

Reversi

Cseppentő Viktor, Kőrösi Dániel, Ország Bence

2020. 05. 22.

1 Dokumentáció

1.1 Áttekintés

1.1.1 Az alkalmazás funciói

1. Egy- és többjátékos mód

A felhasználó játszhat egyjátékos módban AI ellen és többjátékos módban egy számítógépen egy másik játékosal, illetve online helyi hálózaton egy másik játékosal. A hálózati játék megkezdése előtt a host megadja a felhasználónevét és a játékbeállításokat, a kliens pedig a nevét és a host IP címét, mely segítségével kapcsolódik a host-hoz.

A játékmódot és a beállításokat a főmenüben adhatjuk meg.

2. Többszintű nehézségi szint

A felhasználó az egyszemélyes játékmódban több nehézségi szinten tud a számítógép ellen játszani. Minden módban lehetőség van maximális lépésre fordítható idő beállítására, ezzel is finomhangolva a nehézségi szintet.

3. Mentés és betöltés

Egyjátékos módban lehetőség van a játékállás elmentésére és későbbi betöltésére. Minden módban elmentésre kerülnek a végeredmények, amik megjeleníthetők a program "Statistics" fülén.

1.2 Felhasználói felület

1.2.1 Menü

A játék indítása és a beállításai is mind grafikus felületen történnek.

A főmenüben három tab van, ezek a "Singleplayer", "Multiplayer", "Statistics".

1. Singleplayer

Ezen a tabon lehet kezdeményezni számítógép elleni játékot. Amennyiben az idő 0-ra van beállítva, nincs időkorlát.

Az elérhető beállítások:

- Játékos színe
- Nehézségi szint (könnyű, közepes, nehéz)
- Időzítő (0-20 perc)

2. Multiplayer

Ezen a tabon lehet elindítani lokális vagy helyi hálózati játékot.

A kapcsolódás megkönnyítése érdekében megjelenítésre kerül a számítógép lokális IP címe, ami megadható az ellenfél játékosnak.

Megadható név is, de ha üresen hagyjuk, akkor a játékot kezdeményező játékos neve "Host", a kihívott játékos neve "Client" lesz.

A kezdeményező játékos által állítható paraméterek:

- Az ő színe
- Név
- Időzítő

Ha a beállítások megfelelőek a "Host Game" gombra kattintva lehet várakozni az ellenfél játékos csatlakozására.

Az ellenfél játékosnak pedig meg kell adnia a host IP címét, és a "Connect" gombra kattintva tud csatlakozni a játékhoz.

3. Statistics

Az korábbi játékok eredményét tartalmazó tab.

Egy befejezett játékról tárolásra kerül

- a játék típusa
- a játékosok nevei
- a nyertes

1.2.2 Játék közben

A játék megkezdésekor megjelenik a tábla, az egyes korongok a táblán lévő mezőkre kattintással helyezhetők el. Kiemelésre kerülnek azon mezők, amelyekre érvényes lépés tehető.

A képernyő jobb oldalán látható a játék aktuális állása, a soron következő játékos minden esetben kijelölésre kerül, így abban az esetben, amikor nincs elérhető érvényes lépés, akkor azonnal a jelenlegi játékos léphet újra.

A játékalás alatt található az "Exit" gomb, amellyel azonnal ki lehet lépni a játékból. Számítógép elleni játék esetén a kilépés során a játékalás automatikusan mentésre kerül, ami később visszaállítható. Helyi, vagy hálózati játék esetén a kilépés a játék feladásával jár.

1.3 Főbb osztályok és feladataik

- MenuController
 - Főmenü megjelenítése
 - Beállítások módosítása
 - Játék elindítása
- GameController
 - Az aktív játék során kezeli a grafikus felületet, azaz megjeleníti az aktuális játékalást, a hátralévő időt, és a játékosok neveit.
 - Kezeli a kilépést.
- PopupController
 - A hálózati játék esetén a játékkérélmeket jelzi
- Drawer A játékeret egy ún. AnimationTimer modulból származtatott le, így ez is megvalósít egy időzítést. A játéktér frissítése is ebből származik, körülbelül 50 FPS-enként frissíti a játékeret. Minden frissítésnél kirajzoljuk a mezőket és a rajta lévő korongokat.

A játék befejezésénél megáll ez a külön szálon futó folyamat, és a végső frissítésnél kiírja a győztest.
- GameLoop
 - A játék vezérlése
 - * Tábla létrehozása a játék megkezdésekor

- * Egyjátékos mód esetén fogadja a mesterséges intelligencia válaszait
 - * Többjátékos mód esetén feldolgozza a másik játékos lépéseit
- Board
 - A játéklógika megvalósítása
 - Az érvényes lépések meghatározása
 - Az aktuális játékállás eldöntése
- Player
 - Absztrakt osztály, ebből származik le a LocalPlayer, AiPlayer és az OnlinePlayer
 - Egyszemélyes mód esetén a játéktábla alapján a mesterséges intelligencia következő lépését adja meg
- NetworkConnection
 - Felveszi a kapcsolatot a másik játékosal
 - Várja a másik játékos lépéseit

1.4 Használt technológiák

- A grafikus megjelenítéshez JavaFX könyvtár és Scene Builder program használata
- Külső beállítások beolvasására Properties API
- Hálózati kapcsolat kezelésére a java.net könyvtárat használjuk

1.5 Állítható táblaméret

A játék elindulásakor beolvasásra kerül a data/config.properties fájl, amit a Properties API-val dolgozunk fel. Ez a fájl az összes játékra érvényes beállításokat tartalmazza, jelenleg csak táblaméretet.

1.6 Szálak

Összesen 4 fő szálon fut az alkalmazás, ezek az alábbiak:

- Fő szál: a játékos inputját várja a grafikus felületen, majd feldolgozza azt, vagy átadja egy másik szálnak.

- Rajzoló: a játéktáblát rajzolja ki aktív játék közben. Ez a szál futtatja a képfrissítésért felelős "handle" függvényt. A szálát a JavaFX kezeli.
- Játék: a játék megkezdésekor jön létre, és a játéklógika futtatásáért felelős, így egészen a játék végéig él. Felelős a rajzoló szál elindításáért, és leállításáért.
- Hálózati: a netes játék esetén jön létre, az üzenetek küldéséért és fogadásáért felelős.

1.7 Mesterséges intelligencia

Gép elleni játékhoz Monte Carlo szimulációt alkalmaztunk Zobrist hash-eléssel. Az AI-nak véges számú lépése van egy körben. Minden lehetséges lépéshez sokszor végigszimuláljuk a játékot véletlenszerű lépésekkel. Minden szimulációnál feljegyezzük, hogy ki nyerte a játékot és ez alapján hozzáadunk egy értéket a kezdeti lépéshez (1 ha az AI nyert, 0.5 ha döntetlen, 0 ha a játékos nyert). Ezt az értéket hozzáadjuk még minden köztes játékálláshoz is, emiatt a játékállásokat el kell menteni egy lookup table-be. A lookup table-be a játékállásokból egy hash-t képzünk, hogy kevesebb memóriát használjunk, és a hash segítségével jegyezzük fel a hozzá tartozó értéket. A hash-elésre Zobrist hash-t használunk, így nem kell minden álláshoz újragenerálni a hash-t, hanem az előző állásból XOR műveletek segítségével a módosult korongok alapján ez megtehető. Egy következő állásban az AI-nak lehet olyan lépése, melyet már korábbi szimulációban vizsgált, ezért nagyobb biztonsággal megtudja állapítani, hogy mennyire jó az a lépés a lookup table segítségével. Az AI azt a lépést választja, melyhez nagyobb UCT (Upper Confidence Bound 1 applied to Trees) érték tartozik. $UCT = \frac{w}{n} + c\sqrt{\frac{\ln(N)}{n}}$, ahol w a lépéshez tartozó érték, n a lépéshez tartozó szimulációk száma, N a szülőállapothoz tartozó szimulációk száma, c pedig egy konstans paraméter, általában $\sqrt{2}$.

A nehézségi szintek közötti különbség ott nyilvánul meg, hogy a magasabb szintű AI több ideig szimulál, így nagyobb eséllyel találja meg az optimális lépést.

1.8 Hálózati kapcsolat implementációja

A hálózati kapcsolathoz a java.net csomagot használjuk és eseményvezérelt programozást alkalmazunk. A host játékos létrehoz egy TCP szerverszocketet, melyhez a kliens kapcsolódhat. Az üzenetküldést és fogadást egy külön szálon implementáltuk, a küldendő, illetve fogadott üzeneteket egy-egy FIFO-ba tesszük, melyekhez több szálon keresztül hozzáférünk. A FIFO-kat a Net-

workBroker osztály deklarálja, és ezt az osztályt használja minden üzenetküldő, fogadó objektum figyelve a konkurens használatra. String üzeneteket használunk és minden üzenettípushoz rendelünk eseményfüggvényeket, melyeket megfelelő érkezett üzenet alapján meghívunk.

Négyféle üzenettípus van:

- serverstart: akkor küldjük, amikor a kliens kapcsolódott a host-hoz és a host elküldi a játékbeállításokat
- clientstart: akkor küldjük, amikor a kliens elfogadta a játékot, visszaküldi a kapott beállításokat és a nevét
- move: ezzel küldjük el a lépett lépést (pl. move;3;4)
- stop: akkor küldjük ha vége a játéknak, vagy elhagytuk a játékot. Ilyenkor minden játék, és hálózat szálát befejezünk.

Egy üzenettípushoz több esemény is tartozhat.

2 Tesztelés

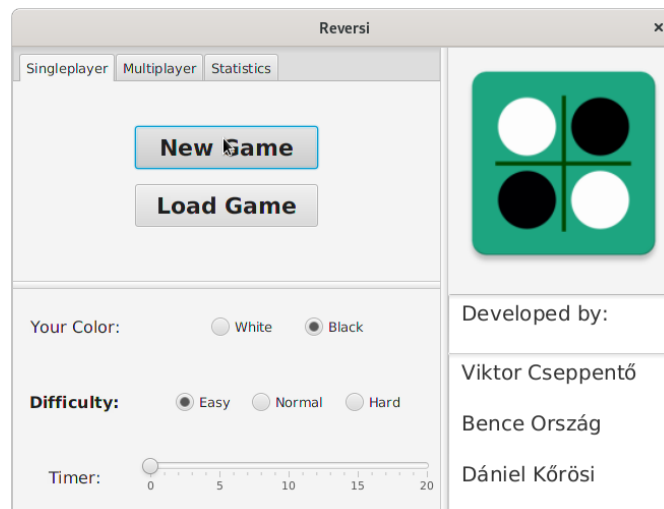
Az alkalmazás funkcióinak helyes működése egyszerű teszteléssel került megálapításra. A tesztelés során a grafikus felületen hajtottunk végre input szekvenciákat, amelyeknek feljegyeztük a hatását, és hogy mi az elvárt működés az adott esetben.

2.1 Egyjátékos mód

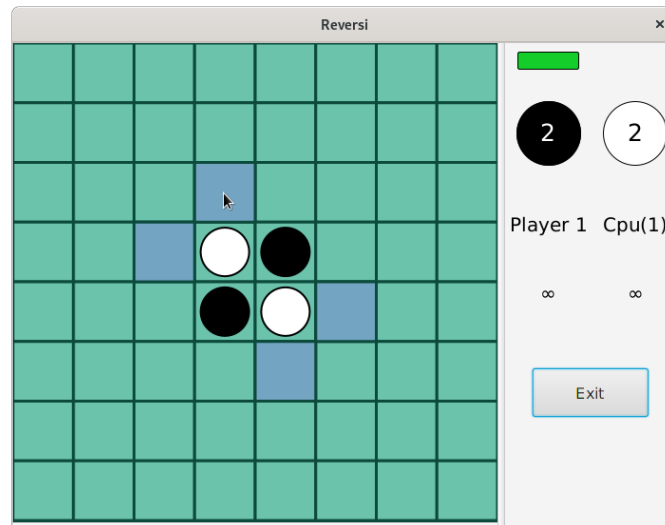
Ebben az esetben azt ellenőrizzük, hogy az egyjátékos mód megfelelően elindul-e, illetve a játékállás elmentése, majd visszaállítása helyesen történik-e meg.

A felületen végrehajtott lépések:

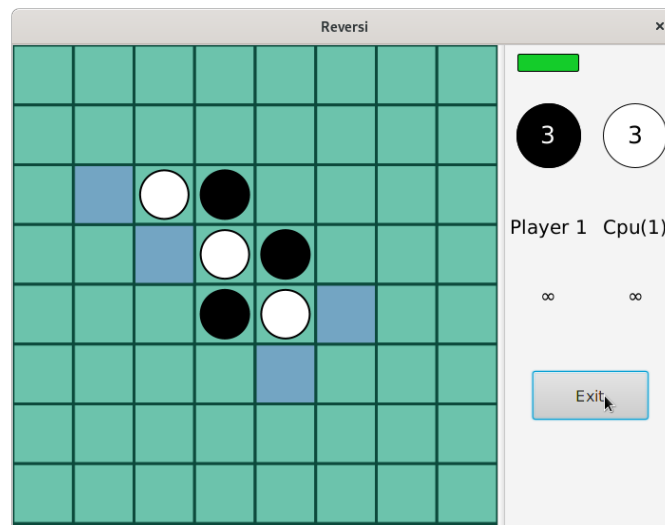
- A játék elindítása alapbeállítások mellett.



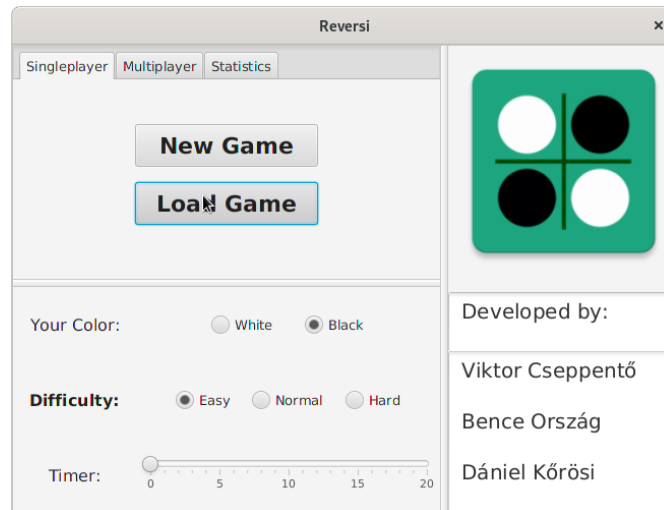
- Egy lépés megtétele.



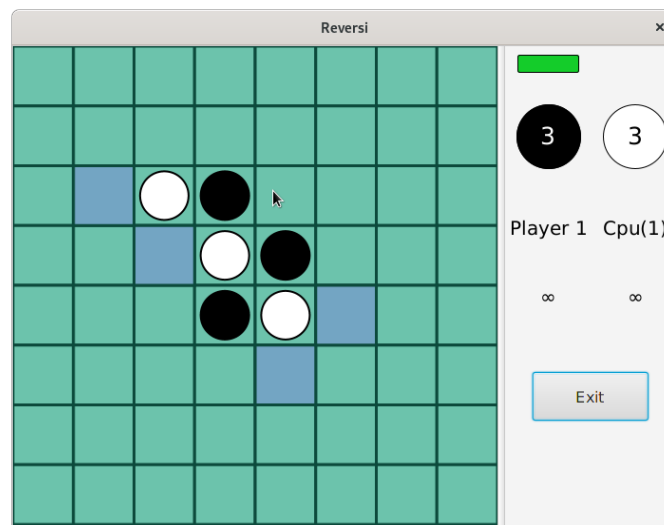
- Kilépés a játékból.



- Az elmentett játékállás visszatöltése



- A visszatöltött állapot ellenőrzése.

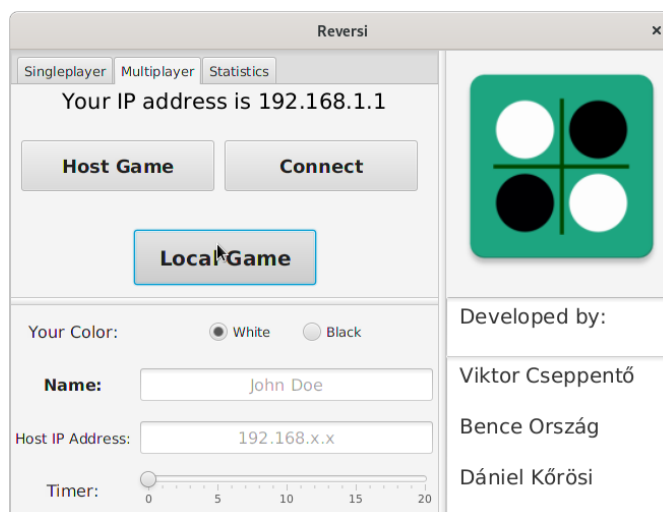


2.2 Lokális játék

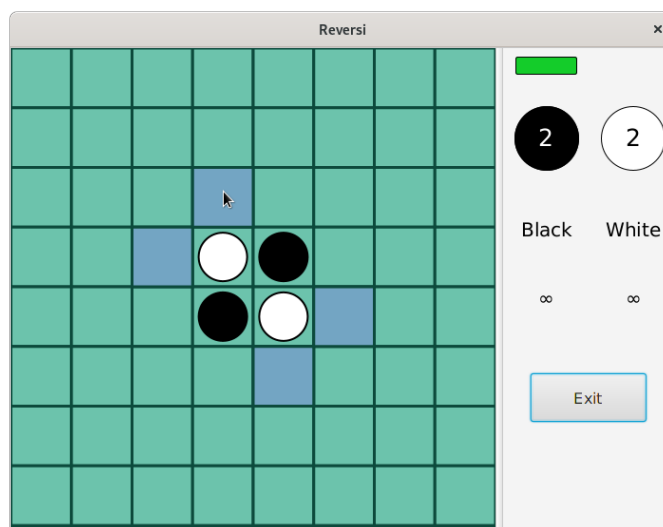
A teszt során a statisztikák frissítését ellenőrizzük. Az egyik játékos kilép, ezzel feladva a játékot.

A felületen végrehajtott lépések:

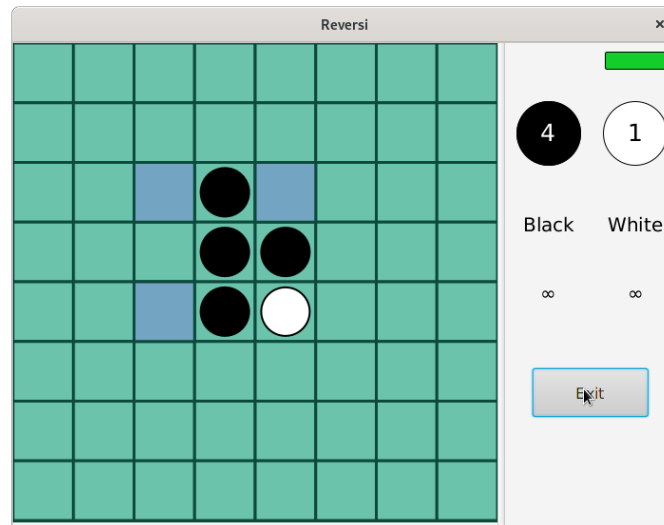
- A játék elindítása alapbeállítások mellett.



- Egy lépés megtétele után kilépés a játékból, ezzel feladva azt.



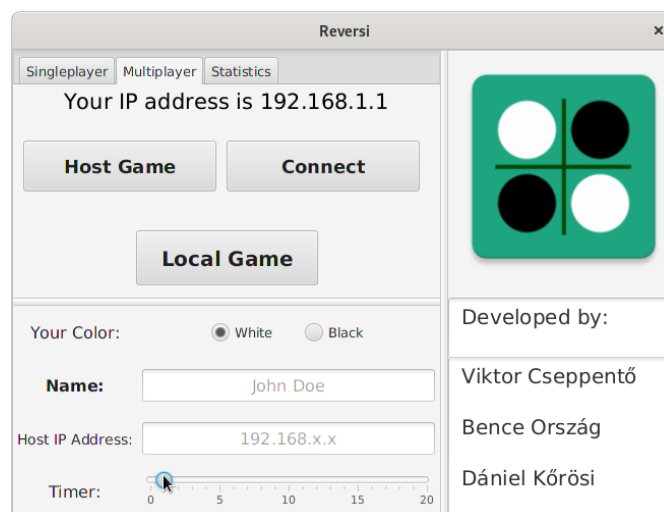
- A "Statistics" tabon ellenőrizni, hogy jelen van-e az új játék eredménye.



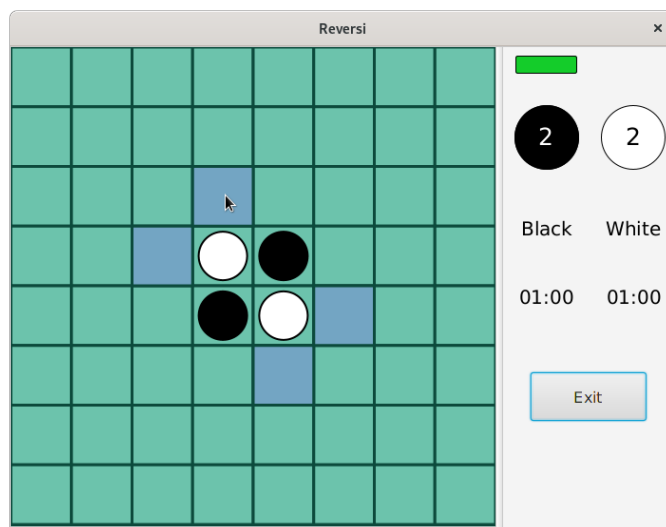
2.3 Lokális játék időzítővel

A teszt során az időzítő helyes működését ellenőrizzük. A felületen végrehajtott lépések:

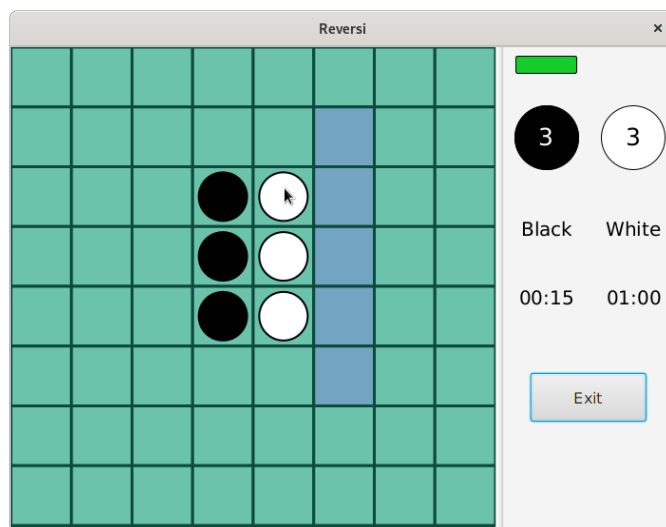
- Lokális játék elindítása alapbeállítások mellett.



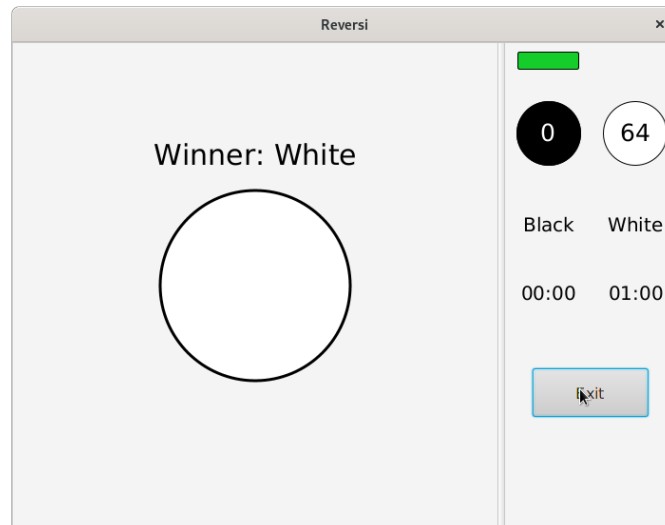
- Egy lépés megtétele, hogy elinduljon az időzítő.



- Az időzítő megvárása



- Ellenőrizni, hogy valóban az ellenfél győzött-e.



2.4 Hálózati játék

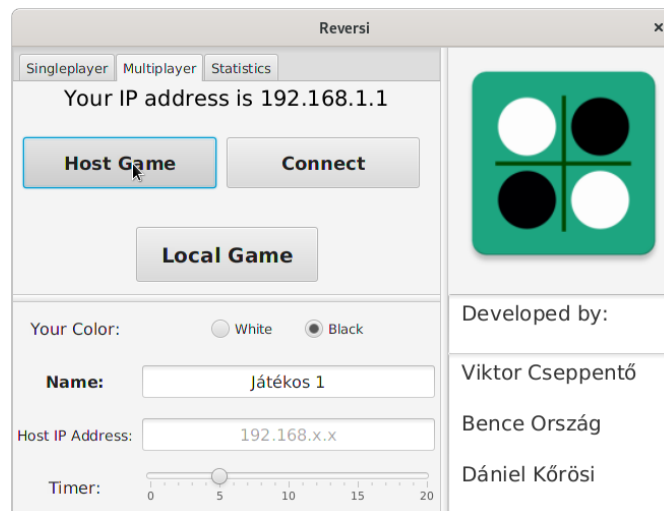
Az alkalmazást két példányban elindítva (a fejlesztői környezet ezt lokálisan képes megtenni).

Az egyik példány indítja el a játékot, a kezdő színe a fekete, a beállított név "Játékos 1", és az időzítő 5 perc.

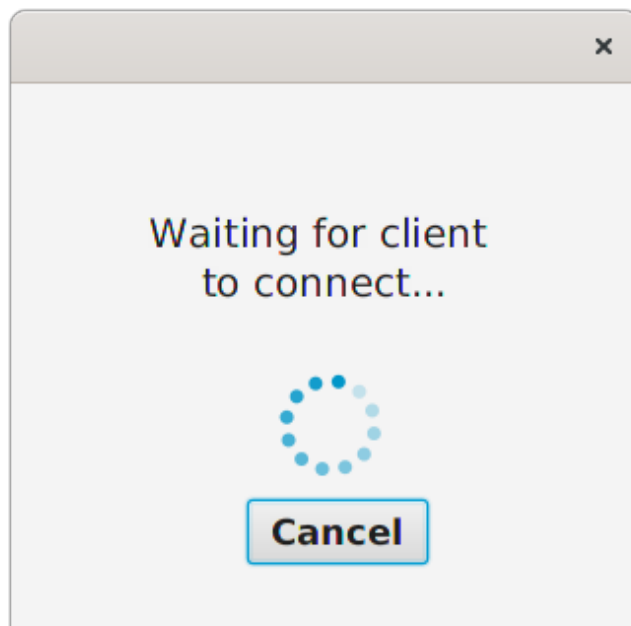
A másik példányban a nevet kell beállítani("Játékos 2"), és az IP címét a másiknak (ami éppen a localhost, hiszen egy gépen futnak, így üresen is hagyható a mező).

Az első felületen végrehajtott lépések:

- Multiplayer tabon a fenti beállítások megadása. "Host Game" gombra kattintással a várakozást jelző pop-up ablak előhívása.

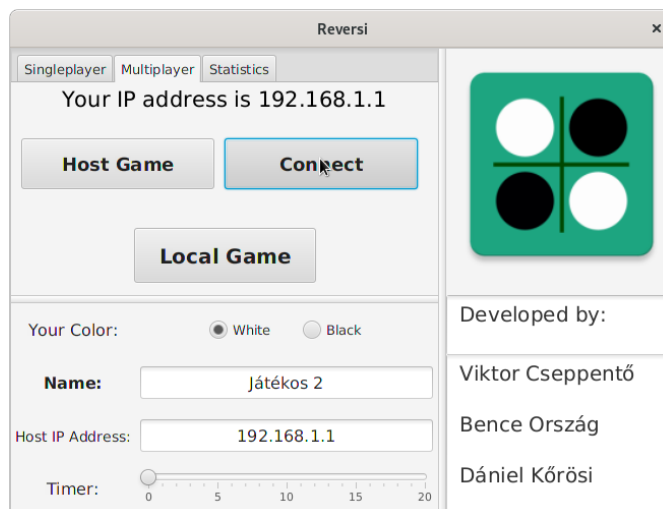


- Várakozás a másik játékos kapcsolódására.

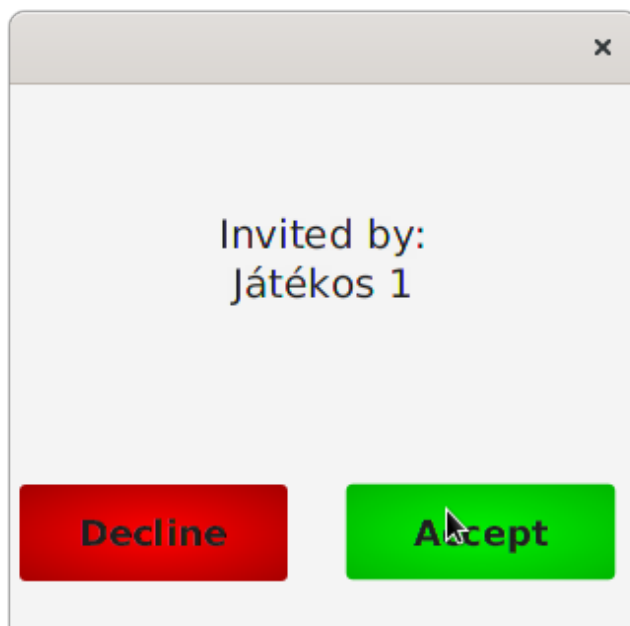


A második felületen végrehajtott lépések:

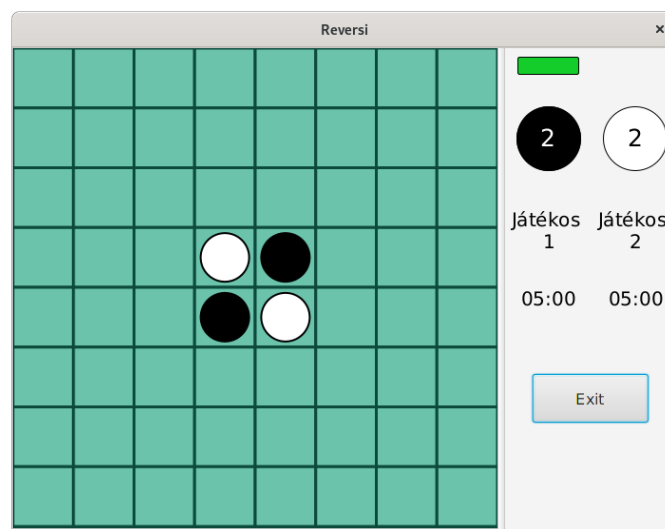
- Várakozás amíg a kezdeményező játékos elindítja a várakozást bejövő kapcsolatra.
- Csatlakozás



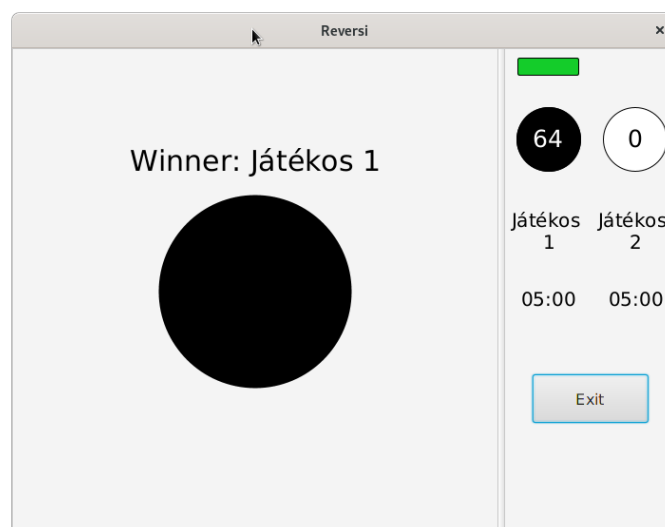
- Majd a kapcsolat elfogadása.



- Játék



- Kilépés



2.5 Tábla méretének módosítása

A végrehajtott lépések:

- /data/config.properties fájl módosítása: `board_dim= 8 => 12`
- Alkalmazás elindítása
- Lokális játék indítása

