SZÉCHENYI EGYETEM
UNIVERSITY OF GYŐR

INFORMATIKA TANSZÉK
DEPARTMENT OF INFORMATICS

# SZAKDOLGOZAT

# PlanIt – Play your cards right

**Rozs Norbert**

**Mérnökinformatikus BSc szak**

**2025**

# Feladatkiírás

**Diplomamunka címe:** Online römi

**Hallgató neve:** Rozs Norbert **Neptun-kód**: FY8769

**Szak**: Mérnökinformatikus BSc **Tagozat**: nappali

**Kezdő tanév és félév**: 2024/25 II. **Nyelv**: magyar

**Típus**: nyilvános

**Feladat**:

A römi kártyajáték megvalósítása webes környezetben.

**Részfeladatok**:

1. A szakdolgozat elkészítéséhez szükséges résztevékenységek: A megfelelő szakirodalom megismerése. A játékszoftver főbb funkciói: játékszobák, játékosok (személyek, gépi játékosok) kezelése, többféle játék (pl. rablórömi) és nehézségi szint (pl. kezdő, haladó) biztosítása, súgó (pl. szabályok, használat módja) megvalósítása. A megoldás hátterében lévő adatbázis megtervezése és létrehozása. A játékosok „minősítéséhez" szükséges adatok gyűjtése és azokból statisztikák készítése.
2. **A szakdolgozat főbb részei**: A römi kártyajáték rövid ismertetése (pl. szabályok, játékmenet). A feladatra megtervezett és elkészített programrendszer dokumentációja.

**Belső konzulens**: Pusztai Pál e. adjunktus, Matematika és Számítástudomány Tanszék

# Nyilatkozat

Alulírott, Rozs Norbert (FY8769) Mérnökinformatikus, BSc. szakos hallgató kijelentem, hogy a „PlanIt – Play your cards right” című szakdolgozat feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Győr, 2025 _____

hallgató

# Tartalomjegyzék

## Tartalom

# Bevezetés

A digitális technológia fejlődésével a klasszikus társas és kártyajátékok is egyre inkább áthelyeződnek az online térbe. Ezek az alkalmazások nemcsak szórakozási lehetőséget kínálnak, hanem lehetőséget teremtenek a programozási készségek fejlesztésére, a felhasználói élmény optimalizálására, valamint a többjátékos hálózati kapcsolatok és mesterséges intelligencia integrálására is. A jelen szakdolgozat célja egy olyan webes kártyajáték-platform megvalósítása, amely néhány ismert játék, például römi, uno és pasziánsz, digitális változatát foglalja magába, modern webes technológiák felhasználásával.

A választott játékok különböző típusú játékmenetet képviselnek: a römi stratégiai és logikai gondolkodást igénylő többjátékos játék, az uno egy gyors, pörgős partijáték, míg a pasziánsz egy egyjátékos, türelmet és figyelmet igénylő kihívás. Ezek együttes fejlesztése lehetőséget ad arra, hogy különböző interakciós modelleket, játékszabályokat és felhasználói élményeket vizsgáljunk, valamint ezekhez illeszkedő technikai megoldásokat dolgozzunk ki.

A szakdolgozat során a cél nem csupán a játékok működő webes implementációja, hanem az átgondolt rendszertervezés, a jól strukturált felhasználói felület kialakítása, és a játéklogikák hatékony megvalósítása is. A megvalósítás során figyelmet fordítok a moduláris felépítésre, újrafelhasználható komponensek létrehozására, valamint a skálázhatóság és a karbantarthatóság biztosítására.

A dolgozat első része bemutatja a játékok szabályait, a tervezés során alkalmazott technológiákat és architektúrát, majd részletesen tárgyalja az implementáció folyamatát, beleértve a kliens- és szerveroldali működést, valamint a felhasználói interakciókat. A záró fejezet a tesztelési tapasztalatokat, felhasználói visszajelzéseket és jövőbeli fejlesztési lehetőségeket foglalja össze.

A projekt célja, hogy egy élvezetes, jól használható és technikailag megalapozott webes játékplatformot hozzon létre, amely nemcsak szórakoztató, de fejlesztési szempontból is értékes példája a modern webalkalmazások készítésének.

## Römi

A römi[1] (*angolul Rummy*) egy klasszikus kártyajáték-család, amely világszerte népszerű, és számos változatban ismert. Eredete a 19. századra vezethető vissza, valószínűleg Mexikóból vagy az Egyesült Államokból származik, és valamilyen formában a kínai Mahjong játékból is inspirációt merített. Az idők során számtalan variációja alakult ki, például az amerikai Gin Rummy vagy az indiai Rummy 500, de az egyszerű römi ezek leegyszerűsített, könnyen tanulható változata, amely ideális kezdők számára és gyors játékmenetével könnyen digitalizálható.

## A játék célja

Az egyszerű römi célja, hogy a játékos elsőként szabaduljon meg az összes lapjától, mégpedig úgy, hogy érvényes sorozatokat[1] és szetteket[2] alakít ki a kezében lévő lapokból. A játék végén a kézben maradt lapok pontlevonást eredményeznek, a győztes pedig bónuszpontokat kap.

## Játékosok száma

2–6 játékos

## Pakli

Két standard 52 lapos francia kártyapakli

## Alapfogalmak

[1]Sorozat: legalább három egymást követő értékű lap azonos színből (pl. ♠5, ♠6, ♠7),

[2]Szett: három vagy négy azonos értékű, de különböző színű lap (pl. ♠7, ♥7, ♦7)

## A játék menete

A játék kezdetén minden játékos meghatározott számú lapot kap: az első játékos 15 míg a többi 14 lapot. Az osztás után a megmaradt kártyák képpel lefelé kerülnek az asztalra, ezek alkotják a húzópaklit.

A játék az osztótól balra ülő játékossal kezdődik, és az óramutató járásával megegyező irányban halad. Minden játékos a saját körében három lépést hajt végre: először húz egy lapot, majd opcionálisan lerakhat kombinációkat, például három azonos értékű, de különböző színű lapot (szett), vagy legalább három egymást követő azonos színű lapot (sorozat), végül eldob egy lapot a dobópaklira.

A játékosok csak akkor kezdhetik meg a kombinációk lerakását, ha az általuk létrehozott csoportosítások megfelelnek a szabályoknak. A játék későbbi szakaszában lehetőség van arra is, hogy egy játékos ne új kombinációkat rakjon le, hanem korábban lerakott sorozatokhoz vagy szettekhez illesszen hozzá további lapokat. Ez a rugalmasság lehetővé teszi a stratégiai játékot, hiszen nemcsak saját kombinációkat kell építeni, hanem érdemes figyelni a többiek által lerakott lapokat is.

A játék addig folytatódik, amíg valamelyik játékos az utolsó lapját is sikeresen eldobja, és így "kiáll" a játékból. Ezzel a kör véget ér, és sor kerül a pontszámok kiszámítására. A győztes játékos pontot nem kap, míg a többiek a kézben maradt lapjaik értéke alapján vesznek el pontokat. A számkártyák értékük szerint, a képes lapok (J, Q, K, A) 10 pontot, a Joker pedig 50 pontot ér.

Ez az egyszerű menet rendkívül jól alkalmas arra, hogy digitális formában is megvalósuljon, mivel a játékszabályok logikusan lemodellezhetők, és a felhasználói interakciók egyértelműen leképezhetők webes felületen keresztül.

# Uno

Az UNO[2] egy modern, pörgős kártyajáték, amelyet 1971-ben az Egyesült Államokban alkotott meg Merle Robbins. A játék a klasszikus „Crazy Eights” egyszerűsített és színesített változata, melyet eredetileg családi szórakozásra szántak, ám hamar világsikerré vált. Az UNO-t később a Mattel játékkonszern vásárolta meg és tette világszerte elérhetővé. Könnyen tanulható szabályai és dinamikus menete miatt rendkívül népszerű különböző korosztályok körében.

## A játék célja

A játékos célja, hogy elsőként szabaduljon meg az összes lapjától, miközben taktikusan használja az akció[1]- és speciális[2] kártyákat, hogy nehezítse az ellenfelek dolgát. A forduló végén a győztes pontokat kap a többi játékos kezében maradt lapok értékei alapján.

## Játékosok száma

2–8 játékos

## Pakli

Egy 108 lapos egyedi UNO pakli:

- 4 szín (piros, kék, zöld, sárga)
- Színekre bontva:
    - Számkártyák (0-9, 0 kivételével mindenből két darab)
    - Akciókártya[2]: Kihagyás, Megforgatás, Húzz kettőt (2 darab/szín)
- Speciális[1]:
    - Színváltó kártya: 4 darab
    - Színváltó és Húzz 4et kártya: 4 darab

## Alapfogalmak

[1]Speciális kártyák:

- Színválasztó: kijátszója új színt választ
- Színválasztó és Húzz 4et: kijátszója színt választ, az ellenfél 4 lapot húz

[2]Akciókártyák:

- Kihagyás: a következő játékos kimarad
- Megfordítás: a játék iránya megváltozik
- Húzz kettőt: a következő játékos két lapot húz, és kimarad

### A játék menete

Minden játékos 8 lapot kap, a többi a húzópakliba kerül. Egy lapot képpel felfelé tesznek le kezdőlapnak, ez a dobópakli első lapja. A játék az osztótól balra kezdődik, és az óramutató járásával megegyezően halad, hacsak egy Megfordítás kártya másként nem rendelkezik.

A soron lévő játékos a kezében lévő lapok közül olyat tesz le, amelyik színben, számban vagy típusban megegyezik a dobópakli tetején lévő lappal. Ha nincs ilyen lap, akkor egyet húz a húzópakliból.

A játék során bármely játékos kijátszhat speciális kártyát, ha a szabályok megengedik. Ha valakinek csak egy lap marad a kezében, kötelező hangosan mondania: "UNO!" Ha ezt elmulasztja, és más játékos ezt észreveszi, akkor büntetésből két lapot kell húznia.

A játék akkor ér véget, amikor egy játékos az utolsó lapját is kijátszotta. Ezután pontozás következik.

## Pasziánsz

A pasziánsz[3] (angolul Solitaire), egy klasszikus egyjátékos kártyajáték, amelynek története egészen a 18. századi Európáig nyúlik vissza. Egyes források szerint Franciaországban vagy Skandináviában játszották először, és a játék népszerűsége azóta is töretlen. A digitális korszakban különösen nagy figyelmet kapott, mivel a Microsoft Windows operációs rendszerek egyik beépített játéka lett az 1990-es évektől kezdődően. A legismertebb változata a Klondike passziánsz, amely a számítógépes változatok alapját is képezi.

### A játék célja

A játék célja, hogy az összes kártyát négy különálló alappakliba[1] rendezve elhelyezzük, egyet-egyet minden színhez (♠, ♥, ♦, ♣), növekvő sorrendben ásztól királyig. A játékosnak logikusan és előrelátóan kell átrendeznie a lapokat, hogy hozzáférjen az eltemetett kártyákhoz, és fokozatosan felépítse a négy színsorozatot.

### Játékosok száma

1 játékos

### Pakli

Egy standard 52 lapos francia kártyapakli (Joker nélkül)

### Alapfogalmak

[1]Alappaklik: négy külön gyűjtőpakli, ahová a lapokat ásztól királyig kell sorrendben és szín szerint elhelyezni.

[2]Játéktér: hét oszlop, ahová a lapokat váltakozó színnel és csökkenő érték szerint lehet lerakni (pl. fekete 7-re piros 6).

[3]Kihúzópakli: a maradék, még nem használt lapokat tartalmazó pakli, amelyből új lapokat húzhatunk.

### A játék menete

A játék kezdetén a paklit megkeverik, és az első hét oszlopba leosztanak egyre több lapot: az elsőbe 1-et, a másodikba 2-t, és így tovább, a hetedikbe 7-et. Minden oszlopban csak a legfelső lap van felfordítva, a többi lefordítva marad.

A játékos célja, hogy a játéktér lapjait sorrendben, váltakozó színekkel rendezze, miközben a lapokat fokozatosan áthelyezi az alappaklikba. Csak váltakozó színű és csökkenő értékű lapokat lehet egymásra helyezni a táblán (pl. fekete 8-ra piros 7), és teljes sorozatok is mozgathatók, ha a szabályok engedik. Ha egy oszlop teljesen kiürül, oda csak királyt lehet letenni (vagy királlyal kezdődő sorozatot).

A maradék lapokat a kihúzópakliból lehet egyesével vagy hármasával felfordítani, attól függően, melyik változatot játsszuk. A felfordított lapokat felhasználhatjuk a játéktéren vagy közvetlenül az alappaklikban is, ha szabályos a lerakásuk.

A játék akkor ér véget, ha minden lap az alappaklikba kerül, vagy ha már nincs több érvényes lépés.

# Tervezési dokumentáció

## Technológia háttér

A projekt fejlesztése során a cél egy modern, megbízható, bővíthető és platformfüggetlen webes kártyajáték-rendszer létrehozása volt, amely támogatja a többjátékos interakciót, a pontszámok mentését, illetve egy későbbi online multiplayer architektúra alapjait is leteheti. Ennek érdekében a rendszer full-stack JavaScript környezetben készült, kihasználva a Next.js[4], Node.js[5] és MongoDB[6] technológiák előnyeit.

### Frontend

A felhasználói felület a legújabb verziójú Next.js keretrendszerrel valósul meg, amely a React alapú fejlesztésre épít, de több fontos előnnyel is kiegészíti azt:

- SSG/SSR[7] támogatás: A Next.js rugalmassága révén lehetőség van a

statikus vagy szerveroldali renderelésre, például a ponttáblázat vagy toplisták kiszolgálása esetén.
- App Router architektúra: A legújabb Next.js verzióban bevezetett app/ mappaalapú struktúra lehetővé teszi a könnyebb komponensszervezést, a layout.tsx és page.tsx fájlok révén pedig világos hierarchiát biztosít.
- Képek és statikus fájlok kezelése: A kártyák megjelenítéséhez SVG/PNG formátumú képek használhatók, amelyeket a Next.js Image komponense optimalizáltan tölt be.
- Tailwind CSS[8]: A vizuális stílushoz egy utility-first CSS keretrendszer is bevethető a gyorsabb és konzisztens UI-fejlesztéshez.

## Backend

A szerveroldali műveletek a Next.js API-rendszerén keresztül valósulnak meg, Node.js környezetben:

- API Routes[9]: A játék állapotának mentése, betöltése, a játékosok pontszámainak kezelése és a jövőbeni multiplayer funkcionalitás REST-alapú végpontokon keresztül történik (/api/ útvonalak).
- Hitelesítés: Beépítésre került egyedi és harmadik féltől származó autentikációs megoldás is.

## Adatbázis

Az alkalmazás adatait, például játékállapotok, felhasználók pontjai, statisztikák, egy MongoDB dokumentumalapú adatbázisban tároljuk.

- Mongoose ODM: A MongoDB-hez a Mongoose segédkönyvtár biztosít strukturált sémákat és validációt a backend logikában.
- Kapcsolódás: A mongoose.connect() hívással, környezeti változók (pl. .env.local) segítségével biztonságosan kapcsolódunk a MongoDB Atlas felhőalapú vagy helyi példányához.
- Adatmodellek: Külön sémák definiálják a játékosokat, a játékpartikat és az eseménynaplót, így jól strukturáltan lehet kezelni az adatokat.

## Egyéb fejlesztési eszközök

- TypeScript[10]: A típusbiztonság érdekében a projekt TypeScript-tel készül, amely csökkenti a futási hibák esélyét.
- ESLint[11]: A kód stílusának és minőségének fenntartására automatikus linting és formázás van beállítva.
- Git verziókezelés[12]: A projekt fejlesztése verziókövetéssel, GitHub repóban történik, amely átlátható módon rögzíti a munkafolyamatokat.
- Visual Studio Code[13]: A program megírására szolgáló alkalmazás, amely bővítményeivel segíti a munkát.

## Felhasználói szerepkörök

A fejlesztett webes kártyajáték-alkalmazás többféle felhasználótípust támogat. A rendszer célja, hogy különböző típusú játékosok számára biztosítson játszható környezetet, mind emberi, mind mesterséges szereplők esetén. Az alábbiakban részletezem a rendszerben részt vevő szereplők funkcióit és jogosultságait.

### Játékos

A játékos a rendszer tényleges felhasználója, aki saját eszközéről, böngészőn keresztül csatlakozik az alkalmazáshoz. Csak regisztrált fiókkal rendelkező felhasználóként veheti igénybe.

Jogosultságai és funkciói:

- Saját játék indítása vagy csatlakozás meglévő játékhoz
- Ellenfél választása: másik játékos vagy bot
- Részvétel a játékban: kártyák húzása, lerakása, dobása
- Saját pontszám megtekintése
- Játék közbeni visszajelzések (pl. szabálytalan lépés, győzelem)
- Többféle játékmód közötti választás: Römi, Uno, Pasziánsz

### Mesterséges játékos

A bot szerepkör a gépi ellenfelet jelenti, amely előre meghatározott logika szerint, programozott szabályok alapján vesz részt a játékban. A botok célja, hogy egyedül játszó felhasználók számára is biztosítsák a teljes játékélményt.

Jellemzői és funkciói:

- A rendszer automatikusan hozza létre a botot a játékos választása alapján
- A bot saját "kézben tartott" lapokkal rendelkezik
- Döntéshozatal algoritmus alapján (pl. egyszerű stratégia vagy véletlenszerű döntések)
- Képes kártyát húzni, lerakni, kombinációkat felismerni
- Nem igényel külön hitelesítést vagy UI-t

### Rendszer

A rendszer, vagy más néven játékmotor felel a játék teljes lebonyolításáért és szabályainak betartatásáért. Ez nem egy külön felhasználó, hanem egy belső vezérlő szereplő, amely kezel minden játékállapotot, eseményt és interakciót a felhasználók között.

Feladatai:

- Pakli létrehozása, keverés, osztás
- Játékállapot nyilvántartása (pl. melyik játékos következik, aktuális kéz)
- Érvényes lépések ellenőrzése (pl. csak szabályos lerakás engedélyezése)
- Győzelemfeltételek ellenőrzése
- Pontszámítás, eredmények mentése adatbázisba
- Játék újraindítása, bezárása, hibakezelés

A rendszer Node.js backend oldalon fut, API-kon és WebSocket[14]-en keresztül kommunikál a frontenddel, és MongoDB-ben tárol minden releváns adatot (játékosok, állapotok, statisztikák stb.).

# Irodalomjegyzék


[1] Römi – Wikipédia: (https://hu.wikipedia.org/wiki/Römi), 2025

[2] UNO – Wikipédia: (https://hu.wikipedia.org/wiki/UNO_(kártyajáték)), 2025

[3] Pasziánsz– Wikipédia: (https://hu.wikipedia.org/wiki/Pasziánsz), 2025

[4] Next.js–The React Framework: (https://nextjs.org), 2025

[5] Node.js– Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine: (https://nodejs.org/en), 2025

[6] MongoDB – The Developer Data Platform: (https://www.mongodb.com), 2025

[7] Static Generation vs Server-side Rendering – Next.js Documentation: (https://nextjs.org/learn/pages-router/data-fetching-two-forms), 2025

[8] Tailwind CSS – Rapidly build modern websites without ever leaving your HTML: (https://tailwindcss.com), 2025

[9] API Routes – Next.js Documentation: (https://nextjs.org/docs/pages/building-your-application/routing/api-routes), 2025

[10] TypeScript – Typed JavaScript at Any Scale: (https://www.typescriptlang.org), 2025

[11] ESLint – Find and fix problems in your JavaScript code.: (https://eslint.org), 2025

[12] Git – Distributed Version Control System.: (https://github.com), 2025

[13] Visual Studio Code – Code editing. Redefined.: (https://code.visualstudio.com), 2025

[14] WebSocket – MDN Web Docs. (https://developer.mozilla.org/en-US/docs/Web/API/WebSocket), 2025