**ZÁRÓDOLGOZAT**

**Diczenty Áron**

**szoftverfejlesztő**

**BAJAI SZC TÜRR ISTVÁN TECHNIKUM**

**Szoftverfejlesztő**

ZÁRÓDOLGOZAT

Air race

**Diczenty Áron**

2024

**NYILATKOZAT A ZÁRÓDOLGOZATRÓL**

Alulírott *Diczenty Áron* (név) tanuló

**kijelentem, hogy**

*Air Race* című záródolgozatomat (nyomtatott és elektronikus formában) a Bajai SZC Türr István Technikum oktatói és tanulói:

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

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

Baja, 2024

aláírás

Tartalomjegyzék

[Tartalomjegyzék 4](#_Toc165271674)

[1. Bevezetés 6](#_Toc165271675)

[1.1. A dolgozat célkitűzései 6](#_Toc165271676)

[1.2. A dolgozat felépítése 6](#_Toc165271677)

[2. Felhasználói dokumentáció 7](#_Toc165271678)

[2.1. Rendszerkövetelmények 7](#_Toc165271679)

[2.2. Játék indítása 7](#_Toc165271680)

[2.3. Játék használata 8](#_Toc165271681)

[2.3.1. Felhasználói felület használata 9](#_Toc165271682)

[Főmenü 9](#_Toc165271683)

[Regisztráció 10](#_Toc165271684)

[Belépés 12](#_Toc165271685)

[2.3.2. Játékmechanika 16](#_Toc165271686)

[3. Fejlesztői dokumentáció 17](#_Toc165271687)

[3.1. Fejlesztés során használt szoftverek és kiegészítők. 17](#_Toc165271688)

[3.1.1. Visual Studio 2017 17](#_Toc165271689)

[3.1.2. XAMPP 8.2.4 17](#_Toc165271690)

[3.1.3. Unity 2022.3.21f1 18](#_Toc165271691)

[3.1.4. Blender 3.6 18](#_Toc165271692)

[3.1.5. Verziókövetés 18](#_Toc165271693)

[3.2. C# scriptek 19](#_Toc165271694)

[3.2.1. Adatbázis kezelése 20](#_Toc165271695)

[3.2.2. Reg.cs 20](#_Toc165271696)

[3.2.3. Login.cs 21](#_Toc165271697)

[3.2.4. GameManager.cs 23](#_Toc165271698)

[3.2.5. Kikepzes.cs 24](#_Toc165271699)

[3.2.6. verseny.cs 25](#_Toc165271700)

[3.2.7. Eredmenyek.cs 25](#_Toc165271701)

[3.2.8. PlaneController.cs 26](#_Toc165271702)

[3.3. Adatbázis 27](#_Toc165271703)

[3.4. 3D modellezés 28](#_Toc165271704)

[4. Összegzés 30](#_Toc165271705)

[4.1. Vélemény 30](#_Toc165271706)

[4.2. Tovább fejlesztési lehetőségek 30](#_Toc165271707)

[5. Ábrajegyzék 31](#_Toc165271708)

[6. Források megjelölése 32](#_Toc165271709)

# Bevezetés

A játék neve „Air Race”. Azért neveztem el így, mert ez a név utal a hidegháborúra, ezen belül a légi harcieszközök fejlesztési versenyére.

A játékban egy MIG-21 es típusú vadászrepülőgépet vezethetünk. Ez egy második generációs vadászrepülőgép, ami nagyon nagy előrelépés volt a repülés történetében. A MIG-21 es vadászrepülő tömeggyártásra alkalmas volt, és ezért volt olyan jó. Amíg alkalmazták, körülbelül 13-14 ezer ilyen típusú gépet gyártottak le, és a saját idejében csúcsmodern repülőgépnek számított. Ezt erősíti meg az a tény is, hogy szolgálatállításától (1959-től) 2002-2005-ig gyártották.

Azért ezt a témát választottam, mert mindig is érdekelt a repülés, ezen belül a vadászrepülőgépek, és szerintem ez az a vadászrepülőgép, amire ránéznek és felismerik az ikonikussága miatt. Ezt az is segítheti, hogy hazánkban is szolgálatba volt helyezve ez a típusú vadászgép.

## A dolgozat célkitűzései

A játékot nemcsak szórakoztatásra készítem, hanem hogy ez idő alatt a játékosok tanuljanak a vadászgépről. Ezen belül a játékban lehetséges legyen regisztrálni, bejelentkezni, eredményeket megnézni, pályákat végig játszani. Mindemellett a játékosok adatait elmenteni adatbázisba.

## A dolgozat felépítése

Dolgozat felépítése:

* Első rész: Bevezetés.
* Második rész: Felhasználói dokumentáció.
* Harmadik rész: Fejlesztői dokumentáció.
* Negyedik rész: Összegzés.
* Ötödik rész: Ábrajegyzék.
* Hatodik rész: Források megjelölése.

# Felhasználói dokumentáció

Ebben a fejezetben az olvasó megismerheti, hogy honnan érhető el a játék, miről szól a játék, az ajánlott rendszerkövetelményeket, a játék indítását és annak használatát.

## Rendszerkövetelmények

A rendszerkövetelmény azért szükséges, mert így lehet garantálni, hogy a program hibátlanul fusson, és a játékélmény ne vesszen el.

*Szoftverkövetelmény:*

* Operációs rendszer: Microsoft Windows 10 Pro x64, Microsoft Windows 10 x64.
* Adatbázis szerver: XAMPP 8.2.4 vagy frissebb verziója.

*Hardverkövetelmény:*

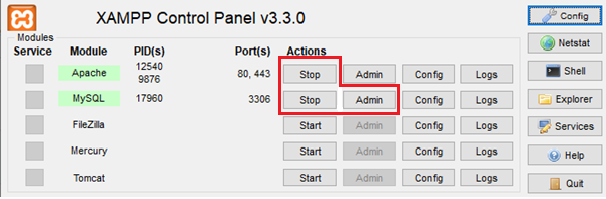
* Processzor: Intel(R) Core(TM) i5-4200U, 1.6GHz, 2 mag, vagy jobb változat. Nem muszáj, hogy Intel márkájú processzor legyen, csak a számítási értéke legyen egyező vagy több.
* Memória méret: Minimum 8 GB.
* Videókártya: Minimum Intel(R) HD Graphics family, 1GB memória. Ha igazán erős processzora van, akkor nem muszáj, hogy legyen videókártyája.
* Tárhely: Minimum 5GB szabad tárhely álljon rendelkezésre.

## Játék indítása

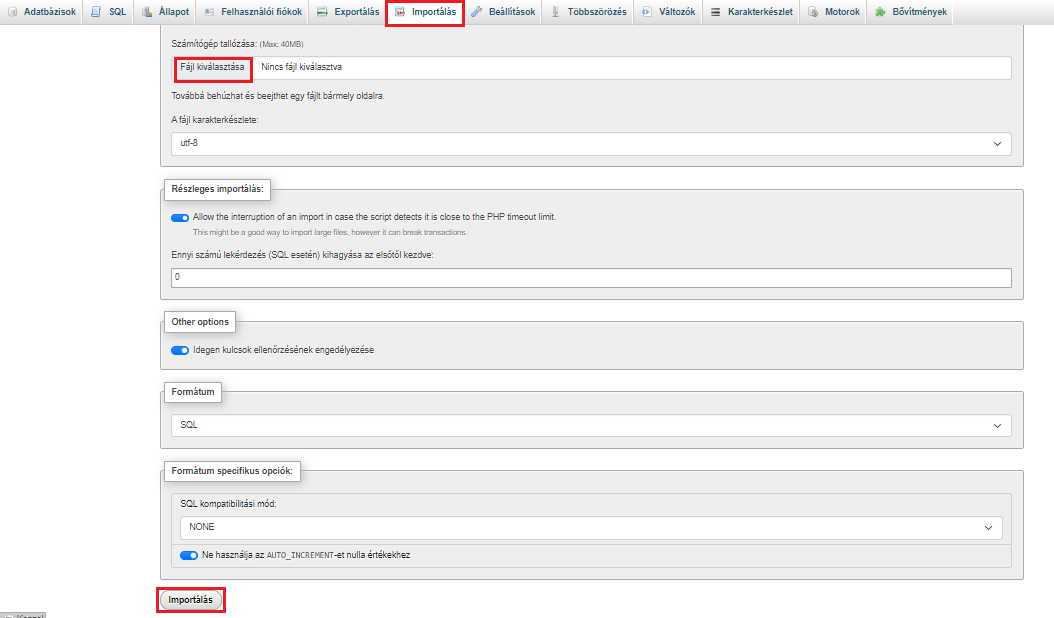
Töltse le a játékot GitHub ról, amit megtalál ezen a linken <https://github.com/DicesAron/AirRace-_Jatek-adatbazis>.

A letöltött ZIP tartalmazza a játékot és a hozzá tartozó adatbázist.

Nyissa meg az XAMPP nevezetű szoftvert, indítsa el az adatbázis szervert, lépjen be az admin felületbe és importálja az adatbázist, amit tartalmaz az airrace.sql fájl.



2.1. ábra: Szerver elindítása.



2.2. ábra: Adatbázis importálása.

Miután ezeket elvégezte, futtassa az AirRace.exe állományt.

## Játék használata

**Röviden a játékról**

Az Air Race egy 3D-s játék, ahol egy MIG-21 es vadászgépet vezethetünk, és pályákat teljesíthetünk. Ahhoz, hogy játszhassunk regisztrálni kell, és bejelentkezni. Belépés után a játék menüjében találhatjuk magunkat, ahol olvashatunk érdekességeket a vadászgépről, beléphetünk a kiképzés és a verseny menüjébe, illetve az eredmények menübe. A kiképzés és a verseny menüjében válaszhatjuk ki és indíthatjuk el a pályákat. Az eredmények az eredmények menüben tekinthetőek meg.

## Felhasználói felület használata

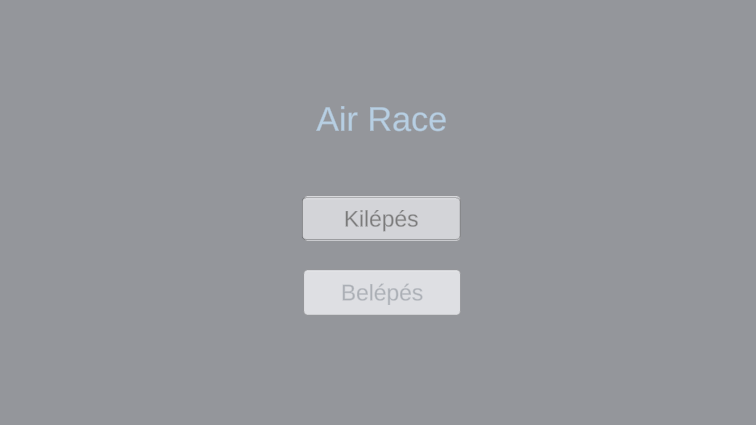
## Főmenü

Belépéskor a Főmenü nyílik meg, ahol két gomb látható, „Regisztráció”, „Belépés”.

Mindkét gomb egy-egy menübe vezet. A „Regisztráció” a regisztráció menüjébe, a „Belépés” a belépés menüjébe. Az „Esc” billentyű lenyomásával megjelenik a kilépés opció.



2.3. ábra: Főmenü.



2.4. ábra: Kilépés opció.

## Regisztráció

A regisztráció egy fontos eleme a játéknak, hisz ha nincs felhasználói fiókunk, akkor nem tudunk belépni a játékba. A regisztráció menüben bal oldalt tájékozódhatunk, hogy mi kell a sikeres regisztráláshoz. A „Regisztráció” szöveg alatt 4 bemeneti mezőt és két gombbot láthatunk.

Mind a 4 mezőt ki kell töltenünk, hogy regisztrálhassunk.

*Bemeneti mezőknek tartalmaznia kell:*

* Felhasználónév: Játékosnév.
* Email cím: Érvényes email cím, például „[teszt@gmail.com](mailto:teszt@gmail.com)”.
* Jelszó: Minimum 6 karakter hosszú jelszó, ami tartalmaz kis és nagybetűt meg számot.
* Jelszó megerősítése: Az előbbiekben beírt jelszó.



2.5. ábra: Regisztrációs felület

Ha minden mezőt kitöltött és a mezők tartalma megfelel a baloldalon látott feltételeknek, akkor a „Regisztráció” gombra kattintva regisztrálhat. Bal oldalt egy eset nincs megemlítve, mégpedig az, hogy nem regisztrálhat már regisztrált játékosnévvel. Nincs megemlítve hisz a játékos nem köteles tudni, hogy mit tartalmaz az adatbázis.

Esetleges hibaüzenetek:



2.6. ábra: Nem töltött ki minden mező hibat.

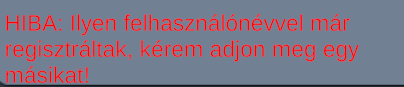
2.7. ábra: Nem megfelelő email formátum hiba.

2.8. ábra: Nem egyező jelszó hiba.

2.9. ábra: Nem megfelelő jelszóhossz hiba.

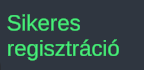
2.10. ábra: Hosszú felhasználónév hiba.

2.11. ábra: Hosszú email cím hiba.

2.12. ábra: Rossz jelszóformátum hiba.

2.13. ábra: Létező felhasználónév hiba.

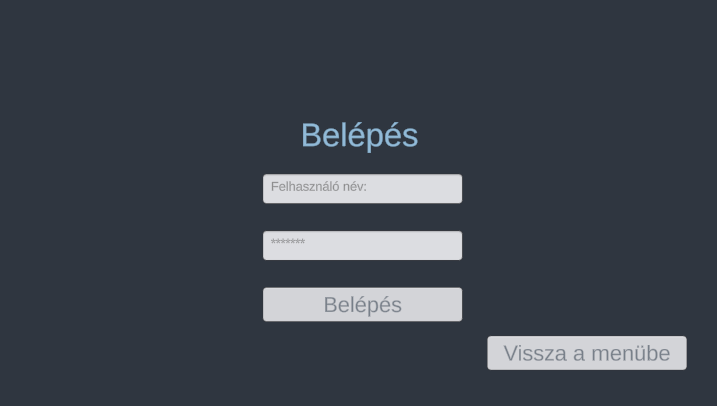
Sikeres regisztráció esetén megjelenik egy „Sikeres regisztráció” felirat jobb oldalt és a játék automatikusan átirányít a Főmenü be.



2.14. ábra: Sikeres regisztráció.

## Belépés

Ebből a menüből lehet belépni a játék menüjébe. Hogy ez meg is történjen, be kell írnunk a „Felhasználó név:” beviteli mezőbe egy már regisztrált felhasználónevet és a hozzá tartozó jelszavát a „jelszó” mezőbe. Ha ez megtörtént, akkor a „Belépés” gombra kattintva a játék átirányít a „játék menüjébe”. A „Vissza a menübe” gombbal visszatérhetünk a Főmenübe. Esetleges hiba esetén a hibaüzenet bal oldalt jelenik meg.



2.15. ábra: Belépés menü.

Esetleges hibaüzenetek:

* „Hiba: Rossz jelszó”.
* „Hiba: Nincs ilyen felhasználó”.
* „Hiba: Minden mezőt ki kell tölteni”.

**Játék menü**

A „játék menü” két részre osztható. Bal és jobb, bal oldalt 3 gomb látható, amik átirányítanak egy másik menübe. Jobb oldalt a vadászgépről lehet érdekességeket olvasni. Összesen 4 oldal van, ahol érdekességet lehet olvasni. Ha az első oldalon vagyunk, akkor nem jelenik meg a „<” gomb, és az utolsó oldalon a „>” gomb.

*Gombok szerepe:*

* „Kiképzés”: Átirányít a kiképzés menübe.
* „Verseny”: Átirányít a kiképzés verseny.
* „Eredmények”: Átirányít a kiképzés eredmények.
* „<”: Balra lapozza a szöveget.
* „>”: Jobra lapozza a szöveget.

Az „Esc” billentyű lenyomásával megjeleni 2 opció. „Kilépés” és „Kijelentkezés”.

„Kilépés”-re kattintva bezárjuk a játékot.

„Kijelentkezés”-re kattintva visszatérünk a Főmenübe.



2.16. ábra: Játék menü



2.17. ábra: Játék menü „Esc”.

**Kiképzés**

A játékos innen indíthatja el az első és második pályát, amik teljesítése után részt vehet a játékosok közötti versenyben. Ha rákattintunk valamelyik „pálya” gombra, akkor az adott pályának a küldetési feltételei jelennek meg alatta. Ha nincs teljesítve az első pálya és rákattintunk a második pályagombra, akkor az jelenik meg, hogy „teljesíteni kell az előző pályát” és nem tudja elindítani. Miután kiválasztottuk a pályát, meg kell nyomni a küldetés megkezdése gombot, ekkor az adott pálya elindul. Legelső belépéskor ebbe a menübe érdemes elolvasni az „Információk” részt, mert itt tudomást szerezhet az irányítás, játék mechanikáról.

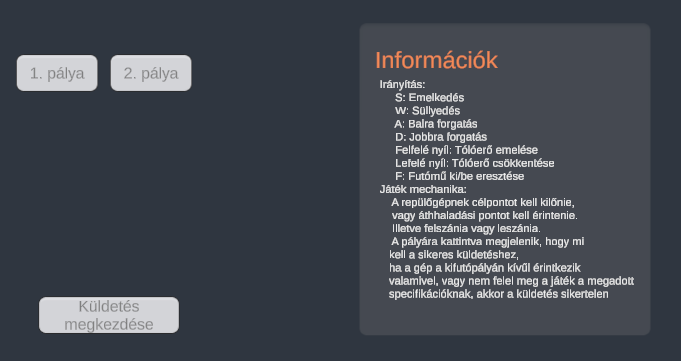
Az „Esc” billentyűt megnyomva 3 opció jelenik meg.

Opciók:

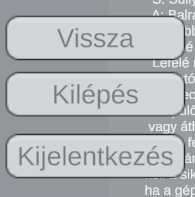
„Vissza”: Rákattintva a játék visszairányít a „Játék menü”-be.

„Kilépés”: Rákattintva bezárja a játékot.

„Kijelentkezés”: Rákattintva a játék visszairányít a „Főmenü”-be.

****

2.18. ábra: Kiképzés menü.



2.19. ábra: Kiképzés menü „Esc”

**Verseny**

A versenymenüt nem szeretném részletezni, hiszen a működési elve megegyezik a „Kiképzés” menüével.

**Eredmények**

Itt tekinthetőek meg a verseny eredményei. Betöltéskor a jobb oldalon lévő rész üres a címsoron kívül. A baloldalon három gomb van és mindegyik a ráírt pálya eredményeit mutatja. Amikor rákattintunk, a jobboldali mezőbe beírja az eredményeket. Mindig a legjobb 3-mat jeleníti meg. Csak az idővel rendelkező játékosokat jeleníti meg.



2.20. ábra: Eredmények menü

Az adott „2.20” ábrán látható, hogy csak 2 játékosnak van időeredménye.Az „Esc” billentyű lenyomásával ugyanaz a 3 opció jelenik meg, mint a „Kiképzés” menüben.

## Játékmechanika

Amikor elkezdődik egy küldetés, vagy megkezdünk egy versenyt, mindig ugyanaz fogad minket. Egy repülőgép a kifutón. A felfelé nyíllal emelnünk kell a tolóerőt, és a „s” billentyűt nyomni, hogy felszállhassunk.

Felszállás után el kell végezni a pálya küldetését, hogy sikeres legyen a küldetés. Sikeres küldetés esetén a játék megváltoztatja az adott pályának az adatait (teljesítve, idő) és megjelenik egy sikeres küldetés felirat és egy gomb. A gombra kattintva visszatérünk a játékmenübe. Ellenkező esetben megjelenik egy sikertelen felirat és nem módosít semmit a játék. Az itt lévő gombkattintásra ugyan az történik, mint a sikeres küldetés esetében.

Ha a küldetések között leszállás is van, akkor ki kell engedni a futóművet, a lefelé nyíllal csökkentjük a tolóerőt és a földet érés után 0-ra állítjuk.

Kiképzéskor a teljesített adatot változtatja meg a játék az adatbázisban, versenykor a ment időnket is módosítja.

Sikertelen küldetés lehet ha:

* A gép a kifutón kívül akármivel érintkezik.
* A gép a kifutóval érintkezik nem teljesen kihajtott futóművel.

# Fejlesztői dokumentáció

Ebben a fejezetben megismerheti, hogy a fejlesztés során milyen szoftvereket és kiegészítőket használtam, a C# scriptek működését és hogy hogyan zajlott a fejlesztés.

A Unity project-et, modelleket, teszteket megtalálja ezen a linken:

<https://github.com/DicesAron/AirRace>

## Fejlesztés során használt szoftverek és kiegészítők.

Minden felhasznált szoftverrel a tanulmányaim során találkoztam, és ezért esett ezekre a választásom. Mindegyik szoftver ingyenesen használható.

## Visual Studio 2017[[1]](#footnote-1)

A Visual Studio a Microsoft által kiadott fejlesztőkörnyezet. Jelenleg a C#, C++, F#, Visual Basic, XML-t és még sok más programozási nyelvet támogat, összesen 36-ot. Segít a megírt kódot gyorsan futtatni, tesztelni, kijavítani. A konzolos programozás mellett lehet számítógépes programot, webes alkalmazást és mobilalkalmazást fejleszteni benne.

C# kódállományok írására és módosítására használtam.

## XAMPP 8.2.4[[2]](#footnote-2)

Apache Friends által kifejlesztett szoftver. A szoftver neve egy rekurzív mozaikszó, ami ezeknek a szavaknak a kezdőbetűje. ["XAMPP Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server) + [MariaDB](https://en.wikipedia.org/wiki/MariaDB) + [PHP](https://en.wikipedia.org/wiki/PHP) + [Perl](https://en.wikipedia.org/wiki/Perl)". Az XAMMP-ot weboldal tervezőknek és programozóknak fejlesztették ki, hogy a lehető leg egyszerűbben tesztelhessék munkájukat internetelérés nélkül.

Az adatbázis futtatására használtam, ami egy webszerveren a saját számítógépemen futott.

## Unity 2022.3.21f1[[3]](#footnote-3)

A Unity egy videójáték-motor, amelyet a Unity Tehcnologies fejlesztetett ki. A szoftver segítségével három, kétdimenziós videójátékokat hozhatunk létre. Saját felhasználásra ingyenes, viszont ha kiadjuk a játékot és egy bizonyos összeget meghalad a bevétel, akkor fizetnünk kell a Unity-nak.

3D-s játékkészítésre használtam és működését C# nyelven írtam meg.

## Blender 3.6[[4]](#footnote-4)

A Blender egy 3D számítógépes grafikus szoftvereszközkészlet, amelyet animációs filmek, 3D-s modellezés, vizuális hatások, textúrázás és még sok más 3D-s munkákra használják.

3D-s objektumok létrehozására és textúrázására használtam.

**BlenderKit**

Ez egy letölthető funkció a Blenderhez, ami egy keresőmezőt implementál az editorba, ahonnan különböző ingyenes és fizetős asset-eket lehet letölteni.

## Verziókövetés[[5]](#footnote-5)

Verziókövetésre a Git-et használtam. A Git egy verziókezelő rendszer, arra szolgál, hogy fileok (programok, dokumentációk, stb) különböző verzióit kordában tartsa, elkönyvelje, tárolja és megossza. Nyomon követi a változtatásokat a könyvtárban és meg is jeleníti. Ha kell, akkor vissza tudja állítani bármelyik commit-olt verziót.

**GitHub[[6]](#footnote-6)**

A Git szoftvert használja, hogy a fejlesztők projektjüket létrehozhassák, tárolják, kezeljék és meg osszák. A Github egy webes szoftver, ami egyszerűvé teszi az előbbiekben felsorolt műveleteket.

Ide töltöttem fel a játéknak a le buildelt változatát, az adatbázist, a Unity project-et, a modelleket és a dokumentációkat.

**GitHub Desktop[[7]](#footnote-7)**

Ezt az asztali alkalmazást azért hozták létre, hogy a fejlesztőnek az alkalmazáson belül kommunikálhasson a GitHub-al, és elvégezzék a legtöbb Git- parancsot anélkül, hogy konzolt vagy webes felületet használna. Könnyű kezelése és letisztultsága miatt élvezet használni.

## C# scriptek

Ahhoz hogy a játék működjön, az összes script elengedhetetlen. Csak az igazán fontos scripteket magyarázom el, mert a pár soros scripteket, magyarázat nélkül is tudja értelmezni az adott programozó, aki megtekinti a forráskódokat.

A forrásállományok elérhetők „[AirRace](https://github.com/DicesAron/AirRace/tree/main)/[Unity](https://github.com/DicesAron/AirRace/tree/main/Unity)/[AirRace](https://github.com/DicesAron/AirRace/tree/main/Unity/AirRace)/[Assets](https://github.com/DicesAron/AirRace/tree/main/Unity/AirRace/Assets)/Scripts” mappában.

*Scriptek feladatai a programban:*

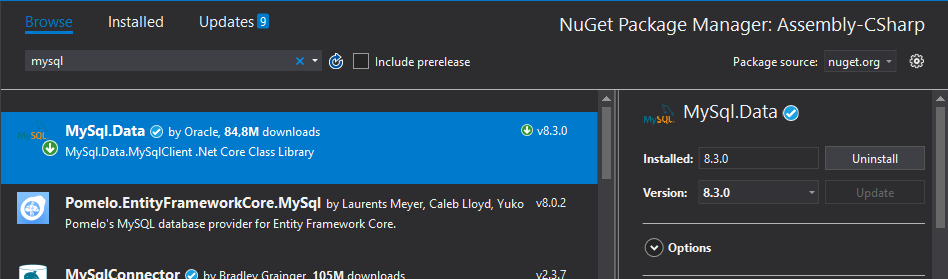
* „Login.cs”: Bejelentkezés.
* „GameManager.cs”: Az eredmények feltöltése adatbázisba.
* „PlaneController.cs”: A repülőgép irányítása, lapátok mozgatása, érintkezések kezelése:
* „verseny.cs”: Verseny menü kezelése.
* „Kikepzes.cs”: Kiképzés menü kezelése.
* „Esc.cs”: Esc billentyű lenyomásakor megjelenő opciók működése.
* „SceneLoader.cs”: Menük betöltése gombnyomásra.
* „Reg.cs”: Regisztráció.
* „CameraControll.cs”: A kamerának a helyzetét átalakítja, hogy kövesse a vadászgépet.
* „erdekesseg.cs”: Érdekessége közötti lapozás.
* „Eredmények.cs”: Megjeleníti az adott pályához tartozó eredményeket.
* „fegyver.cs”: Géppuska tüzelése.
* „TankController.cs”: A tank kilövésének kezelése.
* „palya.cs”: tárolja az adott pályának az ID-t.
* „LovController.cs”: A lövedék törlése, ha érintkezik a célponttal.
* „KorController.cs”: Áthaladási pont kezelése.

## Adatbázis kezelése

A program három adatbázis műveletet használ. Select, Update, Insert.

Ahhoz, hogy C#-ban használhassunk adatbázis kezelést, elsősoron a projectünknek tartalmaznia kell a MySql.Data NuGet-et. Ezzel a NuGet segítségével képesek vagyunk két szükséges „namespace” használatára. A „Mysql.Data” és a „Mysql.Data.MySqlClient”. már megírt fügvényeket és metódusokat használhatunk, amik segítenek az adatbázis elérésben és kezelésével.

Letöltését a Project<Manage NuGet Packages… opcióban tehetjük meg.



3.1. ábra: Visual Studio NuGet Package Manager.

Ahol meghívjuk a többi „namespace”-t, be kell illesztenünk ezt a kettőt:

-„using MySql.Data”

-„using Mysql.Data.MySqlClient”

További lépések példával együtt a 3.2.2 fejezetben olvasható.

## Reg.cs

Ez a script végzi a felhasználó felvételét az adatbázisba. A „Regisztráció” gombra kattintva meghívódik a regisztralas() függvény. Először leellenőrzi. hogy az összes mező ki van-e töltve és, hogy azok az adatok megfelelnek-e az előírtaknak. Ezt a kód 11 feltétellel oldja meg. Ha akármelyik feltétel hamis, akkor a „else” ágban megjelenítésre kerül a hozzá tartozó hibaüzenet.

A 10. feltételben lekérdezésre kerülnek a felhasználónevek egy listában

string connStr = "server=localhost;user=root;database=airrace;port=3306;password=";

MySqlConnection conn = new MySqlConnection(connStr);

try

{

conn.Open();

string sql = "SELECT `Nev` FROM `user`;";

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataReader rdr = cmd.ExecuteReader();

while (rdr.Read())

{

nevek.Add(rdr[0].ToString());

}

rdr.Close();

}

catch (System.Exception ex)

{

hiba.text = $"HIBA: {ex.ToString()}";

}

Ez a kódrészlet létrehozza az adatbázissal a kapcsolatot és a megadott sql parancsot végrehajtja. Miután végrehajtotta az rdr változóba menti a lekérdezett felhasználóneveket. Egy ciklus végigmegy a változón, és az elemeit elmenti a nevek listába. Az utolsó feltétel, hogy a megadott nevet, amit beírt a felhasználó nem tartalmazza a nevek lista.

Igaz feltétel esetén a kód be inzertálja az adatbázisba a felhasználónak az adatait a user táblába és létrehozza az 2 akadálypályát és a 3 versenypályát alapértelmezett adatokkal az akadály, kiképzés táblában. Hamis feltétel esetén megjelenik a hibaüzenet.

Lefutó sql parancsok sikeres regisztrációkor:

-INSERT INTO `user`(`Nev`, `Email`, `Pass`) VALUES ('{nev}','{email}','{pass1}')

-SELECT `ID` FROM `user` WHERE `Nev` = '{nev}'

-INSERT INTO `kikepzes`(`userID`, `palya`, `tejesitve`) VALUES ('{id}','{i}','{0}')//kétszer fut le ciklus segítségével

-INSERT INTO `akadaly`(`userID`, `palya`, `ido`, `teljesitve`) VALUES ('{id}','{i}','0','0')//háromszor fut le ciklus segítségével

Sikeres regisztráció után az Update függvény feltétele igaz lesz és visszairányít a fő menübe.

## Login.cs

Ez a script gondoskodik, hogy beléphessünk a játék menüjébe. A belep() függvény elött két rövidebb függvény látható. Az „inputNev” és az „inputPass”. Mindkét függvény akkor hívódik meg, amikor a hozzá csatolt inputfield tartalma megváltozik. Meghívás esetén a függvény elmenti az inputfield értékét a hozzá tartozó változóba;

„belep()” függvény a hozzácsatolt gomb lenyomásakor fut le. Ilyenkor először egy lekérdezés futtatódig le, ami lekérdezi a beírt névnek az azonosítóját és jelszavát, amit lement a „felhasználó” stringbe. Ha mindkét mezőt kitöltötte és a string nem üres és a beírt jelszó egyezik a lekérdezett jelszóval, akkor a lekérdezett azonosító beírásba kerül a user.txt-be és megjelenik a játék menü. Ellenben hibaüzenetet jelenít meg a „hiba” textbox-ba.

MySqlConnection conn = new MySqlConnection(connStr);

try

{

conn.Open();

string sql = $"SELECT `ID`,`Pass` FROM `user` WHERE `Nev` = '{nev}'";

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataReader rdr = cmd.ExecuteReader();

while (rdr.Read())

{

felhasznalo=rdr[0].ToString() + " " + rdr[1].ToString();

}

rdr.Close();

}

catch (System.Exception ex)

{

hiba.text = $"HIBA: {ex.ToString()}";

}

conn.Close();

if (nev!="" || pass!="")

{

if (felhasznalo != null)

{

if (felhasznalo.Split(' ')[1] == pass)

{

StreamWriter fel = new StreamWriter("AirRace\_Data/user.txt");

fel.Write(felhasznalo.Split(' ')[0]);

fel.Close();

UnityEngine.SceneManagement.SceneManager.LoadScene("JatekMenu");

}

else

{

hiba.text = $"HIBA: Rossz jelszó";

}

}

else

{

hiba.text = $"HIBA: Nincs ilyen felhasználó";

}

}

else

{

hiba.text = "HIBA: Minden mezőt ki kell tölteni";

}

}

## GameManager.cs

Ez a script az összes pályán rá van húzva egy gameobjectre. A „tankController.cs” és a „korController.cs” emeli meg az „allas” változóját. A „korController” akkor emeli mikor ezt a változót, amikor a játékban érintkezünk egy áthaladási ponttal. A „tankController” abban az esetben emeli ezt a változót, ha ötször eltaláltuk.

Elindulásakor lefut a Start() függvény, ami kiolvassa a user.txt tartalmát és elmenti ezt a „felhID” változóba.

A lekerdez() függvény két paramétert kér be. A pálya azonosítóját és az időeredményt. Ezek után az akadálytáblából lekérdezi az adott pályán lévő idejét a felhasználónak az adatbázisból és összeveti a mostani időeredményével. Ha kisebb a mostani időeredménye, akkor igaz eredményt küld vissza, ellenkezőleg hamisat.

Update() függvényben ellenőrzi a script, hogy ez hányadik pálya. Ahol igaz a feltétel az adott küldetés alapján ellenőrzi, hogy mennyi az állás. Ha az állás megegyezik a pálya specifikációjában leírtakkal, akkor frissíti az adatbázisban az adott pályának a teljesített mezőjét igazra és betölti a „sikeres” menüt.

if (palya.palyaid == 1)

{

szoveg.text = $"Állapot: {allas}/3";

if (allas == 3)

{

string connStr = $"server=localhost;user=root;database=airrace;port=3306;password=";

MySqlConnection conn = new MySqlConnection(connStr);

conn.Open();

string sql2 = $"UPDATE `kikepzes` SET `tejesitve`='1' WHERE `userID`='{felhID}'AND `palya`=1";

MySqlCommand cmd = new MySqlCommand(sql2, conn);

cmd.ExecuteNonQuery();

allas = 0;

UnityEngine.SceneManagement.SceneManager.LoadScene("sikeres");}}

Ammenyiben a pálya azonosítója nagyobb, mint kettő, és az állás megegyezik a megadott összeggel, akkor az előző folyamat kibővül a „feltido” változóval, ami tartalmazza az időt, és a „lekerdez” függvénnyel. Meghívásával eldöntésre kerül, hogy ez az időeredmény jobb-e, mint az előző.

Ha igen, akkor frissíti az adott pályának a teljesítve mezőjét igazra, és a játékos idejét a mostani időre az adatbázisban.

if (allas == 10)

{

feltido = string.Format("{0:00}:{1:00}", perc, masodperc);

lekerdez(2, feltido);

if (kisebb)

{

string connStr = $"server=localhost;user=root;database=airrace;port=3306;password=";

MySqlConnection conn = new MySqlConnection(connStr);

conn.Open();

string sql2 = $"UPDATE `akadaly` SET `ido`='00:{feltido}',`teljesitve`='1' WHERE `userID`='{felhID}' AND palya='2'";

MySqlCommand cmd = new MySqlCommand(sql2, conn);

cmd.ExecuteNonQuery();

allas = 0;

UnityEngine.SceneManagement.SceneManager.LoadScene("sikeres");

}

else

{

allas = 0;

UnityEngine.SceneManagement.SceneManager.LoadScene("sikeres");

}

}

## Kikepzes.cs

Ez a script a „Kiképzés” menühöz van csatlakoztatva és ez a script által kezelhetjük, hogy melyik kiképzés pályát kezdjük el.

*Függvények:*

btn1(): Az „1. pálya” gombra kattintva hívódik meg, és megváltoztatja a „pálya” változó értékét 1-re és kiírja ez első küldetésnek a specifikációját.

btn2():Az „2. pálya” gombra kattintva hívódik meg, létrehoz egy string változót, és utána lekérdezi adatbázisból az első pálya „teljesitve” mezőjét. Miután lekérdezte, a string változó értékét a lekérdezett mező értékére állítja be. Ha a lekérdezés sikeres volt, akkor egy feltételben megnézi, hogy igaz e. Ha nem, akkor kiírja azt az üzenetet, hogy „Teljesíteni kell az előző pályát”. Amennyiben a változó értéke igaz, átállítja a „palya” változó értékét igazra, és kiírja a második küldetésnek a specifikációját.

betolt(): Az „palya” változó értéke alapján betölti a megfelelő pályát, ha a változó értéke megegyezik nullával, akkor megjeleníti a „Nem választott pályát” feliratot.

## verseny.cs

Működési elve majdnem azonos a „Kikepzes.cs” működésével, ezért nem tartom részletes bemutatását fontosnak. A „verseny.cs” állományban megírt függvényekkel a „Verseny” menüt tudjuk kezelni.

## Eredmenyek.cs

Ezzel a scripttel kezeljük az „Eredmények” menüt. Feladata, hogy megjelenítse a kiválasztott pályának azt a három játékosát, akinek az időeredménye a legkisebb.

*Függvények:*

btn1(): Az „1.pálya” gomb lenyomásával hívódik meg. A „palya” változó értékét átállítja 1-re és meghívja a Lekerdez(palya) függvény a „palya” paraméterrel

btn2(): Az „2.pálya” gomb lenyomásával hívódik meg. A „palya” változó értékét átállítja 2-re és meghívja a Lekerdez(palya) függvény a „palya” paraméterrel

btn3(): Az „3.pálya” gomb lenyomásával hívódik meg. A „palya” változó értékét átállítja 3-ra és meghívja a Lekerdez(palya) függvény a „palya” paraméterrel

Lekerdez(): Létrehozza a kapcsolatot az adatbázissal, és végrehajt egy több táblás lekérdezést az adatbázisból. Sikeres lekérdezés esetén egy while ciklussal végigmegy az rdr változón és az adatokat hozzáfűzi a „tobbi” TextMeshPro változóhoz.

string connStr = "server=localhost;user=root;database=airrace;port=3306;password=";

MySqlConnection conn = new MySqlConnection(connStr);

try

{

tobbi.text = "";

int i = 1;

conn.Open();

string sql = $"SELECT user.Nev,akadaly.ido FROM `akadaly`,user WHERE user.ID=akadaly.userID AND `palya`='{palya}' AND akadaly.ido!='00:00:00' GROUP by user.ID ORDER BY `akadaly`.`ido` ASC LIMIT 3;";

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataReader rdr = cmd.ExecuteReader();

while (rdr.Read())

{

tobbi.text += $"#{i}\t{rdr[0].ToString()}\t {rdr[1].ToString().Split(':')[1]}:{rdr[1].ToString().Split(':')[2]}\n";

i++;

}

rdr.Close();

}

catch (System.Exception)

{

}conn.Close();

## PlaneController.cs

A játék szinte legfontosabb scriptje, ezzel a scripttel lehet irányítani a repülőgépet. 27 változóval ez a legtöbb változóval rendelkező script.

A script indulásakor az Awake() függvény futtatódik le. Itt a „repulo” változót egyenlővé teszem a „Getcomponent<Rigidbody>()”-val. Ezáltal hozzárendelem a vadászgép rigitbody-át. A „hajtomu” változót egyenlővé teszem a „GetComponent<AudioSource>()”-al. A játékban végűl nincsenek hangok, mert a hanganyagot sehogy sem tudtam megvágni úgy, hogy az ismétlődésnél jól hangzzon.

IranyitasInput() függvény nagyon fontos, hisz itt rendelem a változókhoz az input manager-ből az „axis”- okat. Emellett a tolóerő csökkentését és emelését, a leszállás ellenőrzését, és a lapátok forgatását is itt végzem el.

Lapátok és futómű forgatását egyszerűen oldottam meg. Ellenőrzőm, hogy a játékos melyik gombot nyomja le az irányításkor, és ha az adott gomb lenyomásra kerül, megváltoztatom a lapátok rotációjának koordinátáit és megemelem a hozzátartozó változó értékét eggyel. Ahhoz hogy ne forogjanak körbe-körbe az adott változó leellenőrzésre kerül, hogy kisebb-e egy adott értéknél mielőtt elforgatom. Ha nincs lenyomva gomb, akkor mindegyik változót addig csökkentem, míg nem éri el a nulla értéket és ugyanannyival csökkentem a rotációjának koordinátáit, mint amennyivel emeltem.

A futóműnek van egy bool változója, ami az „f” billentyű lenyomásával az értékét változtatja. Ezt az értéket a true-ra állítja a program a kód elindulásakor.

A Kijelzo() függvénnyel jelenítem meg a gép értékeit a bal alsó sarokban(tolóerő, sebesség, magasság).

FixedUpdate() felel a vadászgép rigitbody komponensének rendelni gépre ható erőket.

Update() fügvényben hívom meg a IranyitasInput() és a Kijelzo() fügvényt.

## Adatbázis

A projektben fontos szerepet játszik az adatbázis. Itt tárolom a felhasználók adatait és a hozzátartozó pályák értékeit.

Az adatbázist az XAMPP szoftverrel készítettem el és futtattam. A játékhoz tartozó adatbázis neve airrace. Három InnoDB táblát tartalmaz melyek a user, akadaly, kikepzes táblák. Kódolásuk utf8\_hungarian\_ci.

*user tábla tartalma:*

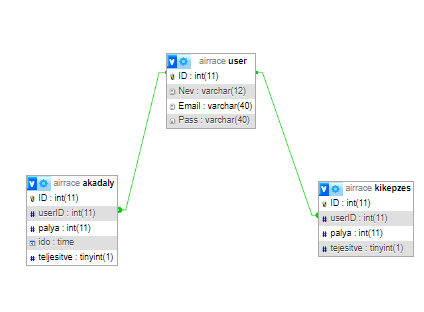
* ID: int típusú, hossza 11 karakter, AI, elsődleges kulcs.
* Nev: varchar típusú, hossza 12 karakter.
* Email: varchar típusú, hossza 40 karakter.
* Pass: varchar típusú, hossza 40 karakter.

*kikepzes tábla tartalma:*

* ID: int típusú, hossza 11 karakter, AI, elsődleges kulcs.
* UserID: int típusú, hossza 11 karakter, index.
* palya: int típusú, hossza 11 karakter.
* tejesitve: tinyint típusú.

akadaly tábla tartalma:

* ID: int típusú, hossza 11 karakter, AI, elsődleges kulcs.
* UserID: int típusú, hossza 11 karakter, index.
* palya: int típusú, hossza 11 karakter.
* ido: time típusú.
* tejesitve: tinyint típusú.



3.2. ábra: Adatbázis kapcsolat

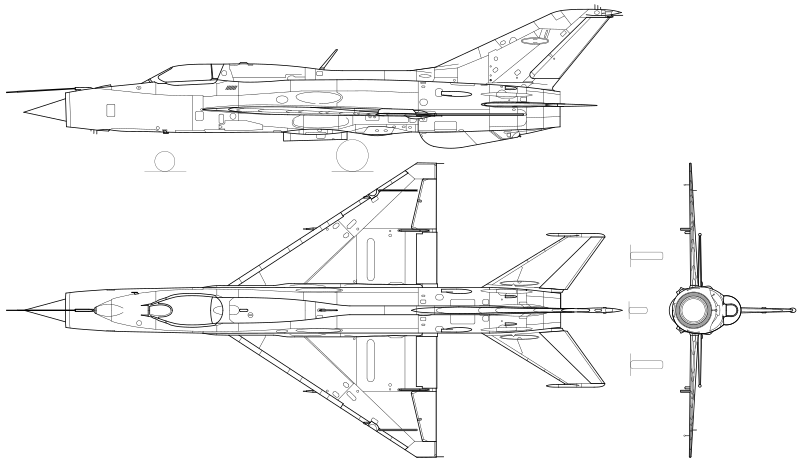
## 3D modellezés

A 3D modellezést a Blender szoftverrel, azon belül a 3.6-os verzióval készítettem. Két modellt készítettem itt, a repülőgépet és a felszállópályát. Legelső modellem a MIG-21-es ideiglenes változata volt, hogy tudjam tesztelni a játékban a repülőgép irányítását. Második modellként a Blender Kit weboldalon ingyenesen felhasználható Tigris II harckocsit töltöttem le, hogy a célpontot le tudjam tesztelni és egy végleges kinézetet adni neki. A harmadik modell, amit én készítettem az a reptér. Az utolsó modell, amit készítettem az a MIG-21-es végleges változata volt, amit már közvetlen úgy modelleztem, hogy a lapátok külön objektumok legyenek, hogy a játékon belül mozgatni tudjam azokat.



3.3. ábra: MIG-21-es vadászgép a blenderben.

A repülő tervezésekor referenciaként egy MIG-21 es tervrajzot használtam.



3.4. ábra: MIG-21 tervrajza

# Összegzés

Az összegzés rész a tartalmazza a projectről alkotott véleményemet és a tovább fejlesztési lehetőségeket.

## Vélemény

Leszámítva a sokórányi fejvakargatást, hogy miért nem működik a program, úgy ahogy azt én szeretném, pozitívumként él bennem a project. Rengeteg új dolgot tanulhattam, például a Blender és a Unity professzionálisabb használatát, jobb rálátásom lett a verziókövetésre és a problémamegoldóképességem fejlődött. Minden egyes megoldott probléma után jobban éreztem magam, és büszkeséget éreztem, mert önállóan oldottam meg. Ezeket a project alatt megtanult dolgokat, majd a munkaerőpiacon hasznosítani tudom majd, aminek nagyon örülök.

## Tovább fejlesztési lehetőségek

Tovább fejlesztési lehetőségek:

* Realisztikusabb irányítás.
* Hangok.
* Effektek.
* Részletesebb 3D-s modellek és textúrák.
* Több függvény használata a kódban.

Az utolsó pontot leszámítva mindegyik elem emeli a játékélményt. Realisztikusabb irányításnál a fizikai együtthatók-ra gondolok, például irányváltásakor a sebesség csökken, tolóerővisszavételkor nem egyből csökken a sebesség, a szárnyak mérete és a gép sebesség alapján felhajtóerőt számolni, ami sokat segítene a leszállásban a „vitorlázás” segítségével. A fejlesztés során implementáltam hangokat és effekteket a játékba, de ezek nem működtek tökéletesen és úgy döntöttem, hogy kiveszem ezeket a játékból, hogy a játékélményt ne rontjam el.

# Ábrajegyzék

[2.1. ábra: Szerver elindítása. 8](#_Toc165249765)

[2.2. ábra: Adatbázis importálása. 8](#_Toc165249766)

[2.3. ábra: Főmenü. 9](#_Toc165249767)

[2.4. ábra: Kilépés opció. 10](#_Toc165249768)

[2.5. ábra: Regisztrációs felület 11](#_Toc165249769)

[2.6. ábra: Nem töltött ki minden mező hibat. 11](#_Toc165249770)

[2.7. ábra: Nem megfelelő email formátum hiba. 11](#_Toc165249771)

[2.8. ábra: Nem egyező jelszó hiba. 11](#_Toc165249772)

[2.9. ábra: Nem megfelelő jelszóhossz hiba. 12](#_Toc165249773)

[2.10. ábra: Hosszú felhasználónév hiba. 12](#_Toc165249774)

[2.11. ábra: Hosszú email cím hiba. 12](#_Toc165249775)

[2.12. ábra: Rossz jelszóformátum hiba. 12](#_Toc165249776)

[2.13. ábra: Létező felhasználónév hiba. 12](#_Toc165249777)

[2.14. ábra: Sikeres regisztráció. 12](#_Toc165249778)

[2.15. ábra: Belépés menü. 13](#_Toc165249779)

[2.16. ábra: Játék menü 14](#_Toc165249780)

[2.17. ábra: Játék menü „Esc”. 15](#_Toc165249781)

[2.18. ábra: Kiképzés menü. 16](#_Toc165249782)

[2.19. ábra: Kiképzés menü „Esc” 16](#_Toc165249783)

[2.20. ábra: Eredmények menü 17](#_Toc165249784)

[3.1. ábra: Visual Studio NuGet Package Manager. 22](#_Toc165249785)

[3.2. ábra: Adatbázis kapcsolat 29](#_Toc165249786)

[3.3. ábra: MIG-21-es vadászgép a blenderben. 30](#_Toc165249787)

[3.4. ábra: MIG-21 tervrajza 31](#_Toc165249788)

# Források megjelölése

Videók:

*Simple Physics Based Plane Controller in Unity:*

<https://www.youtube.com/watch?v=fThb5M2OBJ8> [2024.04.01]

*Smooth Camera Follow in Unity – Tutorial:*

[https://www.youtube.com/watch?v=MFQhpwc6cKE[2024.04.06](https://www.youtube.com/watch?v=MFQhpwc6cKE%5b2024.04.06)]

*Simple Shooting | 3D | Bullets | Unity Game Engine:*

[https://www.youtube.com/watch?v=EwiUomzehKU[2024.0.6](https://www.youtube.com/watch?v=EwiUomzehKU%5b2024.0.6)]

Weboldalak:

*model: Panzerkampfwagen Tiger Ausf. B:*

<https://www.blenderkit.com/asset-gallery?query=category_subtree:ground+order:-created+is_free:true> [2024.04.6]

*War fx:*

<https://assetstore.unity.com/packages/vfx/particles/war-fx-5669>

Színhasználat:   
<https://flatuicolors.com/>

*Terrain Sample Asset Pack:*

<https://assetstore.unity.com/packages/3d/environments/landscapes/terrain-sample-asset-pack-145808>

*Visual Studio:*

<https://en.wikipedia.org/wiki/Visual_Studio>

*XAMPP:*

<https://en.wikipedia.org/wiki/XAMPP>

*Unity (játékmotor):*

<https://hu.wikipedia.org/wiki/Unity_(j%C3%A1t%C3%A9kmotor)>

*Blender (software):*

<https://en.wikipedia.org/wiki/Blender_(software)>

*Github:*

<https://en.wikipedia.org/wiki/GitHub>

*Verziókövetés:*

<https://people.maths.bris.ac.uk/~mb13434/git/miagit.html>

*Girhub Desctop:*

<https://docs.github.com/en/desktop>

*Vadászgép tervrajz:*

<https://hu.wikipedia.org/wiki/MiG%E2%80%9321#/media/F%C3%A1jl:Mikoyan->Gurevich\_MiG-21PF\_3-view.svg

1. <https://en.wikipedia.org/wiki/Visual_Studio> [↑](#footnote-ref-1)
2. <https://en.wikipedia.org/wiki/XAMPP> [↑](#footnote-ref-2)
3. <https://hu.wikipedia.org/wiki/Unity_(j%C3%A1t%C3%A9kmotor)> [↑](#footnote-ref-3)
4. <https://en.wikipedia.org/wiki/Blender_(software)> [↑](#footnote-ref-4)
5. <https://people.maths.bris.ac.uk/~mb13434/git/miagit.html> [↑](#footnote-ref-5)
6. <https://en.wikipedia.org/wiki/GitHub> [↑](#footnote-ref-6)
7. <https://docs.github.com/en/desktop> [↑](#footnote-ref-7)