Tilitoli játék – Programfejlesztési dokumentáció

(Angster Erzsébet – 2002. szeptember 1.)

1. Követelményspecifikáció

A feladat leírása

A feladat a Tili-toli logikai játék szimulálása számítógépen. Az n*n-es rács pontjaiban számok találhatók 1-től számozva, az egyik helyen lyuk van. Kezdetben a számok össze vannak keverve. A játék célja a számok növekvő sorrendbe rendezése (sorfolytonosan) úgy, hogy a lyuk az utolsó helyre kerüljön (lásd ábra). Ha egy számra rákattintunk, akkor a szám beugrik a lyuk helyére, feltéve, hogy a lyuk a szám szomszédja. Az eltelt időt és az aktuális lépésszámot a program kijelzi. A játékot nem lehet felfüggeszteni, de bármikor újra lehet kezdeni.

A játékot háromféle módban lehet játszani: 3*3-as, 4*4-es és 5*5-ös módban. A program minden egyes módhoz egy toplistát (eredménylistát) tart karban. Amikor vége van a játéknak, és az elért eredmény alapján a játék bekerülhet a toplistába, a program megkérdezi a játékos nevét. Ha a játékos is úgy akarja, a lejátszott játék bekerül a toplistába a következő adatokkal: név, idő, lépésszám. Annak van jobb eredménye (és így a listában előrébb szerepel), aki

kevesebb idő alatt rakja sorrendbe a számokat; egyenlő idő esetén a lépésszám dönt. A toplistában egy név többször is szerepelhet. A programban be lehet állítani a toplisták maximális elemszámát – ez alapértelmezésben minden lista esetén 10.

Befejezéskor a program lemezre menti, majd a legközelebbi indításkor betölti a következő adatokat:

- `results.dat` állomány: A három toplistát tartalmazza.
- `config.dat` állomány: A játék konfigurációs adatait tartalmazza: a játék módját (3*3, 4*4 vagy 5*5), a toplisták maximális elemszámát (a győztesek számát), valamint a fő ablak helyzetét és méretét. Az ablak adatai mindig a legutoljára elmentett állapot szerintiek.

Használati esetek

A program indítása: megjelenik a főablak, és elindul az Új játék funkció.

- **Új játék:** Gombok keverése, idő és lépésszámlálás indul – e két adatot a program folyamatosan kijelzi. A játékot nem lehet megszakítani, az időmérést semmilyen módon nem lehet felfüggeszteni. Akkor van vége a játéknak, ha a számok helyes sorrendben vannak, és a lyuk az utolsó helyen áll. Ha a játék bekerülhet a toplistába, a program kér egy nevet. Ha beütik, akkor a játék bekerül a listába (csak külön kérésre jelenik meg).
- **Eredménylista (toplista):** Az aktuális mód toplistájának megjelenítése. A lista egy sora egy sikeresen lejátszott játék (név, idő, lépésszám). A lista rendezett idő, azon belül lépésszám szerint. A listát fel-le lehet görgetni. A listát nem lehet manipulálni, de az egész listát ki lehet törölni. A fő ablakhoz csak a listaablak becsukásával lehet visszatérni. Az eredmény megtekintése nem indukál új játékot.
- **Beállítások:** Be lehet állítani a játék módját (3*3, 4*4 vagy 5*5), valamint a program által megjegyzendő győztesek számát. A beállítás végeztével új játék indul.
- **Kilépés:** A program futása befejeződik.

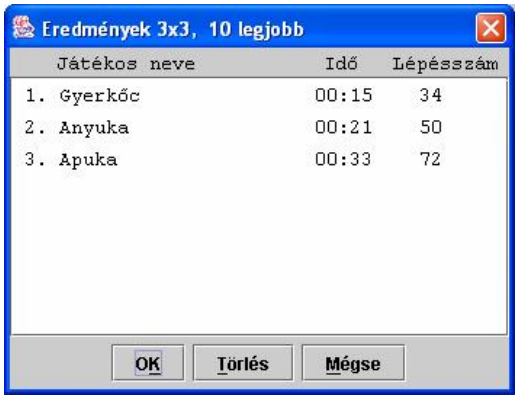
Menüszerkezet

```
Játék
  Új játék
  Eredménylista
  Beállítások
  Kilépés
Súgó
  Használat
  Névjegy
```

Képernyőképek

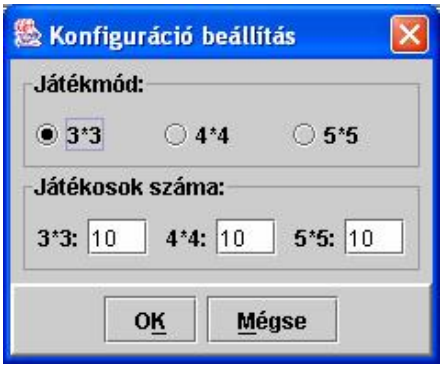
Kezdő, főablak:

Eredménylista (toplista):

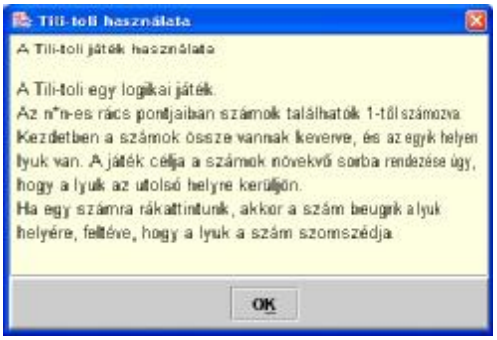


A győztes nevének bekérése:

Beállítások:

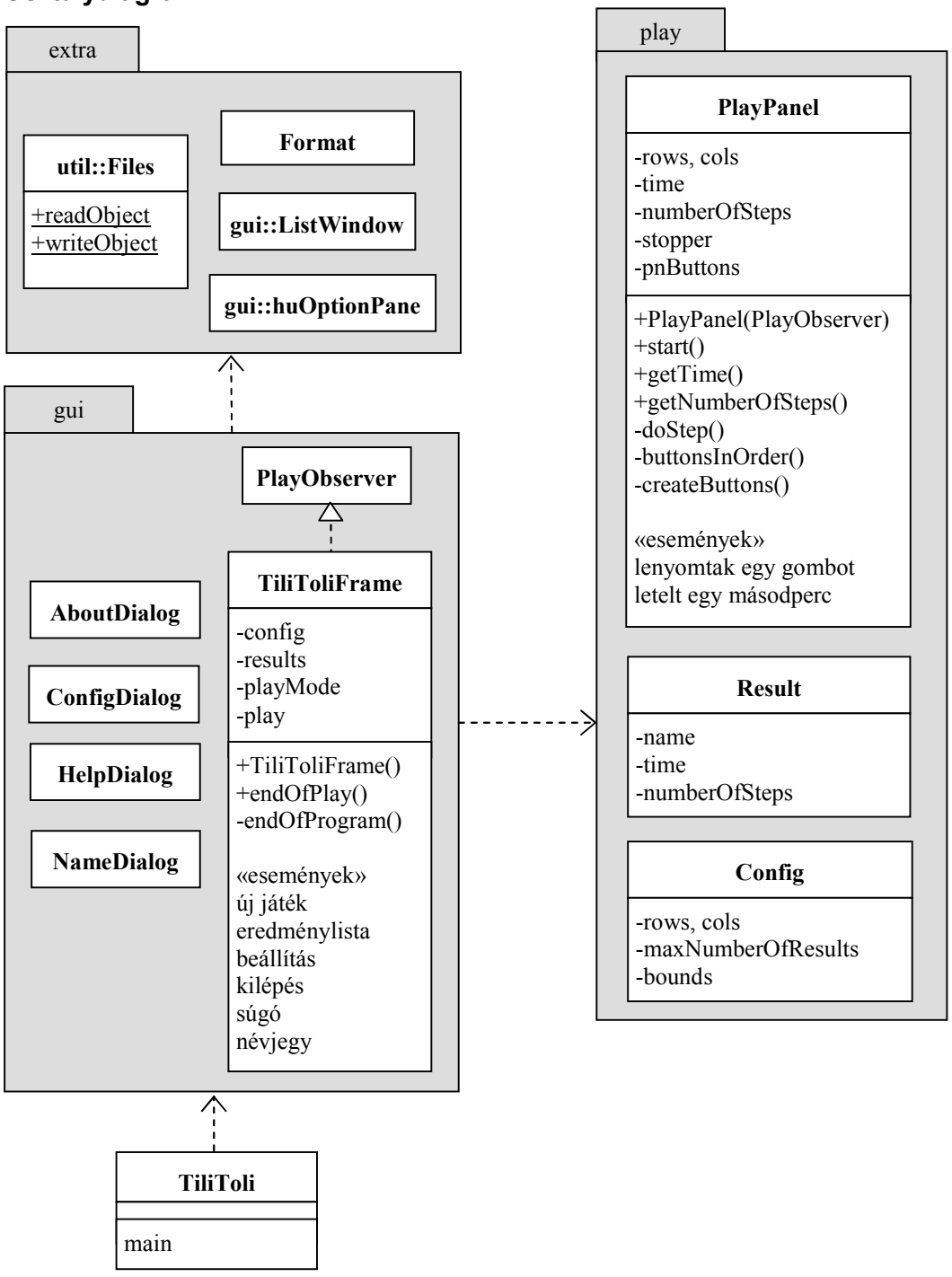


Segítség, névjegy:



2. Programterv

Osztálydiagram



Osztályleírások

Tili_Toli

Indító osztály. Metódusa:

- `main`
  Létrehozza a program fő ablakát, a `TiliToliFrame`-et.

play csomag

Config

Konfigurációs osztály. A programnak a felhasználó által beállítható adatait tartalmazza. Csak alapértelmezett konstruktorral rendelkezik. Szerializálható (fájlba menthető). A játékban egyetlen `Config` objektum van, melyet a program lemezen tárol a `'config.dat'` állományban.

Adatok:

- `playMode: int`
  A játék módja. Lehetséges módok: 1: 3*3-as játék, 2: 4*4-es játék, 3: 5*5-ös játék.
- `maxNumberOfResults: int[3]`
  Játékmódonként azon játékosok maximális száma, akiknek eredményei eltárolásra kerülnek. Más szóval: játékmódonként a toplista hossza.
- `bounds: Rectangle`
  A játék fő ablakának helyzete és mérete. A program megjegyzi a felhasználó által áthelyezett, illetve átméretezett ablak adatait.

Metódusok:

- `rows(): int`
- `cols(): int`
  Megadja az aktuális játékmódhoz a sorok és oszlopok számát.

Result

A játék eredményét (név, idő, lépések száma) eltároló objektum. Rendezési szempont: idő+lépések száma. A toplistába ilyen objektumok kerülnek.

Adatok:

- `name: String`
  A játékot játszó játékos neve.
- `time: int`
  A játék lejátszásának ideje másodpercben.

- `numberOfSteps: int`
  Lépések száma. Ennyi lépésre volt szükség a számok sorbarakásához. Egy lépés egy szám áthelyezése.

PlayPanel

Játékpanel. Egy terület, melyen a gombok és a lyuk elhelyezkedik. Felelős egy játék lejátszásáért. Létrehozáskor megadjuk a sorok és oszlopok számát. Konstruktorában meg lehet adni egy játékot figyelő objektumot. Ha vége van a játéknak, akkor a figyelő értesítést kap, melynek paramétere a játék eredménye: `endOfPlay(Result)`. Az eredményben a név nincs kitöltve, csak az idő és a lépések száma.

A játékpanel egy rácsos terület, ahol a rács pontjaiban gombok és egy lyuk helyezkednek el. A játék indításakor elindul a stopper és a lépésszámláló, ezeket a panel folyamatosan kijelzi. Ha a játékos rákattint egy lyukkal szomszédos számra, a szám helyet cserél a lyukkal. Ha a gombok sorrendje helyes, akkor leáll a stopper és a lépésszámláló, a játék inaktívvá válik, és értesíti a játék figyelőjét.

Privát adatok:

- `rows, cols: int`
  A játék sorainak és oszlopainak száma.
- `time: int`
  Játékidő másodpercben.
- `numberOfSteps: int`
  Lépések száma. Csak az számít lépésnek, ha történt gombmozgás.
- `stopper: Timer`
  Stopper, mely méri a játékidőt. Másodpercenként automatikusan növeli a `time` adatot.
- `pnButtons: JPanel`
  A játék gombjai, a panel konténer `PlayButton` elemei. `PlayButton` egy belső osztály.

Konstruktor, metódusok:

- `PlayPanel(PlayObserver playObserver, int rows, int cols)`
  Konstruktor. Létrehozza a játékpanelt a figyelővel, sor és oszlopszámmal.
- `start()`
  Játék indítása. Véletlenszerűen elrendezi a gombokat, elindítja a stoppert, és engedélyezi a gombokat.

Privát metódusok:

- `doStep(PlayButton btPressed)`
  Elvégzi a lépést. A lépésszámot eggyel növeli.

- `boolean buttonsInOrder()`
  true, ha a gombok jó sorrendben vannak.
- `createButtons()`
  Létrehozza a gombokat, és meg is keveri azokat.

Események:

- `Megnyomtak egy számgombot`
  Ha a szám a lyuk szomszédja: doStep
- `Letelt egy másodperc (Timer eseménye)`
  Idő növelése és kijelzése.

PlayButton – belső osztály

Egy gombot reprezentál. A gombnak van száma. Ismeri a játékpanelt, így annak `pnButtons` gombjait. Meg tudja határozni sorfolytonos pozícióját a játékpanelen, valamint azt, hogy hányadik sorban, illetve oszlopban van.

gui csomag (programspecifikus dialógusok)

TiliToliFrame

A jákék fő ablaka. Tartalmazza a menüt, a játékot, és a szükséges dialógusokat. Implementálja az `Observer` interfészt.

Privát adatok:

- `config: Config`
  A program konfigurációja. Befejezéskor a program automatikusan lemezre menti, indításkor pedig betölti.
- `results: Vector[3]`
  Három darab eredménylista (toplista) a három módhoz.
- `playMode: int`
  Az aktuális mód. 0: 3*3, 1:4*4, 2:5*5
- `play: PlayPanel`
  A játékpanel. Program kezdetekor és módváltáskor új játékpanel jön létre.

Metódusok:

- `TilitoliFrame()`
  Konstruktor. Létrehozza a menüt. Létrehozza a játékot és elindítja. Háttérben létrehozza az összes dialógusablakot is.

- `void endOfPlay(int time, int numberOfSteps)`
  A játék hívja meg. Ha nem elég jó a játék eredménye, akkor közli, hogy: "Jó, jó, de nem az igazi!". egyébként bekéri a nevet. Ha beütik, akkor az eredményt beteszi az aktuális toplistába, rendezetten.

Privát metódusok:

- `void endOfProgram()`
  Program befejezése. A `config` és `results` objektumokat lemezre írja, és befejezi a programot.

Események:

- `Új játék`
  play.start()
- `Eredménylista`
  Eredménylista címébe a mód beírása: resultWindow.showList(results[playMode])
- `Beállítások`
  config objektum szerkesztése a ConfigDialog segítségével.

  Ha változott a konfiguráció, akkor új játékobjektum létrehozása és elindítása: play.start()
- `Kilépés`
  endOfProgram()
- `Súgó`
  helpDialog.show()
- `Névjegy`
  aboutDialog.show()

ConfigDialog

Dialógus ablak a konfigurációs (Config osztályú) objektum szerkesztésére.

Metódusok:

- `ConfigDialog(Frame parent)`
  Konstruktor.
- `Config getConfig(Config oldConfig)`
  Ha az Ok-t nyomták le, akkor visszatérési értéke a megváltoztatott config objektum, egyébként marad a régi objektum.

AboutDialog

A program névjegyét megjelenítő dialógus ablak.

HelpDialog

A segítséget megjelenítő dialógus ablak. Csak egy szövegterületet tartalmaz, OK-re az ablak eltűnik.

NameDialog

Dialógusablak a felhasználó nevének bekérésére.

Metódusok:

- `NameDialog(Frame parent)`
  Konstruktor.
- `String getName()`
  Megjeleníti az ablakot, bekér egy nevet, majd eltünteti az ablakot. Visszaadja a beütött nevet. A Mégse lenyomása esetén null-t ad vissza.

extra.gui csomag (általános dialógusok)

ListWindow

Dialógus ablak, mely egy listát képes megjeleníteni. A konstruktorban megadható a lista címe és fejléce. A lista törölhető.

Metódusok:

- `ListWindow(Frame owner, String title, String header)`
  Konstruktor. Megadható a lista címe és fejléce.
- `showList(Vector list)`
  Megjeleníti a lisát. A lista törölhető. Az Ok vagy a Mégse leütésére eltünteti az ablakot. Mégse esetén az esetleges törlés figyelmen kívül marad.

extra.util csomag

Files

Objektum lemezre való mentésére és az onnan való beolvasására szolgáló statikus metódusok.

- `static void writeObject(String path, Serializable object)`
  A paraméterben megadott szerializálható objektum kiírása lemezre a `path` útvonallal azonosított fájlba.
- `static Object readObject(String path)`
  Objektum beolvasása lemezről a `path` útvonallal azonosított fájlból. A visszatérési érték a beolvasott objektum, olvasási hiba esetén null.