

Programozási technológia 1 - 2. beadandó

Sana Norbert

2016. június 1.

Tartalomjegyzék

1. Feladatkiírás	1
1.1. A feladat szövege	1
1.2. Használati utasítás	2
1.2.1. TL;DR	2
2. Megoldás	2
2.1. Megoldás terve	2
2.1.1. GUI funkcionális terve	2
2.1.2. Osztálydiagramm	4
2.1.3. Osztályok absztrakt leírása	4
2.2. Tesztelési terv	5
2.2.1. GUI tesztelés	5
2.2.2. Működés tesztelése	6

1. Feladatkiírás

1.1. A feladat szövege

7. feladat - Áttörés

Készítsünk programot, amellyel a következő kétszemélyes já tékot lehet játszani. Adott egy $n \times n$ mezőből álló tábla, ahol a két játékos bábúi egymással szembe helyezkednek el, két sorban (pont, mint egy sakktáblán, így mindkét játékos $2n$ bábuval rendelkezik, ám mindegyik bábu ugyanolyan típusú). A játékos bábúival csak előre léphet e gyenesen, vagy átlósan egy mezőt (azaz oldalra, és hátra felé nem léphet), és hasonlóan ütheti a másik játékos bábúját előre átlósan (egyenesen nem támadhat). Az a játékos győz, aki a lőszőr átér a játéktábla másik végére egy bábuval.

A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (6×6 , 8×8 , 10×10), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, melyik játékos győzött, majd automatikusan kezdjen új játékot.

1.2. Használati utasítás

A program fordításához és futtatásához Oracle Java 8 szükséges!

A beadandó programot a *gradle build system* segítségével fordítom, csomagolom és futtatom a teszteket. Ehhez a repó rootjában mellékelve találhatóak a `gradlew` illetve a `gradlew.bat` scriptek, melyek az ún. *gradle wrapper* scriptek. Ezek valamelyikét futtatva (`./gradlew` linuxon, macen vagy `gradlew.bat` windowson) a wrapper script automatikusan letölti a megfelelő gradle verziót, a fordításhoz és a teszteléshez szükséges librarykat a mavencentral repóikból, buildel majd pedig futtatja a teszteket is. A gradle build script úgy van konfigurálva, hogy ehhez nem szükséges paramétereket megadni, ez az ún. *default task* amit a gradle végrehajt.

1.2.1. TL;DR

1. Linux vagy mac

```
./gradlew
```

2. Windows

```
gradlew.bat
```

2. Megoldás

A beadandó feladat (a követelményeknek megfelelően) egy SWING-ben implementált grafikus Java program.

2.1. Megoldás terve

2.1.1. GUI funkcionális terve

Az alkalmazás két fajta ablakot jelenít meg: egyik a főprogram ablaka, a másik a játék ablaka.

1. A főprogram

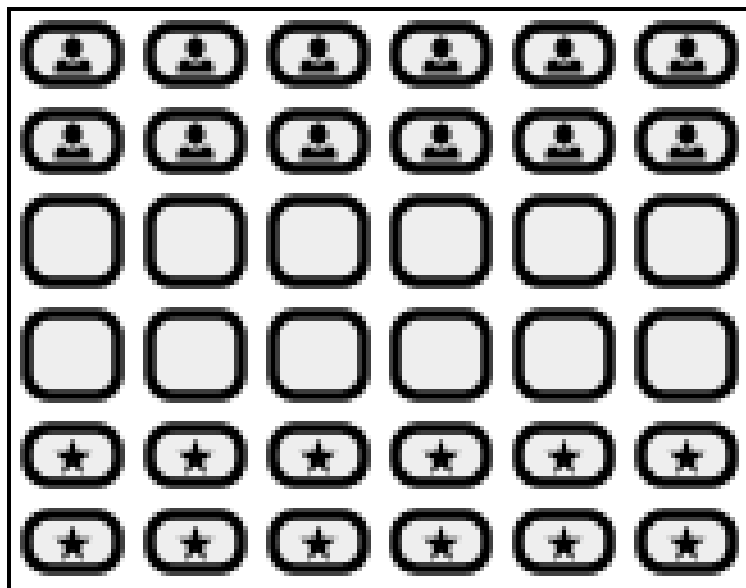
A főprogram ablakában van lehetőségünk beállítani a tábla méretét és új játékot kezdeni.

Tábla méret	6x6 ▼	Új játék kezdése
-------------	-------	------------------

2. A játék

A játék ablakában a beállításunknak megfelelő négyzetrácson kétállású gombok (toggle button) segítségével van ábrázolva a táblajáték és a rajta elhelyezkedő bábuk.

A játékosok úgy mozgathatják a bábjaikat, hogy először a mozgatni kívánt bábra kattintanak, majd pedig a célnak kijelölt mezőre.



2.1.2. Osztálydiagramm

2.1.3. Osztályok absztrakt leírása

1. **Main** A program indulási pontja és a főablak egyben. Feladata megjeleníteni a játék konfigurációját és az adott konfigurációval elindítani egy új játékot.
2. **GameConfig** A játék konfigurációs objektuma. Feladata a játéktábla méretének beállítása.

3. **GameConfigView** A hozzá tartozó **GameConfig** objektum megjelenítéséért és vezérléséért felelős.
4. **GameView** A játék ablaka. A bekonfigurált játékobjektumot jeleníti meg, figyel a játék eseményeire, így frissíti a grafikus nézetet.
5. **GameModel** A játék logikát tartalmazza. A konfiguráció alapján felépít egy mátrixot, melyben a játéktábla állapotát tárolja. Továbbá tárolja az éppen soron következő játékost (kin van a sor, ki lépjen legközelebb) és vizsgálja a játék végét. Ha a játék véget ért (valamelyik játékos nyert) akkor egy felugró ablakkal jelzi.
6. **GameField** A játéktábla egy mezőjének logikáját tartalmazza. Egy mező tartozhat valamely játékoshoz (ekkor bábuja van az adott mezőn a játékosnak) vagy egyik játékoshoz sem (ekkor szabad mező a játék szempontjából).
7. **FieldView** A **GameField** megjelenítésére és vezérlésére szolgál. A mező kiválasztásának eseményét triggereli a hozzá tartozó **GameField** objektumon, melyet a **GameModel** is figyel, így módosítja a játék állapotát a játékszabályok szerint.
8. **Player** Egy enum típus, a játékosok egyszerűbb megkülönböztetésére szolgál.

2.2. Tesztelési terv

2.2.1. GUI tesztelés

1. Főablak
 - A **GameConfigView**-n található legördülő menü a megfelelő elemeket tartalmazza, ezek kiválaszthatóak.
 - A start gomb megnyomásával a megadott konfigurációjú játék elindul.
2. Játék ablak
 - A konfigurációnak megfelelő méretű ablak jelenik meg
 - A játéktábla a megfelelő számú mezőt tartalmazza
 - Kezdőállapotban a játékosok bábuja a megfelelő mezőkön helyezkednek el

2.2.2. Működés tesztélése

1. Konfiguráció

- A kiválasztott konfigurációnak megfelelő méretű játék indul el

2. A játék menete

- Fehér játékos kezd
 - A fehér csak a Déli irányokba tud lépni
 - csak szomszédos mezőre tud lépni
- a játékosok felváltva kerülnek sorra
 - A fehér játékos után a fekete következik
 - a fekete játékos csak az Északi irányokba tud lépni
- Bábu ütése
 - Csak a soron következő játékos tud ütni bábut
 - Csak az ellenfél bábuját tudja ütni
 - az ellenfél bábuját szemből nem tudja ütni
 - az ellenfél bábuját átlósan tudja ütni
 - csak szomszédos bábut tud ütni
- Játék vége
 - Amint az egyik játékos elér a tábla vele átellenes részének utolsó sorába
 - A játék vége esemény aktiválódik
 - * A játék vége és a nyertes játékos megjelenik egy felugró ablakon