Open();  
}

A scriptben található a HashHelper osztály amelyhez a CalculateSHA256 statikus string változó tartozik, amely a jelszó titkosításáért felel.

Továbbá a következő függvények vesznek részt a felhasználókezelésben:

* Start() - a bejelentkezett felhasználó felhasználónevének megjelenítése a Főmenü menüben (bejelentkezés után)
* Registration() - regisztrációs folyamatok, validáció
* Login() - bejelentkezési folyamatok, validáció
* GetLoggedInUsername() - bejelentkezett felhasználó felhasználónevének lekérdezése a felhasználói felületen megadott felhasználónév mező értéke alapján
* LoadFomenu() - Főmenü menü betöltése

## 3.5.2 MenuControl.cs

Ebben a scriptben történik a felhasználói felületek kezelése, a jelenetek (scene), menük betöltése. Összesen 12 scene betöltésének függvénye található, amelyből 5 a pályáké.

//Bejelentkezés menü betöltése (belépés gomb)  
public string Bejelentkezes;  
public void LoadBejelentkezes()  
{  
SceneManager.LoadScene(Bejelentkezes);  
}

Az előbbi forráskódban látható, hogyan néz ki egy scene betöltése a menüknél, ugyanígy a többi, az adott scene nevének megfelelően.

//Első pálya betöltése  
public string Palya1;  
public void LoadPalya1()  
{  
 SceneManager.LoadScene(Palya1);  
 Control.selectedLevel = "Palya1";  
}

Ebben a forráskódban pedig a pályák betöltése látható. A menük scenejének betöltésében annyival különbözik, hogy a Control.cs selectedLevel változójának adja át a “Palya[pálya száma]” értéket, későbbi funkciókhoz.

## 3.5.3 Control.cs

A játék mondhatni fő scriptje a Control.cs, amelyben a játékállapot változása, játékos vezérlése, visszaszámláló illetve idő számláló kezelése, animációk lejátszása, szöveg objektumok megjelenítésének vezérlése, illetve az eredmények mentése történik.

A script függvényei:

* Start() - a játékban a pályák indulásának folyamatai, például a karakter sebessége, kurzortulajdonságok
* Update() - a játék fő folyamatai itt történnek, azaz a 3 másodperces visszaszámlálás a pálya indulásakor, majd az idő számlálása, a játékos irányítása, és a szüneteltetés.
* OnControllerColliderHit() - a karakter ütközéseinek kezelése adott objektumokkal, például Obstacle (akadály) taggel ellátott objektum esetén a játékos veszít, és meghívódik a gameOver() függvény.
* Finish() - az OnControllerColliderHit() hívja meg a Finish (cél) taggel ellátott objektumal érintkezés esetén. A függvény legfontosabb részei az idő megállítása majd elmentése.
* SaveToLocalFile() - szintén az egyik legfontosabb függvény, mivel itt történik az idő és dátum elmentése a pálya teljesítésekor. Az eredmények a játék perzisztens mappájába (Appdata/LocalLow/CsakElore), *[bejelentkezett felhasználó neve]\_[pálya neve]\_savedData.txt* fájlnéven kerül mentésre. Például: Teszt1\_Pálya1\_savedData.txt

A fájlba a következő eredmények kerülnek mentésre: *idő;dátum*

*idő* formátuma: 00:00:000 (perc:másodperc:ezredmásodperc)

*dátum* formátuma: YYYY:MM:DD (év:hónap:nap)

Például: 01:10:543;2024:04:20

* gameOver() - a játékban az Obstacle taggel ellátott objektumok ütközése során illetve a talajról leesés (y koordináta -100) során a játékos veszít, így megjelenik a “Vesztettél felirat”, “Újrakezdés” és “Kilépés” gomb.
* PauseGame() - a felhasználó a játékfolyamatot bármikor megállíthatja az ESC billentyű megnyomásával. Ilyenkor megjelenik a “Beállítások” menü, megáll az idő, és a karakter mozgása.
* ResumeGame() - ha a felhasználó ismét megnyomja az ESC billentyűt, a játékfolyamatban folytatódik az idő számolása, a karakter irányítása (megtartva pozícióját).
* RestartGame() - vesztésnél az “Újrakezdés” gomb lenyomásakor újraindul a pálya, tehát újból visszaszámlálás, újból idő számolás, és a karakter a pálya elejéről indul.

seconds, milliseconds);  
   
 isGrounded = Physics.CheckSphere(transform.position, .1f, groundLayer);  
 if (isGrounded && velocity.y < 0)  
 {  
 velocity.y = -1;  
}   
 //animator változók - animációkhoz, sebességhez; irányítás  
...

## 3.5.4 MusicControl.cs

Ebben a scriptben a játék háttérzenéjének vezérlése történik. A játékban 2 hangfájl játszódik le egymás után. Mivel nem egy hangfájl van a játékban, ezért tömbben tárolom őket. Úgy oldottam meg, hogy ha több (és esetleg később még több) zenét szeretnék lejátszani, akkor azok, mint egy lejátszási listában legyenek, így egy zene lejátszása után ne játszódjon le újra ugyanaz a zene, hanem csak miután véget ért a “lista”. A játék indulásakor véletlenszerűen indul a listából egy zene, és amíg véget nem ér, a játékban mindenhol hallatszik, ha pedig véget ér, egy másik szólal meg.

void Start()  
{ audioSource = GetComponent<AudioSource>();  
 beenPlayed = new bool[Songs.Length];  
 if (!audioSource.isPlaying)  
 {  
 ChangeSong(Random.Range(0, Songs.Length));  
 }  
}   
void Update()  
{ audioSource.volume = Volume;  
 if (audioSource.isPlaying)  
 {  
 trackTimer += 1 \* Time.deltaTime;  
 }  
 if (!audioSource.isPlaying || trackTimer >= audioSource.clip.length)  
 {  
 ChangeSong(Random.Range(0, Songs.Length));  
 }  
 ...  
}  
public void ChangeSong(int songPicked)  
{  
 if (!beenPlayed[songPicked])  
 {  
 ...  
 }  
 else  
 {audioSource.Stop();}}

* Start() - hang forrás komponens inicializálása, már lejátszott hangfájl bool tömbváltozóhoz adása
* Update() - hangerő beállítás, új zene lejátszása, már lejátszott hang ellenőrzése
* ChangeSong() - már lejátszott zene esetén megáll a zene, másik kezdődik helyette

Ezenkívül:

* Awake() - mivel a játék első (Belepes) scene-jében indul a zene, így alap esetben scene váltáskor megszünne a zene. Az Awake()-ben megoldottam, hogy új scene betöltésénél folytatódjon a hang, a DontDestroyOnLoad paraméterrel.

## 3.5.5 EredmenyRead.cs

Ez a script az “Eredmények” menüben használatos a pályák teljesítése után fájlba mentett adatok beolvasására szolgál. A folyamatban fontos hogy az adott eredmény a megfelelő helyen jelenjen meg a felületen. Mivel a pályák különbözőek, így a kupákat az idő alapján pályánként eltérően kell beosztani, azaz a megfelelő text objektumhoz.

A fájl elérési útvonala megegyezik a mentési útvonallal.

A beolvasás után történik a kupa meghatározása illetve a kupa nevéhez megfelelően a kupa képe, a beolvasott idő alapján, majd az eredmények ezután jelennek meg a felületen.

A script függvényei:

* Start() - bejelentkezett felhasználóneve felhasználónevének lekérése PlayerPrefs-ből, elérési útvonal meghatározása, fájl létezésének ellenőrzése, adatok feldolgozása (Split()), kupa meghatározása, adatok megfelelő text objektumhoz rendelése, adott kupához megfelelő kép hozzárendelés
* KupaHatarozas() - idők meghatározása adott kupa szerzéséhez, minden pályánál külön

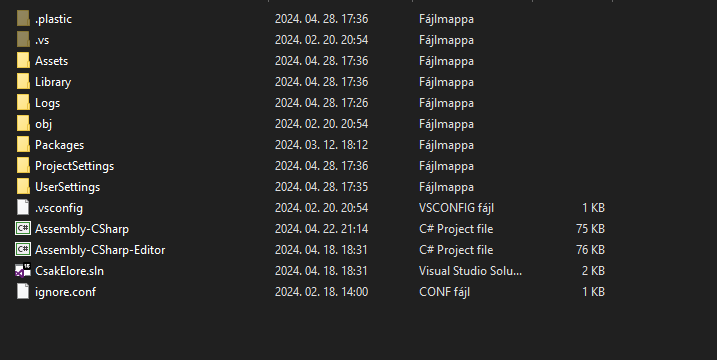
void Start()  
{  
 string loggedInUserName = PlayerPrefs.GetString("LoggedInUsername");  
 string directoryPath = Application.persistentDataPath;   
 for (int i = 1; i <= 5; i++)  
 { string fileName = $"{loggedInUserName}\_Palya{i}\_savedData.txt";  
   
 string filePath = Path.Combine(directoryPath, fileName);  
   
 // Fájl létezése  
 if (File.Exists(filePath))  
 {  
 // Fájl beolvasása  
 string data = File.ReadAllText(filePath);  
 // Adatok feldolgozása  
 ...   
 // Kupa meghatározása  
 string kupa = KupaHatarozas(elapsedTimeString, i);  
   
 // Adatok megfelelő text-hez rendelése  
 ...  
  
 // Képek megjelenítése a megfelelő kupákhoz  
 switch (kupa)  
 {  
 ... }  
 }  
 else  
 {  
 Debug.Log($"Nincs elérhető eredmény a(z) {fileName} fájlhoz.");  
}}}  
  
string KupaHatarozas(string ido, int palyaSzam)  
{  
 string[] idoElemek = ido.Split(':');  
 int perc = int.Parse(idoElemek[0]);  
 int masodperc = int.Parse(idoElemek[1]);   
 switch (palyaSzam)  
 {  
 ...   
 }

## 3.6 Fájlstruktúra

A játék fejlesztési mappájának neve *CsakElore*, ebben található, a (rejtett) *.git*, *.gitignore* és a *README* fájl, illetve ezen belül szintén egy *CsakElore* mappa, amely a Unity Projektet tartalmazza.

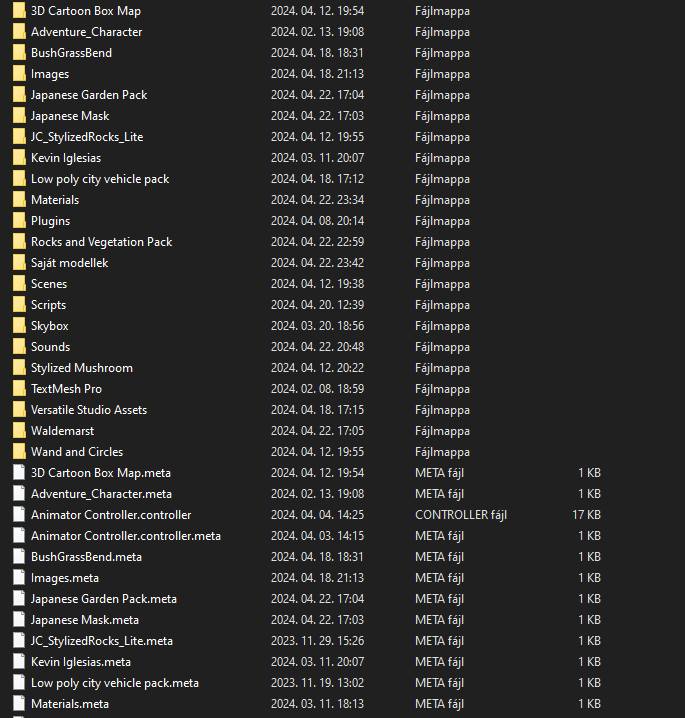


19. ábra Kép a CsakElore mappáról



20. ábra Kép a projekt mappájáról

A projekt mappájában a legfontosabb mappa az *Assets*, ez tartalmazza a Unity Assets Storeból beszerzett assetek mappáit, illetve képek, materialok, pluginek, saját modellek, scenek, scriptek, hangok és egyéb a projektben használt dolgok mappái.



21. ábra Kép az Assets mappa tartalmáról.

# 4. Összegzés

Zárásként következzen a projekt összegzése.

## 4.1 A terv és az elkészült projekt különbségei

A tervezett és az elkészült játék között több különség is született, ezeket kivitelezési problémák, egyéni véleményváltozás és a beadási határidő közeledése is okozta.

A terv szerint először is a projekthez weboldal is tartozott volna, ám mivel ez nem volt követelmény a záródolgozathoz, és a weboldal funkciói teljesen megegyeztek volna a játékban alapból szereplő funkciókkal (felhasználó- és eredménykezelés), elvetettem az ötletet.

A játékbeli eredmények mentése, lekérdezése és kezelése adatbázisban történt volna, ennek meghiúsulását kivitelezési problémák okozták, véleményem szerint ennek teljesítése érdekében bővebb tudásra van szükségem, és bár sok órát szántam a kivitelezésre, végül megváltoztattam az eredmények eltárolási illetve megjelenítési módját, jelenleg ezen adatok fájlbaírva és onnan beolvasva érhetőek el. Annak ellenére, hogy ez a megoldás remekül működik (és számos népszerű alkalmazás is ezt a módszert használja), én jobban preferálom az adatbázisban történő adatkezelést, elérhetőség, kompatibilitás, a kezelés egyszerűsége miatt. Viszont mivel a weboldal ötlete sem valósult meg, emiatt jobb megoldásnak bizonyult projektemben a fájlba mentés.  
Így tehát a tervezett három adatbázis tábla helyett csak egy táblára van szükség.

A játékfolyamatban is történtek változások, például a külön sprintelési (futás) lehetőség a kezdetekben nem volt cél, de van hogy egyes pályák kialakítása megköveteli a különböző sebességgel haladást, ezért adtam hozzá a sprintelés lehetőségét a játékban. Eredetileg két féle kamera opció lett volna, a külső nézet (Third Person) mellett még belső nézetes (First Person) is lett volna a játék, viszont úgy véltem a fejlesztés során, hogy ez inkább zavaró lenne játék közben, mint hasznos, főként a pályák kialakítása miatt.

A tervekben szerepel, hogy a játékban zenék és hangeffektek lesznek, amelyeknek hangerejét lehet csökkenteni és növelni a “Beállítások” menüben. A fejlesztés során számtalanszor teszteltem a játékfolyamatot illetve a pályákat és úgy tapasztaltam, hogy nem ront, hanem épp hogy javít a játékon, ha nem zene, hanem - a játékhoz is jobban passzoló - valóságbeli “háttérzajok” fognak szólni a játék alatt. Így történt, hogy kettő hangfájl hangzik el a játékban, amelyekben madárcsicsergés és a szél által keltett hangok hallhatóak, amelyek viszonylag alacsony hangerőn játszódnak le, így nem gondoltam szükségesnek a játék alatt megváltoztatni ezek hangerejét. A hangeffektek is kivitelezési problémák miatt kerültek eltávolításra a játékból, volt hogy az egész játékot használhatatlanná tették (zenék megállása, a hangeffektek pontatlan lejátszása, de a legnagyobb probléma a karakter mozgásának akadályozása volt), számomra ismeretlen okokból.

Egy-két apróbb változtatás érte még a tervezett menüket, gombokat, a szövegeket, elrendezéseket.

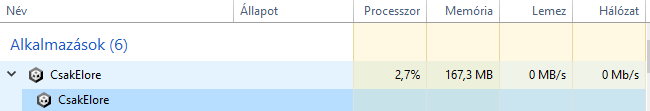
## 4.2 Személyes tapasztalatok és problémák

A fejlesztés során a legtöbb problémát az adatbázisban történő eredmény kezelés, az animációk tökéletes lejátszása, és a hangeffektek beállítása jelentette.  
Az animációk jelenleg sem tökéletesen játszódnak le, ennek optimalizálása közismerten egy nehéz feladat. Ezeknek a javítására napokat szántam, sajnos eredménytelenül, de ezek a problémák, illetve az egész fejlesztési menetre elmondható, hogy remek tapasztalatszerzés számomra, a projekt elkészülése előtt sokkal kisebb elvárásaim voltak a projektel kapcsolatban, így sok örömöt leltem a sikeres kivitelezések során. Rájöttem hogy rengeteget számít egy ilyen projektnél az előre tervezés, a folyamatos projektmenedzsment, a határidők betartása, illetve az, hogy ha sikerül időre elkészülni, mennyivel jobban sikerülhet a projekt. Úgy gondolom nem véletlen, hogy a népszerű játékok megjelenése általában hónapokat, éveket késnek, és fontos kiemelni hogy azokat a játékokat tapasztalt programozói, fejlesztői és tesztelői csapatok készítik, különféle anyagi és erőforrási támogatásokkal. Összességében elmondhatom, hogy bár voltak nehézségek, élveztem a fejlesztés menetét, a problémák megoldását és a kihívások leküzdését, illetve ezek miatt elképzelhető hogy a szoftverfejlesztés világában a videójátékok készítésének tanulására több energiát fogok szánni a jövőben.

## 4.3 Tesztelés

Egyes teszteseteket a *Felhasználói dokumentáció 2.2.1* részében, azaz a 6.-14. ábrákon láthatóak, további teljes tesztelési dokumentáció a GitHub repository Dokumentumok mappájában érhető el.

A játékot továbbá konzulensem, ismerősök, családtagok és osztálytársaim próbálták ki, jó véleménnyel voltak mind a játékélményről, a látványról, és a funkciókról egyaránt.  
A játék többszörös futtatása, tesztelése közben akadásokat, drasztikusan zavaró tényezőket nem észleltem.



22. ábra A program futása közbeni erőforrás használat saját számítógépemen.

A *22. Ábrán* látható értékek mondhatóak átlagos erőforrás használtanak. A processzor használat legalább 2%, legfeljebb 9% volt, a memória használat legalább 150 MB, legfeljebb 700 MB, lemez és hálózat használat 0 MB/s volt.

## 4.4 Fejlesztési lehetőségek

Az elsőszámú továbbfejlesztési feladatoknak mindenképp az adatbázisban történő eredménykezelést, az animációs problémák javítását és a hangeffektek hozzáadását mondanám. A játékélmény javítása érdekében pedig hasznos lenne a pályák illetve pályaválaszték bővítése, részletes optimalizálások, grafikai elemek kidolgozása, illetve élnék “határ a csillagos ég” kifejezéssel. Nyílvánvalóan egy nagyobb kaliberű fejlesztésnél már elengedhetetlenek a finanszírozási, és megfelelő felszerelés minőségi tényezők.

# 5. Források, mellékletek

## 5.1 Források

Luvseatshawty - How To Add Music Playlist To Your Unity Game

<https://www.youtube.com/watch?v=_FlgWmTITcI&t=372s>

Ekkor: 2024.04.21.

Rehope Games - How to Add MUSIC and SOUND EFFECTS to a Game in Unity | Unity 2D Platformer Tutorial #16

<https://www.youtube.com/watch?v=N8whM1GjH4w>

Ekkor: 2024.04.21.

Omogonix - Basic Third Person Character Controller in Unity - Unity C# Tutorial 2022

<https://www.youtube.com/watch?v=cEqjkubspGo&t=360s>

Ekkor: 2024.02.22.

Rehope Games - Make a TIMER & COUNTDOWN in 5 Mins | Unity Tutorial for Beginners

<https://www.youtube.com/watch?v=POq1i8FyRyQ&t=168s>

Ekkor: 2024.04.15.

Jamie Dunbar - Blender 4.0 - Texture Painting quick start guide

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

Ekkor: 2024.03.10.

Maksim Bugrimov – Adventure Character

<https://assetstore.unity.com/packages/3d/characters/humanoids/humans/adventure-character-201384>

Ekkor: 2024.02.10.

Kevin Iglesias – Basic Motions FREE

<https://assetstore.unity.com/packages/3d/animations/basic-motions-free-154271>

Ekkor: 2024.02.10.

WAND AND CIRCLES - Polygon City Pack - Environment and Interior [Free]

<https://assetstore.unity.com/packages/3d/polygon-city-pack-environment-and-interior-free-101685>

Ekkor: 2024.04.12.

JustCreate – Stylized Low Poly Rocks

<https://assetstore.unity.com/packages/3d/environments/landscapes/stylized-low-poly-rocks-271334>

Ekkor: 2024.04.12.

Layer Lab – 3D Props – Cartoon Box Map

<https://assetstore.unity.com/packages/3d/environments/dungeons/3d-props-cartoon-box-map-50743>

Ekkor: 2024.04.12.

Pixel Games – Glowing Stylized Mushroom

<https://assetstore.unity.com/packages/3d/vegetation/plants/glowing-stylized-mushroom-56048>

Ekkor: 2024.04.12.

Polygonal Assets - Low poly city vehicle pack

<https://assetstore.unity.com/packages/3d/vehicles/land/low-poly-city-vehicle-pack-217344>

Ekkor: 2024.04.21.

GapperGames Studios – Rocks and Vegetation Pack URP

<https://assetstore.unity.com/packages/3d/environments/rocks-and-vegetation-pack-urp-243519>

Ekkor: 2024.04.21.

Forge Creative – Japanese Zen Garden Pack

<https://assetstore.unity.com/packages/3d/props/japanese-zen-garden-pack-69167>

Ekkor: 2024.04.22.

Hideout Studio – Free Japanese Mask

<https://assetstore.unity.com/packages/3d/props/free-japanese-mask-66432>

Ekkor: 2024.04.22.

Waldemarst – Japanese Garden Pack

<https://assetstore.unity.com/packages/3d/vegetation/trees/japanese-garden-pack-179492>

Ekkor: 2024.04.22.

NznCG - Nissan GTR R35 nismo Special Edition 2022 Free 3D mode

<https://www.cgtrader.com/free-3d-models/car/sport-car/nissan-gtr-r35-nismo-special-edition-2022>Ekkor: 2024.04.23.

dixigunnpler - Berserk - Dragon Slayer Sword - Asset Fan Model Art Free low-poly 3D model

<https://www.cgtrader.com/free-3d-models/military/gun/berserk-dragon-slayer-sword-asset-fan-model-art>

Ekkor: 2024.04.23.

6lucius - RIFLE | AK-47 Weapon Model (CS2)

<https://sketchfab.com/3d-models/rifle-ak-47-weapon-model-cs2-6b2244ba66274c71abdd194d0b04f731>

Ekkor: 2024.04.23.

AhriFox950 – Visual Studio Logo

<https://www.thingiverse.com/thing:934102>

Ekkor: 2024.04.23.

Snailpot MayChan - Japanese Oni Mask

<https://sketchfab.com/3d-models/japanese-oni-mask-9df68b42f2044ca2866fe99217ba343b>

Ekkor: 2024.04.22.

Bulat.Shakirov - Hannya Mask

<https://sketchfab.com/3d-models/hannya-mask-96340255c7e94659bb97e8736f5907e2>

Ekkor: 2024.04.22.

gkari – Katana Onimaru Kunitsuna Valorant

<https://sketchfab.com/3d-models/katana-onimaru-kunitsuna-valorant-73ae18bf62f249f9ba6a3824201ce8b0#download>

Ekkor: 2024.04.22.

## 5.2 Mellékletek

Saját modell – 206osajtaja

<https://github.com/K3WiiiN/CsakElore/blob/main/CsakElore/Assets/Saj%C3%A1t%20modellek/206osajtaja.fbx>

Saját modell – boostpad

<https://github.com/K3WiiiN/CsakElore/blob/main/CsakElore/Assets/Saj%C3%A1t%20modellek/boostpad.blend>

Saját modell - tablo

<https://github.com/K3WiiiN/CsakElore/tree/main/CsakElore/Assets/Saj%C3%A1t%20modellek/tablo>

Saját modell – c#

<https://github.com/K3WiiiN/CsakElore/blob/main/CsakElore/Assets/Saj%C3%A1t%20modellek/c%23.fbx>

Háttér hangok

<https://github.com/K3WiiiN/CsakElore/tree/main/CsakElore/Assets/Sounds/Music>

Plugin – MySql.Data.dll

<https://github.com/K3WiiiN/CsakElore/blob/main/CsakElore/Assets/Plugins/MySql.Data.dll>

Képek  
<https://github.com/K3WiiiN/CsakElore/tree/main/CsakElore/Assets/Images>