**Quiz játék**

**3.csapat**

**Projektzáró**

**Dokumentáció**

Tartalomjegyzék

[Bevezető 4](#_Toc196470996)

[A projekt célja 5](#_Toc196470997)

[A projekt időtartama 5](#_Toc196470998)

[Projektvezető kinevezése 5](#_Toc196470999)

[A projektcsapat 6](#_Toc196471000)

[Kompetencia mátrix 7](#_Toc196471001)

[Tevékenység-felelős mátrix 8](#_Toc196471002)

[Kommunikációs terv 9](#_Toc196471003)

[A rendszer részletes leírása: 9](#_Toc196471004)

[Modul- és osztályleírások 10](#_Toc196471005)

[Adatkezelés 13](#_Toc196471006)

[questions.json – kérdések 13](#_Toc196471007)

[scores.json – eredmények 13](#_Toc196471008)

[Tesztelés 15](#_Toc196471009)

[Belső tesztelés (fejlesztők) 15](#_Toc196471010)

[Külső tesztelés (ismerősök, osztálytársak) 15](#_Toc196471011)

[Javított hibák 15](#_Toc196471012)

[UML Osztálydiagram 16](#_Toc196471013)

[Fontosabb kódrészletek 17](#_Toc196471014)

[A program működésének folyamata 18](#_Toc196471015)

[Követelmények feltárása 19](#_Toc196471016)

[Funkcionális követelmények 19](#_Toc196471017)

[Nem funkcionális követelmények 20](#_Toc196471018)

[Felhasználói követelmények 20](#_Toc196471019)

[Technikai követelmények 20](#_Toc196471020)

[Projektterv és mérföldkövek 21](#_Toc196471021)

[Logikai keretmátrix 22](#_Toc196471022)

[Használati útmutató kivonat 23](#_Toc196471023)

[További fejlesztési lehetőségek: 23](#_Toc196471024)

[Összefoglalás 24](#_Toc196471025)

[Nehézségek, problémák: 24](#_Toc196471026)

[Tapasztalatok: 24](#_Toc196471027)

[Csapatmunka értékelése: 24](#_Toc196471028)

[Projekt zárása 25](#_Toc196471029)

# Bevezető

Az informatika tanulása sokak számára kihívást jelenthet – különösen akkor, ha az oktatás frontális, motivációhiányos vagy monoton. Projektünk célja, hogy ezt a tanulási folyamatot élvezetesebbé és interaktívabbá tegye egy olyan quiz alapú játék segítségével, amely ötvözi a klasszikus Honfoglaló (mai nevén Triviador) stratégiai elemeit a modern Kahoot-stílusú kérdés-válasz játék dinamikájával.

A fejlesztett alkalmazás nem csupán tudásmérő eszközként szolgál, hanem egyben játékélményt is nyújt, ezáltal ösztönözve a diákokat a tananyag aktív elsajátítására. A kérdések informatika témakörre fókuszálnak – például algoritmusok, programozási alapok, hálózati ismeretek –, és többjátékos módban is játszhatók, így versenyszellemet és csapatmunkát is támogat.

Célunk egy olyan platform létrehozása volt, amely nemcsak szórakoztató, hanem hatékony tanulási eszköz is, és amely elősegíti a digitális kompetenciák fejlődését egy játékos, motiváló környezetben.

# A projekt célja

A Quiz játék egy oktatási célú, hacker szimulációs környezetbe ágyazott kvízjáték, amely során a játékosok számítógépes hálózatokat foglalnak el helyes válaszokkal. Acél a tanulás játékos, élményszerű támogatása, különösen szakközépiskolás és elsőéves egyetemi hallgatók számára.

A játék során a felhasználó egy térképes felületen navigál, ahol számítógépeket "támadhat meg" – azaz kérdésekre válaszolva elfoglalhatja őket. Egy-egy kérdés helyes megválaszolásáért pontot kap, hibás válasz esetén a gép zárva marad, és csak egy nehezebb kérdés helyes megválaszolásával próbálkozhat újra. A cél az összes gép elfoglalása a lehető legkevesebb hibával, a legrövidebb idő alatt.

A projekt célja továbbá az is, hogy a diákokat hozzásegítse a digitális gondolkodás, a hálózati szemlélet és az alapvető informatikai fogalmak játékos gyakorlásához**.**

# A projekt időtartama

* Kezdés: 2025. március 4.
* Befejezés: 2025. április 28.
* Időtartam: 8 hét, hetente mért mérföldkövekkel és folyamatos haladással

# Projektvezető kinevezése

Aláírásunkkal igazoljuk, hogy a projekt vezetésével Asboth Noémi bízzuk meg. A projektvezető aláírásával igazolja, hogy vállalja a projekt vezetésével, szervezésével kapcsolatos teendőket:

* + **feladatok kiosztása**
  + **munkálatok koordinálása**
  + **kapcsolattartás a csapattagokkal**

# A projektcsapat

Minden csapattag részt vett a fejlesztési munkában, szerepkörük alapján konkrét feladatokat láttak el. A projektvezető feladata volt az irányítás, tervezés, koordináció, míg a GUI-tervezők a felhasználói felületet és interakciókat dolgozták ki. Az adatkezelést és dokumentációt Kassai László Richárd végezte.

| **Név** | **Neptun kód** | **Feladatkör** |
| --- | --- | --- |
| Asboth Noémi | BKHD50 | Projektvezető, programozás, JSON fájlkezelés |
| Barányi Alexandra | L0T3QD | GUI tervezés, megvalósítás |
| Hajnal Márk Mihály | L5A1D1 | GUI tervezés, megvalósítás |
| Kassai László Richárd | RMD1FZ | Dokumentáció, JSON fájlok, adminisztráció |

# Kompetencia mátrix

A fenti táblázat jól mutatja, hogy a csapattagok kompetenciái széles körűek voltak, és a projekt sikeres megvalósításához minden szükséges részterület lefedésre került.

| **Név** | **Frontend** | **Backend** | **Adatkezelés** | **Dokumentáció** | **Admin** |
| --- | --- | --- | --- | --- | --- |
| Asboth Noémi | **✅** | **✅** | **✅** |  |  |
| Barányi Alexandra | **✅** |  | **✅** |  |  |
| Hajnal Márk Mihály | **✅** |  | **✅** |  |  |
| Kassai László Richárd |  |  | **✅** | **✅** | **✅** |

# Tevékenység-felelős mátrix

A mátrix megmutatja, hogy a projekt során melyik szereplő milyen felelősséget vállalt, illetve mely fázisokban vett részt aktívan.

| **Tevékenység** | **Projektvezető** | **Csapat** | **Megrendelő (tanár)** | **Tesztelők** |
| --- | --- | --- | --- | --- |
| Projektterv | ✅ | ✅ | ✅ |  |
| Feladatkiosztás | ✅ | ✅ |  |  |
| Kódolás | ✅ | ✅ |  |  |
| Dokumentáció | 🔲 | ✅ |  |  |
| Tesztelés | ✅ | ✅ |  | ✅ |
| Projektzárás | ✅ | ✅ | ✅ |  |

# Kommunikációs terv

A projekt során a kommunikációs tervnek megfelelően minden szereplő rendszeresen kapott visszajelzést és státuszjelentést, így a koordináció hatékony volt.

| **Szereplő** | **Információ típusa** | **Gyakoriság** | **Kommunikációs csatorna** | **Válaszidő** |
| --- | --- | --- | --- | --- |
| Megrendelő (tanár) | Mérföldkövek, státusz | Hetente | Email, konzultáció | 7 nap |
| Projektvezető | Fejlesztési állapot | 2-3 naponta | Messenger, email | max. 2 nap |
| Csapattagok | Fejlesztési feladatok, hibák | Napi | Discord, csoportchat | 1 nap |
| Tesztelők | Funkcióteszt visszajelzés | Teszt után | Személyesen, email | Azonnal |

# A rendszer részletes leírása:

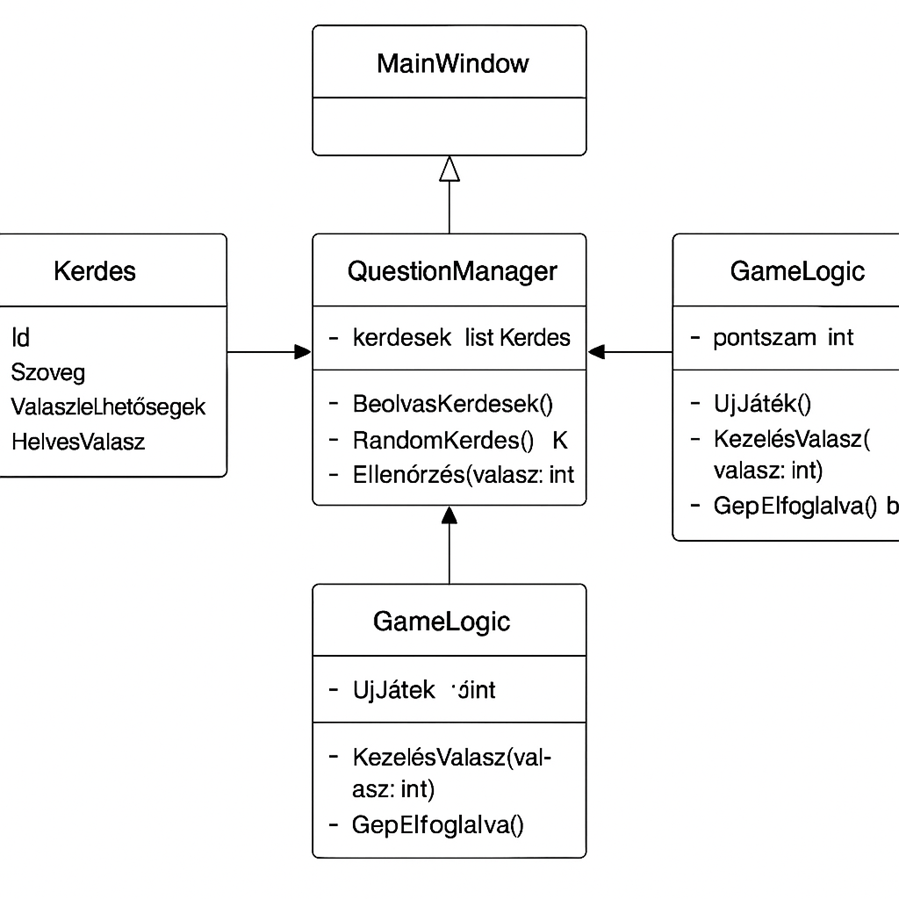
A játék feleletválasztós kérdéseket tartalmaz. Minden kérdéshez négy válaszlehetőség tartozik, amelyek közül egy helyes. A kérdések az informatika témaköréből származnak, azon belül is programozás, hardver, számítógép-ismeret és hálózati ismeretek területekről.

A kérdések megválaszolása után a játék pontozza a válaszokat: a pontérték megegyezik a kérdés nehézségi szintjével (1-3). Hibás válasz esetén az adott gép nem foglalható el, hanem nehezebb szintű kérdés jelenik meg az új próbálkozáshoz.

A játék nem tartalmaz időkorlátot a kérdésekre, de az eltelt időt méri, és a végén megjeleníti, mint rekordidőt.

# Modul- és osztályleírások

A játék felépítése moduláris, minden fontos funkció külön osztályban valósult meg. Ez biztosítja az átláthatóságot, könnyű hibakeresést és későbbi fejleszthetőséget.



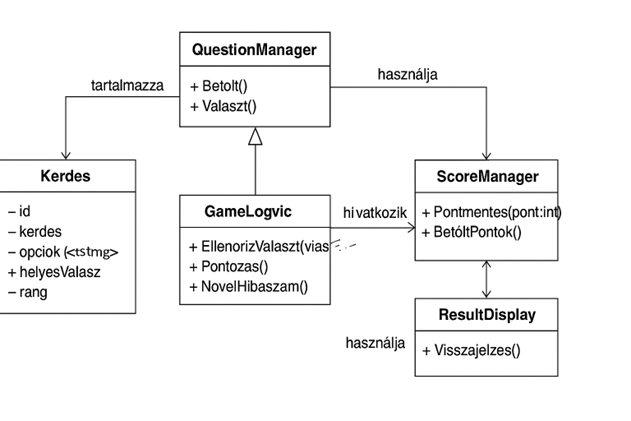
**MainWindow.xaml.cs**

Ez a főablak osztálya, innen indul minden interakció. Kezeli a főmenü megjelenítését, a navigációt, a pályaválasztást és a kilépést. Gombesemények és pályaindítás is innen történik.

**QuestionManager.cs**

Ez az osztály felel a kérdések betöltéséért a questions.json fájlból.  
Képes szűrni kérdéseket nehézségi szint szerint (rank) és véletlenszerűen kiválasztani egyet.  
Függvényei:

* BetoltKerdesek()
* ValasszKerdest(int nehezseg)



**GameLogic.cs**

A játékmenet szabályait kezeli:

* pontozás (+1, +2, +3 a rank szerint),
* hibák számlálása,
* győzelmi / vereségi feltételek ellenőrzése,
* visszajelzés a válasz alapján.

**ScoreManager.cs**

Ez az osztály felelős a pontszámok mentéséért a scores.json fájlba.  
A játék végén az idő és pontszám rögzítésre kerül.

**ResultDisplay.xaml**

Egy UserControl, amely visszajelzést ad:

* színnel (zöld/piros),
* szöveggel (helyes/hibás),
* pont- és hibaszám megjelenítésével.

# Adatkezelés

A Quiz játék adatbázisa két fájlból áll:

## questions.json – kérdések

Minden kérdésnek van:

* id
* question (kérdés)
* options (4 válasz)
* answer (helyes válasz)
* rank (nehézségi szint)

json

MásolásSzerkesztés

{

"id": 1,

"question": "Mi a CPU szerepe?",

"options": ["Számítás", "Kép megjelenítés", "Hang lejátszás", "Tárolás"],

"answer": "Számítás",

"rank": 2

}

## scores.json – eredmények

A pontszám, térképazonosító és idő szerepel:

json

MásolásSzerkesztés

{

"scoreLs": [

{"TimeMinut": 15, "Mep": 1, "Score": 12}

]

}

Az adatok beolvasása és írása a Newtonsoft.Json csomaggal történik.

# Tesztelés

A tesztelés manuálisan zajlott, több körben:

## Belső tesztelés (fejlesztők)

* Főmenü működése
* Pályanavigáció
* Kérdések betöltése, véletlenszerűség
* Válaszlehetőségek működése
* Pontozás, hibák számlálása
* Mentés, betöltés (scores.json)

## Külső tesztelés (ismerősök, osztálytársak)

* GUI átláthatóság
* Használhatóság értékelése
* Hibák: hosszú válasz nem fér ki, helyes válasz után nem záródik a kérdésablak

## Javított hibák

* JSON fájlhiba lekezelése
* Válaszablak bezárása automatikusan
* Hosszú szöveg igazítása

Összegzés: A játék stabil, a célkitűzött funkciók működnek. Egyéni hibák ritkák, a program jól skálázható.

# UML Osztálydiagram

Az alábbi osztályok alkotják az alkalmazás architektúráját:

**Kerdes (class)**

* int id
* string question
* List<string> options
* string answer
* int rank

**ScoreEntry (class)**

* int Mep
* int Score
* int TimeMinut

**QuestionManager (class)**

* List<Kerdes> kerdesek
* BetoltKerdesek()
* ValasszKerdest(int rank)

**GameLogic (class)**

* int pontszam
* int hibak
* bool EllenorizValasz(string adott, string helyes)
* void NyeresVereses()

# Fontosabb kódrészletek

csharp

MásolásSzerkesztés

// Kérdés betöltés

string json = File.ReadAllText("questions.json");

List<Kerdes> kerdesek = JsonConvert.DeserializeObject<List<Kerdes>>(json);

// Véletlenszerű kérdés kiválasztása adott rang alapján

var lista = kerdesek.Where(k => k.rank == rank).ToList();

Random r = new Random();

return lista[r.Next(lista.Count)];

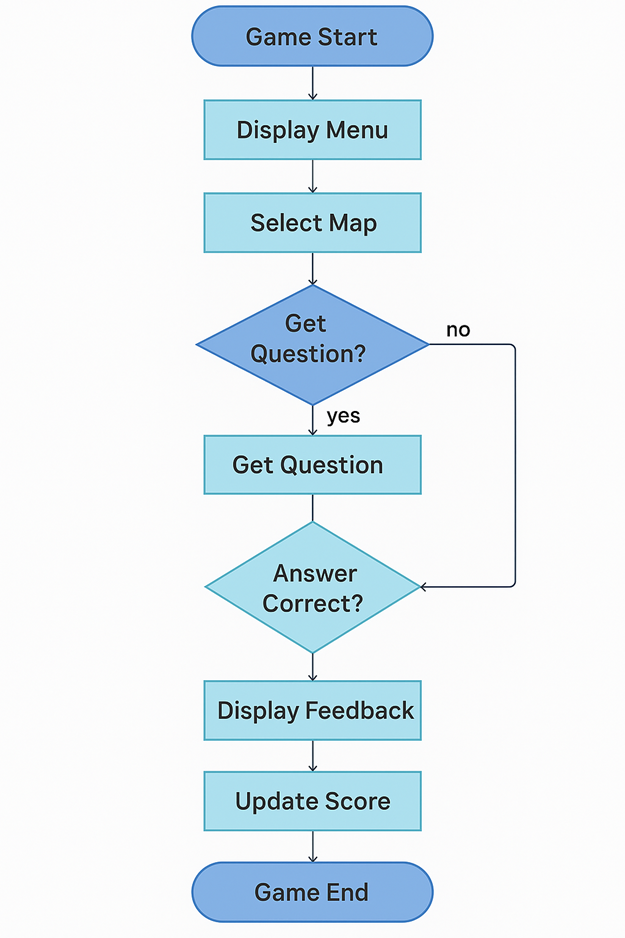
// Válasz ellenőrzése

return adott.Trim().ToLower() == helyes.Trim().ToLower();

// Pontozás

pontszam += kerdes.rank;

# A program működésének folyamata



1. **Főmenü:**  
   Három gomb: *Játék indítása*, *Pályák*, *Kilépés*
2. **Pályaválasztás:**  
   Három nehézségi szint: Map1 (easy), Map2 (normal), Map3 (hard)
3. **Gépkattintás:**  
   A felhasználó kiválaszt egy gépet, ami kérdést generál
4. **Válaszadás:**  
   Négy válasz közül választ, az értékelés azonnali
5. **Pontozás / Hibakezelés:**
   * Helyes válasz: pontszám nő, gép „megszerzett”
   * Hiba: a gép zárolva marad, újra kell próbálni nehezebb kérdéssel
6. **Játék vége:**  
   Ha minden gép elfoglalt vagy a hiba limit túllépett, a játék véget ér
7. **Mentés:**  
   Pont és idő mentésre kerül a scores.json fájlba

# Követelmények feltárása

## Funkcionális követelmények

* Négy válaszlehetőségből választható kvízkérdések
* Véletlenszerű kérdésgenerálás nehézségi szint alapján
* Válasz ellenőrzés, pontozás, hibakezelés
* Több pálya, eltérő nehézségi fokozatok
* Kérdések és pontszámok mentése JSON fájlba
* Highscore lista megjelenítése
* Grafikus felület gombokkal, panelek használatával
* Lehetőség a játék újraindítására, kilépésre

## Nem funkcionális követelmények

* Stabil működés Windows platformon
* Reszponzív GUI, felhasználóbarát megjelenítés
* Hibakezelés: fájl hiánya, hibás adatszerkezet esetén figyelmeztetés
* Gyors betöltés, minimális erőforrásigény

## Felhasználói követelmények

* Nincs szükség regisztrációra vagy tanúsítványra
* Azonnal indítható játékélmény
* Értelmezhető pontozási rendszer
* Helyes és hibás válaszok egyértelmű megjelenítése

## Technikai követelmények

* Fejlesztés: Visual Studio, .NET, WPF, C#
* Adatkezelés: JSON fájlok (questions.json, scores.json)
* Newtonsoft.Json könyvtár szükséges
* Windows 10 vagy újabb operációs rendszer

# Projektterv és mérföldkövek

A fejlesztés során nem történt jelentős csúszás, minden határidő tartható volt.

| **Dátum** | **Tevékenység** |
| --- | --- |
| 2025.03.04 | Projektindítás, csapatalakítás |
| 2025.03.06 | Funkciólista és specifikáció |
| 2025.03.13 | Objektumdiagram, osztályok tervezése |
| 2025.03.20 | GUI és adatmodell fejlesztése |
| 2025.03.27 | JSON kérdésbank feltöltése |
| 2025.04.10 | Pontozás, hibakezelés beépítése |
| 2025.04.17 | Highscore és mentési logika |
| 2025.04.20 | Teljes játék összeállítása |
| 2025.04.24 | Tesztelés, hibajavítás |
| 2025.04.28 | Dokumentáció véglegesítése, beadás |

# Logikai keretmátrix

| **Cél** | **Kimenet** | **Indikátor** | **Ellenőrzés módja** | **Feltételek** |
| --- | --- | --- | --- | --- |
| Oktatási célú játék fejlesztése | Teljes funkcionalitás | Működő játék + mentés | Manuális teszt | Stabil környezet, GUI kész |
| Informatikai tudás fejlesztése | Kérdésbank | Kérdések száma, témakörök | JSON fájl szerkezete | Elég kérdés, jó minőség |
| Projekt határidőre történő lezárása | Beadott játék + dokumentáció | Projektzáró dokumentum | Oktatói értékelés | Sikeres együttműködés |
| Fenntarthatóság és tovább fejleszthetőség | Modularizált forráskód | Osztálystruktúra és MVVM | Kódbázis áttekintése | Jó dokumentáltság |

# Használati útmutató kivonat

A játék elindításakor megjelenik a főmenü:

* **„Játék indítása”** – közvetlenül elkezdi a játékot az első pályán
* **„Pályák”** – kiválasztható: easy (Map1), normal (Map2), hard (Map3)
* **„Kilépés”** – program bezárása

A játék során a térképen számítógépeket lát a játékos. Kattintással indíthat kérdéseket, melyek felugró ablakban jelennek meg.

Válaszadás négy lehetőség közül gombnyomással történik.  
A válasz azonnal értékelésre kerül:

* **Helyes válasz:** +1 / +2 / +3 pont (nehézség alapján)
* **Hibás válasz:** gép zárva marad, új próbálkozás nehezebb kérdéssel

A játék végén megjelenik az összesített:

* **pontszám**
* **hibaszám**
* **időeredmény**

Az eredmények a scores.json fájlban mentésre kerülnek.

# További fejlesztési lehetőségek:

* Időlimites mód bevezetése
* Több kérdéstéma (pl. AI, robotika)
* Többjátékos mód

# Összefoglalás

A Quiz játék fejlesztése a tervezett célokkal összhangban, sikeresen lezárult. A csapat minden tagja aktívan hozzájárult a fejlesztéshez, és a program stabil működése, valamint a teljes dokumentáció biztosítja a beadásra alkalmas állapotot.

Az alkalmazás bemutatásra kész, és további fejlesztési lehetőségekkel rendelkezik.  
A dokumentáció részletesen tartalmaz minden elvárt elemet: funkciók, kódlogika, adatkezelés, UML, tesztelés, kommunikáció, költség és zárás.

## Nehézségek, problémák:

A legnagyobb kihívást a GUI kialakítása jelentette, mivel a játékosbarát és esztétikus felület létrehozása időigényes volt. A csapat napi rutinjai eltérőek voltak, ami néha nehezítette az együttműködést.

## Tapasztalatok:

A projektvezető számára a legnagyobb fejlődési terület a csapatmenedzsment volt: a kommunikáció, feladatkiosztás, ütemezés szervezése. A csapattagok szaktudása jól kiegészítette egymást, és mindenki hozzájárult a közös sikerhez.

## Csapatmunka értékelése:

A munka során mindenki kivette a részét:

* + László: dokumentáció, JSON fájlok
  + Alexandra és Márk: GUI
  + Noémi: háttérkód, logika, JSON fájlkezelés

# Projekt zárása