1. Beadandó feladat dokumentáció

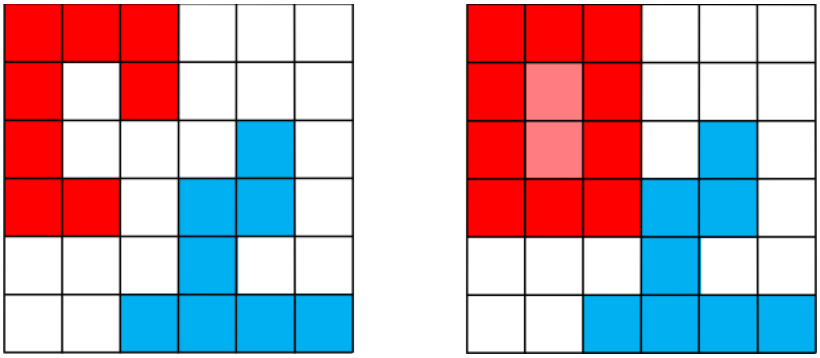
Készítette:

Simon Erhárd Raffael

e-mail: bywhqi@inf.elte.hu

Feladat

Bekerítés: Készítsünk programot, amellyel a következő két személyes játékot játszhatjuk. Adott egy $n \times n$ mezőből álló tábla, amelyre a játékosok 2 × 1-es méretű téglalapokat helyezhetnek el (vízszintesen, vagy függőlegesen). A játékosok felváltva léphetnek. A játék célja, hogy a téglalapokkal elhatároljuk a tábla egy részét (teljesen körbevéve téglalapokkal), amelyben így minden mező a játékosé lesz (beleértve az ellenfél által korábban elfoglalt mezőket is). A program külön jelölje meg a lehelyezett téglalapokat, illetve az elfoglalt területeket, és játék közben folyamatosan jelenítse meg az elfoglalt terület méretét játékosonként. A játék akkor ér véget, ha már további lépés nem végezhető a pályán



A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával (6 × 6, 8 × 8, 10 × 10), valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött. Tipp: a bekerítés ellenőrzéséhez használható például az elárasztásos kitöltés (flood fill) algoritmusa.

Elemzés

• A játékot három pályamérettel játszhatjuk: kicsi (6 x 6), közepes (8 x 8), nagy (10 x 10)

A program indításkor a „Játék indítása” menüpontban kiválaszthatjuk a kívánt pályaméretet, majd azt betölti a program és indulhat a játék!

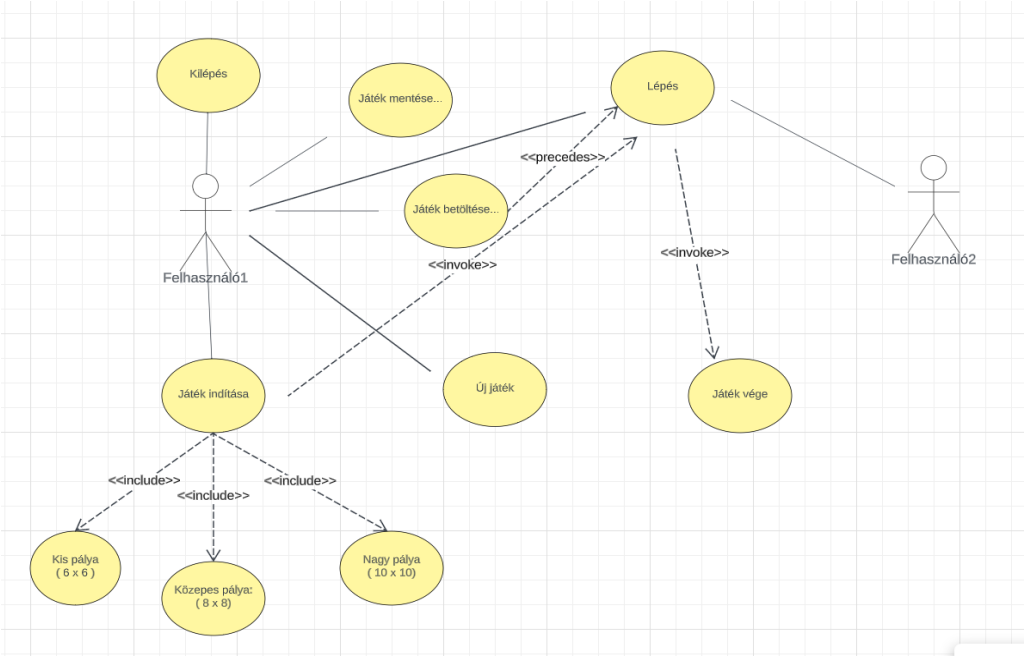
• A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg. • Az ablakban elhelyezünk egy menüt a következő menüpontokkal: File (Új játék, Játék betöltése, Játék mentése, Kilépés), Játék indítása (Kis pálya (6 x 6), Közepes pálya (8 x 8), Nagy pálya (10 x 10). Az ablak alján megjelenítünk egy státuszsort, amely a soron következő játékost, illetve a lehelyezendő elem irányát jelzi.

• A játéktáblát egy $n \times n$ nyomógombokból álló rács reprezentálja. A nyomógomb egérkattintás hatására megváltoztatja a gomb és az irány által kiválasztott másik gomb színét az adott játékos függvényében. Az ablak jobb

oldalán, fent látható a játék állása szöveg, ami alatt látható a két játékos pontszáma. A pontszám a beszínezett felületet takarja játékosonként.

• A játék automatikusan feldob egy dialógusablakot, amikor vége a játéknak (tehát vagy nyert a piros játékos, vagy nyert a kék játékos, vagy a játék döntetlen) Szintén dialógusablakokkal végezzük el a mentést, illetve betöltést, a fájlneveket a felhasználó adja meg.

• A felhasználói esetek az 1. ábrán láthatóak.



1. Ábra: Felhasználói esetek diagramja

Tervezés

Programszerkezet

• A programot háromrétegű architektúrában valósíjuk 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 Bekerítés táblával kapcsolatos információk tárolása, valamint a betöltés/mentés biztosítása.

• A Board osztály egy érvényes Board táblát biztosít. Ez a felhasználó által megadott választás alapján egy (6 x 6), vagy (8 x 8), vagy (10 x 10) táblát jelent. Ez alapértelmezetten 0-ásokkal van feltöltve, ezek jelölik az üres, még be nem színezett területeket. A tábla lehetőséget az állapotok lekérdezésére (getCell), valamint direkt beállításra (setCell).

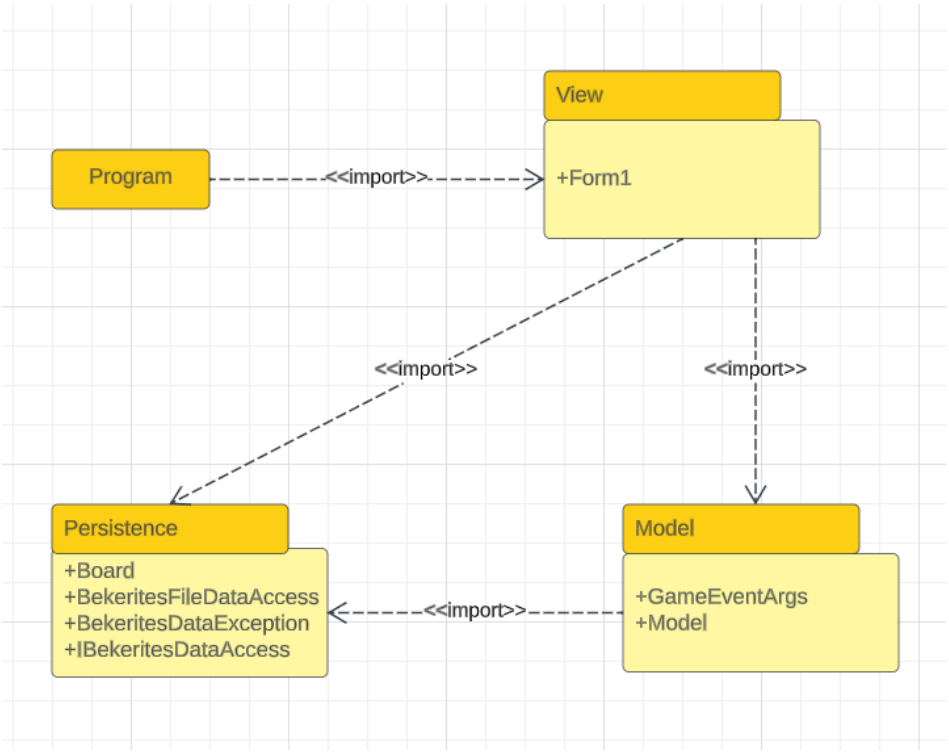
• A hosszútávú adattárolás lehetőségeit az IBekeritesDataAccess interfész

adja meg, amely lehetőséget ad a tábla betöltésére (Load), valamint mentésére (Save).

• Az interfészt szöveges fájl alapú adatkezelésre a BekeritesFileDataAccess osztály valósítja meg. A fájlkezelés során fellépő hibákat a BekeritesDataException kivétel jelzi.

• A program az adatokat szöveges fájlként tudja eltárolni. Ezeket az adatokat a programban bármikor be lehet tölteni, illetve ki lehet menteni az aktuális állást.

•A fájl tartalmazza a táblát, valamint az aktuális játékos meghatározása érdekében a tábla első eleméhez hozzáadunk egy konstanst, amit majd betöltéskor mindig ki is vonunk belőle, így nem változik láthatóan a játék.



2. Ábra: Az alkalmazás csomagdiagramja

Modell

• A modell lényegi részét a Model osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék egyéb paramétereit, úgymint a győztes meghatározása (getWinner), a pontszám kiszámítása (calculateScore), a bezárt alakzatok feltöltése (FillClosedShapes), a játék végének meghatározása (isGameOver), emellett az elemek lerakása (PlaceBlock), új játék kezdésére (newGame).

Új játéknál az előző tábla letörlődik az ablakból ezután a játék indítása menüpontban bármikor új játékot indíthatunk!

• A játékállapot megváltozásáról a WinnerDeclared esemény tájékoztat.

• Minden lépés után a MoveMade esemény változtatja meg a tábla állását.

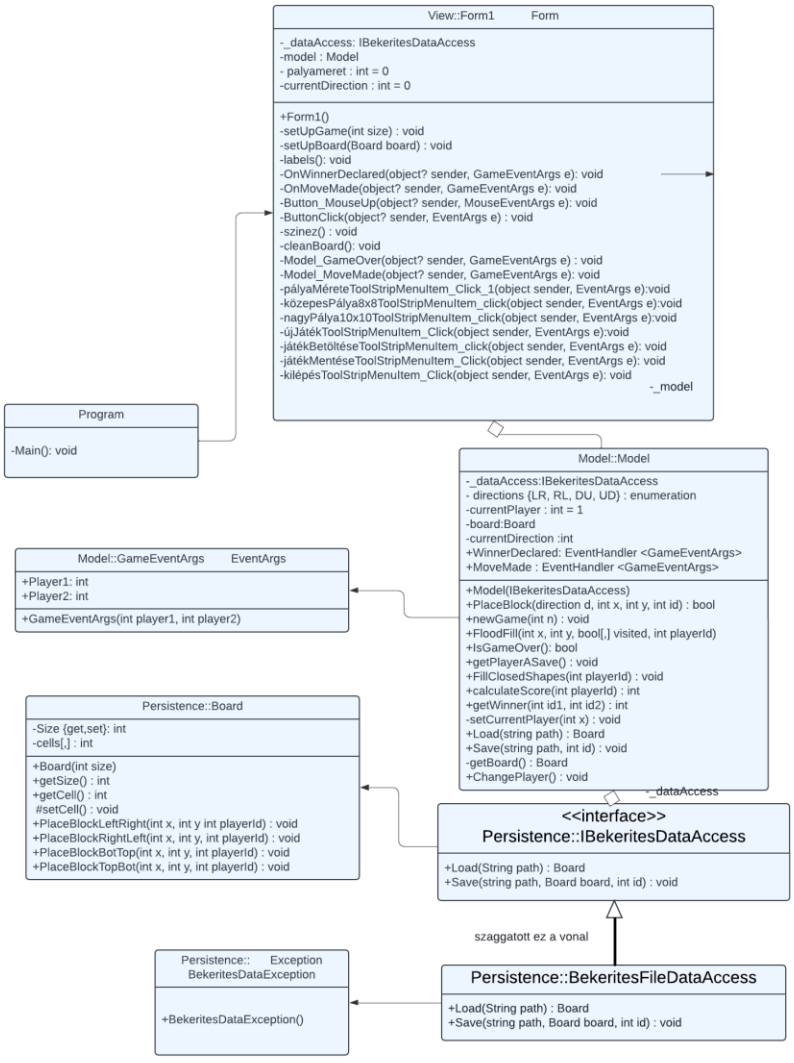
• A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (Load) és mentésre (Save)

Nézet

• A nézetet a Form1 osztály biztosítja, amely tárolja a modell egy példányát (model), valamint az adatelérés konkrét példányát (dataAccess).

• A játéktáblát egy dinamikusan létrehozott gombmező reprezentálja. A felületen létrehozzuk a megfelelő menüpontokat, valamint dialógusablakokat, és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását a Board osztály konstruktora, illetve az értékek beállítását (PlaceBlock) külön metódusok végzik.

• A program teljes statikus szerkezete a 3. ábrán látható.



3.ábra: Az alkalmazás osztálydiagramja

Tesztelés

A modell funkcionalitása egységtesztek segítségével lett ellenőrizve az UnitTest1 osztályban.

Az alábbi tesztesetek kerültek megvalósításra:

Alapvető játékfunkciók tesztmetódusai:

•**GameOverTest1** – A játék abban az esetben ér véget, ha már egyik játékos sem tud több elemet lehelyezni. Ez a teszteset megvizsgálja, hogy egy olyan pályán, ahol még létezik szabályos lépés vége van-e a játéknak.

•**GameOverTest2 –** Ez a teszteset megvizsgálja, hogy vége van-e a játéknak abban az esetben, ha nem létezik szabályos lépés.

•**BekeritesBoardTest –** Ez a teszteset néhány alapvető mechanikáját vizsgálja az új játék indításakor inicializált táblának. Pontosabban azt, hogy a játék elején biztosan az 1. játékos kezd, hogy a játékosoknak nincsen kezdetben pontja, emellett a játéknak nem lehet vége a tábla inicializálásakor.

•**CalculateScore_Player1** –Ez a teszteset néhány lépés megtétele után megvizsgálja, hogy az adott játékosnak a pontszáma megegyezik-e a várttal.

Mivel a játék legfontosabb része az, hogy megfelelően menjen végbe a bekerítés, ezért erre 3 különböző tesztet készítettem. Ezek letesztelik, hogy működik-e a másik játékos bekerítése, ellenőrzik, hogy a tábla sarkát nem lehet bekeríteni, hogy mi történik akkor, ha egy Piros alakzat lezárását egy Kék hajtja végre.

Bekerítés tesztmetódusai:

•**FillTest1** – Ez a teszteset megvizsgálja, történik-e bekerítés abban az esetben, ha a két játékos zár be együtt egy alakzatot.

•**FillTest2** – Ez a teszteset megvizsgálja, hogy a pálya szélein történik-e bekerítés.

•**FillTest3 –** Ez a teszteset megvizsgálja, hogy történik-e bekerítés abban az esetben, ha az egyik játékos keríti be a másikat.