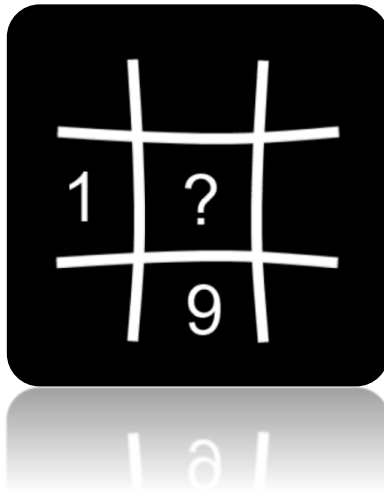


Házi feladat dokumentáció

Android alapú szoftverfejlesztés

[Félév: 2020 tavasz]



Sudoku

Fekete Ákos – N6Z4HP

akosroland25@gmail.com

Laborvezető: Gazdi László



Bemutatás

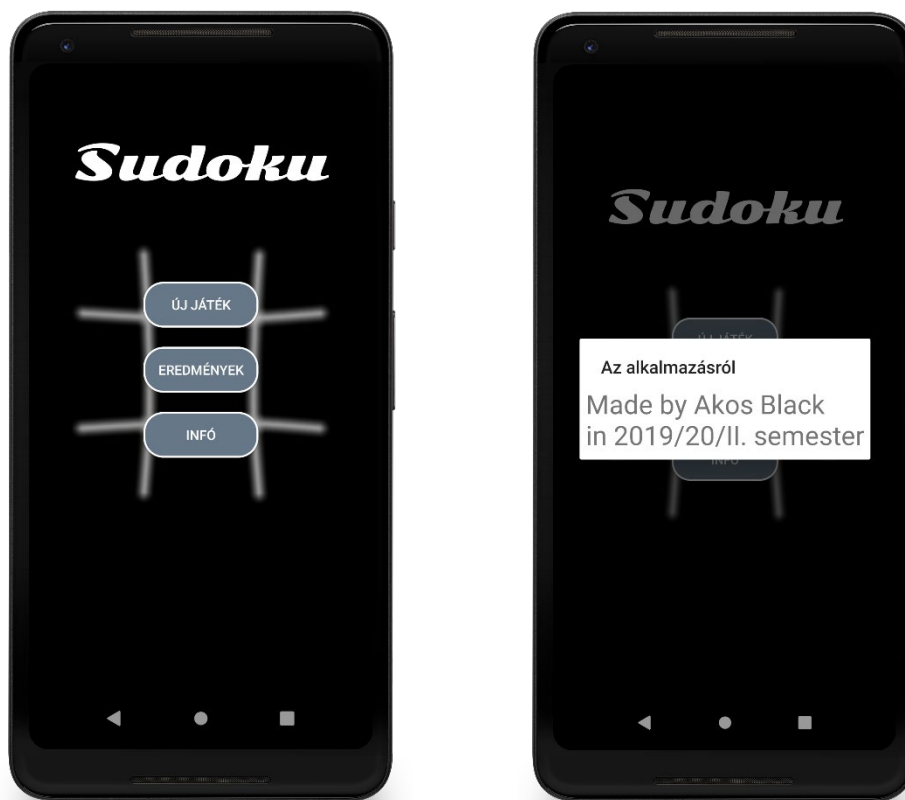
Mivel szeretek sudokuzni, szerettem volna írni egy saját Sudoku játékot Androidra, aminek a pályagenerálásához saját algoritmust használok. Maga a játékot szeretném felhasználó baráttá tenni, így például sok féle képen tud a játékos „segítséget” kapni a játéktól.

Főbb funkciók

Az alkalmazás egy főmenüvel indulni, ami egy ilyen japán, minimalista stílust próbál megidézni, indításkor a játék neve, illetve a háttérben lévő grafika animálva megjelenik. Ebben a főmenüben van lehetőség új játék indítására, illetve a korábbi eredmények megtekintésére, illetve pár információ is elérhető az alkalmazásról.

- A felhasználói interakciókat kezelő felület és azokat feldolgozó algoritmus.
- A Sudoku kitöltésének helyességet kiértékelő algoritmus
- Az eredményeket az eszközön perzisztensen tárolja
- Kitöltést segítő funkciók (a választott szám kiemelése, menet közbeni ellenőrzés)
- Ha a játék 2 napja nem volt elindítva, értesítést küld, játékra ösztönöz
- Ha játék közben meggondoljuk magunkat, és új Sudoku szeretnénk kezdeni, rögtön játék közben a Toolbaron van rá lehetőség
- Különböző nehézségi szintek alapján történő pályagenerálás

Felhasználói kézikönyv

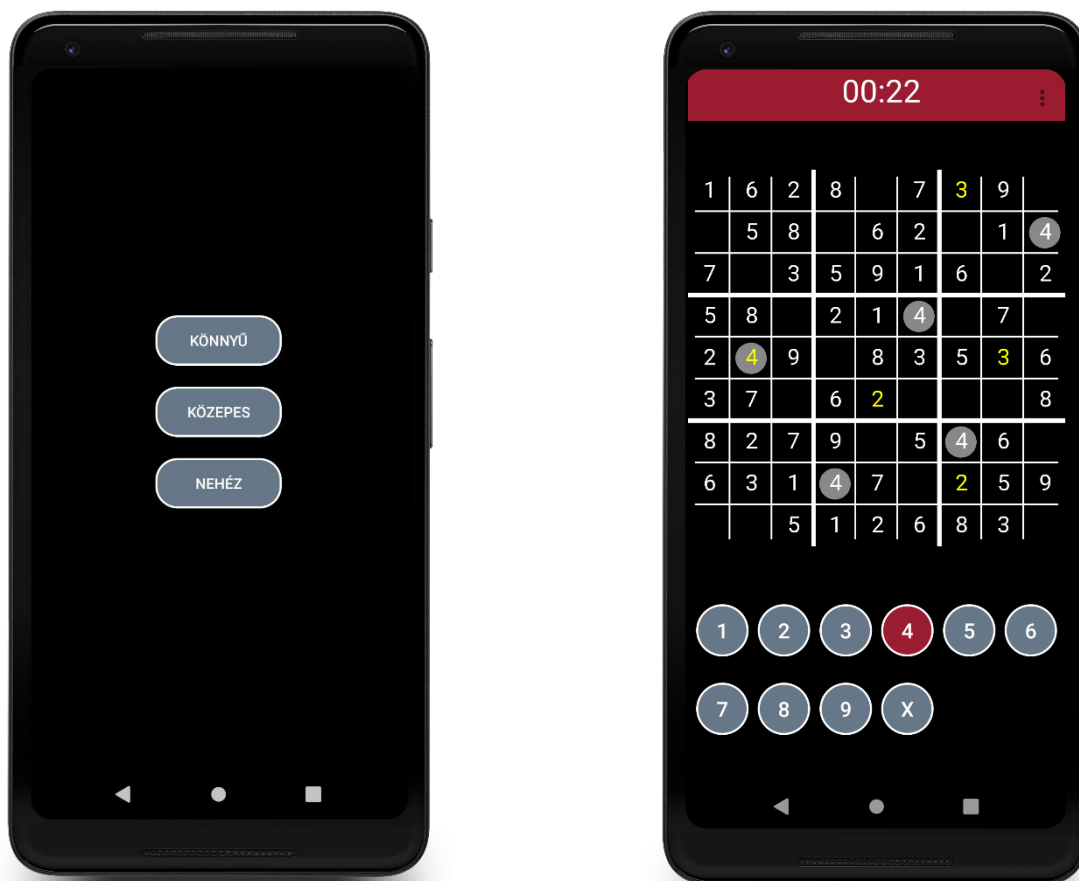


1. ábra: A Főmenü látható, ahonnan a többi almenü pont elérhető, mellette az pár információ az alkalmazásról

Az alkalmazás a Főmenüvel indul, innen a felhasználó további menüpontokra navigálhat, kezdhet új játékot, megnézheti az eredményeket.

Az INFÓ gomb segítségével, az alkalmazásról pár adatot tudhat meg a felhasználó, ahogy az a fenti ábrán is látható.

Ha új játékot szeretnénk indítani, akkor az ÚJ JÁTEK menüpont segítségével tovább jutunk a Nehézség választó felületre.



2. ábra: A Nehézségválasztó látható, ahonnan a kívánt nehézség kiválasztása után, annak megfelelő pályát generál, és visz tovább a játék felületére.

Miután a nehézséget kiválasztottuk, egyből a játéktérbe dob az alkalmazás. Az alsó gombokkal beírhatja a játékos a kívánt számait a Sudoku pályába. A játék segít a megoldásba, amit szeretnénk számot beírni a pályára, abból a már lent lévőket egy szürke körrel ki is emeli, illetve a játékos által felvitt számok, sárgával külön jelezve vannak. Törölni az „X”-el lehet, itt is külön jelzésre került, hogy melyik elem törlése megengedett.

Ha a pályát sikeresen megoldotta a felhasználó, választhat 3 japán harcos maszkja közül, illetve írhat be saját nevet, ami megjelenik majd az Eredmények fül alatt.

Az Eredmények fül alatt az eszközre mentett eredmények vannak, amit görgetni lehet, ha esetlegesen már nem tartunk számot egy eredményre, annak törlésére lehetőség van. Az Eredményeket az alábbi ábrán mutatja.



3. ábra: Az Eredmények menüpontban megtekinthetők az eddigi eredmények, ezt görgetni is lehet.

Felhasznált technológiák

Az alkalmazáshoz elkészítéséhez az alábbi fontosabb technológiákat használtam:

- **Canvas** használata a játék elemeinek a megrajzolásához
- **RecyclerView** használata az Eredmények görgetéshez
- **Internal Storage** az Eredmények tárolásához
- **Notification** használata a játékos ösztönzésére
- **Pending intent** majd **BroadcastReceiver**rel való elkapása
- **Toolbar** használata az játék újratekésítéséhez
- **Saját algoritmus** a Sudoku pálya generálásához

Fontosabb technológiai megoldások

Az elején egy egységes témát akartam adni a programomnak, ami végül egy ilyen minimalista fehér/fekete/piros színekombinációt öltött, illetve egy ilyen japán vonalat is igyekeztem behozni, ami sajnos leginkább csak a választható ikonokban nyilvánult meg.

A legnagyobb nehézsége a feladatomnak mégis a **Sudoku pálya generálása** volt. Ehhez végül egy cselhez folyamodtam, felhasználva azt hogyha egy kész, helyesen kitöltött Sudoku pályát 3-om oszlop és sor csoportra bontva, egy csoporton belül két oszlop/sor megcserélésével nem változik annak helyessége, illetve megoldhatósága sem. Ezek a cserék csoporton belül természetesen véletlenszerűen vannak kiválasztva, véletlen sokszor, de legalább 75-ször, hogy teljesen biztos legyen az eredmény. Ezután a program véletlenszerűen kitöröl elemet a pályáról, annyit, amennyi a nehézségi szintnek megfelelő (sajnos itt valami féle hiba van, mert nem pontosan annyit, de teszteléseim alapján ez maximum 2-4 elembe tér csak el) Így a játékos csak azt veszi észre, hogy mindig egy új, máshogyan kinéző pályát kap. A másik nagy előnye ennek a módszernek, hogy rendkívül gyorsan lefut tud futni algoritmus.