

Endless Survival

Készítette: Mikes Dávid, Dömök Dávid
szoftverfejlesztő 2022-2023
Konzulens tanár: Juhász Zoltán

Tartalomjegyzék

Bevezetés.....	3
Munkamegosztás	3
Fejlesztői dokumentáció: Weboldal	4
1. Alkalmazott fejlesztői eszközök:.....	4
2. Alkalmazott technológiák:	4
3. Adatbázis	4
- Táblák	4
- Tárolt eljárások.....	5
4. Biztonság	6
5. Jogkörök	6
6. Navigáció és elrendezés	6
7. Nyelv.....	8
Felhasználói dokumentáció	9
1. Telepítés útmutató	9
2. Felhasználói kézikönyv és az oldal működése	9
- Bejelentkezés.....	9
- Profil menü	12
- Rangok.....	14
- Lábléc.....	14
Tesztelés	15
Továbbfejlesztés lehetőség.....	15
Fejlesztői dokumentáció: Játék	16
1. Alkalmazott fejlesztői eszközök.....	16
2. Alkalmazott technológiák.....	16
3. Futtatási környezet.....	16
4. Fejlesztői ütemterv.....	16
5. Grafikus részek elkészítése.....	16
6. Minimum viable product.....	16
7. Tesztelés.....	21
Felhasználói dokumentáció	23
1. Rendszerkövetelmények	23

2. Telepítési útmutató	23
3. A program működése, felhasználói kézikönyv	23
Felhasznált források:	24
Ábrajegyzék	25

Bevezetés

Szakedolgozatunk témájának kiválasztásában segített, hogy korábban én már elkezdtem Unity-ben egy játékot fejleszteni. Eredetileg a játék egy looter shooter nek indult, de idővel újra gondoltam az egészet és egy fps survival játék lett belőle.

Az Endless Survival egy egyszemélyes túlélő shooter játék (jövőben többjátékos opció is tervben van). A játék során a játékos minden pálya előtt kiválaszthat egy elsődleges és egy másodlagos fegyvert. A játékosok minden pálya után tapasztalati pontot kapnak elegendő tapasztalati pont összegyűjtése esetén a játékos szintet lép és kap egy képesség pontot. A képesség pontokat pályán kívül lehet elkölteni különféle karakter erősítésekre vagy képességekre. Képesség pontok mellé bizonyos szintek elérésekor új fegyverek is elérhetővé válnak.

A játék célja, hogy a játékos minél tovább próbáljon meg túlélni a folyamatosan erősödő és egyre gyorsabban termőellenségekkel szemben.

A játéknak az Endless Survival nevet választottam, mely egy Windows operációs rendszer alatt futó, c# nyelven írt asztali alkalmazás. A játék a Unity engine-t használja.

Munkamegosztás

A játékot készítette Mikes Dávid.

A weboldalt és az adatbázist készítette Dömök Dávid.

Fejlesztői dokumentáció: Weboldal

1. Alkalmazott fejlesztői eszközök:

Kódszerkesztő: Visual Studio Code 1.77.3

Adatbázis kezelő: phpMyAdmin 5.2.0

Szerver: XAMPP 8.1.10

2. Alkalmazott technológiák:

Web fejlesztő nyelv: php, JavaScript

Web alkalmazás keretrendszer: Laravel 10.x

Fronted eszközkészlet: Bootstrap 5.3

Ikonkönyvtár: Font Awesome 4.7.0

3. Adatbázis

- Táblák

A program adatbázisa phpMyAdmin-al lett létrehozva és a táblák egy része a Laravel adatbázis migrációjával.

Az adatok tárolásához 5 táblát használok melyeknek mezői és tulajdonságai a következők:

users		
id	bigint(20), PK	A felhasználó egyedi azonosítója.
email	varchar(255), FK	A felhasználó email-je. A „password_resets” tábla összekötéséért felelős.
username	varchar(15)	A felhasználó neve.
password	varchar(255)	A felhasználó jelszava.
avatar	text	A felhasználó profilképének a nevét és fájlkiterjesztését tárolja. Alapértelmezetten „default.jpg”
email_verified_at	timestamp, NULL	Az email visszaigazolásának az ideje.
remember_token	varchar(100), NULL	Egy token ami a bejelentkezésnél a „bejelentkezve marad” beállításhoz kell. Alapértelmezetten NULL.
created_at	timestamp, NULL	Regisztrálás ideje. Alapértelmezetten NULL.
updated_at	timestamp, NULL	Az felhasználó adatainak módosításának ideje. Alapértelmezetten NULL.

Ez a tábla tartalmazza a felhasználók személyes és bejelentkezéshez szükséges adatokat.

characters		
id	int(11), PK	A karakter egyedi azonosítója.
uid	bigint(20), FK	A felhasználó id-ja. A „users” tábla összekötéséért felelős.
name	varchar(50)	A karakter neve.
level	int(11)	A karakter szintje. Alapértelmezetten 0.
caste	varchar(100), NULL	A karakter kasztja. Alapértelmezetten NULL.
save	longtext, NULL	A legutóbbi ezzel a karakterrel indított játék mentés betöltéséhez tárol adatokat. Alapértelmezetten NULL.

Ez a tábla tartalmazza a felhasználók karaktereiknek az adatait.

matches		
id	int(11), PK	A meccs egyedi azonosítója.
cid	int(11), FK	A karakter id-ja. A „characters” tábla összekötéséért felelős.
time	time	Túlélési idő.
kills	int(11)	Ölések száma.
death	bit(1)	Meghalt-e a játékos.
primary_weapon	varchar(50)	Elsődleges fegyver neve.
secondary_weapon	varchar(50)	Másodlagos fegyver neve.
damage_taken	int(11)	Szerzett sérülés.
date	datetime	A meccs befejezésének ideje.

Ez a tábla tartalmazza a felhasználók meccseit.

password_resets		
email	varchar(255), PK	A felhasználó email-ja.
token	varchar(255)	A jelszó visszaállításához szükséges token.
created_at	timestamp, NULL	Alapértelmezetten NULL.

Ez a tábla tartalmazza a jelszó visszaállításához szükséges adatokat.

Van még egy „migrations” nevű tábla ez a Laravelnek szükséges adatokat tartalmaz a migrációkról.

- Tárolt eljárások

Összesen két tárolt eljárás van:

add_new_match(user_id, character_name, time, kills, death, primary_weapon, secondary_weapon, damage_taken, date):

Egy új meccset add hozzá az adatbázishoz.

Mivel egy meccs egy karakterhez tartozik ezért az eljárás előbb megnézi hogy van-e tárolva az adott nevű karakter a felhasználónak az adatbázisban és ha nincs akkor létrehoz egyet.

Hibaüzenetet ad vissza ha nem létezik az adott felhasználó.

add_rand_match(username, count):

Megadott mennyiségű meccset add hozzá az adatbázishoz véletlenszerű értékekkel (a fegyverek és a karakter neve mindig ugyan az).

Hibaüzenetet ad vissza, ha nemlétezik az adott felhasználó, vagy ha kevesebb mint 0 meccset akarunk hozzáadni.

4. Biztonság

A Laravel bycrypt hash támogatást biztosít a jelszavak tárolására amiket használtam is.

A felhasználó hitelesítésére a Laravel beépített funkcióit használom.

5. Jogkörök

Két felhasználót különböztetünk meg, aki bevan jelentkezve és aki nincs.

A kettő között csak az a különbség, hogy aki bevan jelentkezve annak elérhető egy menü a saját profilját nézve legalul (Bővebben lásd [itt](#)).

6. Navigáció és elrendezés

Az oldal bal felső sarkán található egy ikon ez a játék parancsikonja.










Az oldal három menüpont található: Endless Survival (főoldal), Rangsor, Rólunk illetve egy Kereső mező, ami mobil nézetben 4. menüpontként jelenik meg.

A főoldal tartalmaz egy leírást a játékról, a telepítés menetéről és itt lehet letölteni a hozzá szükséges fájlokat.









A Rangsor menüpontba megtekinthető a játékosok statisztikái alapján kalkulált rangsor. Itt csak azok a játékosok találhatók meg, akik legalább egy meccset játszottak (bővebben lásd [itt](#)).

A Rólunk menüpont tartalmaz egy rövid leírást a fejlesztőről és a játék keletkezésének okáról.

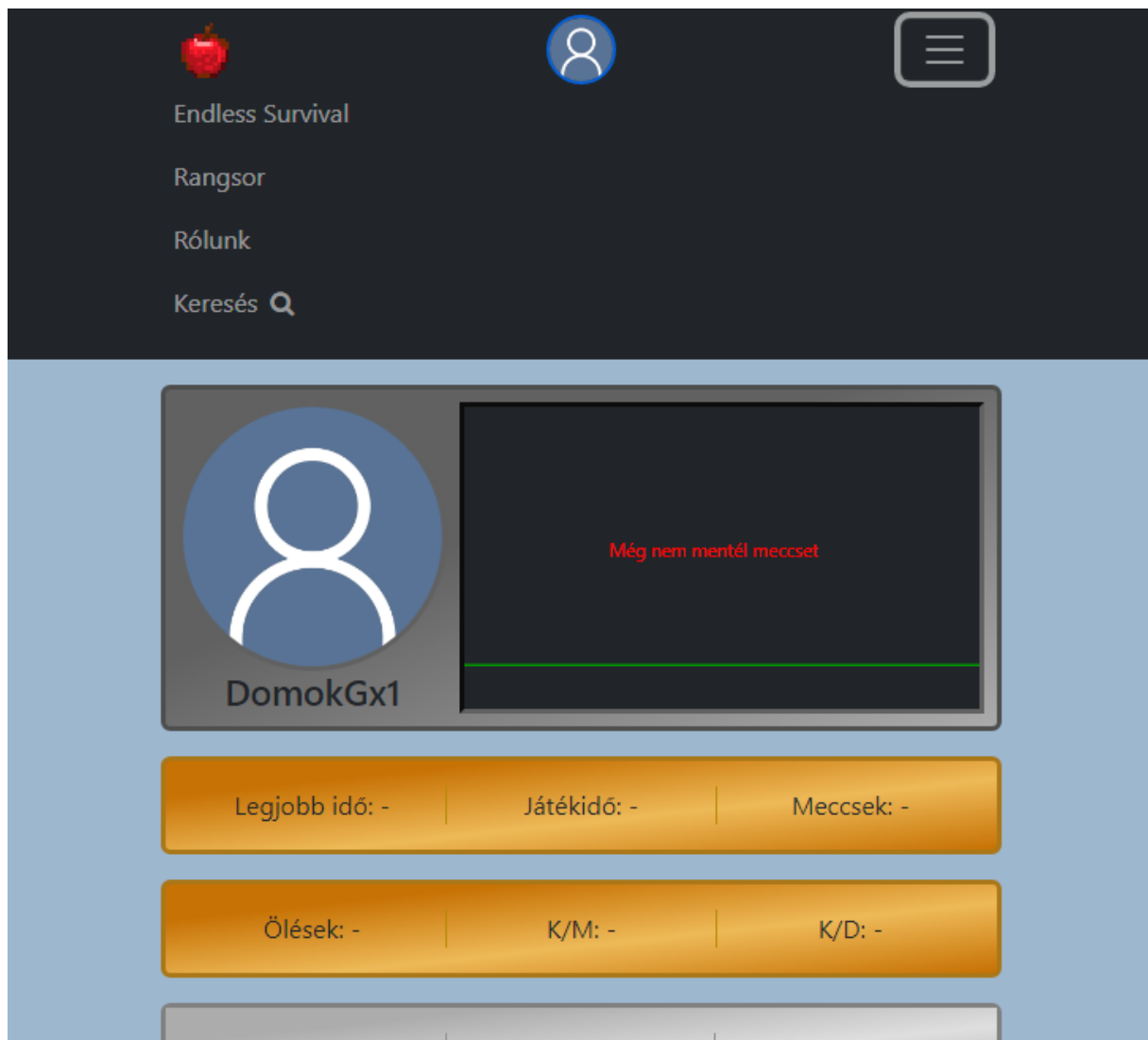
Az oldal illeszkedik a képernyő méretéhez és annak megfelelően lesz elrendezve. Például mobil nézetben a menüpontok egy lenyíló menüben helyezkednek el egymás alatt, hogy elférjenek. Míg asztali számítógép nézetben egymás mellett vannak.

 Endless Survival Rangsor Rólunk <input type="text" value="Felhasználónév"/> <input type="button" value="Keresés"/> 			
Keresés			
Felhasználó	Legjobb idő	Mecsek	Játékidő
 Babi	-	0	00:00:00
 DomokGx1	-	0	00:00:00
 DomokGx2	-	0	00:00:00
 Jóképű Kenyér	-	0	00:00:00
 Kenyér	-	0	00:00:00
 Steve	-	0	00:00:00
 test	00:54:13	35	16:24:11

1. ábra: Kereső asztali nézetben

  			
Keresés			
<input type="text" value="Felhasználónév"/> <input type="button" value="Keresés"/>			
Felhasználó	Legjobb idő	Mecsek	Játékidő
 Babi	-	0	00:00:00
 DomokGx1	-	0	00:00:00
 DomokGx2	-	0	00:00:00
 Jóképű Kenyér	-	0	00:00:00
 Kenyér	-	0	00:00:00
 Steve	-	0	00:00:00
 test	00:54:13	35	16:24:11

2. ábra: Kereső mobil nézetben



3. ábra: Profil mobil nézetben lenyíló menüvel, 0 meccsel

7. Nyelv

Az oldal magyar és angol nyelvet támogat. Ehhez a Laravel egyik nyelvi lokalizációs csomagját használtam. (<https://laravel-lang.com/>)

Az oldal saját szövegeit külön fájlokban vannak tárolva ugyanis a nyelvi csomag frissítésekkor felülíródnának az általam bevitt szövegek. Ezek a fájlok elrendezésben megfelelnek a nyelvi csomagnak.

Az angol nyelvet frissítenem kellett a nyelvi csomaggal, mert abban több szöveg volt megadva, mint az alapértelmezettbe (pl.: a validálásnál az attribútumok).

Több nyelv hozzáadásához frissítse a config/app.php locales listáját és adja hozzá a megfelelő fájlokat a lang mappába.

A Nyelvek betöltéséhez létrehoztam egy Middleware-t Localization.php néven.

Felhasználói dokumentáció

1. Telepítés útmutató

A weboldal működéséhez az alábbi dolgokat kell tenni:

1. Telepítse a [Composer](#), PHP és a [Node](#) legújabb verzióját.
2. Töltse le vagy klónozza a [projektet](#) web részét.
3. Nyissa meg a projekt mappáját és futtassa a „composer install --prefer-source” parancsot terminálban vagy cmd-ben.
4. Másolja a .env.exapmle fájl tartalmát a .env fájlba. Erre használhat parancsot „copy .env.example .env”.
5. Nyissa meg a .env fájl és módosítsa a következő mezőket az önnek megfelelő értékekre:
 - az adatbázishoz:
 - DB_HOST, DB_PORT
 - DB_USERNAME
 - DB_PASSWORD
 - email-es értesítéshez:
 - MAIL_MAILER
 - MAIL_HOST
 - MAIL_PORT
 - MAIL_USERNAME
 - MAIL_PASSWORD
 - MAIL_ENCRYPTION
6. Futtassa a „php artisan key:generate” parancsot.
7. Importálja a szakdgame_dumb.sql fájlt phpMyAdmin-ban.
8. Futtassa a „php artisan serve” parancsot.

Az egyszerűség kedvéért az email mezők már előre ki vannak töltve a .env.example fájlban.

2. Felhasználói kézikönyv és az oldal működése

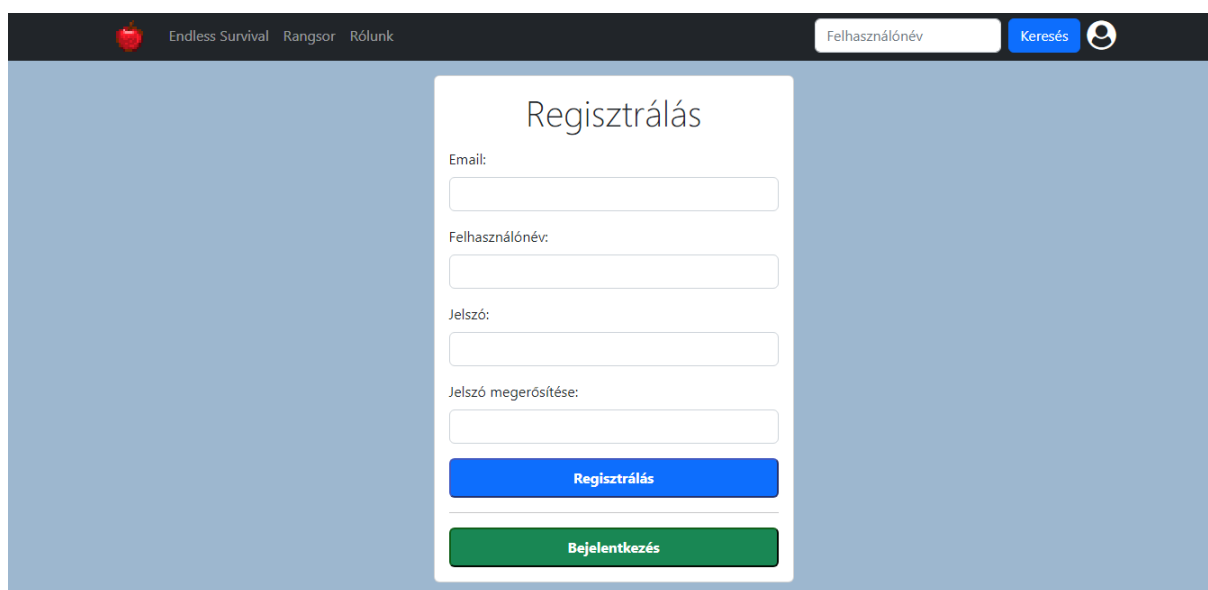
A projektben két Controllert használtam PageController és PostController. A PostController-be ment minden olyan ami post módszert használ. A PageController-be meg mindent ami get-et használ.

- Bejelentkezés

Az oldalon bejelentkezni a jobb felül lévő (mobil nézetben középen) profil ikonra kattintva lehet. Ez elvezet egy bejelentkezési oldalra. Itt lehet bejelentkezni, regisztrálni vagy jelszó visszaállító email-t kérni.



4. ábra: Menüpontok



5. ábra: Regisztráció

A bejelentkezéshez felhasználónevet vagy email címet és jelszót kell használni.

Bejelentkezésnél a felhasználó beállíthatja, hogy bejelentkezve maradjon-e. Ha beállítja, akkor a bejelentkezve tartja a felhasználót addig, amíg ő ki nem jelentkezik vagy törlik az oldal cookie-it. Még akkor is, ha bezárta a böngészőt.

6. ábra: Bejelentkezés

A jelszó visszaállítás email kérésnél felhasználónevet vagy email-t kell megadni.

7. ábra: Jelszó helyreállítás

Ha a felhasználó sikeresen bejelentkezett a profil ikon helyett a felhasználó profilképe jelenik meg és a saját profilját megtekintve láthat 4 menüpontot legalul.

Legjobb idő: -	Játékidő: -	Mecsek: -
Ölések: -	K/M: -	K/D: -
Legjobb idő: -	Játékidő: 00:00:00	Mecsek: 0
Ölések: 0	K/M: -	K/D: -
Halálok: 0	Szerzett sérülés: 0	20. játékos

8. ábra: Profil beállítások menü

- Profil menü

- Profil szerkesztés:

Itt lehet felhasználónevet és profilképet változtatni. A profilkép csak .png, .jpg és .jpeg fájlkiterjesztésű, maximum 2048 KB lehet.

Még megtekinthető itt egyéb adat is köztük, hogy mikor erősítette meg az email fiókját. Ha még ezt nem tette meg akkor újra tudja küldeni az ehhez kellő email-t.

A mentés gombra kattintva elmentheti az új adatokat kivéve, ha valami helytelen. Ilyenkor legalul értesíti a felhasználót hogy mi a probléma.

The screenshot shows the 'Profil szerkesztés' (Profile editing) interface. At the top, there's a navigation bar with 'Endless Survival', 'Rangsor', and 'Rólunk'. A search bar and a user icon are on the right. The main content area features a large circular profile picture placeholder with a white '8' shape inside, labeled 'DomokGx1'. To the right of the picture is a table with the following data:

Felhasználónév:	DomokGx1
Email:	domok.davidka1@gmail.com
Visszaigazolva:	Visszaigazolva Újraküldés
Regisztrálás:	2023-04-15 19:33:28
Profilkép:	Fájl kiválasztása Alapértelmezett

Below the table are two buttons: a red 'Vissza' (Back) button and a green 'Mentés' (Save) button. At the bottom center, the text 'Endless Survival' is displayed.

9. ábra: Profil szerkesztés

- Jelszó változtatás:

A felhasználó megváltoztathatja a jelszavát. A régi jelszó szükséges hozzá.

10. ábra: Jelszó megváltoztatása

- **Fiók törlése:**

A felhasználó végleg törölheti a fiókját. Jelszó szükséges hozzá.

11. ábra: Fiók törlése

- **Kijelentkezés:**

Kijelentkezteti a felhasználót.

Bármelyik játékos statisztikáit meglehet tekinteni akár bejelentkezés nélkül is.

Egy játékos statisztikájának megtekintésére rákereshet a játékos nevére a keresőmezőben vagy megtalálhatja a rangsorban és a profilképére vagy a nevére kattintva elérheti azt.

A játékos profilját megtekintve a profilképe mellet láthatja az összes meccsét egy grafikus táblában. Ebben a táblában egy piros vonal jelzi a játékidőt dátumhoz kötve. Ahogy a felhasználó mozgatja a kurzort a táblában láthatja az egyes meccsek adatait. A tábla bal oldalán megtekintheti a regisztrálás dátumát is ha oda viszi a kurzort.



12. ábra: Grafikus tábla

- Rangok

Összesen 7 rangsor van: legjobb idő, játékidő, meccsek száma, Ölések száma, ölés/halál, ölés/meccs és a 7. az ezekből kiszámolt rang. Nevezzük fő rangnak. Ez a rang alapján vannak felsorolva a játékosok a Ranglistán.

A fő rangot úgy számoljuk ki, hogy összeadjuk a 6 rang számát és ha ez a szám minél közelebb áll a 0-hoz annál nagyobb a fő rang.

Ha a játékos még egy meccset sem játszott, akkor ezt kiírja a grafikus tábla és nincsenek statisztikái.

Előfordulhat, hogy a ranghoz kiszámított érték ugyan az két felhasználónak ilyenkor ugyan azt a rangot kapják meg.

The screenshot shows the 'Rangsor' (Ranking) page. At the top, there is a search bar labeled 'Felhasználónév' and a 'Keresés' button. Below the title 'Rangsor' is a table with the following data:

#	Felhasználó	Legjobb idő	Meccsek	Játékidő
1.	test1	00:59:54	459	225:58:55
2.	test	00:54:13	35	16:24:11
3.	test6	00:57:17	2	01:54:34

13. ábra: Rangsor

- Lábléc

Itt lehet nyelvet váltani (bővebben [itt](#)).

14. ábra: Lábléc

Tesztelés

A tesztelésben segített a Lavravel debug beállítása, amivel a fejlesztés során rengeteg hibát sikerült kijavítani a részletes hibaüzeneteknek köszönhetően.

Az oldalon megjelenő statisztikák, rangok megjelenítéshez szükség volt teszt adatokra. A teszt adatok feltöltéséhez egy [tárolt eljárást](#) hoztam létre az adatbázisban.

Az adatbázis tesztadatai megtalálhatóak test_data.sql néven, amit importálni kell phpMyAdmin-ban.

Továbbfejlesztés lehetőség

A játékban jelenleg csak egy karakterrel lehet játszani, de a jövőben ez változhat. Az adatbázis és a weboldal úgy van megírva, hogy támogassa a több karaktert.

Lehetőség van több nyelv támogatására (bővebben [itt](#)).

Fejlesztői dokumentáció: Játék

1. Alkalmazott fejlesztői eszközök

Játék motor: Unity

Kódszerkesztő: Visual Studio 2022, Visual Studio Code

Grafikai tervező program: Blender

Adatbázis-tervező program: MySql

2. Alkalmazott technológiák

Programozási nyelv: C#, php

3. Futtatási környezet

Az Endless Survival játék Windows operációs rendszerre lett tervezve. A játék internet kapcsolatot igényel. A weboldalról való letöltést és telepítést követően a EndlessSurvival.exe fájlt kell elindítani.

4. Fejlesztői ütemterv

2022 Október	A játék alapjainak megtervezése
2022 November	A játék alap logikájának megírása
2023 Január	A játék menetének kidolgozása
2023 Március	Felhasználói és játék adatok tárolása adatbázisban
2023 Április	Kód refaktorálása és tisztítása, grafikai részek megtervezése

5. Grafikus részek elkészítése

Mivel számomra egy játéknál az egyik legfontosabb elem a grafikai megjelenés stílustól függetlenül. Mivel egy ideje már foglalkozok 3D modellezéssel így egy stylised stílus mellett döntöttem. A grafikai megvalósításhoz a Blender 3D-t használtam, mivel nem értek a 2D-s arthoz így az UI elemekhez is Blender ben készített 3D-s rendereket használtam. A játék műfajának sci-fi-t választottam, de mivel karakter design-hez nem volt sok ötletem így egy ingyenes online AI kép generátor programot használtam (Craiyon), hogy megtervezsem az ellenfeleket, fegyvereket és a player karakterét. Mivel a játék first person nézetben játszódik és még nincs benne multiplayer opció emiatt úgy döntöttem, hogy csak a karakter kezeit modellezem meg. Végül az ellenfelek modelljei nem készültek el így az alap placeholder objektok vannak az ellenfelek helyett. Az animációkat nem sikerült jól megcsinálni így végül a player kezei sincsenek berakva és animációk sincsenek még.

6. Minimum viable product

A könnyebb áttekinthetőség érdekében a scripteket külön mappákba rendeztem. A projekt állományainak architektuális felépítése a következő:

-Enemy

Ez a könyvtár tartalmazza az összes scriptet, ami az ellenfelekhez tartozik.

- Events

Ebben a könyvtárban található a háttérben futó események.

- MenuScript

Ebbe a mappába van tárolva minden script a menühöz.

- SelectMenu

Ez a mappa tartalmazza a karakter és fegyver választáshoz tartozó scripteket.

- Weapon

Ebben a mappában vannak a fegyverekhez tartozó scriptek.

- PlayerScript

Ebben a mappában vannak a játékos hoz tartozó scriptek több almappában.

1. DataPresistance

Ebben a mappában van tárolva minden file, ami a játékos adatainak a mentéséhez kell.

2. LevelSystem

Ebben a mappában van tárolva a játékos szintjéhez tartozó scriptek.

3. Managers

Ebben a mappában vannak a különböző összekötő managerek.

4. Movement

Ebben a mappában van tárolva minden játékos irányításához tartozó script.

5. Skills

Ebben a mappában van tárolva képességekhez tartozó scriptek.

A projekt állományai részletesen:

- Enemy könyvtár:

1. DealDmg

Ez a script felelős azért, hogy az enemy képes legyen a játékost megsebezni

2. EnemyAI

Ez a script felelős az enemy mozgásáért és támadásáért. Mivel a játék csak egy survival móddal rendelkezik így az enemy csak 3 státusszal rendelkezik:

- `playerInSightRange` - Ez a kódrészlet figyeli, hogy a player hatótávolságon belül van e vagy sem és ha hatótávon belül van akkor kezdje el üldözni a playert. (jelenlegi beállításokkal mindig hatótávon belül van, de más játékmódok bevezetésénél lehet módosítani ezen)
- `Environment` - ez a kódrészlet egy raycast-al figyeli, hogy van e a player és az enemy közt valami tereptárgy.
- `playerInAttackRange` – ez a kódrészlet figyeli, hogy a player támadási hatótávon belül van e vagy sem.

Ezek utána ezek a feltételek használatával eldönti az AI, hogy mit csináljon.

3. EnemySpawner

Ez a script felelős az enemyk spawnolásáért. jelenleg 2 megadott koordináta által behatárolt négyzet területen spawnol enemyket. Ez a script össze van kapcsolva az Events-Timer scripttel és ahogy telik az idő csökkenti az enemyk spawnolási idejét

4. Target

Ez a script felelős azért, hogy az enemyt el lehessen találni és el lehessen pusztítani.

- Events könyvtár

1. Timer

Ez a script figyeli, hogy mennyi ideig sikerült a playernek túlélnie. Későbbiekben más játékmódokhoz visszaszámlálóként is képes funkcionálni.

2. Ectraction

Ez a script összekötésben van a Timer hez és ha a timer elér egy x időt megjeleníti a kimenekítési pontot a playernek. A kimenekítési pontban a playernek x ideig túl kell élnie mert ha elhagyja kimenekítési pontot a visszaszámláló újra indul. Ha sikeresen túléli a visszaszámlálót akkor a player nyer és átvált a Summary scene re.

- MenuScript

1. MainMenu

Ez a script felelős a menüért, ami a bejelentkezés után jelenik meg amiben elindíthatjuk a játékot, megnyithatjuk a beállításokat vagy kiléphetünk a programból.

2. Menu

Ez a script felelős az in game menü megnyitásáért.

3. MoveCanvas

Ez a script felelős a lobby/főmenü ben történő karakter választás és képesség fejlesztés nézet közti váltásért.

4. SettingsMenu

Ez a script tartalmazza mind a két menüben a beállítások menüpontot.

5. Summary

Ez a script a játék végén megjelenő scene-el kapcsolatos információkat tartalmazza (Összegzés, meccs végkifejletje), innen lehet újrakezdeni a meccset vagy visszatérni a menübe.

- SelectMenu

1. CharacterSelection

Ez a script követi, hogy melyik karakter lett kiválasztva. (jelenleg inaktív mert csak egy választható karakter van)

2. LoadCharacter

Ez a script tölti be a pályára a kiválasztott karaktert.

3. PrimarySelection

Ez a script követi, hogy melyik elsődleges fegyver lett kiválasztva és tárolja el egy változóban.

4. LoadPrimary

A PrimarySelection scriptben eltárolt változó alapján betölti a pályára a kiválasztott elsődleges fegyvert.

5. SecondarySelection

Ez a script követi, hogy melyik másodlagos fegyver lett kiválasztva és tárolja el egy változóban.

6. LoadSecondary

A SecondarySelection scriptben eltárolt változó alapján betölti a pályára a kiválasztott másodlagos fegyvert.

- Weapon

1. EquiWindow

Ez a script felelős azért, hogy bizonyos fegyvereket a játékos csak bizonyos szintek elérésekor tudjon használni.

2. GunSystem

Ez a script felelős azért, hogy a fegyverek képesek legyenek lőni. különböző beállításokkal lehet állítani a fegyvernek a lövési módjait, és a fegyverekhez tartozó animációk/Fx-eket is ez a script tárolja.

3. WeaponSwitching

Ez a script felelős azért, hogy a player válthasson az elsődleges és másodlagos fegyvere közt.

- PlayerScript
 1. Player

Ez a script tartalmazza a playernek a statjait(élet, shield), tartalmazza a képességeket is amik a player statisztikáit módosítják.

- DataPresistance
 1. Data – GameData

Ez a class tartalmazza a menteni kívánt adatok definiálását.

2. DataPersistenceManager

A DataPersistenceManager egy singleton class. Ez a class felelős a játékos adatainak a mentéséért és betöltéséért 2 scene közt.

3. FileDataHandler

Ez a class felelős a játékos adatainak file-ba való mentéséért.

- LevelSystem
 1. LevelSystem

Ebben a classban vannak definiálva a különböző szintekhez szükséges experience mennyiségek. Először a class leteszteli, hogy a player elérte e a maximum szintet vagy sem. Utána normalizálja az experience értékét és ezt a normalizált értéket használja későbbiekben a LevelWindow class, hogy vizuálisan kijelyezze a jelenlegi haladást a következő szint felé. Az AddExperience metódus adja hozzá a kapott experience pontokat a meglévőkhöz és ha eléri az adott értéket akkor megnöveli a játékos szintjét.

2. LevelSystemAnimated

Ez a class felelős a szintlépési animációért, az xp bár animálásáért.

3. LevelWindow

Ez a class felelős a levelsystem UI elemeinek megjelenítéséért.

- Managers
 1. LevelSystemManager

Ez a class egy összekötőként szolgál a LevelSystem class között és azok a classok közt, amik megvalósítják. Másodsorban ez a class számolja, hogy a játékosnak hány ellenféllel sikerült végeznie.

2. SaveLoad

Ebben a classban van tárolva minden adat, ami mentésre vagy betöltésre kell.

- Movement
 1. MouseLook

Ez a script felelős a játékos kamera mozgásáért.

2. PlayerController

Ez a class felelős a játékos mozgásáért. Egy public enumba vannak deklarálva a különböző state-k. A StateHandler metódus felelős a különböző statek-el kapcsolatos változtatások elvégzéséért.

- Skills
 1. Dashing

Ez a class egy szimpla dash képességet ad a játékosnak. A Dash metódusban a játékos-t a jelenlegi mozgási irányába megadott erővel felfelé és mozgási irányba (vagy ha egyhelyben áll akkor előre) löki a játékost azzal, hogy meghívja a DelayedDashForce metódust. A DelayedDashForce a player rigidbodyd-componentjéhez ad egy impulzus erőt. A GetDirection metódus határozza meg a játékos jelenlegi irányát.

2. PlayerSkills

Ez a class tartalmazza az összes képesség definiálását egy public SkillType nevű enum-ba. Egy SkillType típusú listában tárolja a feloldott képességeket. A CanUnlock metódus ellenőrzi le, hogy a feloldani kívánt skillnek minden követelménye teljesült-e. Ezután a TryUnlockSkill metódus leteszteli, hogy van-e elég skillpontja a játékosnak, hogy feloldja a kéességet. Ezután az UnlockSkill metódus hozzáadja a feloldott képességeket tároló listához az adott képességet.

3. Ui_SkillTree

Ez a class tartalmazza a SkillTree-hez tartozó UI elemeket. A SetPlayerSkills metódus tartalmazza a képességekhez tartozó gomboknak a listáját és hívja meg a 2 vizuális update metódust. Az UpdateVisuals metódus frissíti a UI elemeknek a megjelenítését. A SkillButton class definiálja a különböző képességekhez tartozó gombokat.

7. Tesztelés

A fejlesztés során folyamatosan manuális tesztelésre volt szükség, játék fejlesztése során több probléma felmerült a kódban melyeket javítani kellett. Az első probléma, amibe fejlesztés során belefutottam az a Unity universal renderer pipeline-al történt, csak hosszabb tesztelés után derült ki, hogy a hibát egy elavult VFX pack eredményezte. A második probléma, amibe belefutottam az a mentés és betöltés volt. Tesztelések során arra a megoldásra jutottam, hogy a betöltés során rá kellett rakjak egy delay-t a scriptre mert az előtt akarta feltölteni a másik script adatait, hogy az aktiválódott volna. Sajnálatos módon ez viszont egy másik hibához vezetett, ami miatt mikor hozzáadódik a tapasztalati pont a jelenlegihez akkor a rendszer visszaugrik egyes szintre és onnan szintelteti vissza a megfelelő szintre és tapasztalati pontra a játékost. Ez ahhoz vezetett, hogy a játékos újra megkapta az összes képességpontot, amit azzal tudtam korrigálni, hogy felvettem egy új változót, ami számon tartja, hogy a játékos eddig hány képességpontot költött el. Végül ezt a hibát nem sikerült kijavítani. Hibák miatt a StateManager scriptet többször is újra kellett írni. A levelSystem né apróbb elírási hibákba futottam. Az adatbázisba való írásnál egy hibába ütköztem, aminél a DBReadWrite script nem kapja meg a DBManager értékeit és mivel még nem találtam rá megoldást így az adatbázisba való írás még nem működik. A Lobby nál a UI elemek mozgásánál számos hiba

felmerült, mint a kód szempontjából mind a UI megvalósítása során. Az EnemyAI script megírásánál is több problémába futottam, amiket javítani kellett például, hogy érzékelje az enemy hogy van e fal a játékos és az adott enemy közt. Ezt először egy Raycast-al akartam megoldani, de mivel az enemy egy CheckSphere-el tesztelte, hogy a player lőtávon belül van e vagy sem így bele futottam abba a hibába, hogy ha a játékos túl közel van a falhoz akkor az enemy úgy érzékelte, hogy a fal az köztük van és a ChasePlayer state futott le és nem kezdett el támadni. erre a megoldás az lett, hogy a CheckSphere-t lecseréltem egy Raycast-ra.

Felhasználói dokumentáció

1. Rendszerkövetelmények

Hardver követelmények:

-
-
-

Szoftver követelmények:

-

2. Telepítési útmutató

A weboldalon lévő Github Linkről lehet letölteni a játékot, nem igényel kicsomagolást. Mivel még nincsen szerverre feltöltve így kell hozzá egy localhost pl.: xampp. A játék mellett található lesz egy sqlconnect mappa is, ezt a mappát másolja vagy helyezze át az xampp vagy hasonló programnak a htdocs mappájába.

3. A program működése, felhasználói kézikönyv

A játék fejlesztéséhez a Unity Engine 2022.1.16f1 verzióját használtam. A Unity egy cross-platorm játék motor, amit a Unity Technologies fejleszt, elsődlegesen videójátékok készítéséhez és számítógépes szimulációkhoz használt szoftver gépekhez, konzolokhoz és mobil telefonhoz. A Unity egy több funkciós motor, ami támogatja a 2D és 3D grafikát, drag and drop funkciót és scriptírást c# nyelven.

A játék felhasználói profil alapján készít mentéseket a karakterekhez. A játék automatikusan ment kilépésnél és minden pálya után mikor visszakérül a játékos a Lobby-ba vagy újra indítja a pályát. Későbbiekben lehetőség lesz felhőalapú mentésre is.

Letöltés és telepítés után a játékot az EndlessSurvival.exe file-al lehet elindítani. A felhasználói profilba belépés után 3 opció lesz: A Play gomb, ami tovább viszi a játékost a Lobby-ba, options gomb megnyitja a játék beállításokat végül a Quit gombbal bezárhatja a játékot. Ha még nincs regisztrált fiókja vagy elfelejtette a jelszavát akkor a Registration gombra kattintva a játék megnyitja a weboldalt.

Irányítás: A játékos a WASD billentyűkkel tudja irányítani a karakter mozgását a shift lenyomásával átvált futás módba a karakter, az E-billentyű lenyomásával tud dashelni a játékos, ha feloldotta a képességet, a SPACE billentyű lenyomásával ugrani tud a játékos. A játékos az 1 es és 2 es billentyűk lenyomásával tud váltani az elsődleges és másodlagos fegyver közt. Későbbiekben a 3-9 es billentyűkre lesznek beállítva az aktív képességek.

Fegyverek: Bal egérgomb lenyomásával lehet tüzelni a fegyverekkel, az R billentyű megnyomásával lehet újra tölteni a tárat. Későbbiekben a jobb egérgomb lenyomásával lehet majd zoom-olni. A C

billentyű lenyomásával bizonyos fegyvereknél alternatív lövésmódot aktivál.

A Lobby-ban van lehetősége a játékosnak kiválasztani az elsődleges és másodlagos fegyvereit. Ha már lépet szintet akkor a Skills And Passives gombra rányomva tudja elkölteni a képességpontjait passzív és aktív képességekre. Későbbiekben a fegyverekre kattintva lehet majd fejleszteni őket. A

StartGame gombra nyomva tudja elindítani a játékot. Jelenleg csak egy játékmód van, ami egy survival pálya. Későbbiekben tervbe van pályák bővítése és többféle játékmód, multiplayer lehetőség és nehézségi szintek. A StartGame gomb lenyomása után a játékos bekerül a pályára, ahol túl kell élnie az egyre erősödő és egyre gyorsabban érkező ellenfeleket. Ha sikeresen túlél 5 percet akkor megjelenik egy kimenekítési pont, a játékos döntheti el, hogy mikor megy el a kimenekítési pontra, ahol 5 másodpercet túl kell élnie mielőtt kimenekítik a pályáról ez az idő alatt nem hagyhatja el a kimenekítési pontot vagy a számláló újra indul. Ha a játékos sikeresen túléli az 5 másodpercet akkor a játék átviszi a Summary-screen re a játékost, ahol a statisztikáját kiírja és megkapja a tapasztalati pontjait. Viszont, ha a játékosnak nem sikerül a kimenekítés és meghal akkor a játék átviszi a summary-screen re ugyan úgy viszont csak a tapasztalati pontok felét kapja csak meg. A Summary-screen ről lehet visszamenni a Lobby-ba vagy újraindítani a pályát.

HUD: A pályára belépve bal alsó sarokban jelenik meg a játékos életerreje(piros csík) és a shieldje(kék csík). Felül a képernyő közepén a számláló jelzi, hogy mennyi ideje van már a játékos a pályán, ha a játékos belép a kimenekítési pontra akkor bal felső sarokba jelenik meg a kimenekítési visszaszámláló. Jobb felső sarokban található a minimap a minimapp közepén megjelenő zöld pont jelzi a játékost, a piros pontok jelzik az ellenfelek helyét. Amikor a játék eléri az 5 percet akkor a minimappon megjelenik a kimenekítési pontot jelző kék karika és a játékos felett megjelenik egy zöld nyíl, ami a kimenekítési pont irányát jelzi.

Felhasznált források:

- Stack Overflow

youtube:

- Code Monkey
- Brackeys
- Board To Bits Games
- Gabriel Aguiar Prod
- Dave / GameDevelopment

Ábrajegyzék

1. ábra: Kereső asztali nézetben.....	7
2. ábra: Kereső mobil nézetben	7
3. ábra: Profil mobil nézetben lenyíló menüvel, 0 meccsel.....	8
4. ábra: Menüpontok.....	10
5. ábra: Regisztráció	10
6. ábra: Bejelentkezés	11
7. ábra: Jelszó helyreállítás.....	11
8. ábra: Profil beállítások menü	11
9. ábra: Profil szerkesztés.....	12
10. ábra: Jelszó megváltoztatása.....	13
11. ábra: Fiók törlése.....	13
12. ábra: Grafikus tábla	14
13. ábra: Rangsor.....	14
14. ábra: Lábléc	15