Java nagyházi

Németh Dániel, Gelencsér András

2022. március 17.

Verziók

- 1.0 Első verzió
- 1.1 Határidők

Határidők

- **2022.04.04.** 24:00 - **Check Point** (opcionális)
  Aki feltölt valami működő kódot (fordul, fut, kipróbálható) annak véleményezzük.
- **2022.05.16.** 24:00 - **Final**

1. Bevezetés

A nagyházi során a félévben tanult ismeretek segítségével kell elkészíteni egy felhasználói felülettel is rendelkező programot. A program a Color Flip nevű játékot kell, hogy megvalósítsa. Rendelkeznie kell felhasználói felülettel, ahol indítható és játszható a játék. GUI nélküli módban indítva a konzol segítségével is lehessen játszani. Elvárt funkciók: legyen cheat (solver) lehetőség, mely megoldja a játékos helyett a játékot, aktuális játék mentési lehetőség, játék idő mérés.

2. Color Flip játékszabályok

Adott egy $n \times n$-es négyzetes mátrix. Mindegyik cellának két színe lehet. A játék kezdetekor véletlenszerűen töltődnek fel színekkel a cellák. A játék célja, hogy minden cella azonos színű legyen (mindegy, hogy melyik szín). Egy cella megnyomásával az összes, a megnyomott cella sorában és oszlopában lévő cella színt vált.

Segítség: $2n \times 2n$-es mátrix esetén garantált, hogy megoldható a játék, páratlan elemű oldal esetén nincs garantálva, így ekkor úgy kell generálni a pályát, hogy biztosan megoldható legyen a játék. Ehhez érdemes a megoldott pályából visszafele haladni és előállítani a kezdeti „véletlenszerű" állapotot.

3. Tervezett pontozás

Részfeladat	Pontszám
Játék algoritmus	125
Console-os működés	50
GUI működés	100
Solver páros n esetén	75
Solver páratlan n esetén	50
Összesen	400

Fontos megjegyezni, hogy a játéknak minimális szinten működni kell, használhatónak kell lenni ahhoz, hogy pontot lehessen kapni bármely részfeladatra. Tehát például a játék algoritmus működése nélkül a GUI vagy Console-os működés értékelhetetlen. **GUI nélkül a nagyházi értékelhetetlen.**

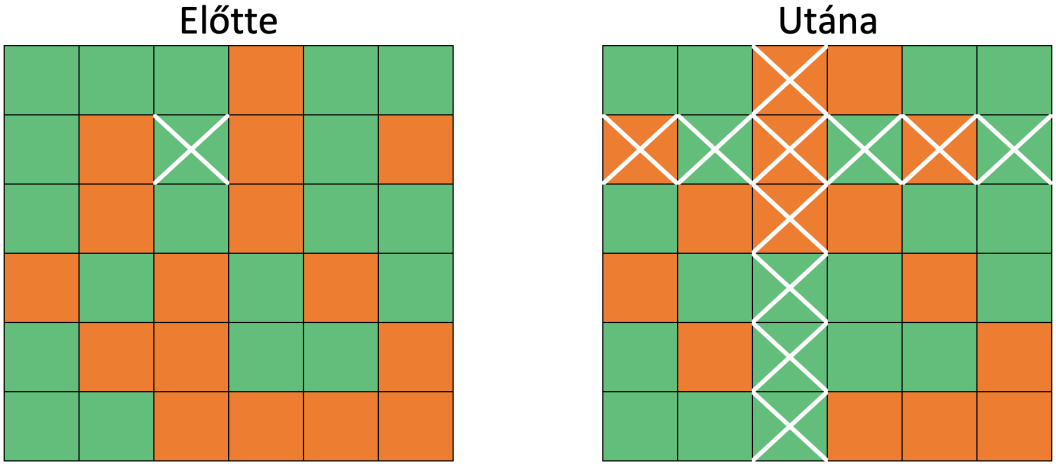
4. Részletes leírás

4.1. Játék működése

Adott egy $n \times n$-es négyzetes mátrix. Mindegyik cellának két színe lehet. A játék kezdetekor random töltődnek fel színekkel a cellák, kivéve páratlan n esetén. Páratlan n esetén megoldott állapotból kell visszafele „generálni" egy kezdeti állapotot, így garantálható, hogy megoldható a játék. Páros n esetén biztosan létezik megoldás, így ekkor random generált cellákból kell kiindulni. A játék célja, hogy minden cella azonos színű legyen.

Egy cella megnyomásával az összes, a megnyomott cella sorában és oszlopában lévő cella színt vált.

Az alábbi ábra mutat erre egy példát:



A baloldali a megnyomás előtti állapotot mutatja ahol fehér X jelöli a megnyomásra kerülő cellát. A jobboldalin a megnyomás utáni állapot látható, a fehér X-ek jelölik cellákat ahol a szín megváltozott.

4.2. Játék menete

4.2.1. Console

Ha a program indításánál meg van adva a `-console` vagy `-nogui` argumentum, akkor a játék GUI nélkül, Console módban indul.

1. Program indítása, az arguments alapján Console működés indítása
2. A játék a `start` paranccsal indítható és az *exit* paranccsal zárható be a program.
   - A program kínálja fel, hogy új játékot vagy egy megkezdett régit szeretne folytatni a játékos.
   - Ekkor új játék választása esetén meg kell adni a pálya méretét. Az elfogadott pályaméretek: $2 \leq n \leq 20$
   - A megfelelő pályaméret megadása után a program legenerálja a pályát:
     - Páros n esetén a pálya celláinak színe random legyen generálva.
     - Páratlan n esetén a megoldott pályából (ugyanolyan színű minden elem) kell random számú (r: $\frac{n}{2} \leq r \leq n^{1.5}$) cellát 'megnyomni' és ezzel előállítani a kezdeti állapotot.
   - A pálya legenerálása/betöltése után a console-ra kell kirajzolni a pályát.
3. A játékos a megnyomandó gomb koordinátáinak megadásával tud cellát megnyomni.
   - Az első koordináta a bal felső saroktól számolt sorindex (az indexelés 1-gyel kezdődjön), a második koordináta a bal felső saroktól számolt oszlopindex (szintén 1-gyel kezdődjön az indexelés)
   - Példa parancs: `1 1`, vagy `4 2`
   - Érvényes parancs/koordináta esetén a program a játékszabályok alapján végrehatja a lépést. Érvénytelen koordináta esetén figyeleztesse a játékost a helyes parancsra.
   - Ha a játéknak nincs vége, akkor a console-ra kell kirajzolni a lépés utáni pályát.
4. Ha a játékos a lépésével megnyerte a játékot (az összes cella azonos színű), akkor erről a program írjon üzenetet a console-ra és írja ki, hogy hány lépésből sikerült megoldani a játékot. Ezután új játék indítható a `start` paranccsal.
5. Ha a játékos a feladja a játékot, az `exit` paranccsal tud kilépni a programból, a `stop` parancssal pedig be tudja fejezni a futó játékot, de új játékot tud indítani a `start` paranccsal. Ha abba kell hagynia a játékot, de később még folytatná, akkor a `save` paranccsal mentheti a pályát és aktuális állapotát.
6. Játék közben a `solve` paranccsal lehet indítani a solver-t ami kiírja a console-ra a megnyomandó cellák koordinátáit, amivel a játékos meg tudja nyerni a játékot.

4.2.2. GUI

A program alapértelmezetten GUI üzemmódban indul. A GUI szálon nem szabad nagyobb számításokat végezni. A GUI-nak reszponzívnak kell lennie, nem szabad lefagyni.

1. Program indítása, az arguments alapján GUI működés indítása

2. A játék jelenítsen meg egy menüt ahol van lehetőség bezárni a programot, új játékot indítani valamint régi játékot betölteni.
   - Új játék esetén meg kell adni a pálya méretét. Az elfogadott pályaméretek: $2 \leq n \leq 20$
   - A megfelelő pályaméret megadása után a program legenerálja a pályát:
     - Páros n esetén a pálya celláinak színe random legyen generálva.
     - Páratlan n esetén a megoldott pályából (ugyanolyan színű minden elem) kell random számú (r: $\frac{n}{2} \leq r \leq n^{1.5}$) cellát „megnyomni" és ezzel előállítani a kezdeti állapotot.
   - A pálya legenerálása után jelenítse meg a pályát, minden cella kattintható legyen. Valamint jelenítsen meg és indítson el egy órát, ami méri a játék során eltelt időt.
3. Az adott cella megnyomásánál a szabályoknak megfelelően változzon a pálya.
4. Ha a játékos a lépésével megnyerte a játékot (az összes cella azonos színű), akkor erről a program értesítse a játékost és jelenítse meg, hogy hány lépésből és mennyi idő alatt sikerült megoldani a játékot. Ezután lehet új játékot indítani.
5. A játék során a játékosnak bármikor van lehetősége, elmenteni, befejezni a játékot vagy a teljes programot. Ezeket az eseteket a programnak tudni kell kezelni.
6. Játék közben legyen lehetőség a solver indítására. Ekkor a program jelezze a cellákon, hogy mely cellák lenyomása vezet győzelemhez. A jelzések frissüljenek ahogy a felhasználó lenyom cellákat. Legyen lehetőség a solver kikapcsolására is.
7. Legyen lehetőség többféle (előre megadott) színpalettát beállítani, illetve ezt a játék futása közben módosítani.

4.3. Egyéb funkciók

4.3.1. Solver

A játéknak fontos tulajdonsága, hogy egy cella kétszeres megnyomása visszaállítja az eredeti állapotot, nem változtat a pálya állapotán. A megnyomások sorrendje pedig kommutatív, azaz felcserélhető. Ebből következik, hogy minden cellát maximum 1-szer kell lenyomni ahhoz, hogy megnyerjük a játékot.

A brute-force megoldás esetén így tehát $2^{N \cdot N}$ nagyságrendű állapotot kéne ellenőrizni a megfelelő cellakombináció megtalálásához. Ez nem elfogadható a megoldás során, hiszen nagyon sok ideig tartana, már egy 4x4-es pálya esetén is 65 536 lehetséges megoldást kéne ellenőrizni.

Ehelyett páratlan n esetén használjuk ki, hogy tudjuk milyen lépésekkel generáltuk le a pályát a játékosnak, valamint a játékos lépéseit is eltárolhatjuk. Ezekből kiszámolható, hogy mely cellák lenyomása kell a játék megoldásához.

Páros n esetén érdemes a játékkal játszva szabályszerűségeket felfedezni. Megadható olyan algoritmus amely pontosan 1 cella állapotát változtatja meg. Ebből pedig összerakható a megoldás, ahol minden cellát legfeljebb 1-szer nyomunk le.

Fontos, hogy a solver olyan megoldást adjon, ahol minden cellát legfeljebb egyszer kell lenyomni.

4.3.2. Saver

A játék biztosítson lehetőséget arra, hogy egy megkezdet pályát folytatni lehessen később is. (A játék program újra indítása utáns is!) A konzolos futtatásnál valamint a GUI menüben is jelenjen meg az új játék kezdése mellett az elmentett játék folytatásának lehetősége is, amennyiben korábban már mentett a játékos pályát. Mentéskor ne csak a pálya állását, de az addig megtett lépésszámot, valamint GUI estén az addig eltelt időt is tároljuk!

4.3.3. Timer

GUI-s futtatás esetén a játék mező mellett jelenjen meg egy kijelző is, mely a pálya indításától számított eltelt időt mutatja óra:perc:másodperc (esetleg századmásodperc) formátumban.