Békéscsabai SZC Nemes Tihamér Technikum és Kollégium

Szakképesítés megnevezése: Szoftverfejlesztő

Azonosító száma: 54 213 05

**Off time**

Konzulens Neve Készítette: Varga András Bendegúz

Gajda Judit 5/13.B

Oktatási azonosító: 72169956207

Békéscsaba, 2021/2022

Tartalom

[1. Bevezetés 3](#_Toc97546278)

[1.1 Köszönetnyilvánítás 3](#_Toc97546279)

[1.2 Indoklás 3](#_Toc97546280)

[2. Fejlesztői dokumentáció 3](#_Toc97546281)

[2.1 Fejlesztői környezet 3](#_Toc97546282)

[2.1.1 Programozási nyelv és környezet 3](#_Toc97546283)

[2.1.2 Játékmotor 3](#_Toc97546284)

[2.2 Adattárolás 4](#_Toc97546285)

[2.2.1 Adatbázisszerkezet 4](#_Toc97546286)

[2.2.2 A users tábla 4](#_Toc97546287)

[2.2.3 A saves tábla 4](#_Toc97546288)

[2.2.4 A wins tábla 4](#_Toc97546289)

[2.2.5 A questions tábla 5](#_Toc97546290)

[2.2.6 Fájlrendszer 5](#_Toc97546291)

[2.2.7 Mentés menete 5](#_Toc97546292)

# 1. Bevezetés

## 1.1 Köszönetnyilvánítás

Ezúton szeretnék köszönetet nyilvánítani tanáraimnak, akik itt létem alatt előkészítették tudásomat, és a programozás iránti érdeklődést felkeltették bennem.

## 1.2 Indoklás

Azért választottam ezt a témát, mert a programozás mindig is közelebb állt hozzám, mint a webfejlesztés, és már régóta szerettem volna saját játékprogramot készíteni. Már a záródolgozat feljövetele előtt célom volt az, hogy C# nyelven megtanuljak programozni, de ebben a legtöbb esetben limitált és unalmas online talált feladatok nem sokat segítettek. Ezzel ellenben, egy logikai fejtörő/kvízjáték elkészítése megfelelő kihívásnak tűnt a képességeim fejlesztéséhez. Sok hasonló fejlesztő és ismeretterjesztő tartalom létezik már, de a választék bővítése sosem árt, és egy interaktív játékprogram könnyebben megtartja az emberek figyelmét mint például egy weboldal.

# 2. Fejlesztői dokumentáció

## 2.1 Fejlesztői környezet

### 2.1.1 Programozási nyelv és környezet

A program C# nyelven írtam, hiszen a Unity ezt használja. A nyelv alapból is jó választás lett volna, mivel széleskörűen támogatott, objektum orientált és sok hasznos alapkönyvtárral rendelkezik. Mindezek mellett egy könnyű nyelv, nem olyan gépközeli mint például a C++, a program futásával kapcsolatos dolgokat kezeli helyettünk, de futási sebessége mégis elég gyors. Fejlesztési környezetnek Visual Studio-t használtam, ami a Unity-vel összekapcsolódva még könnyebbé teszi e fejlesztést.

### 2.1.2 Játékmotor

Játékmotornak a Unity-t választottam, mivel 2D játékok készítéséhez kitűnő, és a scriptekhez C# nyelvet használ, amiben már van egy kis jártasságom. A program könnyen kezelhető, és alapból kezeli a legtöbb szükséges szimulációt, nekem csak a játékelemekkel és azok interakciójának logikájával kellett foglalkoznom.

## 2.2 Adattárolás

### 2.2.1 Adatbázisszerkezet

A felhasználói adatok és kvíz kérdések tárolására a program a GameDB nevű adatbázist használja. Minden adatbázissal kapcsolatos operációért a DatabaseManager osztály felel. Minden az összes többi osztály, aminek szüksége van az adatbázisra ezen keresztül szerzi meg a szükséges adatokat.

### 2.2.2 A users tábla

Ez a tábla a felhasználók tárolásáért felelős. A felhasználók nevén és hashelt jelszaván kívül még azt is tárolja, hogy hányadik pályánál tartanak.

### 2.2.3 A saves tábla

A saves tábla a mentéseket tartja számon. A userID oszlop tárolja, hogy melyik felhasználóhoz tartozik a mentés. A levelName a pályanevet tárolja, a fileName pedig a mentésfájl kiterjesztés nélküli nevét. Ezek mellett tárolja a nehézséget, mozgások számát, eltelt időt és a mentés idejét.

### 2.2.4 A wins tábla

A felhasználó pályateljesítéseinek adatait tárolja. Ugyanúgy, mint a saves táblánál, a felhasználóra a userID idegen kulccsal hivatkozunk. Tárolja a nehézséget, lépések számát, pályanevet, eltelt időt és a teljesítés idejét is.

### 2.2.5 A questions tábla

Ez a tábla tárolja a kvízeket. Egy kvíz rekord áll egy azonosítóból, nehézségből, kvíz szövegből, jó válaszból és három rossz válaszból. Ezek mellett azt is tárolja, hogy a rekordot felhasználó hozta-e létre.

### 2.2.6 Fájlrendszer

A program az adatbázist és a mentésmappát a felhasználó AppData/LocalLow mappán belüli saját magának létrahozott mappájában tárolja. Az összes felhasználó mentéseit ugyanebben a mentés mappában tárolja, és ezeket az adatbázis saves táblájának segítségével kezeli.

### 2.2.7 Mentés menete

A mentés folyamata akkor kezdődik, amikor a felhasználói felületről egy gombnyomással meghívjuk a GameController objektum Save metódusát. Először címkéjük alapján megkeressük az összes menteni való objektumot, majd egy listába eltároljuk az összes objektum ISaveable interfészének referenciáit.



Ezután példányosítunk egy új SaveData objektumot, és az összes menteni való objektummal hozzáadatjuk a saját adatát.



Az így elkészült SaveData objektumot ezután szerializáljuk, és a FileManager objektum segítségével fájlba írjuk.

# 2.3 Adatbázis kezelés

### 2.3.1