

1. Beadandó feladat dokumentáció

Készítette:

Trembickij Máté

Neptun: miszuu

E-mail: miszuu@inf.elte.hu

Feladat:

Készítsünk programot a közismert Tetris játékra.

Adott egy \times pontból álló tábla, amely kezdetben üres. A tábla tetejéről egymás után új, 4 kockából álló építőelemek hullanak, amelyek különböző formájúak lehetnek (kocka, egyenes, L alak, tető, rombusz). Az elemek rögzített sebességgel esnek lefelé, és az első, nem telített helyen megállnak. Amennyiben egy sor teljesen megtelik, az eltűnik a játéktérrel, és minden felette lévő kocka eggyel lejjebb esik.

A játékosnak lehetősége van az alakzatokat balra, jobbra mozgatni, valamint forgatni óramutató járásával megegyező irányba, így befolyásolhatja azok mozgását. A játék addig tart, amíg a kockák nem érik el a tábla tetejét.

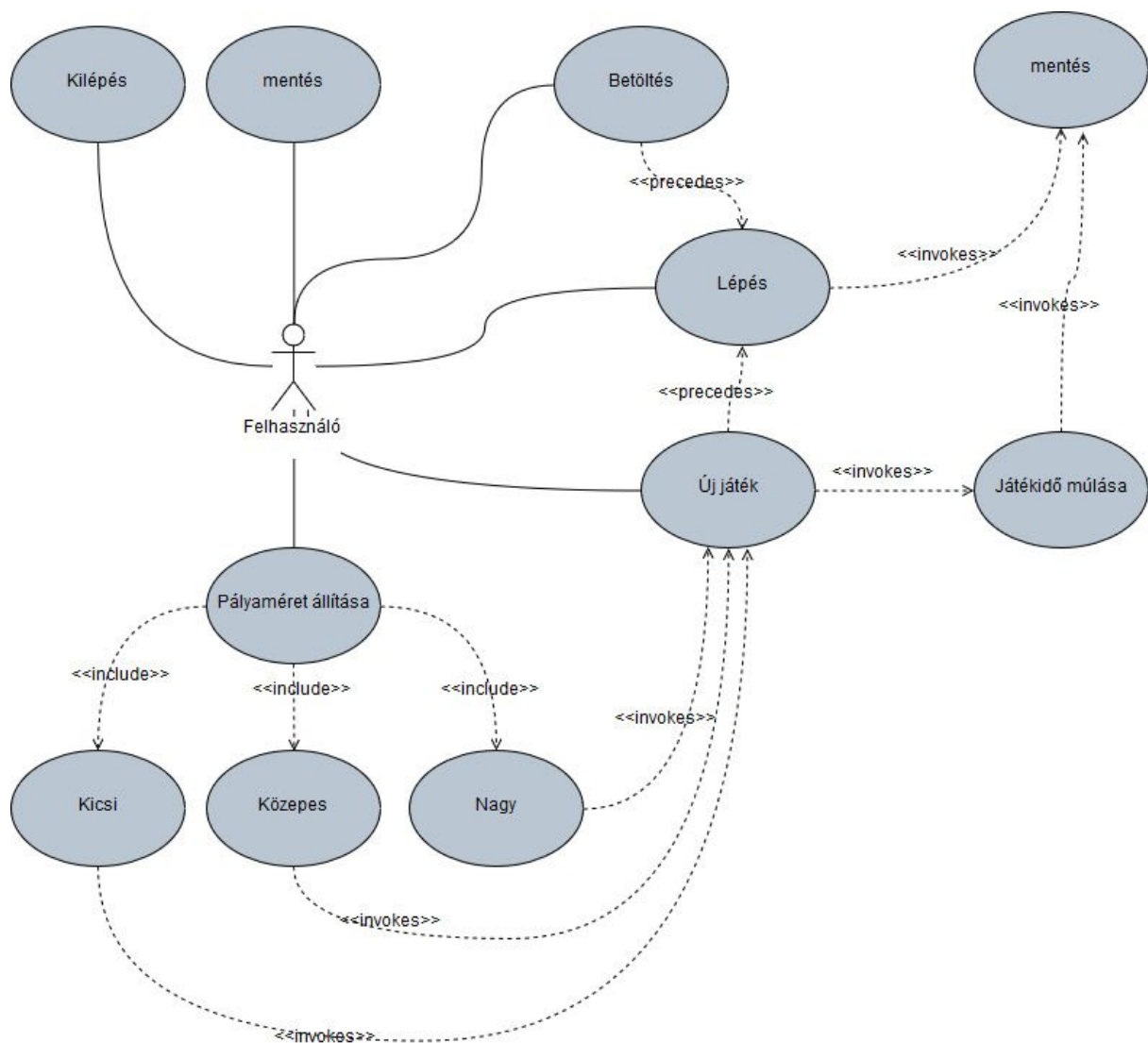
A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával (4×16 , 8×16 , 12×16), valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozognak az elemek). Ismerje fel, ha vége a játéknak, és jelenítse meg, mennyi volt a játékidő. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére.

Elemzés:

- A játékot három pályamérettel játszhatjuk: létezik a kicsi (4 sor, 16 oszlop), közepes (8 sor, 16 oszlop), nagy (12 sor, 16 oszlop). A program indításakor közepes méretű pályamérettel, és

el is indul egy új játék. A játék méretét a felül lévő menüben lehet állítani (size-on belül small/medium/large). A gombok megnyomásával azonnal új játék indul.

- A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.
- Az ablakban felül található egy menü a következő menüpontokkal: File (New Game, Size, Save, Load, Exit), Pause gomb amivel a játékot szüneteltethetjük. Az ablak alján megjelenítünk egy státuszsort, amely az eltelt időt jelzi.
- A játéktáblát egy (4x16, , 8 x 16, 12 x 16) gombokból álló rács reprezentálja. A nyomógombok, nem funkcionálnak másra, csak a játék megjelenítésére. A játékot a billentyűzettel irányítja a játékos azon belül a nyilakkal. A bal, le illetve jobb nyíl a vele megegyező irányba mozgatja az alakzatot, míg a felfelé nyíl forgatja.
- Az alakzatok a leérkezéskor (amikor nem tudnak már tovább leesni), megszilárdulnak, s beleolvadnak a játéktáblába. Ezzel együtt megidéznek egy újabb alakzatot a kezdőpontban, ha nem sikerül ez a cselekedet, a játék befejeződik.
- A játék automatikusan feldob egy dialógus ablakot, amikor vége van a játéknak. A dialógusablak megjeleníti az időt. Szintén dialógusablak az ami kezeli a játék mentését illetve a betöltését. A fájlnevet a felhasználó adhatja meg.
- A felhasználói esetek az 1. ábrán láthatóak.



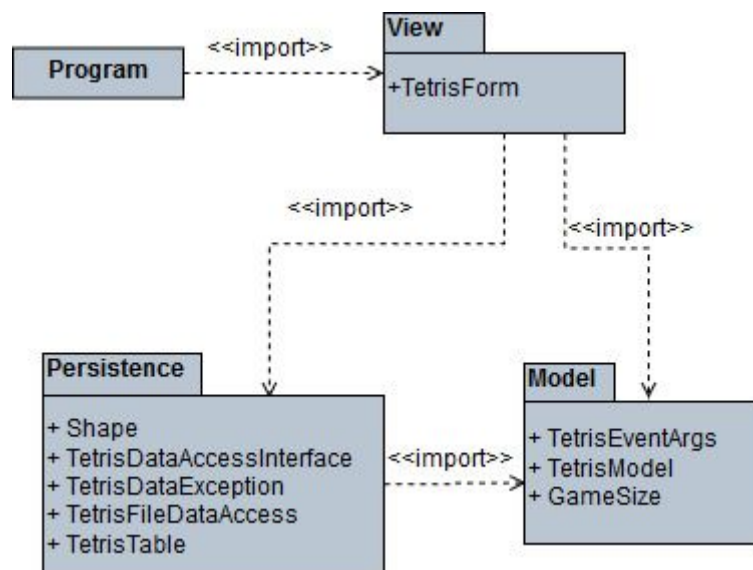
1. ábra: Felhasználói esetek diagramja

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.
- Perzisztencia:
 - Az adatkezelés feladata a Tetris táblával kapcsolatos információk tárolása, valamint a betöltés/mentés biztosítása.
 - A TetrisTable osztály egy érvényes Tetris táblát biztosít, ahol minden mező értéke ismert az értéke (**_tetrisTable**) (tehát hogy van ott valami, fixen van ott valami, vagy

pedig ideiglenesen van-e ott valami), és ismert a típuskódja is (**_typeTable**). Mindkettő a játék kezdetekor generált, kezdőértékekkel. A tábla alapértelmezetten (8+1) x 16-os, ez a konstruktorban változtatható. A tábla lehetőséget ad különböző állapotok elérésére (**Table, TypeTable, ShapeRotation, ShapeCordY, ShapeCordX, Shape, Size, Time**), valamint a jelenlegi alakzat mozgására (**checkAndMoveLeftPosition, chechAndMoveDownPosition, checkAndMoveRightPosition, checkAndRotatePosition**), a sorok ellenőrzésére (**deleteCompletedRows**), új alakzat kreálására (**newShapeSpawn**) és a játék léptetésére (**gameAdvanced**).

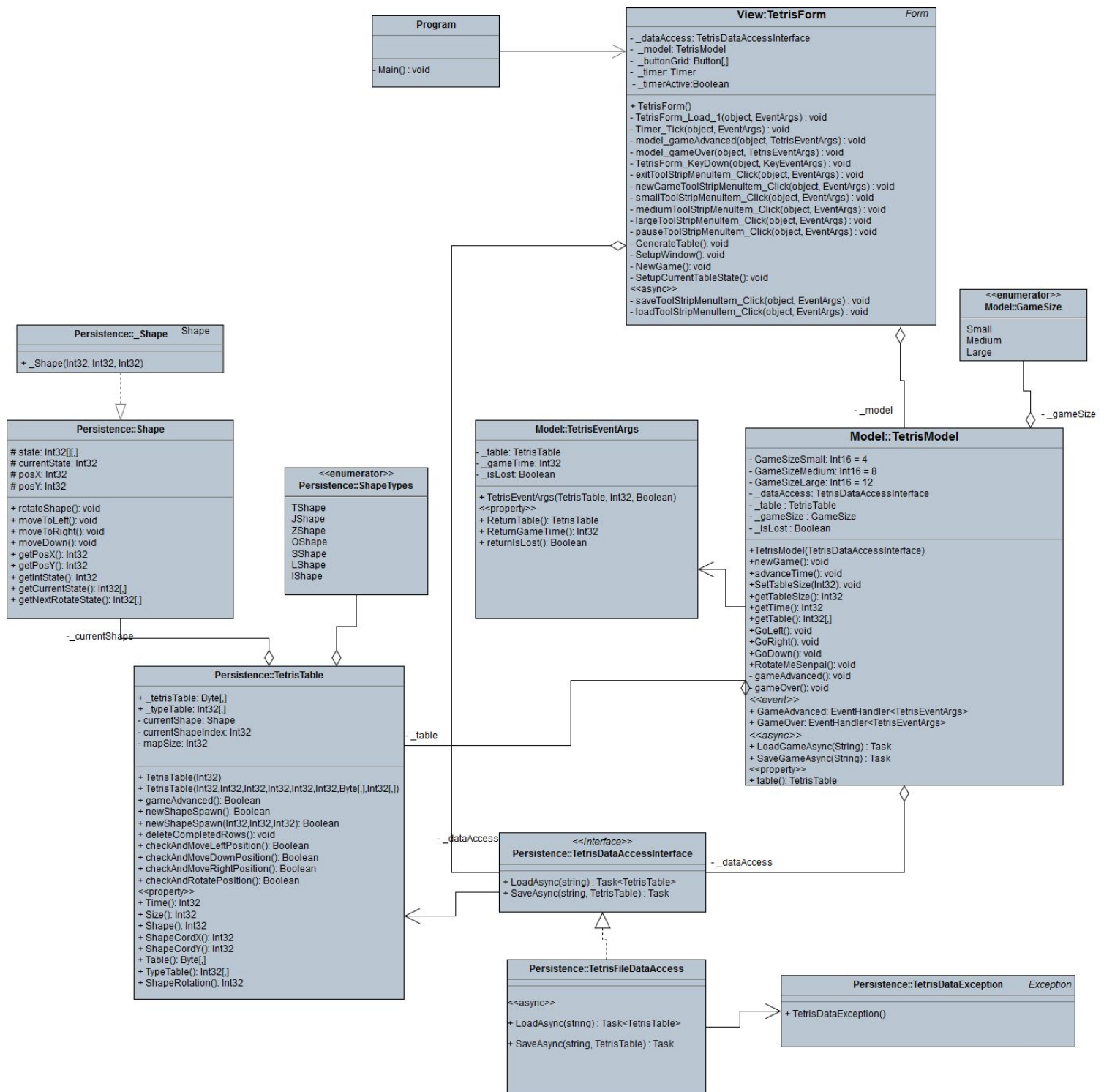
- A **Shape** osztály egy alakzatot reprezentál, ami definiálja a lehetséges állapotait (**state**), jelenlegi állapotindexét (**currentState**), és a koordinátáját (**posX, posY**). Ennek vannak különböző függvényei amik változtatják a különböző tulajdonságait (**moveToLeft, moveToRight, rotateShape, moveDown**), meg különböző függvények, amikkel lekérdezhető az állapotaik (**getPosX, getPosY, getIntState, getCurrentState, getNextRotateState**).
- A hosszú távú adattárolást a **TetrisDataAccessInterface** szolgáltatja, amely lehetővé teszi egy elmentett játékállás betöltését (**LoadAsync**) avagy elmentését (**SaveAsync**). A műveleteket hatékonysági okokból aszinkron módon valósítjuk meg.
- Az interfészt szüveges alapú adatkezelésre a **TetrisFileDataAccess** osztály valósítja meg, a fájlkezelés során történő hibákat pedig a **TetrisDataException** kivétel jelzi.
- A program az adatokat szüveges fájlként tárolja, amelyek a tt kiterjesztést kapják. Ezeket az adatokat a programban bármikor be lehet tölteni, illetve ki lehet menteni az aktuális állást.
- A fájl első sora megadja a tábla méretét (4/8/12), valamint az időt. A következő sorban található az aktuális alakzatról való fontos információk, azaz az alakzat típusa, alakzat x koordinátája, y koordinátája és az alakzat forgásindexe. A fájl többi része izomorf leképezése a játéktáblának, összesen méret+1 sor és 16 oszlop található szöközőkkel elválasztva. Ez a mátrix kétszer található meg, mivel a másodikban a táblán lévőknek a színkódja található (avagy az alakzat típusa).



2.ábra: Az alkalmazás csomagdiagramja

- Modell:
 - A modell lényegi részét a **TetrisModel** osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék egyéb paramétereit mint az, hogy veszített-e az állás (**_isLost**) és a játék méretét (**_gameSize**). Az osztály lehetőséget ad új játék kezdésére (**newGame**), valamint a játék léptetésére (**advanceTime**).
 - A játékállapot változásáról a **gameAdvanced** esemény, míg a játék végéről a **gameOver** esemény ad tájékoztatást. Az események argumentuma (**TetrisEventArgs**) tárolja a győzelem állapotát, a játék idejét, illetve hogy vége van-e játéknak vagy sem.
 - A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad beöltésre (**LoadGameAsync**) és mentésre (**SaveGameAsync**).
 - A játék méretét a **_gameSize** felsorolási típuson át kezeljük, és a **TetrisModel** osztályban konstansként tároljuk az egyes értékeket.
- Nézet:
 - A nézetet a **TetrisForm** osztály biztosítja, amely tárolja a modell egy példányát (**_model**), valamint az adatelérés konkrét példányát (**_dataAccess**).
 - A játéktáblát egy dinamikusan létrehozott gombmező (**_buttonGrid**) reprezentálja. A felületen létrehozunk a menüpontokat, illetve az alul lévő státuszsort, valamint a dialógusablakokat, és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását (**GenerateTable**), illetve az értékek beállítását a (**SetupCurrentTableState**) metódusok végzik. Az ablak méretreszabását a (**SetupWindow**) metódus végzi.

- A játék időbeli kezelését egy időzítő végzi (**_timer**), amelyet a pause gombbal le tudunk állítani, s utána újraindítani is a pause gombbal tudja a felhasználó. Inaktíváljuk, amennyiben a játékos elveszítette a játékot, illetve mentés és betöltés esetén is inaktíváljuk.
- A program teljes statikus szerkezete a 3. ábrán látható.



3. ábra: Az alkalmazás osztálydiagrammja

Tesztelés:

- A modell funkcionalitását egységtesztek segítségével lett ellenőrizve a
- Az alábbi tesztesetek kerültek megvalósításra:
 - **TetrisSmallGame, TetrisMediumGame, TetrisLargeGame:** Pályaméret beállítása, új játék indítása, időellenőrzés és játékméret ellenőrzése. Játéktábla méretének ellenőrzése.
 - **TetrisAdvanceTime:** Itt ellenőrizzük a játék idejének telését. És a játék léptetésének a hatását. Leellenőrizzük a bal, illetve a jobbra való mozgást is.
 - **TetrisLeftSideTest:** Itt leellenőrizzük a játéktábla jobb oldalát. Nem lehet kimenni az alakzattal.
 - **TetrisRightSideTest:** Itt leellenőrizzük a játéktábla jobb oldalát. Nem lehet kimenni az alakzattal, s megáll .
 - **TetrisDownMovement:** A játékbeli lefelé mozgást ellenőrzi, s a játék vége event is meghívódik nagy valószínűséggel.