# 2.Beadandó

Név: **Jőrös Bence Tamás**

ETR-azonosító: **ZHY53N**

E-mail cím: **rexomber@inf.elte.hu**

Kurzuskód: **IP-18ePROGTEG**

Gyakorlatvezető neve: **Nagy András**

Feladatsorszám: **8.**

# Felhasználói dokumentáció

Feladat:

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy 𝑛×𝑛 mezőből álló tábla, amelynek a négy sarkába 2-2 fehér, illetve fekete ló figurát helyezünk el (az azonos színűek ellentétes sarokban kezdenek). A játékosok felváltva lépnek, a figurák L alakban tudnak mozogni a játéktáblán. Kezdetben a teljes játéktábla szürke színű, de minden egyes lépés után az adott mező felveszi a rá lépő figura színét (bármilyen színű volt előtte). A játék célja, hogy valamely játékosnak függőlegesen, vízszintesen, vagy átlósan egymás mellett 4 ugyanolyan színű mezője legyen. A játéknak akkor van vége, ha minden mező kapott valamilyen színt. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (4×4, 6×6, 8×8), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, melyik játékos győzött, majd automatikusan kezdjen új játékot.

**Környezet:**

*IBM PC, legalább Windows XP operációs rendszer*

Használat:

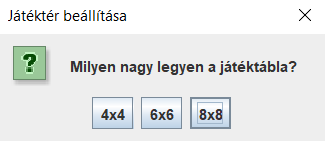
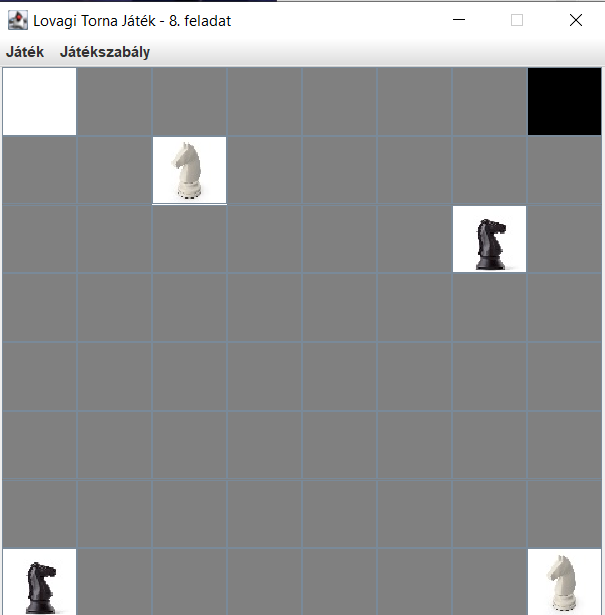
A program indítása:

A program a **knight\_tournament/src/Game** oldalon/néven található, indítható.

A program bemenete:

A program nem kér be adatokat a játékhoz.

Egy lehetséges párbeszéd:

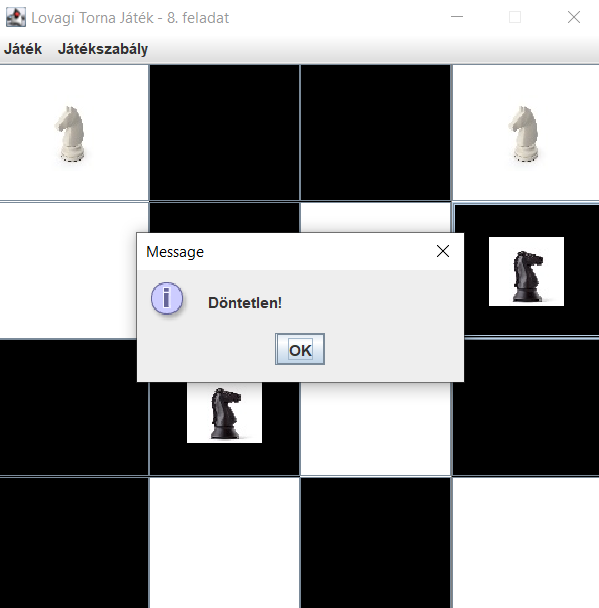
*A program eredménye:*

A program kiírja a győztes nevét, a hozzá tartozó adatokkal. Amennyiben döntetlen van, vagy senki nem maradt életben a verseny végére, úgy ezeket az eseteket is kiírja.

Egy lehetséges kimenet:

A képen Webhely látható

Automatikusan generált leírás



Hibalehetőségek, további fejlesztések:

A program olyan szempontból kicsit átláthatatlan, hogy 2 kattintás kell a lépéshez: kijelölés, majd végrehajtás, de hogy éppen a felhasználó melyik kattintást végzi, azt semmi nem jelöli. Természetesen lehetnek továbbfejlesztések – mentés vagy a pontozás belevétele a játékba.

# Fejlesztői dokumentáció

Feladat:

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy 𝑛×𝑛 mezőből álló tábla, amelynek a négy sarkába 2-2 fehér, illetve fekete ló figurát helyezünk el (az azonos színűek ellentétes sarokban kezdenek). A játékosok felváltva lépnek, a figurák L alakban tudnak mozogni a játéktáblán. Kezdetben a teljes játéktábla szürke színű, de minden egyes lépés után az adott mező felveszi a rá lépő figura színét (bármilyen színű volt előtte). A játék célja, hogy valamely játékosnak függőlegesen, vízszintesen, vagy átlósan egymás mellett 4 ugyanolyan színű mezője legyen. A játéknak akkor van vége, ha minden mező kapott valamilyen színt. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (4×4, 6×6, 8×8), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, melyik játékos győzött, majd automatikusan kezdjen új játékot.

**Specifikáció:**

Bemenet: Game.java, Knight.java, GridLayoutManager.java, StartGameDatas.java

Kimenet: a nyertes**: Szín,** a nyertes lény neve és egyéb adatai.

Előfeltétel: Knight.count > 0 && table = true

Utófeltétel: Nyertes:ColorWhite:boolean (tableField[i]=!grayColor && tableField[j]=!grayColor) && tableField[i][j]=colorWhite && tableField[i+1][j+1]=colorWhite && tableField[i+2][j+2]=colorWhite && tableField[i+3][j+3]=colorWhite && tableField[i+4][j+4]=colorWhite

Környezet:

IBM PC, legalább Windows2000 operációs rendszer

Java, IntelIj fejlesztői környezet.

Forráskód:

A forráskód megtalálható a http://canvas.elte.hucímen a zhy53n.zip fileban, melynek tartalma:

* Knight.java,
* StartGameDatas.java,
* Game.java
* GridLayoutManager.java

Megoldás:

Osztály leírása:

1. Game:
   1. Ez az osztály felel a játk szabályos indulásáért. Példányosít egy objektumot a StartNewGameDatas() osztályból, aminek meghívja a játék elindításáért felelős metódusát, majd magáért a játéktérért felelős osztályt is meghívja (GridLayoutManager()). Ha „kiikszeljük” a játékot a jobb felső sarokban, bezárja a programot.
2. StartGameDatas:
   1. A játéktér szabályos kiválasztásáért felel. 2 privát adattagja: String[] options – amik tartalmazzák a játéktér méretét – illetve a defaultOption, amely az alap választási lehetőséget rögzíti.
   2. Egyetlen metódusa a startGame(), amely feldobja a választópanelt, majd ennek értékét odaadja a GridLayoutManager osztálynak.
3. Knight
   1. A Knight osztály a bábuk tulajdonságaiért felelnek. Csak azt tudják magukról, amit kell nekik: jelenlegi pozíció, szín, sorszám, és az őket reprezentáló képek. Getterekkel és Setterekkel van ellátva.
4. ButtonHandler implements ActionListener
   1. A clickeléskor kiváltott esemény. Számon tartja hogy melyik játékos, hanyadszorra klikkelt, beleértve az érvénytelen klikkeket is. Ezen kívül csak a processClick() metódust hívja meg.
5. GridLayoutManager
   1. A legterjedelmesebb osztály, ahol a játék logikája kerül megvalósításra.

Adattagok:

//táblaméret eltárolása:  
 private final int basicRow;

private final int basicCol;

//Components:

private final JButton[][] squares;

//colors:

private final Color colorBlack = Color.BLACK; //Black squares

private final Color colorWhite = Color.WHITE; //White squares

private final Color colorGrey = Color.GRAY; //Grey squares

private final Color colorRed = Color.RED; //Red squares

//

private int chosenKnight;

private boolean isWhite;

//first click on the Player's turn or the second?

boolean firstClick = false;

List<Knight> knights = new ArrayList<>();

Metódusok leírása:

1. public GridLayoutManager(int basicRow, int basicCol): A játéktábla kirajzolásáért, az eseménykezelők hozzáadásáért, a menükért, a lovagok kezdeti pozíciójáért és a játéktábla méretéért felel.
2. **public boolean hasFourEqualColors():** Minden kör végén keresi azt, hogy hol lehet 4 egyforma szín, ami az egyik játékos számára a győzelmet jelenti.
3. **private boolean fullBoard():** Minden kör végén megnézi, hogy minden táblanégyzet beszíneződött-e, ami a döntetlent jelenti.
4. **private boolean isValidMove(int i, int j)**: A logika, ami validálja egy huszár lépését. Az átadott paraméterek azok a mezők indexei, ahova a lovag lépni szeretne.
5. **private void processClick(int i, int j)**: A click eseménykor kiváltodott metódus, kezeli, hogy mi történjen klikkeléskor. Első és második klikk események vannak lekezelve, illetve az is, ha valaki nem valid módon szeretne lépni. Az átadott paraméterek azok a mezők indexei, ahova a lovag lépni szeretne.

Osztályok:

A képen asztal látható

Automatikusan generált leírás A képen asztal látható

Automatikusan generált leírás A képen asztal látható

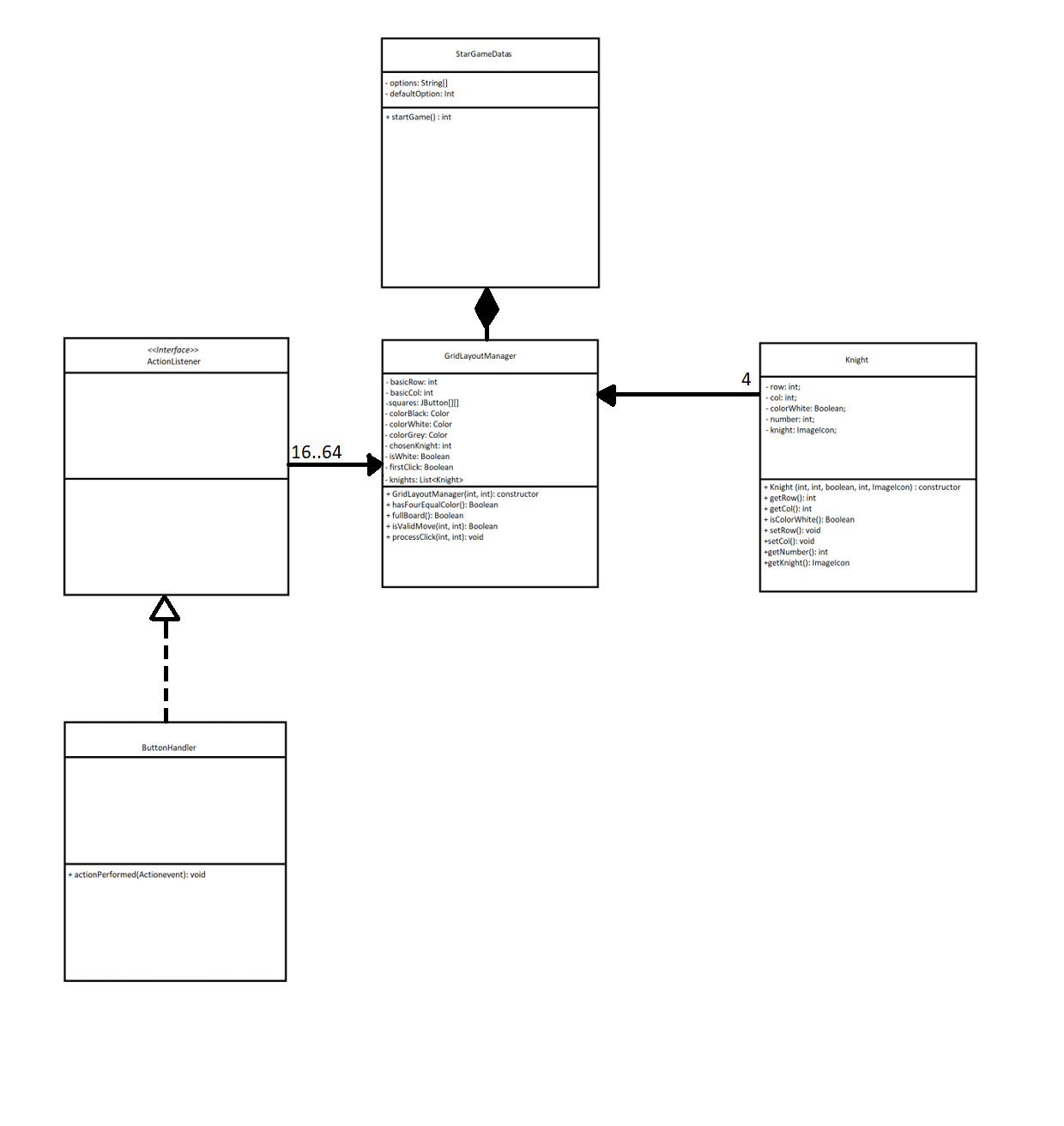
Automatikusan generált leírás

A képen asztal látható

Automatikusan generált leírás A képen Téglalap látható

Automatikusan generált leírás

UML:



Tesztelés:

Érvényes tesztesetek:

|  |  |  |
| --- | --- | --- |
| **Teszteset rövid leírása** | **Bemenet** | **Kimenet** |
| Pályaválasztó | Kiválasztjuk az adott pályaméretet | A játék rendben elindul |
| Első lépés tesztelése  (Pálya kiválasztásával) | A fehér kezd. | A fehér lép (jó helyre tud csak) |
| Fehér majd fekete lép (felváltva) | A fehér után a feketére kattintok | A fekete lép |
| Győzelem (sor/oszlop) | Egy oszlopban csak fehér | Kiírja a győzelmet |
| Győzelem (átló) | Egy átlóban csak fekete | Kiírja a győzelmet |
| Döntetlen | Tele van a tábla, de nincs 4 egymás után ugyanabból a színből | Kiírja a döntetlent |
| Pálya befejezése után kiírás, majd új játék | Győzelem kiírása után új pálya kiválasztása | Ki lehet választani az új pályát és elkezdi a játékot. |

Érvénytelen tesztesetek:

|  |  |  |
| --- | --- | --- |
| **Teszteset rövid leírása** | **Bemenet** | **Kimenet** |
| Fehér után egyből fehérrel szeretnék lépni | Fehérre kattintás | A játék a hibát lekezeli |
| Feketével szeretnénk kezdeni | Feketére kattintás | A játék a hibát lekezeli |