Beadandó feladat dokumentáció

Készítette: Rápli András
Neptun: LBZICA

Feladat:

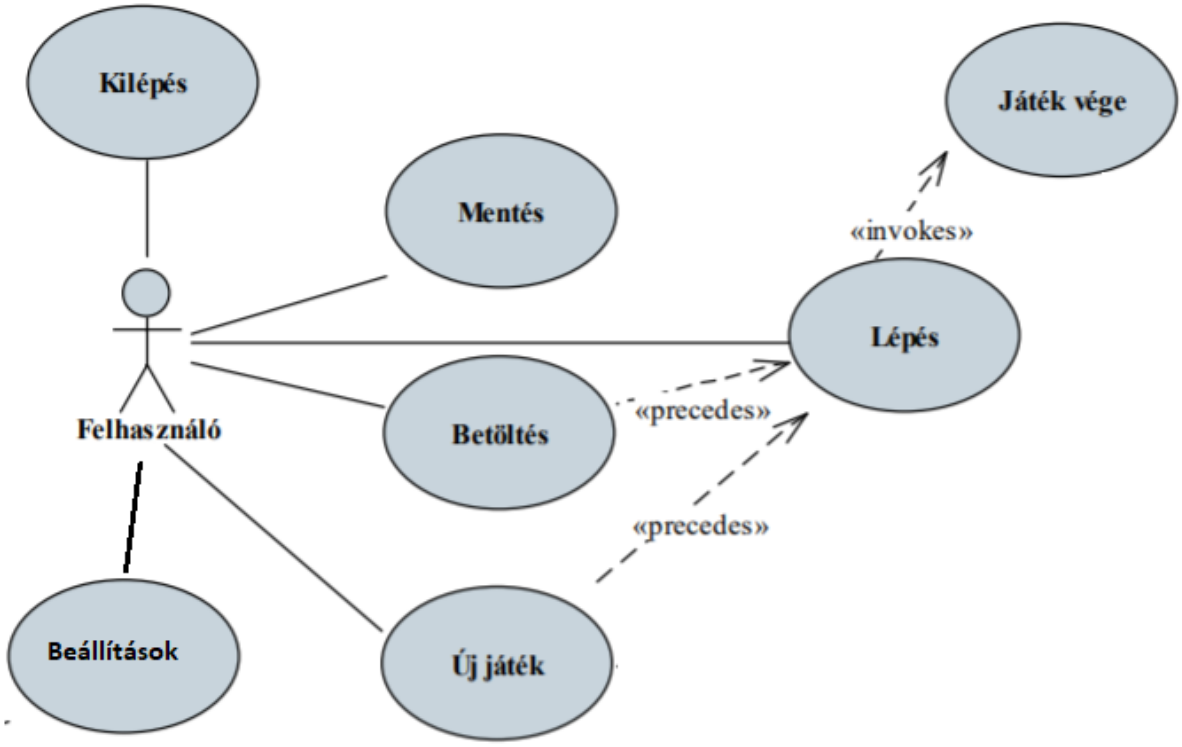
Készítsünk programot, amellyel az alábbi két személyes játékot játszhatjuk. Adott egy $n \times n$ pontból álló játéktábla, amelyen a játékosok két szomszédos pont között vonalakat húzhatnak (vízszintesen, vagy függőlegesen). A játék célja, hogy a játékosok a húzogatással négyzetet tudjanak rajzolni (azaz ők húzzák be a negyedik vonalat, független attól, hogy az eddigieket melyikük húzta). Ilyen módon egyszerre akár két négyzet is elkészülhet. A játék addig tart, amíg lehet húzni vonalat a táblán. A játékosok felváltva húzhatnak egy-egy vonalat, de ha egy játékos berajzolt egy négyzetet, akkor ismét ő következik. A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával (3 × 3, 5 × 5, 9 × 9), játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött (ha nem döntetlen). Játék közben a vonalakat, illetve a négyzeteket színezze a játékos színére.

Elemzés:

- A feladatot kétablakos asztali alkalmazásként Windows Forms grafikus felülettel oldottam meg.
- Az első ablak tartalmazza a játék beállításait, mely a játéktábla betöltése előtt jelenik meg.
- A második ablak maga a játéktábla, ahol a Négyzetek nevű játék játszható.

- A Beállítások ablakban lehetőség van a két játékos neveinek és a játéktábla méreteinek megadására.
- A játék automatikusan feldob egy ablakot, ha a játéktábla betelt, azaz nincs több elérhető lépés. Ekkor megjeleníti a két játékos elért pontszámait.

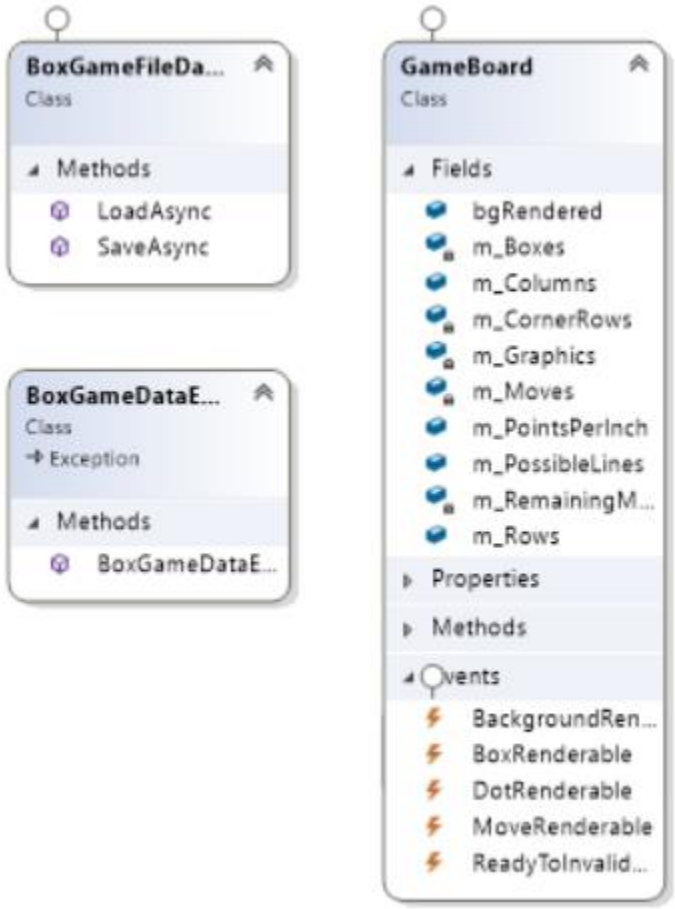
Az alábbi ábra tartalmazza a lehetséges felhasználói eseteket:



Tervezés:

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:



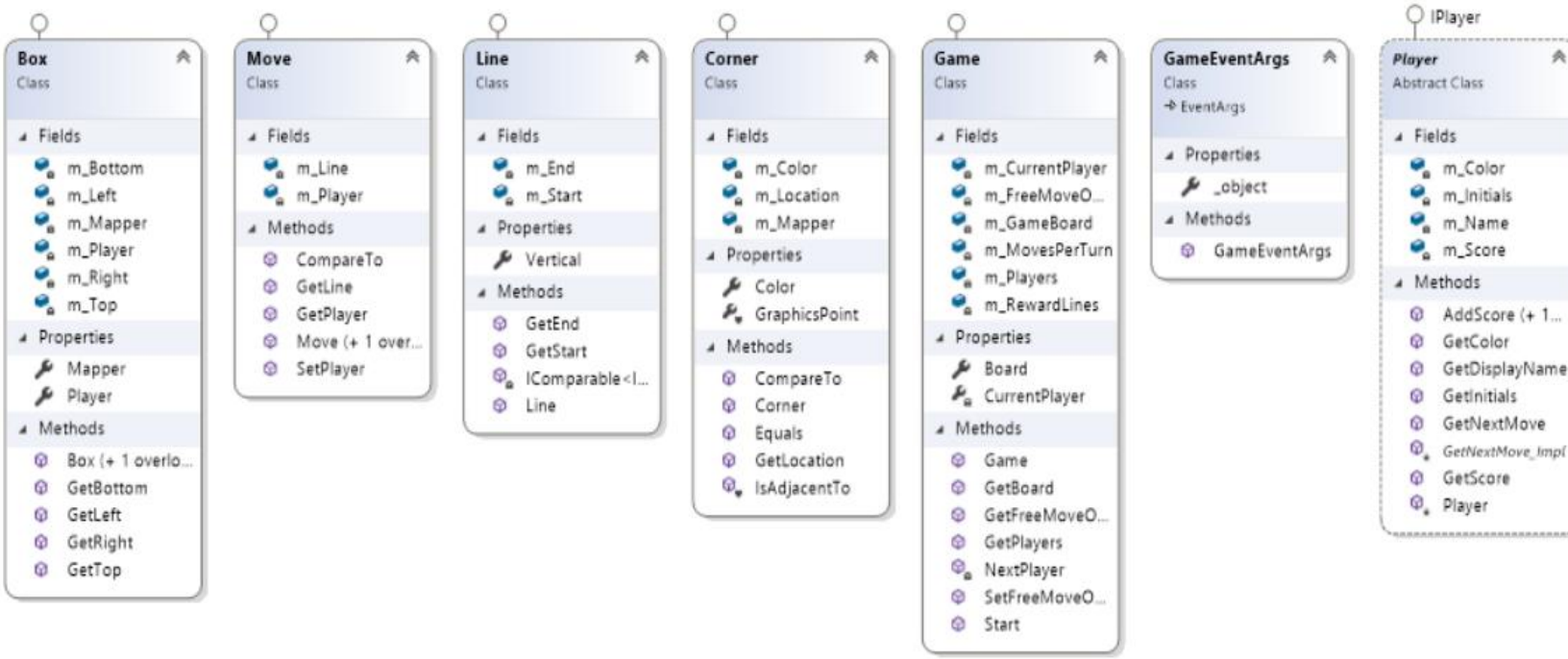
A perzisztencia réteg feladata a játékktáblával kapcsolatos adatok tárolása és a mentés/betöltés biztosítása. A játéktábla megvalósítása a GameBoard osztályban található. A GameBoard tárolja a táblaméretet (m_Rows, m_Columns), a megtett lépéseket (m_Moves), a táblán névő sorokhoz tartozó csúcsokat (m_CornerRows), és a táblára berajzolt négyzeteket (m_Boxes).

A BoxGameFileDataAccess osztály tartalmazza a tábla mentésének és betöltésének logikáját. A LoadAsync() metódus betölt, a SaveAsync() pedig ment a StreamReader és StreamWriter segítségével. Ennek az osztálynak a formális leírását az IBoxGameDataAccess interfész tartalmazza. Fájl írási/olvasási hiba esetén BoxGameDataException típusú kivételt dob a program.

Modell:

A modell az alábbi osztályokat tartalmazza:

1. Game (a játék aktuális állapota: ki következik, hányat léphet)
2. Player (a játékos adatai: neve, színe)
3. Move (egy játékbeli lépés adatai: honnan hova húzunk vonalat)
4. Line (egy vonal, két sorba rendezett pont (Corner))
5. Corner (egy pont a pályán, amiből vagy amibe vonalat lehet húzni)
6. Box (egy behúzott négyzet: mely pontok között van a négyzet és melyik játékos húzta be)
7. GameEventArgs (az eseménykezeléshez szükséges subclass, melyben a fenti osztályokat lehet átadni az esemény paramétereként)

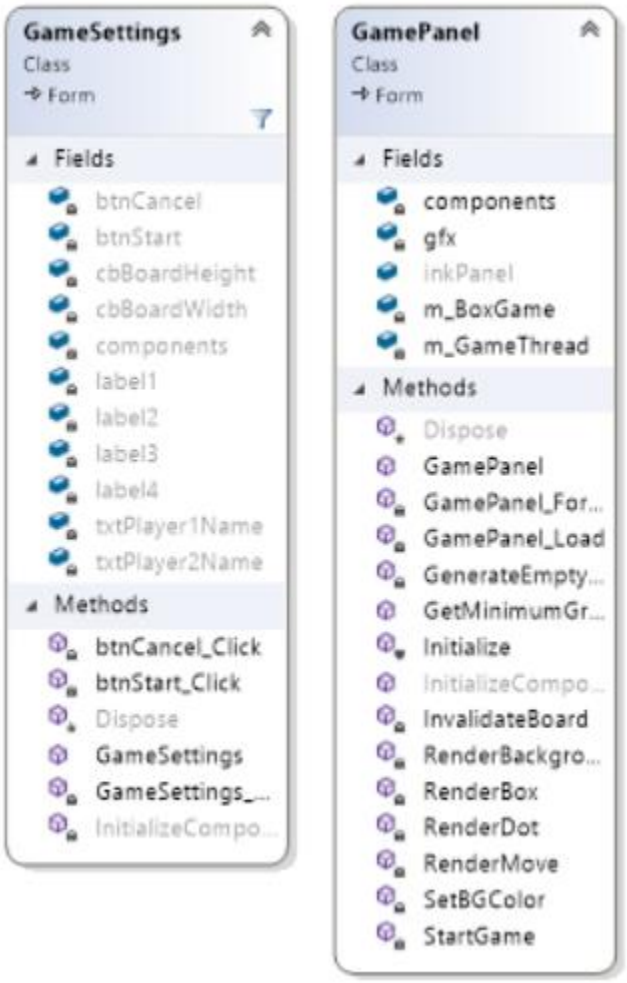


Nézet:

A nézet tartalmazza:

1. GameSettings (a beállítások menüt a lehetséges beállításokkal mint input vezérlőkkel)
2. GamePanel (a játéktáblát a mezőkkel mint input vezérlőkkel)

Mindkét osztály a System.Windows.Forms.Form -ból lett származtatva.



Tesztelés:

Három lényegi pontját teszteltem a játéknak, mellyel a feladat megoldása bizonyítható.

1. **TestGameBoardHasBeenInitialized()** metódussal megvizsgálom, hogy a játéktábla valóban megfelelően lett példányosítva, létezik a megfelelő dimenziókkal, hogy a táblán lehetséges vonalakat behúzni, illetve hogy a skálázás miatt nem mosódik el a felület.
2. **TestMove()** metódussal kipróbálok egy lépést a táblán, megvizsgálom, hogy rögzítve lett-e a lépés a játéktáblán, és a rögzített értékek megfelelőek
3. **TestBox()** metódussal behúzok négy vonalat, melyek négyzetet alkotnak és megvizsgálom, hogy létrejön-e a négyzet objektum, és az valóban ahhoz a felhasználóhoz tartozik-e, aki a legutolsó vonalat behúzta.