

# Első beadandó feladat dokumentáció

Készítette:

**Kocsis Márk**

Feladat:

Készítsünk programot, amellyel a potyogós amőba játékot lehet játszani, vagyis az amőba azon változatát, ahol a jeleket felülről lefelé lehet beejteni a játéklemezőre. A játéklemező itt is  $n \times n$ -es tábla, és ugyanúgy X, illetve O jeleket potyogtathatunk a mezőre. A játék akkor ér véget, ha betelik a tábla (döntetlen), vagy valamelyik játékos kirak 4 egymás melletti jelet (vízszintesen, vagy átlósan). A program minden lépésnél jelezze, hogy melyik játékos következik, és a tábla egy üres mezőjére kattintva helyezhessük el a megfelelő jelet. Természetesen csak a szabályos lépéseket engedje meg a program. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával ( $10 \times 10$ ,  $20 \times 20$ ,  $30 \times 30$ ), játék szüneteltetésére, valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött (a táblán jelölje meg a győztes 4 karaktert). A program folyamatosan jelezze külön-külön a két játékos gondolkodási idejét (azon idők összessége, ami az előző játékos lépésétől a saját lépéséig tart, ezt is mentjük el és töltjük be).

Elemzés:

- A játékot három különböző méretű pályán játszhatjuk. Az alapbeállítás a  $10 \times 10$  négyzet méretű pálya, de választhatunk  $20 \times 20$ , illetve  $30 \times 30$  méretű pályát is.
- A játékot egy ablakban csináljuk meg, Windows Forms grafikus felülettel.
- Az ablakban elhelyezünk egy menüt a következő menüpontokkal: Fájl (Új játék, Játék betöltése, Játék mentése, Kilépés), Beállítások ( $10 \times 10$  tábla méret,  $20 \times 20$  tábla méret,  $30 \times 30$  tábla méret). Az ablak alján megjelenítünk egy státuszsort, amely az egyes játékosok lépéssel töltött idejét számlálja.
- A játéktáblát egy megadott méretű gombráccsal alkotjuk meg. A játékosnak, ha lépni akar, akkor a megfelelő gombra kell rákattintania, és annak függvényében, hogy az első vagy a második játékoson van a sor, ha a lépés szabályos volt, egy 'X' vagy egy 'O' jelenik meg a gombon. Az 'X' az első játékos jele, míg a 'O' a második játékosé minden esetben. A játékosnak ebben a programban muszáj az adott gombra kattintania, ahova szeretné „ejteni” a jelet.
- A játéknak akkor van vége, ha van kijön 4 ugyan olyan jel vízszintesen vagy átlósan, vagy ha betelik a tábla. A játék végét felugró ablakban jelezzük.
- A felhasználó esetei:
  - Kilépés
  - Mentés
  - Új játék
  - Betöltés
  - Lépés
    - A játék véget érhet ebből az eseményből
  - Táblaméret módosítása
    - $10 \times 10$ -es tábla méret

- 20x20 tábla méret
- 30x30 tábla méret

### Tervezés:

- Programszerkezet:
  - A programot háromrétegű architektúrában valósítjuk meg. A megjelenítés a **View**, a modell a **Model**, míg a perzisztencia a **Persistence** névtérben helyezkedik el. A program csomagszerkezete a 2. ábrán látható.
- Perzisztencia:
  - Az adatkezelés feladata a Potyogtató amőba játéktáblájának a tárolása, valamint a mentés és a betöltés.
  - A **Connect4Table** osztály egy érvényes játéktáblát biztosít, amely mindig ellenőrzi, hogy a rajta elvégzett lépés szabályos-e. A van egy paraméteres konstruktora, ez várja a méretet, ami rendre 10, 20 vagy 20 lehet, illetve egy paraméter nélküli konstruktora is van, ez alapjáraton 10x10 méretű táblát hoz létre. A tábla tárolja az utolsónak lépett játékost is **\_lastPlayer** adattagként. További metódusai információ lekérdezésre, értékbeállításra vagy ellenőrzésre szolgálnak: **IsEmpty**, **GetValue**, **SetValue**, **SetValueWhenLoad**, **CheckStep**.
  - A hosszú távú adattárolás lehetőségeit az **IConnect4DataAccess** interfész adja meg, amely lehetőséget ad a tábla betöltésére (**LoadAsync**), valamint mentésére (**SaveAsync**). A műveleteket hatékonysági okokból aszinkron módon valósítjuk meg.
  - Az interfészt szöveges fájl alapú adatkezelésre a **Connect4FileDataAccess** osztály valósítja meg. A fájlkezelés során fellépő hibákat a **Connect4DataException** kivétel jelzi.
  - A program az adatokat szöveges fájlként tudja eltárolni, melyek az **ctl** kiterjesztést kapják. Ezeket az adatokat a programban bármikor be lehet tölteni, illetve ki lehet menteni az aktuális állást.
  - A **ctl** fájl első sorában a táblaméret és az utolsónak lépett játékos sorszáma található, majd a játéktábla egy „méret x méret” típusú mátrix, az üres hely: 0, X jel: 1 és O jel: 2 megfeleltetéssel.
- Modell:
  - A modell lényegi részét a **Connect4GameModel** osztály valósítja meg. Ez az osztály felel a játék logikájáért. Tárolja az aktuális játékost (**\_currentPlayer**), a játékosok idejét (**\_player1Time**, **player2Time**), nyert-e az aktuális játékos, illetve van egy potyogtató amőba játéktábla (**\_table**) típusa is. Az osztály metódusával lehet lépni (**Step**), játékot betölteni és menteni.
  - A modell osztályban van két esemény: a játék vége és a játék előrehaladása, ezek argumentuma külön osztályban van megvalósítva, a **Connect4EventArgs**-ban.
- Nézet:
  - A nézetet a **GameForm** osztály biztosítja, amely tárolja a modell egy példányát (**\_model**), valamint az adatelérés konkrét példányát (**\_dataAccess**).
  - A játéktáblát egy dinamikus létrehozott gombmező (**\_buttonGrid**) reprezentálja. A felületen létrehozuk a megfelelő menüpontokat, illetve státuszsort, valamint dialógusablakokat, és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását (**GenerateTable**), illetve az értékek beállítását (**SetupTable**) külön metódusok végzik.
  - A nézet ablaka automatikusan állítja be a méretét, alkalmazkodva a játéktábla méretéhez. A játéktáblában a tábla nagyságától függ az azt alkotó gombok mérete. A 10x10-es tábla gombjai a legnagyobbak, míg a 30x30-as tábla gombjai a legkisebbek.

A gombok fix mérete miatt az alkalmazás jelenleg csak kellően nagy felbontás mellett működik hiánytalanul (pl. FHD).

- A játék időbeli kezelését egy időzítő végzi (`_timer`), amelyet mindig aktiválunk játék során, illetve inaktiválunk, amennyiben bizonyos menüfunkciók futnak.
- Teszt:
  - A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a **Connect4GameModelTest** osztályban.
  - Az alábbi tesztesetek lettek megvalósítva:
    - **Connect4GameModelLoadTest**
    - **Connect4GameModelNewGame10Test**
    - **Connect4GameModelNewGame20Test**
    - **Connect4GameModelNewGame30Test**
    - **Connect4GameModelStepTest**
    - **Connect4GameModelAdvanceTimeTest**

#### Fejlesztési tervek:

- Kisebb méretű képernyőkre optimalizálás
- Fix méretű ablak, a játéktér változik csak
- Potyogtatós mechanizmus megvalósítása, tehát bármely oszlopra lehet kattintani, ha oda még lehet „jelet” dobni