1.Beadandó feladat

Készítette:

Csóka Máté
E-mail: gt9bq9@inf.elte.hu

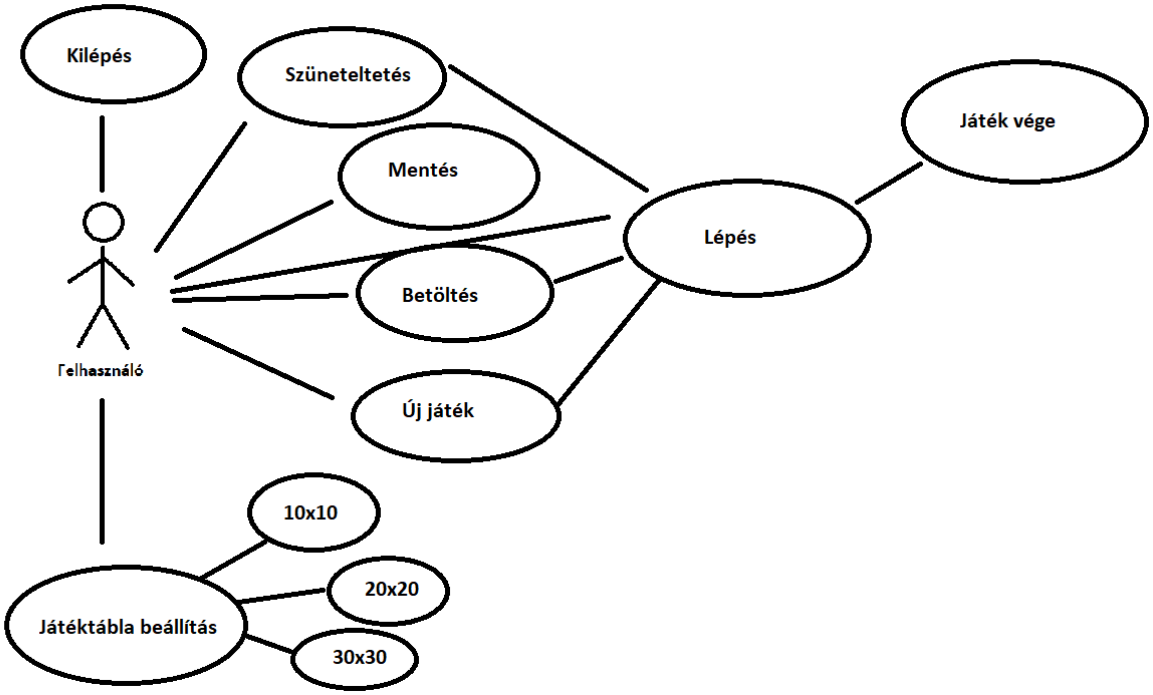
Feladat:

Készítsünk programot, amellyel az alábbi Reversi játékot játszhatjuk. A játékot két játékos játssza $n \times n$-es négyzetrácsos táblán fekete és fehér korongokkal. Kezdéskor a tábla közepén X alakban két-két korong van elhelyezve mindkét színből. A játékosok felváltva tesznek le újabb korongokat. A játék lényege, hogy a lépés befejezéseként az ellenfél ollóba fogott, azaz két oldalról (vízszintesen, függőlegesen vagy átlósan) közrezárt bábuit (egy lépésben akár több irányban is) a saját színünkre cseréljük. Mindkét játékosnak, minden lépésben ütnie kell. Ha egy állásban nincs olyan lépés, amivel a játékos ollóba tudna fogni legalább egy ellenséges korongot, passzolnia kell és újra ellenfele lép. A játékosok célja, hogy a játék végére minél több saját színű korongjuk legyen a táblán. A játék akkor ér véget, ha a tábla megtelik, vagy ha mindkét játékos passzol. A játék győztese az a játékos, akinek a játék végén több korongja van a táblán. A játék döntetlen, ha mindkét játékosnak ugyanannyi korongja van a játék végén. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (10 × 10, 20 × 20, 30 × 30), játék szüneteltetésére, valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött. 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 mentsük el és töltsük be).

Elemzés:

- A játékot három nehézségi szinttel játszhatjuk: könnyű (10x10-es tábla), közepes (20x20-as tábla), nehéz (30x30-as tábla). A program indításkor könnyű nehézséget állít be, és automatikusan új játékot indít.
- A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.
- Az ablakban elhelyezünk egy menüt a következő menüpontokkal: File (Save, Load, Pause, Exit), New Game (10x10,20x20,30x30). Az ablak tetején elhelyezünk 4 labelt, ami a játékosok pontját és idejét jelzi.
- A játéktáblát egy 10x10 (vagy 20x20, 30x30) nyomógombokból álló rács reprezentálja. A nyomógomb egérkattintás hatására megváltoztatja, a megnyomott cellát az adott játékos színére. Csak oda engedünk kattintani, ahol lehetséges lépés található (pirossal jelzett mezők). A már megváltoztatott, vagy érvénytelen lépéseket tartalmazó mezőket nem engedjük változtatni.
- A játék automatikusan kidob egy dialógusablakot, amikor vége a játéknak (megtelt a tábla, vagy 2x passzoltak a játékosok egymás után, tehát nincs több érvényes lépés). Szintén dialógusablakkal végezzük el a mentés, betöltést, illetve a játék szüneteltetését.

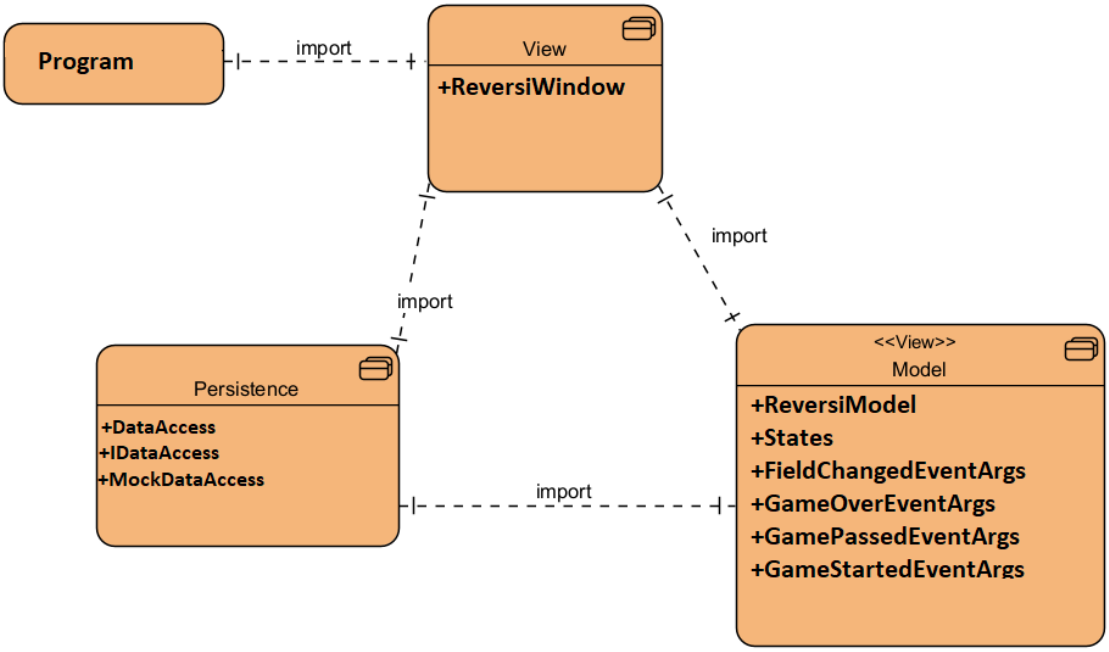
- A felhasználói esetek az 1. ábrán láthatóak.



1.ábra: Felhasználói esetek diagramja

Tervezés:

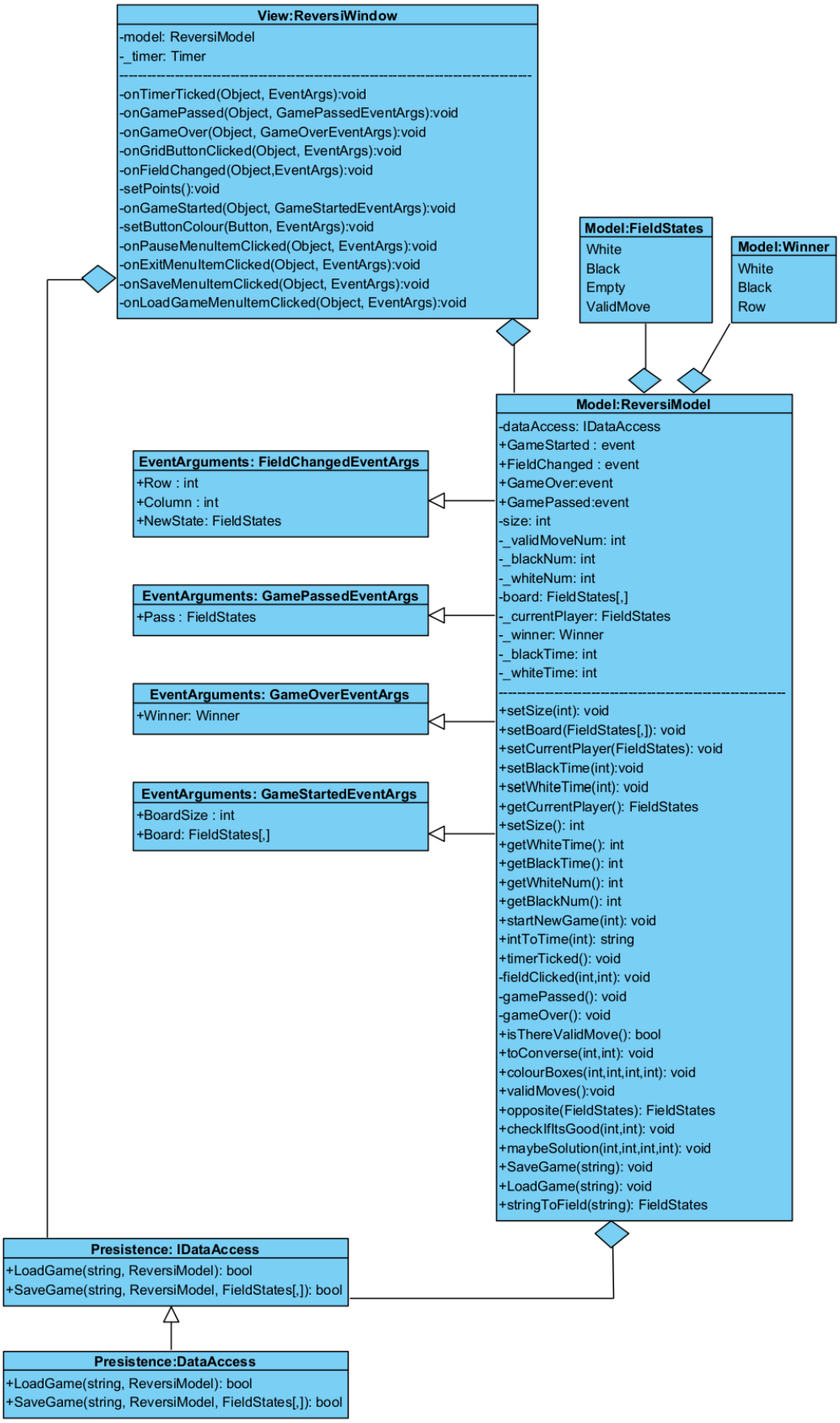
- Programszerkezet:
  - A programot háromrétegű architektúrában valósítjuk meg. A megjelenítés a ***View***, a modell a ***Model***, míg a perzisztencia a ***Persistence*** névtérben helyezkedik el. A program csomagszerkezete a 2.ábrán látható.
  - A program szerkezetét két projektre osztjuk implementációs megfontolásból: a ***Persistence*** és ***Model*** csomagok a program felületfüggetlen projektjében, míg a ***View*** csomag a Windows Formstól függő projektjében kap helyet.
- Perzisztencia:
  - Az adatkezelés feladata a Reversi táblával kapcsolatos információk tárolása, valamint a betöltés/mentés biztosítása.
  - A hosszú távú adattárolás lehetőségeit az ***IDataAccess*** interfész adja meg, amely lehetőséget ad a tábla betöltésére (***LoadGame***), valamint mentésére (***SaveGame***).
  - Az interfész szöveges fájl alapú adatkezelésre a DataAccess osztály valósítja meg.
  - 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 első sora megadja a tábla méretét. A második sor a következő játékost, majd sorban következik: a fekete ideje, a fehér ideje, majd a mátrix értékei.



2.ábra: Az alkalmazás csomagdiagramja

- Modell:
  - A modell lényegi részét a ***ReversiModel*** osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék egyéb paramétereit, úgymint az idő (***timeTicked***) és a valid lépések vizsgálata (***validMoves***). A típus lehetőséget ad új játék kezdésére (startNewGame), valamint lépésre (fieldClicked).
  - A játékállapot változásáról a **FieldChangedEventArg** esemény, míg a játék végéről a **GameOverEventArg** esemény tájékoztat.
  - A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (***LoadGame***) és mentésre (***SaveGame***)
- Nézet:
  - A nézetet a ***ReversiWindow*** osztály biztosítja, amely tárolja a modell egy példányát (**model**), valamint az adatelérés konkrét példányát (**dataAccess**).
  - A játéktábla egy dinamikusan létrehozott gombmező () reprezentálja. A felületen létrehozzuk a megfelelő menüpontokat, illetve státuszsort, valamint dialógusablakokat, és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását (**onGameStarted**), illetve az értékek beállítását (**setButtonColour**) külön metódusok végzik.
  - A játék időbeli kezelését egy időzítő végzi (**_timer**), amelyet mindig aktiválunk játék során, illetve inaktiválunk, amennyiben bizonyos menüfunkciók futnak.

- A program teljes statikus szerkezete a 3. ábrán található



3.ábra: Az alkalmazás osztálydiagramja

Tesztelés:

- A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a **ReversiGameModelTest** osztályban.
- Az alábbi tesztesetek kerültek megvalósításra:
  - **ReversiModelNewGameTenTest, ReversiModelNewGameTwentyTest, ReversiModelNewGameThirtyTest**: Új játék indítása, a mezők kitöltése, valamint a pontszámok ellenőrzése.
  - **ReversiModelStepTest**: Játékbeli lépések hatásainak ellenőrzése.
  - **ReversiModelGameOverTest**: A GameOver metódus meghíváskor a megfelelő játékos nyer-e.
  - **ReversiModelLoadTest**: A játék modell betöltésének tesztelése mockolt perzisztencia réteggel.