Zsíros Gábor Géza, Szilvási Kornél

Kivonat

[Ragadja meg az olvasó figyelmét egy izgalmas kivonattal. Ez többnyire a dokumentum rövid összegzése.   
Ha készen áll a tartalom hozzáadására, kattintson ide, és kezdjen gépelni.]

[Dokumentum címe]

[Dokumentum alcíme]

Tartalom

[Bevezetés 4](#_Toc192663323)

[1. Felhasznált technológiák 6](#_Toc192663324)

[1.1. HTML 6](#_Toc192663325)

[1.2. CSS 6](#_Toc192663326)

[1.3. JavaSript 6](#_Toc192663327)

[1.4. PHP 6](#_Toc192663328)

[1.5. Visual Studio Code 7](#_Toc192663329)

[1.6. XAMPP 7](#_Toc192663330)

[1.7. MySQL 7](#_Toc192663331)

[1.8. GitHub 8](#_Toc192663332)

[1.9. Dbdiagram 8](#_Toc192663333)

[1.10. Unity 8](#_Toc192663334)

[1.11. Aseprite 8](#_Toc192663335)

[2. Felhasználói dokumentáció 9](#_Toc192663336)

[1.1. main.php 9](#_Toc192663337)

[1.2. download.php 10](#_Toc192663338)

[1.3. contact.php 10](#_Toc192663339)

[1.4. register.php 10](#_Toc192663340)

[1.5. login.php 11](#_Toc192663341)

[1.6. player.php 11](#_Toc192663342)

[1.7. settings.php 11](#_Toc192663343)

[1.8. header.php 12](#_Toc192663344)

[1.9. footer.php 12](#_Toc192663345)

[1.10. A játék bemutatása 12](#_Toc192663346)

[3. Fejlesztői dokumentáció 13](#_Toc192663347)

[2.1. Felhasznált technológiák 13](#_Toc192663348)

[2.1.1. VSC – miért jó? milyen extension-t használsz? pl emmet, miért jó, hogyan gyorsítja meg a munkát 13](#_Toc192663349)

[2.1.2. HTML – mit jelent, hányféle html jelölőt használtál 13](#_Toc192663350)

[2.1.3. CSS - mire jó, hogyan csoportosítottad a stílusokat 13](#_Toc192663351)

[2.2. Adatbázis 13](#_Toc192663352)

[2.2.1. Adatbázis célja 13](#_Toc192663353)

[2.2.2. Tervezési lépések 14](#_Toc192663354)

[2.2.3. Egyedtípusok/egyedek meghatározása 14](#_Toc192663355)

[2.2.4. Kapcsolatok meghatározása 20](#_Toc192663356)

[2.2.5. Táblák 20](#_Toc192663357)

[2.2.5.1. Players 20](#_Toc192663358)

[2.2.5.2. PlayerMovement 20](#_Toc192663359)

[2.2.5.3. GameManager 24](#_Toc192663360)

[2.2.5.4. LobbyCamera 25](#_Toc192663361)

[2.2.5.5. CameraFollow 25](#_Toc192663362)

[2.2.5.6. MainMenu 26](#_Toc192663363)

[2.2.5.7. HealthBar 26](#_Toc192663364)

[2.2.5.8. CoinManager 27](#_Toc192663365)

[2.2.5.9. PlayerLobbyMovement 28](#_Toc192663366)

[2.2.5.10. SettingsMenu 29](#_Toc192663367)

[2.2.5.11. PlayerAttack 30](#_Toc192663368)

[2.2.5.12. NextLevel 31](#_Toc192663369)

[2.2.5.13. SettingsMenuManager 32](#_Toc192663370)

[2.2.6. Minta adatok 32](#_Toc192663371)

[2.2.6.1. Players 32](#_Toc192663372)

[2.2.7. Adatbázis diagram 32](#_Toc192663373)

[2.2.8. Adatbázis továbbfejlesztési lehetőségek 33](#_Toc192663374)

[2.3. A programozás folyamata 33](#_Toc192663375)

[2.3.1. tervezés folyamata Figma 33](#_Toc192663376)

[2.3.2. pl regisztrációs oldal programozása 33](#_Toc192663377)

[2.3.3. olyan kódrészletek magyarázatokkal, amit kiemelnél, amire büszkék vagytok 33](#_Toc192663378)

[4. Felmerül akadályok 33](#_Toc192663379)

[Összefoglalás 34](#_Toc192663380)

[5. Források 35](#_Toc192663381)

[5.1. Weboldalak 35](#_Toc192663382)

[5.2. Képek 35](#_Toc192663383)

# Bevezetés

A projekt célja egy platformer típusú játék fejlesztése volt, amelyhez modern fejlesztési technológiákat alkalmaztunk. A játékot a Unity játékmotor segítségével valósítottuk meg, amely egy hatékony és rugalmas környezetet biztosított a gyors és hatékony fejlesztésnek. A projekt másik fontos eleme egy webes felület létrehozása volt, amely lehetővé teszi a játékosok számára, hogy regisztráljanak, bejelentkezzenek, és hozzáférjenek a játékkal kapcsolatos információkhoz. A webes felület megvalósítása során PHP, SQL, valamint HTML technológiákat használtunk.

A projekt két fő részre osztható: a játékfejlesztésre és a webes környezet létrehozására. A két terület külön, de egymással szoros együttműködésben készült, a csapattagok munkamegosztása alapján.

A platformer játékot Unity-ben fejlesztettük, amely lehetőséget nyújtott a grafikus környezet, a játékmechanikák és az interaktív elemek teljes körű kidolgozására. A játékfejlesztési folyamat során az alábbi lépésekre helyeztük a hangsúlyt.

Játékmechanikákhoz tartozik a karakter mozgása, ugrása, interakciók a környezettel, valamint a játékos és az ellenfelek közötti interakciók kidolgozása. Ezeket a funkciókat Unity C# nyelven írt szkriptek segítségével valósítottuk meg.

A játékszintek vizuális megjelenését a Unity Editor segítségével terveztük meg, figyelembe véve a játékélmény szempontjából fontos kihívásokat és navigációs elemeket. Grafikai elemekhez is besorolható a játékhoz szükséges grafikai elemek, például hátterek, karakterek és interaktív objektumok elkészítésére ingyenes és egyedi tartalmakat használtunk fel, biztosítva a vizuális élményt.

Tesztelésbe beletartozik az egyes funkciók integrálása, amiután alapos tesztelést végeztünk, hogy a játékmenet zökkenőmentes és bugmentes legyen.

A játék egyes funkcióit alapos tesztelésnek vetettük alá: A mozgásmechanikák és az interakciók tesztelése annak érdekében, hogy a játékélmény gördülékeny legyen. A teljesítmény optimalizálása, hogy a játék zökkenőmentesen működjön. Illetve a hibák és bugok kijavítása a végleges verzióban.

A játékfejlesztést Zsíros Gábor Géza végezte, aki a Unity által kínált eszközöket maximálisan kihasználva, önállóan valósította meg a játék minden elemét.

A játékhoz kapcsolódó webes felület fejlesztése szintén kiemelt szerepet kapott a projekt során. A webes környezet célja az volt, hogy a játékosok regisztrálhassanak, és a játékkal kapcsolatos adatokat biztonságosan kezeljük.

A webes fejlesztés főbb lépései voltak a frontend fejlesztése. A weboldal megjelenését HTML és CSS segítségével valósítottuk meg. A felhasználóbarát kialakítás érdekében reszponzív dizájnt használtunk, amely különböző eszközökön is megfelelő megjelenést biztosít.

A backend fejlesztést a weboldal dinamikus funkcióit PHP nyelven készítettük el. Ez a technológia tette lehetővé az adatbevitel feldolgozását, például a regisztrációs és bejelentkezési funkciók működését.

Az adatokat egy MySQL alapú adatbázisban tároltuk. Az adatbázis struktúráját úgy alakítottuk ki, hogy a játékosok adatai (például felhasználónév, e-mail cím, jelszó) biztonságosan és hatékonyan kezelhetők legyenek.

Tesztelés során a weboldal minden funkcióját alaposan áttekintettük, hogy biztosítsuk az adatok helyes kezelését és a felület hibamentes működését. A webes fejlesztési munkákat Szilvási Kornél végezte, aki a projekt weboldalának és adatbázisának teljes kialakításáért felelt.

A csapat két fő tagból állt. A játék fejlesztése, beleértve a játékmechanikák kidolgozását, a grafikai elemek integrációját, valamint a játék tesztelését Zsíros Gábor Géza végezte míg a webes felület teljes fejlesztése, beleértve a frontend és backend kialakítását, az adatbázis struktúrájának megtervezését, valamint a weboldal tesztelését Szilvási Kornél végezte.

# Felhasznált technológiák

## HTML

A HTML (Hypertext Markup Language) egy leírónyelv, amely a weboldalak szerkezetének és tartalmának meghatározására szolgál. Ez az alapvető technológia biztosítja a weboldalak építőköveit, lehetővé téve a szövegek, képek, linkek és egyéb elemek elrendezését. A HTML egyszerűsége és áttekinthetősége miatt könnyen elsajátítható, ugyanakkor elengedhetetlen a webfejlesztésben, hiszen minden weboldal alapját képezi.

## CSS

A CSS (Cascading Style Sheets) egy stíluslapnyelv, amely a weboldalak kinézetének és formázásának meghatározására szolgál. Segítségével a tartalom (HTML) és a megjelenés (dizájn) különválasztható, így a fejlesztők könnyen testre szabhatják az oldalak vizuális megjelenését. A CSS segítségével beállíthatóak például a színek, betűtípusok, elrendezések és animációk, amelyek hozzájárulnak az esztétikus és modern weboldalak létrehozásához.

## JavaSript

A JavaScript egy programozási nyelv, amelyet főként weboldalak és webalkalmazások interaktív funkcióinak megvalósítására használnak. Elsődlegesen a böngészőben fut, ami lehetővé teszi a dinamikus tartalomkezelést, az események kezelését és a felhasználói élmény javítását. JavaScript nélkül a weboldalak statikusak lennének, míg ezzel a nyelvvel olyan funkciók hozhatók létre, mint az űrlapellenőrzés, animációk vagy interaktív felhasználói felületek. Ennek köszönhetően a JavaScript kulcsszerepet játszik a modern webfejlesztésben.

## PHP

A PHP (Hypertext Preprocessor) egy szerveroldali szkriptnyelv, amelyet elsősorban dinamikus weboldalak és webalkalmazások fejlesztésére használnak. Lehetővé teszi a programozók számára, hogy dinamikusan generált tartalmakat hozzanak létre, interaktív funkciókat valósítsanak meg, valamint adatbázisokkal kommunikáljanak. Rugalmasságának és sokoldalúságának köszönhetően a PHP az egyik legnépszerűbb nyelv a webfejlesztésben, amely segít az interaktív és felhasználóbarát weboldalak kialakításában.

## Visual Studio Code

A Visual Studio Code (röviden VS Code) egy ingyenesen elérhető, nyílt forráskódú fejlesztői környezet, amelyet széles körben alkalmaznak különféle programozási nyelvekben írt szoftverek készítéséhez. Rendkívül rugalmas és testreszabható, számos beépített funkcióval és bővíthető kiegészítőkkel rendelkezik, amelyek megkönnyítik a kódírást, a hibakeresést és a projektmenedzsmentet. Népszerűségének egyik oka az intuitív felhasználói felület és a hatékony eszközkészlet, amelyek segítségével a fejlesztők gyorsabban és eredményesebben dolgozhatnak.

## XAMPP

A XAMPP egy komplex, nyílt forráskódú szoftvercsomag, amely lehetővé teszi a webfejlesztők számára, hogy helyi szerverkörnyezetet hozzanak létre saját számítógépükön. A neve az alábbi komponenseket foglalja magában: Cross-platform támogatás, Apache webszerver, MySQL adatbázis-kezelő, PHP programnyelv és Perl. Ezzel az eszközzel könnyedén tesztelhetők és fejleszthetők a dinamikus webalkalmazások anélkül, hogy egy távoli szerverre lenne szükség.

## MySQL

A MySQL egy széles körben használt, nyílt forráskódú relációs adatbázis-kezelő rendszer, amely a webalkalmazások egyik alapvető technológiája. Kiemelkedően gyors, megbízható és hatékony módot kínál az adatok tárolására, kezelésére és lekérdezésére. Rugalmasságának és könnyű használatának köszönhetően mind kisebb weboldalak, mind nagyobb vállalati rendszerek számára ideális választás.

## GitHub

A GitHub egy webalapú verziókezelő és együttműködési platform, amely kulcsfontosságú szerepet tölt be a modern szoftverfejlesztésben. Segítségével a fejlesztők könnyedén követhetik a kódváltoztatásokat, együtt dolgozhatnak csapatban, és kezelhetik projektjeiket. A Git verziókezelő rendszerre épülve biztosítja a kód biztonságos tárolását és kezelését, miközben hatékony eszközöket kínál a fejlesztési folyamat optimalizálására.

## Dbdiagram

A Dbdiagram egy online tervezőeszköz, amely vizuális megoldást nyújt relációs adatbázisok sémájának megtervezésére és dokumentálására. Egyszerű, mégis hatékony eszközt biztosít az adatmodellek áttekinthető ábrázolására, megkönnyítve a fejlesztők és adatbázis-tervezők számára a struktúrák kidolgozását. Intuitív felülete lehetővé teszi az adatbázisok gyors és hatékony megtervezését, így felgyorsítva a fejlesztési folyamatot.

## Unity

Az Unity egy fejlett játékmotor, amelyet videojátékok és interaktív alkalmazások fejlesztésére használnak. Lehetővé teszi a fejlesztők számára, hogy 2D és 3D játékokat készítsenek különböző platformokra, például PC-re, konzolokra és mobil eszközökre. Az Unity rugalmas, erőteljes eszközöket biztosít a grafika, a fizika és a szkriptek kezelésére, így népszerű választás a játékiparban.

## Aseprite

Az Aseprite egy népszerű pixel art és animációs szoftver, amelyet kifejezetten pixelgrafikák és sprite-ok készítésére terveztek. A program rendkívül intuitív felhasználói felülettel rendelkezik, amely lehetővé teszi a könnyed rajzolást, animációk készítését és szerkesztését. Aseprite támogatja a rétegeket, a különböző színmódokat és a frame alapú animációt, így ideális választás videojáték fejlesztők, grafikusok és animátorok számára. A program lehetővé teszi a képek exportálását különböző formátumokban, például PNG vagy GIF formátumban, és egyszerűsített munkafolyamatokat kínál a sprite-ok és animációk kezelésére.

# Felhasználói dokumentáció

A weboldalunk hét fő részből áll, amelyek különböző funkciókat és információkat kínálnak a felhasználónak. Először a felhasználó a fő oldalt (main.php), a letöltési oldalt (download.php), a kapcsolat oldalt (contact.php), a registert (register.php) és a login (login.php) oldalt látja. Miután létrehozta a saját fiókját, elérhetővé válik számára a profil oldal (player.php) és a beállítások oldal (settings.php).

## main.php

A főoldalon egy jól strukturált menüsáv található, amely lehetővé teszi a felhasználók számára, hogy könnyedén navigáljanak a weboldal különböző részei között. Ez a menüsáv biztosítja a gyors hozzáférést a letöltési oldalhoz, a letöltéshez és a kapcsolat tartáshoz, így minden fontos funkció egyetlen kattintással elérhető. Hogy a felhasználó visszakerüljön erre az oldalra más menü pontokról ahhoz a „Főoldal” opcióra kell nyomnia. Ez a menüsáv minden oldalon megtalálható.

Közvetlenül a menüsáv alatt kap helyet egy részletes ismertető a játékról, amely célja, hogy a látogatók egy átfogó képet kapjanak a játék világáról, történetéről és főbb mechanikáiról. A kedvcsináló szövegek mellett egy rövid bemutató videó is helyet kap, amely vizuálisan is megismerteti a látogatót a játékmenettel, így könnyebben eldöntheti, hogy érdeklődik-e a játék iránt. A videó bemutatja a legfontosabb funkciókat, a játéktér kialakítását és a karakterek mozgását, valamint betekintést enged az akciódús vagy stratégiai elemekbe, amelyeket a játék kínál.

A bemutató videó alatt egy további vizuális szekció található, ahol a látogatók különféle képernyőképeket tekinthetnek meg a játékról. Ezek a képek a játékmenet különböző aspektusait mutatják be, beleértve a környezetet, a karaktereket és a felhasználói felületet. Ezen kívül a játék fejlesztési folyamata során készült belső munkák is megjelennek, amelyek betekintést engednek a kulisszák mögé. A képek mellett rövid leírások segítik a látogatókat abban, hogy jobban megértsék, mit ábrázolnak a vizuális elemek, és hogyan kapcsolódnak a játék általános élményéhez. Ez a szekció nemcsak információt nyújt, hanem arra is ösztönzi a látogatókat, hogy mélyebben elmerüljenek a játék világában, és még inkább érdeklődjenek a kipróbálása iránt.

A főoldal célja, hogy vizuálisan vonzó és informatív módon mutassa be a játékot, miközben lehetőséget ad a felhasználóknak arra, hogy könnyedén eligazodjanak az oldalon és minden fontos funkcióhoz gyorsan hozzáférjenek.

## download.php

Amikor a felhasználó a „Letöltés” menüpontra kattint, automatikusan a letöltési oldalra kerül. Ezen az tudja a felhasználó letölteni a játékot, így offline is tud játszani. Ehhez elég az oldalon levő gombra kattintani.

## contact.php

Az oldal célja, hogy lehetővé tegye a felhasználók számára, hogy kapcsolatba lépjenek a játék készítőivel. Az oldal információkat tartalmaz az elérhetőségeiről, beleértve az email címet és telefonszámot, valamint egyéb hasznos információkat, hogy a felhasználók könnyedén felvehessék velünk a kapcsolatot. Ez hasznos, ha a felhasználónak bármilyen kérdése vagy problémája akad a játékkal, vagy a weboldallal kapcsolatban.

## register.php

Ezen az oldalon tudja a felhasználó létrehozni saját fiókját, amivel tud majd játszani. Ehhez meg kell adnia egy felhasználónevet, jelszót és email címet és el kell fogadnia a felhasználási feltételeket. Csak olyan felhasználónevet és email címet adhat meg, ami még nem létezik az adatbázisban, különben az oldal hibát fog adni. A jelszónak pedig legalább 3 karakter hosszúnak kell lennie, a felhasználási feltételeket pedig kötelező elfogadni a regisztráláshoz. Ha minden adat stimmel, akkor a „Regisztrálok” gombra kattintva létrejön a felhasználó saját fiókja. Regisztrálás után automatikusan átdobja őt a login.php oldalra. Ha van már fiókja és nem akar újat létrehozni, akkor a „Regisztrálok” gomb alatt is lehetősége van egy kattintással átmenni a login.php oldalra, ehhez a „Bejelentkezés” szóra kell kattintania.

## login.php

Ha már regisztrált a felhasználó, itt tud bejelentkezni a saját fiókjába. Ehhez meg kell adnia a felhasználónevét és a jelszavát helyesen. Ha mindkettő helyes, akkor a „Bejelentkezés” gombra kattintva sikeresen bejelentkezik a felhasználó és új lehetőségek nyílnak meg előtte a weboldalon.

## player.php

Amikor a felhasználó a „Profil” gombra kattint, az oldal automatikusan betölti és megjeleníti a saját játékon belüli adatait. Egy jól áttekinthető, kompakt táblázatban láthatja a legfontosabb információkat, például a játékosnevét, a játékban összegyűjtött érmék számát, a halálok számát és a megölt játékban a megölt ellenségek számát. A számok mellett egy odaillő ikon is megjelenik. Az oldal célja, hogy a felhasználók gyorsan és egyszerűen hozzáférjenek a saját profiljukhoz és nyomon követhessék fejlődésüket a játékban.

## settings.php

A settings.php oldal lehetőséget biztosít a felhasználó számára, hogy módosítsa a fiókjához tartozó adatokat, például a felhasználónevet és a jelszót. Az oldal először ellenőrzi, hogy a felhasználó be van-e jelentkezve. Ha a felhasználó nincs bejelentkezve, akkor átirányítja őt a bejelentkezési oldalra (login.php). Ahhoz, hogy a felhasználó módosítsa a felhasználónevét, meg kell adni a jelenlegi jelszavát. Akkor is szükség van a jelenlegi jelszóra, ha a felhasználó a jelszavát szeretné módosítani, ilyenkor az új jelszavát két helyre is be kell írnia. Ha a jelenlegi jelszó nem egyezik, vagy a felhasználónév amire módosítani akarja, már foglalt, vagy az új jelszó kevesebb mint a 3 karakter, vagy az új jelszó és annak megerősítése nem egyezik, akkor az oldal hibát fog kiírni. Ezen az oldalon a felhasználónak lehetősége van törölni a fiókját, a fiók törlése gombra kattintva feldob egy új ablakot, ahol még vissza tudja vonni ezt a döntését a felhasználó. Ha mégse teszi, akkor a fiókja törlésre kerül.

## header.php

Ez tartalmazza minden másik oldal navigációs sávját. Megtalálható rajta a Főoldal, Letöltés, Kapcsolat, Bejelentkezés és Regisztráció gomb. Ezekre kattintva elviszi a felhasználót a megfelelő oldalra. Ha a felhasználó bejelentkezik saját fiókjába, akkor a Bejelentkezés és Regisztráció gomb helyett megjelenik egy beállítások ikon, amire ha ráhúzza a kurzort, egy legördülő lista jelenik meg, amik a profil, beállítások és a Kijelentkezés gomb szerepel. Ezekre kattintva a felhasználót átkerül az adott oldalra, illetve a Kijelentkezés gomb segítségével kijelentkezik a felhasználó a fiókjából.

## footer.php

Majdnem mindegyik oldal legalján szerepel egy footer, kivéve a bejelentkezés és regisztrációs oldal alján. Ez egy sort tartalmaz „© 2025 Last Knight. Minden jog fenntartva.”.

## A játék bemutatása

A játék neve Last Knight. Ez egy 2D platformer játék, ami egy pixeles grafikával rendelkező történetet mutat be. Célja hogy az általunk irányított karakter történetét végig vigye a játékos. A játék során a felhasználó külön-külön féle akadályokkal kell megbirkóznia mint pl: puzzle rejtvényekkel vagy egyszerű titkos mechanikák keresésével.

# Fejlesztői dokumentáció

A gyökérkönyvtárban találhatók az alapvető beállítások és konfigurációs fájlok, mint például a .env, amely környezeti változókat tárol, valamint a composer.json és composer.lock, amelyek a PHP csomagkezeléshez szükségesek. Az admin mappa az adminisztrációs funkciókhoz tartozó PHP fájlokat tartalmazza, míg a backend a háttérfolyamatokat és az üzleti logikát kezeli. Ezen belül a footer és header almappák külön fájlokban tárolják a weboldal láblécét és fejlécét, ami segíti a moduláris felépítést és az újrafelhasználhatóságot. A főoldali működéshez kapcsolódó PHP fájlok közvetlenül a gyökérkönyvtárban helyezkednek el, ide tartozik például a main.php, a login.php, a register.php, valamint a settings.php, amelyek a felhasználói hitelesítést és beállításokat kezelik. A css mappa a különböző oldalakhoz tartozó stíluslapokat tárolja, mint például a main.css, loginstyle.css és registerstyle.css, biztosítva az egységes megjelenést. A js mappa a JavaScript fájlokat tartalmazza, amelyek az oldal interaktivitását segítik, például a dropdown.js a lenyíló menük kezeléséért felelős. A downloads és images mappák letölthető fájlokat és a weboldalon használt képeket tárolják. Összességében a projekt mappaszerkezete jól elkülöníti az egyes funkcionális egységeket, ami megkönnyíti a karbantartást és a továbbfejlesztést.

## Felhasznált technológiák

### VSC – miért jó? milyen extension-t használsz? pl emmet, miért jó, hogyan gyorsítja meg a munkát

### HTML – mit jelent, hányféle html jelölőt használtál

### CSS - mire jó, hogyan csoportosítottad a stílusokat

## Adatbázis

### Adatbázis célja

A játék adatbázisának célja, hogy tárolja a játékosok adatait és a játékmenettel kapcsolatos információkat. Minden adat, illetve a játékosok statisztikái itt kerülnek eltárolásra. Az adatbázis tervezésénél arra törekedtünk, hogy minimalizáljuk a redundanciát, biztosítva ezzel a gyors adatlekérdezéseket és az adatok konzisztenciáját. Mivel a játékosok személyes adatokat is regisztrálnak, fontos volt a biztonság, hogy minden szenzitív adat védett módon legyen tárolva.

### Tervezési lépések

A tervezés kezdetén meghatároztuk azokat a táblákat, amelyek szükségesek a játék működéséhez. Az alapadatok, mint a játékosok személyes adatai és a játékmenet adatai, mind szerepeltek az adatbázisban.

A tervezési folyamat elején a legnagyobb kihívás a kapcsolatok megfelelő definiálása volt. A táblák közötti összekapcsolások segítettek abban, hogy a játékosok és játékmenet adatai könnyen kezelhetők és lekérdezhetők legyenek. A tervezés folyamán fontos szerepet kapott a vizualizáció, így egy online adatbázis-tervező eszközt, a dbdiagram.io-t használtuk, hogy gyorsan és hatékonyan ábrázoljuk az adatbázis struktúráját.

A Unity játékmotorban a játék fejlesztése során különböző adatokat tárolunk a játékosról és magáról a játékról. Ezek az adatok tartalmazhatják például a játékos profilját, elért eredményeit, mentett állásait.

A weboldalon ezekhez az adatokhoz egy JSON fájlformátumon keresztül férünk hozzá. A játék az adatokat JSON formátumban exportálja, amelyet a weboldal képes feldolgozni és megjeleníteni. Ez lehetővé teszi, hogy a játékosok a weboldalon is nyomon követhessék statisztikáikat, eredményeiket anélkül, hogy közvetlenül a játékba kellene belépniük.

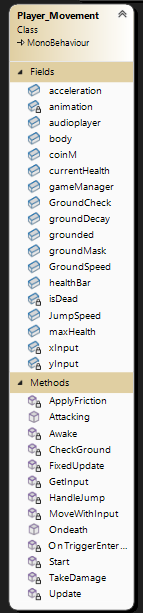
Ez a megoldás biztosítja az adatok könnyű kezelését és átjárhatóságát a játék és a webes felület között, miközben minimalizálja az adatbázis-terhelést.

### Egyedtípusok/egyedek meghatározása

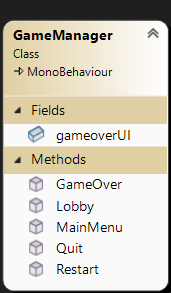
#### Players

A Players tábla tárolja a játékosok adatait, mint például a nevüket, e-mail címüket és jelszavukat és azt, hogy a felhasználó user vagy adminként van bejelentkezve. Ez a tábla biztosítja, hogy minden játékos rendelkezzen egy egyedi azonosítóval.

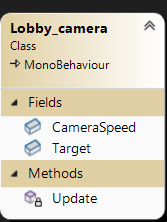
#### PlayerMovement



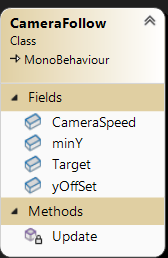
#### GameManager



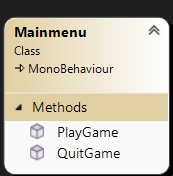
#### LobbyCamera



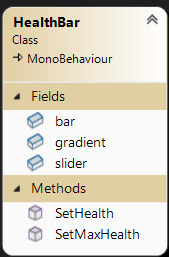
#### CameraFollow



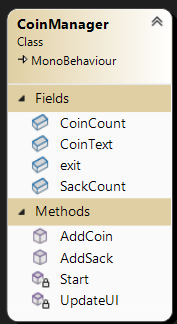
#### MainMenu



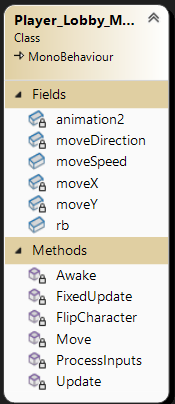
#### HealthBar



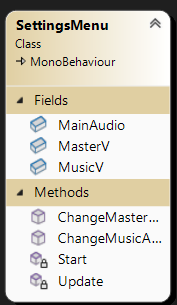
#### CoinManager



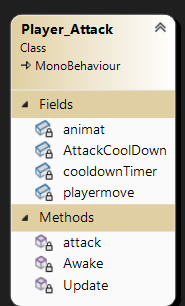
#### PlayerLobbyMovement



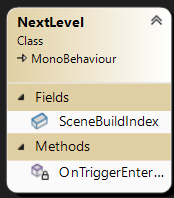
#### SettingsMenu



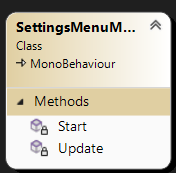
#### PlayerAttack



#### NextLevel



#### SettingsMenuManager



### Kapcsolatok meghatározása

### Táblák

### Players

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| ID | int | játékos azonosítója | elsődleges kulcs |
| Name | varchar(255) | játékos neve |  |
| Email | varchar(255) | játékos email címe |  |
| Password | varchar(255) | játékos jelszava |  |
| Rool | bool | felhasználó admin vagy user |  |

### PlayerMovement

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| acceleration | Float | a sebesség változásának sebességére utal |  |
| animation | Animator | hivatkozik a játékos egy helyben álló animációjára |  |
| audioplayer | AudioSource | Hivatkozik a játékon belüli hangokra |  |
| body | RigidBody2D | Hivatkozik a játékos karakterre |  |
| coinM | CoinManager | Hivatkozik az érme számláló függvényre |  |
| currentHealth | Int | Megszabja a játékos jelenlegi életmennyiségét |  |
| gameManager | Gamemanager | A játékban lévő interaktív elemekért felelős függvényre hivatkozik |  |
| GroundCheck | BoxCollider2D | Ellenőrzi hogy a játékos a földön van-e |  |
| GroundDecay | Float | hatások szimulálása, mint a talajerózió |  |
| grounded | Bool | igaz vagy hamis értékkel megszablya hogy a játékos ugorani tud-e |  |
| groundMask | LayerMask | tipikusan egy LayerMask változó, a talaj észlelésére használnak |  |
| GroundSpeed | Float | a játékos gyorsasága a földön |  |
| healthBar | HealthBar | hivatkozás a játékos életét tároló / kijelző sávra |  |
| isDead | Bool | igaz vagy hamis értékkel meghatározza hogy a játékos meghalt-e vagy sem |  |
| JumpSpeed | Float | az ugrás ereje |  |
| maxHealth | Int | Egy adott értéken meghatározott élet mennyiség |  |
| xInput | Float | a játékos irányításához szükséges x tengely mennyisége |  |
| yInput | Float | a játékos irányításához szükséges y tengely mennyisége |  |
| ApplyFriction | void | fokozatosan lelassítja a karakter mozgását, amikor a földön vannak |  |
| Attacking | void |  |  |
| Awake | void | Meghatározza a játékos mozgásához való animációt |  |
| CheckGround | void | Ellenőrzi hogy a játékos a földön van-e és csak akkor tudja engedélyezni az első ugrást |  |
| FixedUpdate | void | A CheckGround, a MoveWithInput és a ApplyFriction metódukos hivatkozása |  |
| GetInput | void | A mozgást határozza meg vízszintes és függőleges irányba |  |
| HandleJump | void | Ellenőrzi hogy a játékos az ugrás gombot nyomta meg és hogy az ugrások száma kevesebb-e mint 2. Amennyiben az ugrások száma már 1 akkor a második ugrás 1.3 szor erősebb. |  |
| MoveWithInput | void | Ellenőrzi hogy a vízszintes tengelyen lévő mozgás nagyobb-e mint 0 |  |
| OnDeath | void | Ellenőrzi hogy az “isDead” létezik-e és  elpusztítja a játékost és elindít egy háttér zenét jelezve hogy a játékos meghalt |  |
| OnTriggerEnter | void | Ellenőrzi hogy a játékos érintkezik bizonyos ID-val rendelkező elemekkel mint pl: “coin, sack..” és ezeket azoknak megfelelő módon kezelje őket. |  |
| Start | void | Beállítja és megjeleníti a maximális élet mennyiséget |  |
| TakerDamage | void | Levon a jelenlegi életből és ezt ki is mutatja annak megfelelően |  |
| Update | void | Hivatkozik a GetInput és a HandleJump metódusokra. Illetve Ellenőrzi hogy a jelenlegi élet mennyiség több mint 0. Ha nem akkor hivatkozik a GameOver és Ondeath metódusokra |  |

### GameManager

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| gameoverUI | Gameobject | A képernyőn való adatok megjelenítése |  |
| GameOver | void | aktiválja azt a boolen értéket hogy a játék végetért |  |
| Lobby | void | Gomb ami visszatéríti a játékost a játéklobbiba |  |
| MainMenu | void | Gomb ami visszatéríti a játékost a főmenübe |  |
| Quit | void | Gomb ami kilépteti a játékost |  |
| Restart | void | Gomb ami újra indítja a játékot |  |

### LobbyCamera

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| CameraSpeed | float | meghatározott sebesség a kamera számára |  |
| Target | Transform | meghatározza hogy az általunk megadott célpontot kövesse a kamera |  |
| Update | Void | Folyamatosan frissíti hogy létezik-e a célpont és hogy merre van |  |

### CameraFollow

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| CameraSpeed | float | meghatározott sebesség a kamera számára |  |
| minY | float | A minimális Y tengelyen lévő szint ameddig még mehet |  |
| Target | Transform | meghatározza hogy az általunk megadott célpontot kövesse a kamera |  |
| yOffSet | float | Folyamatosan frissíti hogy a kamera ne süllyedjen egy bizonyos szint alá |  |
| Update | Void | Folyamatosan frissíti hogy létezik-e a célpont és hogy merre van |  |

### MainMenu

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| PlayGame | void | Betölté a következő jelenetet, tehát elindítja a játékot |  |
| QuitGame | void | Kilép a játékból |  |

### HealthBar

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| bar | Image | Kimutatja az élet mennyiséget |  |
| gradient | Gradient | keretbe foglalja a sávot |  |
| slider | Slider | Ezen jelenik meg a bar mennyisége |  |
| SetHealth | void | beállítja hogy a jelenlegi élet mennyiség jelenjen meg a sávon |  |
| SetMaxHealth | void | beállít egy általunk megadott maximális élet mennyiséget |  |

### CoinManager

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| CoinCount | int | Megszámolja az összeszedett érme mennyiségeket |  |
| CoinText | Text | Kiírja az érme mennyiséget |  |
| exit | GameObject | Kiléptet a szintről amint a játékos célba ér |  |
| SackCount | void | Zsákokat számolja |  |
| AddCoin | void | Hozzáadja az érmét egy általunk adott értékkel (+1) |  |
| AddSack | void | Hozzáadja az zsákot egy általunk adott értékkel (+10) |  |
| Start | void | Lekérdezi az UpdateUI-t |  |
| UpdateUI | void | Ellenőrzi hogy a játékos összegyűjtött-e egy adott érme mennyiséget hogy a kijárat megnyíljon |  |

### PlayerLobbyMovement

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| animation2 | Animator | Hivatkozik az animátorra |  |
| moveDirection | Vector2 | A játékos melyik irányba halad |  |
| moveSpeed | float | A játékos mozgási sebessége |  |
| moveX | float | A játékos vízszintes mozgása |  |
| moveY | float | A játékos függőleges mozgása |  |
| rb | Rigidbody2D | A rigidbody2D komponens |  |
| Awake | void | lehívja az animációt a játékosra |  |
| FixedUpdate | void | hivatkozik a Move metódusra |  |
| FlipCharacter | void | Ellenőrzi hogy a játékos a vízszintes tengelyen + vagy - értékben mozog és aszerint fordítja a játékost |  |
| Move | void | A mozgás megvalósítása |  |
| ProcessInputs | void | lekérdezi a mozgást és normalizálja azon értékeit |  |
| Update | void | Hivatkozik a ProcessInputra és a FlipCharacter-re és emellet beállítja a mozgást a bemenet szerint |  |

### SettingsMenu

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| MainAudio | AudioMixer | vezérli és kezeli a hangot a játékban |  |
| MasterV | Slider | Állítható értéket ad a fő hangerő sávnak |  |
| MusicV | Slider | Állítható értéket ad a zene hangerő sávnak |  |
| ChangeMasterAudio | void | Változtatni lehet a fő hang erejét |  |
| ChangeMusicAudio | void | Változtatni lehet a zene hangerejét |  |
| MainAudio | AudioMixer | vezérli és kezeli a hangot a játékban |  |
| MasterV | Slider | Állítható értéket ad a fő hangerő sávnak |  |

### PlayerAttack

|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| animat | Animator | hivatkozik a megfelelő animációra |  |
| AttackCoolDown | float | A támadások között, általunk megadott eltelt idő |  |
| cooldownTimer | float | Az időzítő értéke |  |
| playermove | PlayerMovement | Hivatkozás a PlayerMovement scriptre |  |
| attack | void |  |  |
| Awake | void | Kettő kömponenst értékesítünk a játékos mozgását és az ahoz tartozó animációt |  |
| Update | void | Egy gomb lenyomásával aktiválja a támadás funkciót |  |

### NextLevel

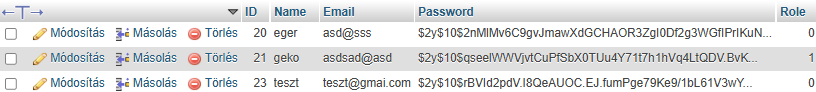
|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| SceneBuildIndex | int | A Unity Editorban lévő következő jelenet lekérdezése |  |
| requiredObjectName | string | Az első szint meghatározása |  |
| requiredObjectName2 | string | Az második szint meghatározása |  |
| OnTriggerEnter2D | void | Ellenőrzi hogy a játékos ütközött a Collider2D elemekkel akkor vigye el a következő szintre |  |

### SettingsMenuManager

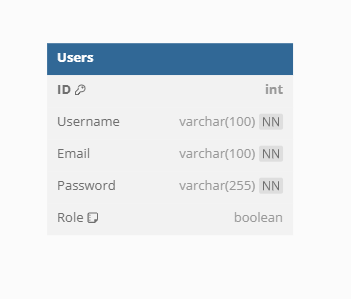
|  |  |  |  |
| --- | --- | --- | --- |
| **mező neve** | **mező típusa** | **mező leírása** | **megjegyzés** |
| Start |  |  |  |
| Update |  |  |  |

### Minta adatok

### Players



### Adatbázis diagram



Az adatbázis diagram **Users** tábla adatstruktúráját mutatja be dbdiagram.io formátumban. A tábla célja, hogy a felhasználói adatokat biztonságosan és konzisztensen tárolja. A tábla struktúrája a következőképpen épül fel:

A **Users** tábla elsődleges kulcsa az **ID** oszlop, amely automatikusan növekvő egész számként biztosítja az egyedi azonosítókat minden rekord számára. Ezen felül két fontos, egyedi azonosító mező található: a **Username** és az **Email**. Mindkét oszlop esetében a UNIQUE és a NOT NULL korlátozások vannak érvényben, ami azt jelenti, hogy nem engedélyezett az üres vagy ismétlődő értékek tárolása, így garantálva a felhasználók egyediségét a rendszerben.

A **Password** oszlop egy 255 karakter hosszúságú mező, amely elegendő kapacitást biztosít a biztonságos jelszóhash-ek tárolásához, ezzel támogatva a rendszer biztonsági követelményeit. Végül a **Role** oszlop boolean típusú, alapértelmezett értéke false, amely segít megkülönböztetni például az átlagos felhasználókat az adminisztrátori jogosultsággal rendelkező felhasználóktól.

### Adatbázis továbbfejlesztési lehetőségek

A rendszer további fejlesztésére számos lehetőség van.

* A mentések és játékos statisztikája kezelése mellett a jövőben például a játékosok közötti interakciók (pl. üzenetek, barátok) tárolására is szükség lehet.

## A programozás folyamata

### tervezés folyamata Figma

### pl regisztrációs oldal programozása

### olyan kódrészletek magyarázatokkal, amit kiemelnél, amire büszkék vagytok

# Felmerül akadályok

problémák: pl. nem sikerült képfeltöltés, mi volt a gond, hogyan oldottad meg, itt lehet olyat, hogy elengeded, mást csinálsz helyette, és ezt leírod, hogy nem sikerült megoldani

# Összefoglalás

Elért eredmények, tovább fejlesztési lehetőségek

közös munka értékelése

# Források

## Weboldalak

bootstrap input group:

[Input group · Bootstrap v5.3](https://getbootstrap.com/docs/5.3/forms/input-group/)

css animációk:

<https://www.w3schools.com/css/css3_animations.asp>

css grid:

[CSS Grid Layout Guide | CSS-Tricks](https://css-tricks.com/snippets/css/complete-guide-grid/)

php functions:

<https://www.w3schools.com/php/php_functions.asp>

php változó elérése javascriptben:

<https://jscript.blog.hu/2022/06/12/php_valtozo_javascript_valtozoba>

## Képek

háttérkép:

<https://th.pngtree.com/freebackground/autumn-beautiful-scenery-game-background_2508913.html>

abc betűrendben: Szerző, cím, megjelenés dátuma, ha van, link (excelben érdemes gyűjteni és sorba rendezni)

képek forrása külön