SZAKDOLGOZAT

|  |
| --- |
| Váci Szakképzési Centrum  Boronkay György Műszaki Technikuma és Gimnáziuma |
| **VIZSGAREMEK**  Arcalite |
| Konzulens: Készítők:  Gyombolainé Cserny Zsuzsanna Bakó Borka  Bali István Gábor  Katona Bálint  Vác  2025 |

Hallgatói nyilatkozat

Alulírottak, ezúton kijelentjük, hogy a szakdolgozat saját, önálló munkánk, és korábban még sehol nem került publikálásra. Szakdolgozatunk a Váci Szakképzési Centrum Boronkay György Műszaki Technikum és Gimnázium Szoftverfejlesztő és tesztelő technikus képzésén készítettük. Tudomásul vesszük, hogy szakdolgozatunkat a Váci Szakképzési Centrum Boronkay György Műszaki Technikum és Gimnázium tárolja.

Bakó Borka Bali István Gábor Katona Bálint

Konzultációs lap

Vizsgázók neve:

* Bakó Borka
* Bali István Gábor
* Katona Bálint

Szakdolgozat címe: **Arcalite**

Program által nyújtott szolgáltatások:

* Platformer játék
* A játékbeli haladás nyomonkövetése a weboldalon
* Tudástár a játékban lévő ellenfelekhez és tárgyakhoz
* E-mailek (visszaigazoló, jelszó módosítás)

|  |  |  |
| --- | --- | --- |
| **Sorszám** | **A konzultáció időpontja** | **Konzulens aláírása** |
| 1. | 2024. október 11. |  |
| 2. | 2024. november 15. |  |
| 3. | 2024. december 13. |  |
| 4. | 2025. január 17. |  |
| 5. | 2025. február 14. |  |
| 6. | 2025. március 14. |  |

A szakdolgozat beadható: A szakdolgozatot átvettem:

Vác, 2025. Vác, 2025.

Konzulens A szakképzést folytató

intézmény felelőse

Tartalomjegyzék

[Hallgatói nyilatkozat 3](#_Toc194394904)

[Konzultációs lap 4](#_Toc194394905)

[Tartalomjegyzék 5](#_Toc194394906)

[1. Témaválasztás 7](#_Toc194394907)

[2. Fejlesztői dokumentáció 8](#_Toc194394908)

[2.1. Fejlesztői környezet 8](#_Toc194394909)

[2.1.1. Visual Studio Code 8](#_Toc194394910)

[2.1.2. XAMPP 8](#_Toc194394911)

[2.1.3. Visual Studio 8](#_Toc194394912)

[2.1.4. GODOT 8](#_Toc194394913)

[2.2. Használt technológiák, nyelvek 9](#_Toc194394914)

[2.2.1. HTML 9](#_Toc194394915)

[2.2.2. CSS 9](#_Toc194394916)

[2.2.3. JavaScript 9](#_Toc194394917)

[2.2.4. jQuery 9](#_Toc194394918)

[2.2.5. JSON 9](#_Toc194394919)

[2.2.6. Bootstrap 10](#_Toc194394920)

[2.2.7. PHP 10](#_Toc194394921)

[2.2.8. HTTP, REST API 10](#_Toc194394922)

[2.2.9. MySQL 10](#_Toc194394923)

[2.2.10. C# 11](#_Toc194394924)

[2.2.11. Git, GitHub 11](#_Toc194394925)

[2.2.12. ChatGPT 11](#_Toc194394926)

[2.3. Adatszerkezet 12](#_Toc194394927)

[2.3.1. Alapadatok 12](#_Toc194394928)

[2.3.2. Kapcsolati ábra 12](#_Toc194394929)

[2.3.3. Táblák 13](#_Toc194394930)

[2.4. A weboldal felépítése 21](#_Toc194394931)

[2.4.1. A weboldal szerkezete és formázás 21](#_Toc194394932)

[2.4.2. Saját felugróablakok 21](#_Toc194394933)

[2.4.3. Az API és a weboldal közti kommunikáció 22](#_Toc194394934)

[2.4.4. Egyéb funkcionalitás a weboldalon 22](#_Toc194394935)

[2.5. Az API felépítése 23](#_Toc194394936)

[2.5.1. A cél 23](#_Toc194394937)

[2.5.2. Hibakezelés és közös metódusok 23](#_Toc194394938)

[2.5.3. A végpontok 24](#_Toc194394939)

[2.6. Az alkalmazás felépítése 35](#_Toc194394940)

[2.7. Tesztelés 36](#_Toc194394941)

[2.7.1. A játék tesztelése 36](#_Toc194394942)

[2.7.2. Frontend tesztelés 36](#_Toc194394943)

[2.7.3. Backend tesztelés 36](#_Toc194394944)

[2.8. Továbbfejlesztési lehetőségek 37](#_Toc194394945)

[3. Felhasználói dokumentáció 38](#_Toc194394946)

[3.1. A program rövid ismertetése 38](#_Toc194394947)

[3.2. A program feltelepítése 39](#_Toc194394948)

[3.2.1. A játék telepítése 39](#_Toc194394949)

[3.2.2. A weboldal üzembe helyezése 39](#_Toc194394950)

[3.2.3. Az adatbázis felállítása 39](#_Toc194394951)

[3.3. Használati útmutató 40](#_Toc194394952)

[3.3.1. A játék irányítása 40](#_Toc194394953)

[3.3.2. A weboldal használata 40](#_Toc194394954)

[4. Mellékletek 41](#_Toc194394955)

# Témaválasztás

Ide egy szép szöveg, hogy miért csináljuk ezt <3

# Fejlesztői dokumentáció

## Fejlesztői környezet

### Visual Studio Code

A Visual Studio Code (VSC) egy nyílt forráskódú kódszerkesztő alkalmazás. Nagy mértékben testre szabható, illetve rengeteg kiegészítő tartalom készült hozzá, melyekkel könnyen fejleszthetünk a legtöbb programozási nyelven. Mi a weboldal HTML vázához, formázásához, a funkcionalitást biztosító JavaScript szkriptekhez, illetve az adatbázissal kommunikáló PHP RestAPI fejlesztéséhez használjuk.

### XAMPP

A XAMPP egy nyílt forráskódú és platformfüggetlen webszerver-szoftver csomag. A csomag fő elemei – amelyek miatt mi is használjuk – az Apache szerver, PHP támogatás, a MariaDB adatbázis-motor, illetve a phpMyAdmin adatbázis-kezelőfelület.

### Visual Studio

A Visual Studio egy fejlesztői kódszerkesztő, amit kód szerkesztésére, debug-olására, de akár alkalmazásunk publikálására is használhatunk. Az IDE a standard kódszerkesztésen és debug-oláson felül tartalmaz fordítókat, kódkiegészítő, grafikus tervező, és még sok egyéb hasznos eszközzel, melyek elősegítik a fejlesztési élményünket. Ebben a projektben a játék kódjának megírásához használtuk.

### GODOT

A Godot egy játékfejlesztő motor, mellyel egy egységes felületen készíthetünk akár 2D-s, akár 3D-s platformfüggetlen játékokat. A motor egy mindent átfogó eszközcsomagot kínál, hogy a fejlesztők a játék fejlesztésére fókuszálhassanak, és ne kelljen már megoldott problémákat újból megoldani. A program teljesen ingyenes, és nyílt forráskódú, többnyire a közösség által fejlesztett.

## Használt technológiák, nyelvek

### HTML

A HTML (HyperText Markup Language) egy struktúrált leírónyelv, a weboldalak szerkezetének kialakításához használt szabvány. Általában együtt használatos a CSS és JavaScript nyelvekkel, mely három a webfejlesztés alapjaként szolgál. Míg a HTML a struktúra kialakítására használják, a CSS-t a formázásra, a JavaScript-et pedig a funkcionalitás kialakítására használják.

### CSS

A CSS (Cascading Style Sheets) egy stíluslapnyelv, mellyel a weboldalak megjelenítését és formázását alakíthatjuk ki. Lehetővé teszi, hogy a fejlesztők a formázást és a struktúrát elválasszák egymástól, továbbá, hogy a weboldalak különféle környezetekben is egységesen jelenjenek meg.

### JavaScript

A JavaScript egy dinamikus szkriptnyelv, amelyet általában weboldalak fejlesztéséhez használnak. Az egyik legelterjedtebb és legjelentősebb programozási nyelv a webfejlesztésben. Lehetővé teszi, hogy a weboldal interaktív legyen és dinamikusan reagáljon a felhasználó bemenetére. Emellett teljes értékű alkalmazásokat, játékokat is lehet benne készíteni, ilyenkor a weboldal egy megjelenítő szerepet vet fel. Mi a JavaScriptet a weboldal interaktív részének kialakításán felül az adatok API-ból történő lekérésére és betöltésére használjuk.

### jQuery

A jQuery egy könnyűsúlyú, gyors és keresztplatformos JavaScript könyvtár, amelyet főleg webfejlesztés során használnak. Célja, hogy egyszerűbbé és hatékonyabbá tegye a JavaScript alapú webfejlesztést. Népszerűsége és kényelme miatt sok webfejlesztő használja azáltalános feladatok megkönnyítésére és az alkalmazások gyors fejlesztésére.

### JSON

A JSON (JavaScript Object Notation) egy könnyen olvasható adatcsere formátum, amely gépi feldolgozásra alkalmas. A JSON adatokat egy szöveges formátumban ábrázolja, és adatstruktúrákat, például objektumokat és tömböket képes reprezentálni. Mi az API-ban vesszük hasznát, mivel mind a PHP, mind a JavaScript támogatja a formátumot, beépített funkcióival könnyen átalakítható és kezelhető a továbbított adat.

### Bootstrap

A Bootstrap egy ingyenes és nyílt forráskódú frontend keretrendszer weboldalak és webalkalmazások fejlesztéséhez. Elsősorban HTML, CSS és JavaScript alapú. A Bootstrap célja, hogy segítsen az egyenletes és gyorsan fejlődő webes projektek létrehozásában, minimalizálva a tervezési időt és növelve a projekt hatékonyságát. A keretrendszer egy nagyon népszerű eszköz a webfejlesztők körében, mivel lehetővé teszi az egyszerű és gyorsan fejlődő webes projektek létrehozását, miközben biztosítja a reszponzív design és a konzisztens megjelenés előnyeit.

### PHP

A PHP egy általános szerveroldali szkriptnyelv, amelyet dinamikus weblapok készítésére fejlesztették ki. A hagyományos HTML lapokkal ellentétben a kiszolgáló a PHP-kódot nem küldi el az ügyfélnek, hanem a kiszolgáló oldalán a PHP-értelmező motor dolgozza fel azt. A programokban lévő HTML elemek érintetlenül maradnak, de a PHP kódok lefutnak. A kódok végezhetnek adatbázis-lekérdezéseket, létrehozhatnak képeket, fájlokat olvashatnak és írhatnak, kapcsolatot létesíthetnek távoli kiszolgálókkal. A PHP-kódok kimenete a megadott HTML elemekkel együtt kerül az ügyfélhez.

### HTTP, REST API

A REST (Representational State Transfer) egy elterjedt architektúra az alkalmazás-programozási interfészek (API-k) tervezéséhez és megvalósításához. A REST API-k az erőforrásokat (adatokat, szolgáltatásokat) ábrázolják, és az HTTP protokollt használják a kliens és a szerver közötti kommunikációhoz. A REST API-k állapotmentesek, azaz a kliens kérésének tartalmaznia kell az összes szükséges információt. A REST API-k népszerűségét az egyszerűség, az egységes interfész és a skálázhatóság adja, amely lehetővé teszi a nagyobb terhelések kezelését.

### MySQL

A MySQL egy ingyenes, nyílt forráskódú relációs adatbázis-kezelő rendszer. A MySQL könnyen használható, széles körben elterjedt és skálázható, ami azt jelenti, hogy nagy adatmennyiségeket is képes hatékonyan kezelni. Az adatok egyszerű kezelése és a megbízhatósága miatt a MySQL-t széles körben használják az üzleti alkalmazásokban, weboldalakban, mobilalkalmazásokban, játékokban és még sok más területen.

### C#

A C# (C-sharp) egy modern, objektumorientált programozási nyelv. Elsősorban a .NET platformhoz kapcsolódik, és az egyik leggyakrabban használt nyelv a Windows alkalmazások, webalkalmazások és szolgáltatások fejlesztéséhez. A C#-t széles körben használják üzleti alkalmazások, játékok, webalkalmazások, adatszolgáltatások és még sok más fejlesztésére.

### Git, GitHub

A Git egy nyílt forráskódú, elosztott verziókezelő szoftver, vagy másképpen egy szoftverforráskód-kezelő rendszer, amely a sebességre helyezi a hangsúlyt. A GitHub ingyenes internetes szolgálatatás, amely a Git segítségével szoftverfejlesztési verziókövetés-szolgáltatást nyújt. A grafikus felület segítségével egyszerűen hozhatunk létre és kezelhetünk Git repository-kat.

### ChatGPT

A ChatGPT (Generative Pre-trained Transformer)[1] az OpenAI mesterséges intelligencia (MI) kutató laboratórium által kifejlesztett chatbot, mely a felhasználókkal való folyamatos kommunikáció automatizálása során értelmezőmodelleket használ, melyek segítségével a bevitt információkat azonnal interaktívan kezeli. Segítségével a sokszor hosszas keresést, interneten kutakodást lerövidíthetjük, egy-egy problémára másodperceken belül választ kaphatunk. Emellett kisebb algoritmusokat, mint tesztadatok generálása, vagy bizonyos adatok, esetleg kódrészletek helyességének ellenőrzése, is könnyebb, és még inkább gyorsabb elvégezni a ChatGPT segítségével.

## Adatszerkezet

Adatbázisunk célja a felhasználók adatainak, játékbeli állásuknak, illetve a játék általános információinak tárolása

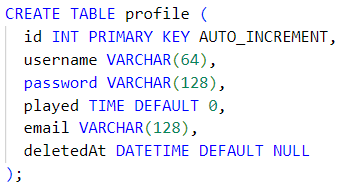
### Alapadatok

|  |  |
| --- | --- |
| MySQL kliens verzió | 10.4.28 - MariaDB |
| Adatbázis neve | arcalite |
| Tárolómotor | InnoDB |
| Alapértelmezett illesztés | UTF-8 |

### Kapcsolati ábra

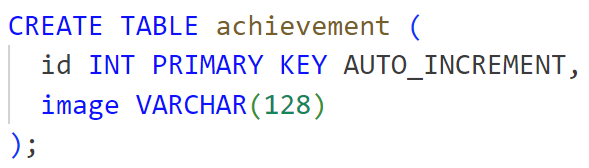
### Táblák

#### A profile tábla

A felhasználók regisztrációs adatait tárolja, illetve az összes játszott időt.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| id | Egész | PK |  |
| username | Szöveg |  | A fiók felhasználóneve |
| password | Szöveg |  | A fiók jelszava (PASSWORD függvénnyel hash-elve) |
| email | Szöveg |  | A fiók e-mail címe |
| played | Idő |  | Az összes játszott idő |
| deletedAt | Dátum és idő |  | A fiók törlésének időpontja |

#### Az achievement tábla

Az elérhető mérföldköveket tartja számon. Egyelőre nincs sokféle adat benne, későbbi fejlesztések miatt lett létrehozva.

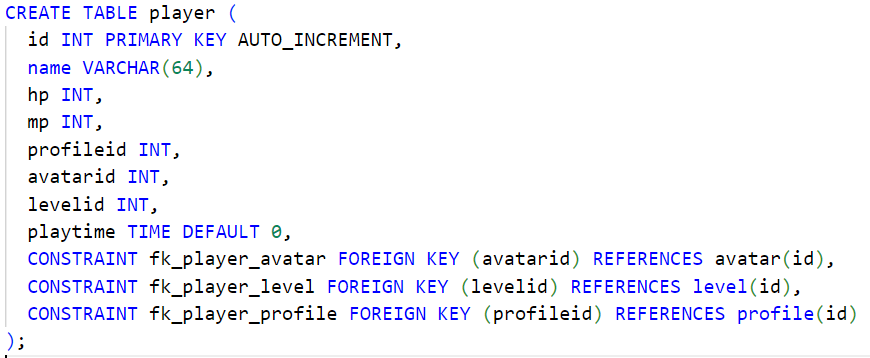
|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| id | Egész | PK |  |
| image | Szöveg |  | A mérföldkőhöz tartozó ikonkép helyét tárolja a szerveren. |

#### A proach tábla

Az adott felhasználó által elért mérföldköveket tartja számon.

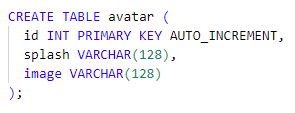
|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| profileid | Egész | PK, FK | A fiók kódja. Kapcsolat a *profile* táblával. |
| achievementid | Egész | PK, FK | A mérföldkő kódja. Kapcsolat az *achievement* táblával. |

#### A player tábla

A felhasználó által készített karakterek adatait tárolja.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| id | Egész | PK |  |
| name | Szöveg |  | A karakter neve |
| hp | Egész |  | A karakter életpontjainak maximális értéke |
| mp | Egész |  | A karakter manapontjainak maximális értéke |
| profileid | Egész | FK | A hozzátartozó fiók kódja. Kapcsolat a *profile* táblával. |
| avatarid | Egész | FK | A karakter avatárképének kódja. Kapcsolat az *avatar* táblával. |
| levelid | Egész | FK | A legutoljára elért szint kódja. Kapcsolat a *level* táblával. |
| playtime | Idő |  | A karakterrel játszott idő. |

#### Az avatar tábla

A karaktereknek választható avatár képek elérési útját tárolja.

|  |  |  |  |
| --- | --- | --- | --- |
| Mezőnév | Típus | Kulcs | Leírás |
| id | Egész | PK |  |
| splash | Szöveg |  | Az avatárhoz tartozó splash art elérési útja a szerveren. |
| image | Szöveg |  | Az avatárhoz tartozó kép elérési útja a szerveren. |

#### A level tábla

A szintekhez tartozó képek elérési útját tárolja.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| id | Egész | PK |  |
| image | Szöveg |  | A szinthez tartozó kép elérési útja a szerveren. |

#### Az enemy tábla

A játékban lévő ellenfelek adatait tárolja.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| id | Egész | PK |  |
| hp | Egész |  | Az ellenfél életpontjainak maximális értéke. |
| image | Szöveg |  | Az ellenfélhez tartozó kép elérési útja a szerveren. |

#### Az enemplay tábla

A karakter által már felfedezett ellenfeleket tárolja. Az adott karakter „csak ezeket ismeri”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| playerid | Egész | PK, FK | A karakter kódja. Kapcsolat a *player* táblával. |
| enemyid | Egész | PK, FK | Az ellenfél kódja. Kapcsolat az *enemy* táblával. |

#### Az item tábla

A játékban megszerezhető tárgyak adatait tárolja

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| id | Egész | PK |  |
| image | Szöveg |  | A tárgyhoz tartozó kép elérési útja a szerveren. |

#### Az itemplay tábla

A karakter által már megszerzett/felfedezett tárgyakat tárolja. Az adott karakter „csak ezeket ismeri”.

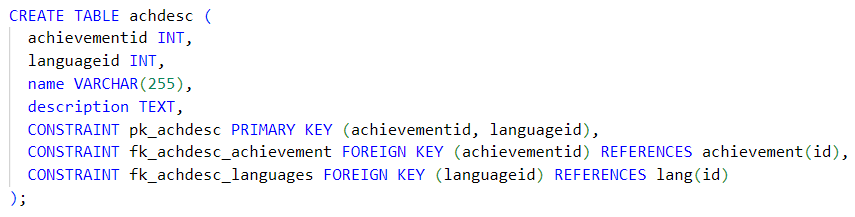
|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| playerid | Egész | PK, FK | A karakter kódja. Kapcsolat a *player* táblával. |
| itemid | Egész | PK, FK | A tárgy kódja. Kapcsolat az *item* táblával. |

#### A lang tábla

Az elérhető nyelveket tárolja.

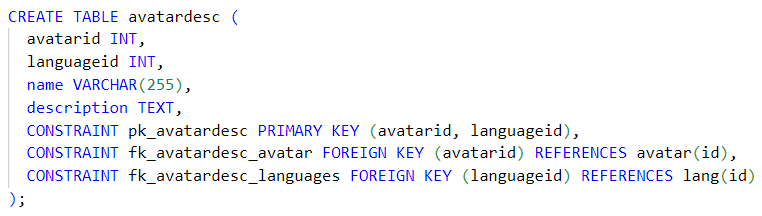
|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| id | Egész | PK |  |
| name | Egész |  | A nyelv megnevezése |

#### Az achdesc tábla

A mérföldkövek nevét és leírását tárolja adott nyelveken.

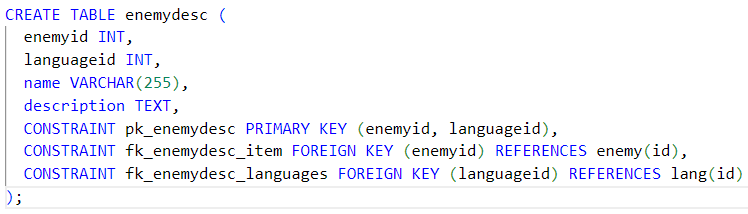
|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| achievementid | Egész | PK, FK | A mérföldkő kódja. Kapcsolat az *achievement* táblával. |
| languageid | Egész | PK, FK | A nyelv kódja. Kapcsolat a *lang* táblával. |
| name | Szöveg |  | A mérföldkő megnevezése az adott nyelven. |
| description | Szöveg |  | A mérföldkő leírása az adott nyelven. |

#### Az avatardesc tábla

Az avatárok nevét és leírását/háttértörténetét tárolja adott nyelveken.

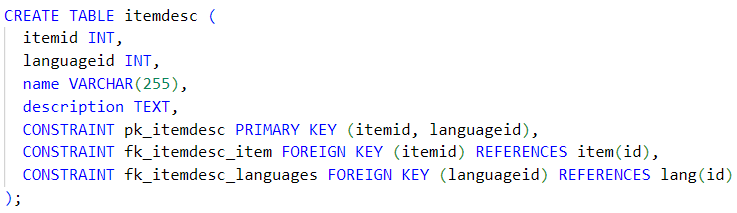
|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| avatarid | Egész | PK, FK | Az avatár kódja. Kapcsolat az *avatar* táblával. |
| languageid | Egész | PK, FK | A nyelv kódja. Kapcsolat a *lang* táblával. |
| name | Szöveg |  | Az avatár megnevezése az adott nyelven. |
| description | Szöveg |  | Az avatár leírása az adott nyelven. |

#### Az enemydesc tábla

Az ellenfelek nevét és leírását tárolja adott nyelveken.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| enemyid | Egész | PK, FK | Az ellenfél kódja. Kapcsolat az *enemy* táblával. |
| languageid | Egész | PK, FK | A nyelv kódja. Kapcsolat a *lang* táblával. |
| name | Szöveg |  | Az ellenfél megnevezése az adott nyelven. |
| description | Szöveg |  | Az ellenfél leírása az adott nyelven. |

#### Az itemdesc tábla

A tárgyak nevét és leírását tárolja adott nyelveken.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| itemid | Egész | PK, FK | A tárgy kódja. Kapcsolat az *item* táblával. |
| languageid | Egész | PK, FK | A nyelv kódja. Kapcsolat a *lang* táblával. |
| name | Szöveg |  | A tárgy megnevezése az adott nyelven. |
| description | Szöveg |  | A tárgy leírása az adott nyelven. |

#### A saves tábla

Az adott karakter mentési fájljait tárolja el.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezőnév** | **Típus** | **Kulcs** | **Leírás** |
| playerid | Egész | PK, FK | A karakter kódja. Kapcsolat az *player* táblával. |
| time | Dátum és idő |  | A mentés időpontja. |
| save | Longblob |  | A mentési állomány. |

## A weboldal felépítése

### A weboldal szerkezete és formázás

A weboldal 5 darab HTML-fájlból áll, viszont a formázás egy központi CSS fájlban, a *style.css*-ben található, illetve a formázáshoz kapcsolódó JavaScript függvények a *script* mappában a *styleScript.js*-be kerültek.

A weboldalon használt színek többségét CSS-változókkal adjuk meg, és ezek segítségével alakítottunk ki egy sötét és világos témát az oldalnak. Ezt az oldal tetején a menüsávon található kapcsolóval állíthatjuk át. A kapcsoló a háttérben meghívja a *themeChange* függvényt, mely egy cookie-ban eltárolja a jelenleg kiválasztott témát, majd a *setTheme* függvény meghívásával beállítja azt. Az oldal *body* eleme kap egy *bsTheme* adattagot, mely értéke a jelenlegi téma, és amely alapján állítja be a CSS a változóit.

### Saját felugróablakok

A weboldalon hibaüzenetek megjelenítésére, illetve a profilműveletek szükséges kommunikációra saját készítésű felugróablakokat használunk. Maga a felugróablak egy div, benne egy címmel, paragrafussal, szöveges bemeneti mezővel és három gombbal. A közös *common.js* szkriptfájlban találhatjuk a felugróablak működéséhez és használatához szükséges függvényeket:

* **showPopup**: Az oldal tetejére csúsztatja a felugróablakot.
* **closePopup**: Az oldal teteje fölé csúsztatja („elrejti”) a felugróablakot.
* **setupPopup**: A felugróablak mindig megjelenő részeinek beállítását végzi el. Ezek a címszöveg, az üzenet, a „*Mégsem*” gomb beállítása, illetve a többi gomb funkciójának reset-elése.
* **setAlert**: Egy bemenet nélküli felugróablak beállítását végzi el. Paraméterekből beállítja a „*Rendben*” gomb tényleges szövegét, funkcióját, illetve elrejti a nemhasznált részeket.
* **setPrompt**: Egy bemenetet váró felugróablak beállítását végzi el. Paraméterekből beállítja az „*Elküld*” gomb tényleges szövegét, funkcióját, illetve megjeleníti a szükséges elemeket, és elrejti a nemhasználtakat. A paraméterek között szerepel a bemenet helyességét ellenőrző függvény, illetve a helytelen bemenetre visszaadott hibaüzenet is.
* **Alert**: A bemenet nélküli felugróablak használatát lehetővé tevő függvény. Meghívja a megjelenítést, beállítást és az esetleges előző ablak bezárását is.
* **Prompt**:A bemenetet váró felugróablak használatát lehetővé tevő függvény. Meghívja a megjelenítést, beállítást és az esetleges előző ablak bezárását is.

### Az API és a weboldal közti kommunikáció

Az API és a kliens közti kommunikációt JavaScript-tel kezeljük, a jQuery *$.ajax()* függvényének segítségével.

Több végpontnak is szüksége van információkra, mint a felhasználó azonosítója, neve, vagy a megjelenítés nyelvének azonosítója (jelen állapotban mindig 1 (magyar)). Ezeket az oldal bejelentkezéstől kezdve cookie-kban tárolja, és elérésükhöz a common.js állományban áll rendelkezésünkre a *getCookie()* függvény, mely a megadott nevű cookie értékét, illetve ha nincs ilyen nevű cookie, null értéket ad vissza.

A tényleges API kérések függvényei a weboldalakhoz tartozó szkriptekben találhatóak.

|  |  |
| --- | --- |
| **Fájl** | **Feladat** |
| index.js | A főoldalon megjelenő karakterek adatait kéri le és tölti be. |
| login.js | A bejelentkezéshez, regisztrációhoz és új jelszó kéréshez szükséges bemenet ellenőrzéseket és API-hívásokat intézi el. |
| profile.js | A profil oldalon megjelenő adatokat kéri le és tölti be. |
| ranking.js | A rangsor táblákba kerülő adatokat kéri le és tölti be. |
| wiki.js | A lexikon adatait kéri le és tölti be. |

### Egyéb funkcionalitás a weboldalon

A menüsávon a bejelentkezésfüggő elemek beállítását elvégző függvények a *common.js* fájlban találhatóak. Itt ellenőrzi az oldal, hogy van-e *userid* eltárolva: ha nincs, a *Lexikon* menüpont nem lesz kattintható, illetve a *Bejelentkezés* menüpont fog megjelenni; ha van *userid*, a *Lexikon* elérhető lesz, illetve a *Bejelentkezés* helyett a *Profil* oldalra tudunk eljutni a menüből.

Kijelentkezéskor a már sokszor említett cookie-kat egyszerűen töröljük (felhasználó azonosító, -név), erre a *common.js* fájlban található a *logout* függvény.

## Az API felépítése

### A cél

A weboldal, illetve a játék és az adatbázis közötti kommunikációt hivatott az API ellátni. Az API PHP nyelven íródott, a visszatérési adatokat JSON formátumban adja vissza.

### Hibakezelés és közös metódusok

Mivel minden végpont meghívásakor az első lépés a bemenő adatok ellenőrzése, az adatbázis kapcsolat felállítása, illetve hibák esetén hasznos visszajelzés küldése, ezért ezek két „közösen használt” PHP fájlba kerültek.

A *config.php* a bemenő adatok validálására (*hasProperFields* és *checkValidity*) és a visszatérési értékek visszaadására (*ReturnMessage*, *ReturnResult*, *ReturnQuery*) való függvényeket tartalmazza. Utóbbiak mind valamiféle adatot (akár egy üzenetet, akár adatbázisból lekért adathalmazt), mind egy HTTP válaszkódot is visszaadnak, így hibaüzenetek küldésére is ezeket használják a végpontok.

A *connection.php* az adatbázis kapcsolatot hozza létre a *mysqli* osztály példányosításával. Ezzel központilag, egy helyen lehet megadni a kapcsolathoz szükséges információkat (*$CONFIG* tömb), mint az adatbázis neve, a szerver címe stb. Kapcsolódási hiba esetén hibát dob vissza, melyben megadja az SQL hibakódot és -üzenetet is.

#### A config.php által küldhető hibakódok

|  |  |  |
| --- | --- | --- |
| Válaszkód | Üzenet | Megjegyzés |
| 500 | Hiba az SQL lekérdezésben: (Hibakód: …) … |  |
| 405 | Hiba az API-hívásban. | Hibás metódus esetén |
| 400 | Hiba az API-hívásban. | Hibás bemenet esetén |

Ezeket a hibakódokat a többi végpont is adhatja, ott nincsenek ezek külön felsorolva.

### A végpontok

#### login.php

Leírás: A felhasználó bejelentkeztetését végző végpont.

Metódus: **GET**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| email | szöveg | A bejelentkezéshez használt e-mail cím. |
| password | szöveg | A fiókhoz tartozó jelszó. |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| id | egész | A fiók egyedi azonosítója. |
| username | szöveg | A fiókhoz tartozó felhasználónév. |
| email | szöveg | A fiókhoz tartozó e-mail cím. |

Hibakódok:

|  |  |
| --- | --- |
| Válaszkód | Üzenet |
| 401 | Nincs ehhez az e-mail címhez regisztrált fiók! |
| 401 | Hibás jelszó! |
| 401 | Több fiók bejelentkezési adatai egyeznek! |

#### register.php

Leírás: Az újonnan regisztráló felhasználó adatait rögzítő végpont.

Metódus: **POST**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| username | szöveg | A fiókhoz tartozó felhasználónév |
| email | szöveg | A fiókhoz tartozó e-mail cím. |
| password | szöveg | A fiókhoz tartozó jelszó. |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| code | egész | HTTP válaszkód |
| message | szöveg | A válaszüzenet |

Hibakódok:

|  |  |
| --- | --- |
| Válaszkód | Üzenet |
| 409 | Már van fiók regisztrálva ehhez az e-mail címhez! |
| 409 | Már van fiók ilyen felhasználónévvel! |
| 500 | Hiba az adatok feltöltése közben. |

#### email\_exists.php

Leírás: Ellenőrzi, hogy egy adott e-mail címhez tartozik-e aktív felhasználó.

Metódus: **GET**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| email | szöveg | A fiókhoz tartozó e-mail cím. |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| exists | logikai | Létezik-e a megadott e-mail címhez tartozó fiók. |
| username | szöveg | A fiókhoz tartozó felhasználónév. Ha az *exists* hamis, NULL. |
| email | szöveg | A fiókhoz tartozó e-mail cím. Ha az *exists* hamis, NULL. |

Hibakódok:

A *config.php* által adhatók.

#### new\_password.php

Leírás: Egy fiók új jelszavának beállítását elvégző végpont.

Metódus: **POST**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| email | szöveg | A fiókhoz tartozó e-mail cím. |
| password | szöveg | A beállítandó jelszó. |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| code | egész | HTTP válaszkód |
| message | szöveg | A válaszüzenet |

Hibakódok:

|  |  |
| --- | --- |
| Válaszkód | Üzenet |
| 500 | Hiba a frissítés közben! Az új jelszó nem került beállításra! |

#### profile.php

Leírás: A bejelentkezett felhasználó karaktereinek adatait kéri le.

Metódus: **GET**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| userid | egész | A lekérni kívánt fiók egyedi azonosítója. |
| langid | egész | A lekért adatok nyelve. |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| - | tömb[karakteradat] | Egy tömb a lekért karakteradatokkal. |

*A karakteradatok felépítése:*

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| id | egész | A karakter egyedi azonosítója. |
| name | szöveg | A karakter neve. |
| hp | egész | A karakter maximum életpontja. |
| mp | egész | A karakter maximum manapontja. |
| level | egész | A karakter legmagasabb elért szintje. |
| playtime | idő | A karakterrel játszott idő. |
| image | szöveg | A karakterhez tartozó splashart. |

Hibakódok:

A *config.php* által adhatók.

#### update.php

Leírás: Egy megadott felhasználó e-mail címét vagy felhasználónevét megváltoztatja a megadottakra.

Metódus: **POST**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| userid | egész | A fiókhoz tartozó egyedi azonosító |
| username | szöveg | Az új felhasználónév. Ha üres szöveg, ignorálja a végpont. |
| email | szöveg | Az új e-mail cím. Ha üres szöveg, ignorálja a végpont. |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| username | szöveg | A frissített rekordhoz tartozó felhasználónév. |
| email | szöveg | A frissített rekordhoz tartozó e-mail cím. |

Hibakódok:

A *config.php* által adhatók.

#### delete\_profile.php

Leírás: Megadott fiók törlését elvégző végpont.

Metódus: **POST**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| userid | egész | A fiókhoz tartozó egyedi azonosító |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| code | egész | HTTP válaszkód |
| message | szöveg | A válaszüzenet |

Hibakódok:

|  |  |
| --- | --- |
| Válaszkód | Üzenet |
| 500 | Hiba a törlés során! A profil nem került törlésre! |

#### index.php

Leírás: A játékban játszható karakterek adatainak lekérése.

Metódus: **GET**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| langid | egész | A lekért adatok nyelve |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| - | tömb[karakteradat] | Egy tömb a lekért karakteradatokkal. |

*A karakteradatok felépítése:*

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| id | egész | A karakter egyedi azonosítója. |
| image | szöveg | A karakterhez tartozó avatárkép elérési útja a szerveren. |
| name | szöveg | A karakter neve. |
| description | szöveg | A karakterről szóló leírás. |

Hibakódok:

A *config.php* által adhatók.

#### ranking.php

Leírás: A ranglistákhoz szükséges adatokat kéri le.

Metódus: **GET**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| type | szöveg | A ranglista szempontja. |
| langid | egész | A lekért adatok nyelve. |

Kimenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| - | tömb[adat] | Egy tömb a lekért adatokkal. |

*Az adatok felépítése, ha a type „Profile”:*

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| Felhasználónév | szöveg | A fiókhoz tartozó felhasználónév. |
| Játékidő | idő | A játékos összegzett játékideje. |
| Elért\_mérföldkövek | egész | A játékos által elért mérföldkövek száma. |
| Végigjátszások | egész | A játékos végigvitt játékainak száma. |

*Az adatok felépítése, ha a type „GameThrough”:*

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| Karakter | szöveg | A karakter neve. |
| Profil | szöveg | A karakterhez tartozó fiók neve. |
| Avatár | szöveg | A karakterhez tartozó avatár neve. |
| Játékidő | idő | A karakterrel játszott idő. |
| Elért\_szint | egész | A karakterrel elért legmagasabb szint. |
| Felfedezett\_ellenfelek | egész | A karakterrel felfedezett ellenfelek száma. |
| Felfedezett\_tárgyak | egész | A karakterrel felfedezett tárgyak száma. |

Hibakódok:

A *config.php* által adhatók.

#### wiki.php

Leírás: A lexikonhoz szükséges adatokat kéri le.

Metódus: **GET**

Bemenet:

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| userid | egész | A lekérni kívánt fiók egyedi azonosítója. |
| langid | egész | A lekért adatok nyelve. |
| request\_type | szöveg | A lekérés típusa. |

Kimenet:

*Ha a request\_type „STATISTICS”:*

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| enemyFound | egész | A játékos által már felfedezett ellenfelek száma. |
| enemyAll | egész | Az ellenfelek összdarabszáma. |
| itemFound | egész | A játékos által már felfedezett tárgyak száma. |
| itemAll | egész | A tárgyak összdarabszáma. |

*Egyébként:*

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| - | tömb[adat] | Egy tömb a lekért karakteradatokkal. |

*Az adatok felépítése, ha a request\_type „ENEMY”:*

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| id | egész | Az ellenfél egyedi azonosítója. |
| name | szöveg | Az ellenfél neve. |
| hp | egész | Az ellenfél maximum életpontja. |
| desc | idő | Az ellenfél leírása. |
| image | szöveg | Az ellenfélhez tartozó kép. |

*Az adatok felépítése, ha a request\_type „ITEM”:*

|  |  |  |
| --- | --- | --- |
| Megnevezés | Típus | Leírás |
| id | egész | A tárgy egyedi azonosítója. |
| name | szöveg | A tárgy neve. |
| desc | idő | A tárgy leírása. |
| image | szöveg | A tárgyhoz tartozó kép. |

Hibakódok:

|  |  |
| --- | --- |
| Válaszkód | Üzenet |
| 400 | Hiba az API-hívásban. |

## Az alkalmazás felépítése

A játék forráskódjának általunk írt fájljai a játék mappáján belül a *Script* mappában találhatóak. A feladat megszabásai és tanulmányaink miatt C# nyelven írtuk a forráskódot. A *cs* fájlok a *Script*-en belül feladatuk szerint csoportosítva vannak. A mappák feladatait az alábbi táblázat szemlélteti.

|  |  |
| --- | --- |
| **Mappa** | **Benne lévő fájlok feladata** |
| Game | A játék elemeinek, szereplőinek működését biztosító szkriptek. |
| Globals | Minden más által elérhető funkciók szkriptjei. Ilyen pl. az adatbázis-kapcsolatot létesítő fájl vagy a mentést és betöltést végző fájl. |
| Maps | A pályák váltásának kezelését végző szkriptek. |
| Mechanics | A játékban található mechanikák és logikák kezelését végző szkriptek. |
| Menus | A különféle menük és egyéb GUI elemek kezelését végző szkriptek. |

A szkriptfájlok mellett egy *.cs.uid* fájl is található, mely a GODOT játékmotor miatt kell, mivel ő ezalapján ismeri fel és használja a *.cs* fájlokat.

### MainNode.cs

A GODOT-ban a játék mindegyik eleme egy úgy nevezett *Node*. Ez gyakorlatban egy *Node2D* nevű osztályból való származottságot jelenti. A *MainNode* a játék indításakor hozza létre a fő *Node*-ot, mely alá a játék többi eleme tartozni fog. Emellett a töltőképernyő és a főmenü *Node*-jait is legenerálja, illetve elindítja az adatok betöltését.

### A Game mappa

#### BasicProjectile.cs

#### ChargeProjectile.cs

#### Item.cs

#### Player.cs

#### SpellOracle.cs

### A Game\Enemies mappa

### A Game\Enemies\Bosses mappa

### A Game\Enemies\Projectiles mappa

### A Globals mappa

#### ConfigFileHandler.cs

#### DBConnector.cs

A fájlnak alapvetően három feladata van: Az első az adatbázissal való kapcsolat létrehozása. Az ehhez szükséges adatok (szervercím, port, adatbázisnév stb.) az osztály változóiban vannak eltárolva, melyek segítségével épül fel a connection string. Ez alapján az osztály a MysqlConnection osztály példányosításával létrehozza a kapcsolatot. A másik két funkció is ezt használja.

A második feladat a bejelentkezéskor a felhasználó adatainak lekérése. Lekérésre kerül a játékos azonosítója és felhasználóneve, illetve a külön játékmeneteinek adatai (név, játszott idő, stb.) és a hozzátartozó legutolsó mentés. A lekért adatok eltárolására a fájlban található két struct segítségevel történik. A UserData a játékos adatainak, illetve a CharacterData a játékos mentéseinek tárolására szolgál.

Az utolsó feladat az újonnan mentett játék feltöltése. \*\*\*Itt kipótolod mert a függvények random helyre megírása és máshol meghívása átlátható 🫶.

### A Maps mappa

### A Mechanics mappa

### A Menus mappa

## Tesztelés

A szoftverfejlesztés nélkülözhetetlen lépése a tesztelés. Ezzel biztosíthatjuk, hogy a fejlesztett szoftver megbízhatóan működjék, illetve megfeleljen a célnak és a megrendelő elvárásainak.

A tesztelés elvégzését sokféleképpen megtehetjük, kezdve az egyszerű manuális tesztekkel, ahol mi magunk találunk ki tesztadatokat és teszteljük ki őket, egészen a tesztelésre készített programokig, melyek akár maguknak generálnak tesztadatokat, majd ellenőrzik, hogy az elvárt kimenetet kapták-e. Mi a projekt jellege és keretei alapján a manuális tesztelés mellett döntöttünk.

### A játék tesztelése

### Frontend tesztelés

### Backend tesztelés

Az API teszteléséhez a ReqBin tesztelőalkalmazást használtam. A ReqBin egy ingyenes online tesztelőeszköz, mely egy böngészős bővítménnyel a helyi szerverre is tud API-hívásokat küldeni. A kezelőfelületen egyszerűen megadható az URL, a meghívás típusa és törzse, illetve egyből látjuk a kérésre kapott választ, így könnyen és gyorsan tudjuk az API-nkat tesztelni. Fontos megjegyezni, hogy az oldal tényleges API-hívásokat intéz, így azok ténylegesen módosíthatják az adatainkat – amennyiben ilyen funkciót is kínál az API-nk.

Backend: restapi program? would be easiest

frontend és játék: kézzel

képek az eredményekről!!!

## Továbbfejlesztési lehetőségek

Példa hibás bemeneti adatokra

Példa hibás metódusra (GET helyett POST)

# Felhasználói dokumentáció

## A program rövid ismertetése

## A program feltelepítése

### A játék telepítése

### A weboldal üzembe helyezése

### Az adatbázis felállítása

## Használati útmutató

### A játék irányítása

### A weboldal használata

# Mellékletek