

1. Beadandó feladat dokumentáció

Készítette

Szabó-Galiba Máté

NTBGJ1

Feladat

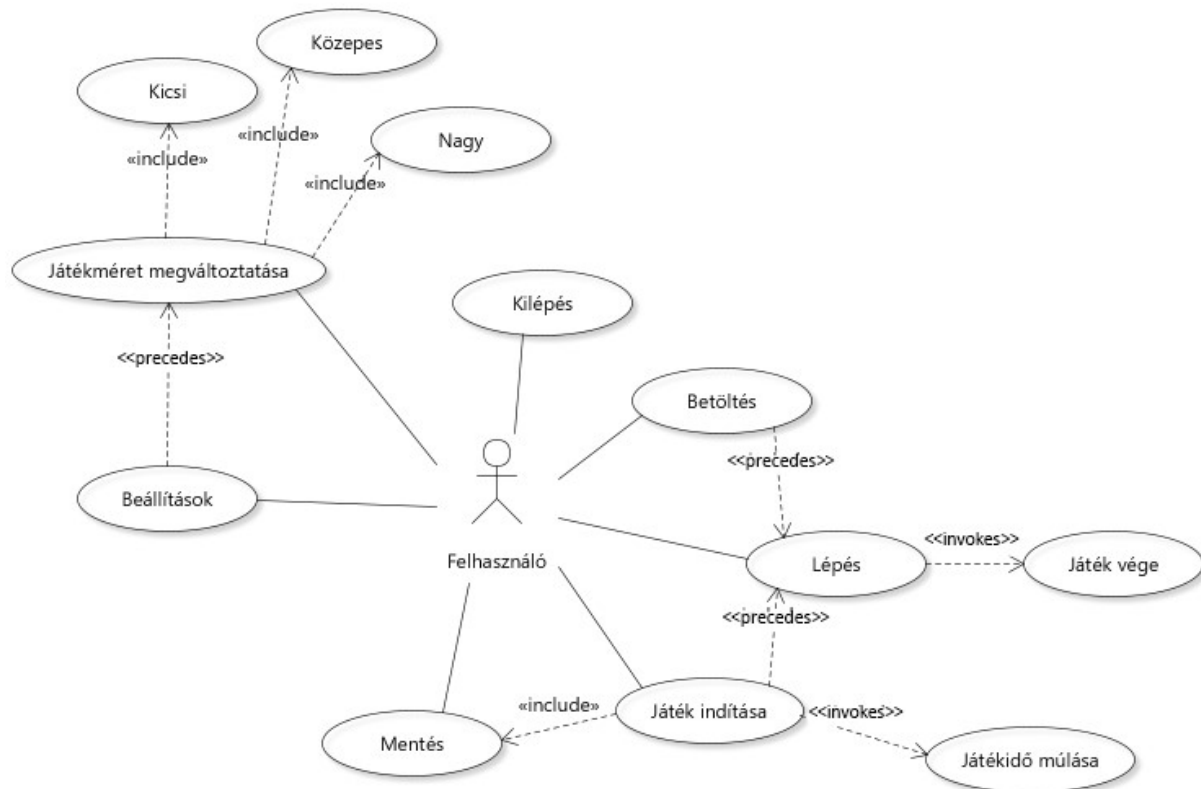
Potyogós amőba

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 (10×10, 20×20, 30×30), játék szüneteltetésére, valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött (a 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 mentsük el és töltsük be).

Elemzés

- A játékot három különböző méretben játszhatjuk: kicsi (10×10), közepes (20×20), és nagy(30×30). A program indításakor a játék megkezdése előtt lehetőséget ad a választásra, és alapértelmezett értéként kicsi méretet használ.
- A feladatot Windows Forms grafikus felület használatával valósítjuk meg.
- A menü ablakban játék betöltésére, új játék kezdésére, és a játék beállításainak a módosítására kapunk lehetőséget.
- A beállítások ablakban megváltoztathatjuk az újonnan kezdett játékok pályaméretét.
- A játék ablakban lesz egy menüsáv „Fájl” lehetőséggel, ahol el lehet menteni a játékot.
- A játék ablak alján egy státuszsávban lesz lehetőség a játék szüneteltetésére. Itt írja a játék állapotát is, illetve az egyes játékosok (X, O) eltelt idejét.
- A játéktáblát egy nyomógombokból álló rács reprezentálja. Adott nyomógombra bal egérgombbal kattintva a soron következő játékos új „korongot” illeszthet be a gomb oszlopának a tetejére, amennyiben az nem teli, ezzel befejezve a játékot, vagy átadva az irányítást a következő játékosnak.
- A játék végén (betelt tábla, vagy egy játékos sikeresen összegyűjtött 4 egy vonalban lévő, egymást követő korong) a státuszsávban a játék állapotát jelölő szöveg mutatja a játék eredményét. Ez után egyik játékos sem tehet több mozdulatot.
- A felhasználói esetek az 1. ábrán láthatók.

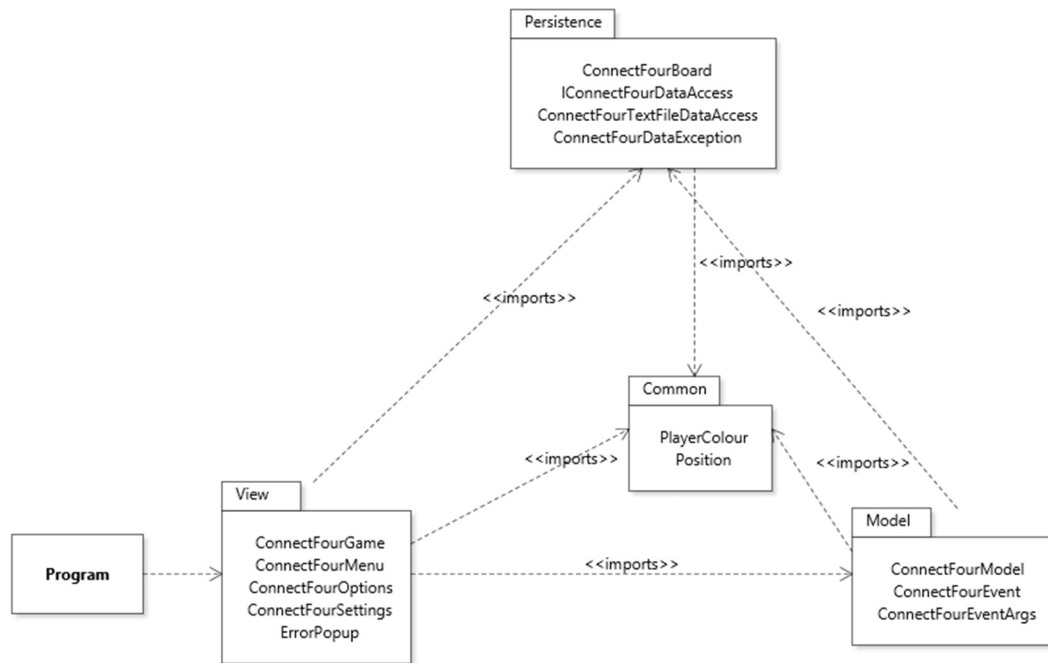


1. Ábra – Felhasználói esetek

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 **Common** névtérben néhány mindhárom réteg által egyszerű adatszerkezetet valósítunk meg. A program csomagszerkezete a 2. ábrán található.
- A program szerkezetét két projektre osztjuk implementációs megfontolásból: a **Common**, **Persistence** és **Model** csomagok a program felületfüggetlen projektében, míg a **View** csomag a Windows Formstól függő projektben kap helyet.



2. Ábra – Csomagdiagram

1. Perzisztencia:

- Az adatkezelés feladata a potyogós amőba (továbbiakban, illetve programkódban: **ConnectFour**) táblával kapcsolatos információk tárolása, valamint a betöltés/mentés biztosítása.
- A **ConnectFourBoard** osztály egy Connect Four táblát biztosít. Alkalmas alapvető információk lekérdezésére (**Width**, **Height**, **IsFull**, **IsOver**), tárolja a megtett lépések listáját (**Moves**), illetve az egyes játékosok által gondolkozással töltött időt (**PlayerTime**). Új lépés megjátszása az **Insert** metóduson keresztül lehetséges. Új lépés kijátszása előtt alkalmas ellenőrizni, hogy az adott lépés győzelmet eredményezne-e az azt megjátszó játékosnak, illetve hogy milyen pozíciókon lennének a győztes kombinációba tartozó mezők (**WinningMove**).
 - Az osztálynak vannak alosztályai (**Column**, **WinCache**) amelyek segítik a működésben. A **Column** osztály egy oszlop reprezentációja, míg a **WinCache** egy lépés eredményeként keletkező győzelem számításában segít.
- A hosszú távú adattárolás lehetőségeit az **IConnectFourDataAccess** interfész adja meg, amely lehetőséget ad a tábla betöltésére (**LoadAsync**) és a tábla mentésére (**SaveAsync**). A műveleteket hatékonysági okból aszinkron módon valósítjuk meg.
- Az interfészt szöveges fájl alapú adatkezelésre a **ConnectFourTextFileDataAccess** osztály valósítja meg. A fájlkezelés során fellépő hibákat a **ConnectFourDataAccessException** jelzi.
- A program az adatokat szöveges fájlként tudja eltárolni, melyek a **.cfs** kiterjesztést kapják. Ezeket az adatokat a játék során (akár a játék befejeződése után) lehet menteni, a főmenüből betölteni.
- A fájl felépítése a következő:
 - Az első sorban a mentés során használt **.cfs** formátum verziószáma van. Jelenleg csak egy verzió létezik, így ez mindig **1.0**.
 - A második sorban a tábla méretei vannak (szélesség, magasság) egy szóközzel elválasztva.

- A harmadik sorban a játékosok (X, O) gondolkozással töltött idejei vannak egész milliszekundumban, szóközzel elválasztva.
- Innentől kezdve soronként egy-egy lépés van, ahogy a játék folyamán időrendi sorrendben követték egymást. Egy lépés az oszlop sorszámaival van jelölve (0-tól kezdve), ahol az aktuális lépést tette a játékos.
- Végül a fájl egy üres sorral zárul.

2. Modell:

- A modell lényegi része a **ConnectFourGameModel** osztály, amely a tábla és a nézet közötti kommunikációért felelős. Fő feladata az események kiváltása, emellett az idő elteltét jelzi a táblának.
 - Új játék kezdetekor a tábla méretét **BoardSize** paraméterként adjuk át a konstruktornak.
 - Játék betöltésekor a betöltött táblát a **LoadBordAsync** statikus metódusból kapjuk meg, majd ezt adjuk át a konstruktornak paraméterként.
- Az események argumentuma **ConnectFourEventArgs** típusúak, ez tárolja az esemény típusát (**EventType**, **ConnectFourEvent** felsorolástípus), az aktuális játékos gondolkozással töltött idejét (**GameTime**), az esetlegesen megtett lépés pozícióját (**Position**), illetve győzelem esetén a győztes cellák pozícióit a táblázaton belül (**Winner**).
- A játékállapot változásáról a **GameEvent** esemény értesíti a nézetmodellt – ez paramétereként kap egy **ConnectFourEvent** felsorolási típust, aminek az értékei **PLACEMENT**, **PAUSE**, **WIN**, és **DRAW** lehetnek.
- Az idő elteltét a **TimeAdvanced** esemény jelzi, ebben az esetben a **ConnectFourEvent** értéke **TIME_ELAPSED**.

3. Nézet

- A program indításakor a **ConnectFourMenu** ablak jelenik meg a felhasználó számára, itt navigálhat az új játék betöltése, a beállítások módosítása, és a játék indítása lehetőségek között.
- A „Beállítások” gombra kattintva a **ConnectFourSettings** ablak jelenik meg. Ez egy **ConnectFourOptions** típusú objektumot kap paraméterként, amibe a módosított beállítások mentésekor az új preferenciák kerülnek.
- A „Játék betöltése” gombra kattintva egy fájl választó dialógus jelenik meg. Amennyiben a felhasználó bezárja, nem megy végbe megfigyelhető művelet, viszont ha a felhasználó kiválaszt egy fájlt, a program megpróbálja betölteni azt. A sikertelen betöltést egy felugró **ErrorPopup** ablak jelzi. Sikeres betöltés esetén a játék ott folytatódik, ahol mentéskor abbahagyták.
- Az „Új játék” gombra kattintva a játékos új **ConnectFourGame** játéklapot a beállításokban megadott táblázatmérettel (alapértelmezettként kicsi, 10x10). Az játékhoz használt modell a **_model** adattagban van elmentve, ebben van a tábla. A tábla egy **TableLayoutPanel**-ben van megjelenítve, ahol minden cella saját **ConnectFourButton** (**Button**-ból örökölt) típusú. Az idő egy **ToolStrip**-ben van megjelenítve illetve itt lehet szüneteltetni is a játékot. A játék állapotát egy címke jelzi ebben a sávban.

Tesztesetek

1. Egyszerű teszt a győzelemre
2. Egyszerű teszt a döntetlenre

A tesztek magukban foglalják az események ellenőrzését.