

**Készítette:**

Décsei Dorka

[aeikir@inf.elte.hu](mailto:aeikir@inf.elte.hu)

**Feladat:**

Tetris

Készítsünk programot a közismert Tetris játékra. Adott egy  $n \times m$  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 játéklemezről, 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 pálya méret megadásával ( $4 \times 16$ ,  $8 \times 16$ ,  $12 \times 16$ ), valamint játék szüneteltetésére (ekkor nem telik az idő, Eseményvezérelt alkalmazások 2025/2026 őszi félév 5 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.

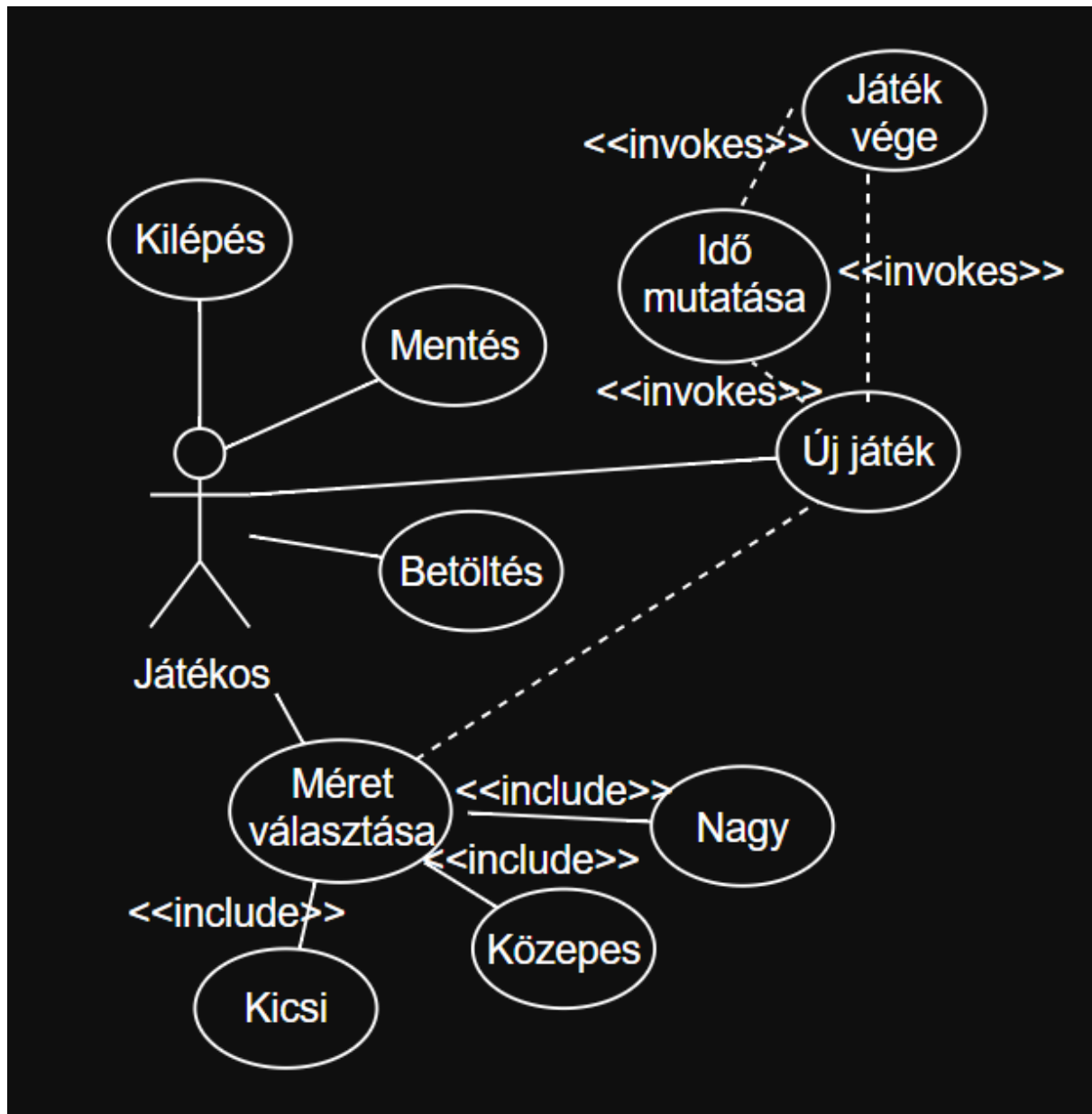
**Elemzés:**

A feladat egy Tetris játékprogram megvalósítása.

A játékmenet főbb jellemzői:

- **Játéktér:** A tábla mérete új játék indításakor választható három lehetőség közül:  $4 \times 20$ ,  $8 \times 20$ ,  $16 \times 20$
- **Irányítás:** A játékos a billentyűzet segítségével befolyásolhatja az alakzatok mozgását: balra és jobbra mozgathatja, valamint az óramutató járásával megegyező irányba forgathatja őket. Lehetőség van az alakzatok gyorsított leejtésére is.
- **Játékmechanika:** Az elemek folyamatosan, rögzített sebességgel esnek. Ha egy alakzat földet ér vagy egy másik alakzatba ütközik, megáll és rögzül. Amennyiben egy sor teljesen megtelik elemekkel, az a sor törlődik, és a felette lévő sorok lejjebb csúsznak.
- **Játék vége:** A játéknak akkor van vége, ha az újonnan érkező alakzat már nem fér el a táblán (eléri a tetejét). Ekkor a program megjeleníti a játékidőt.
- **Funkciók:** A program lehetőséget biztosít a játék szüneteltetésére (ekkor az időmérés és az elemek mozgása is megáll), valamint a játékállapot mentésére és betöltésére.

Felhasználói esetek:



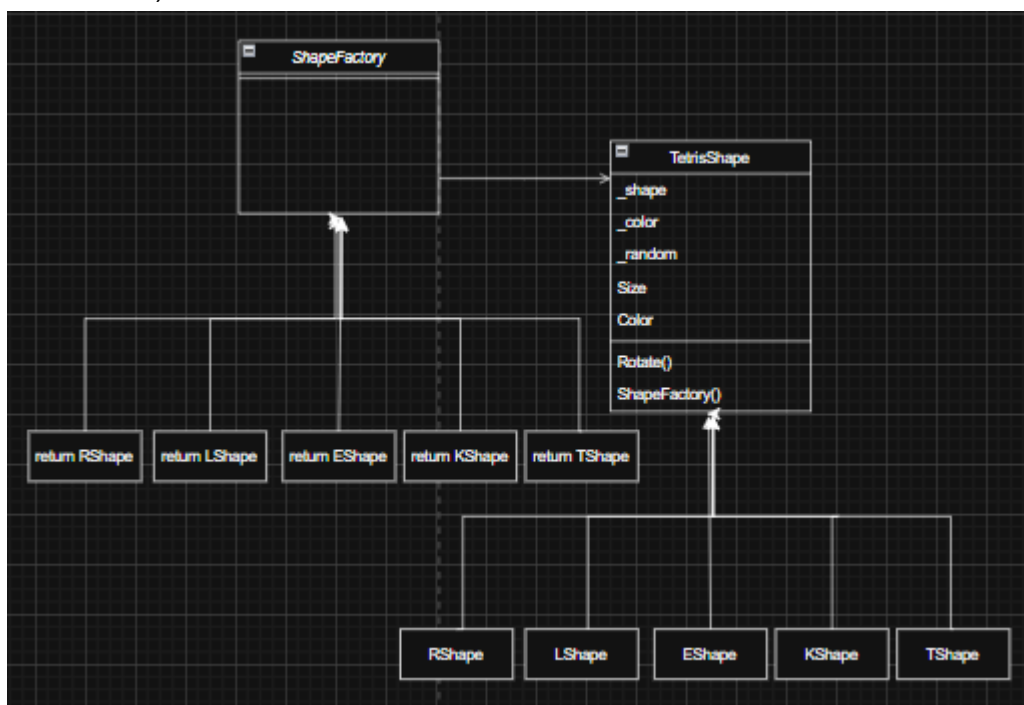
## Tervezés és a program szerkezete

A programot **MVVM (Model-View-ViewModel)** architektúrában valósítjuk meg, szétválasztva a megjelenítést, az modellt és az adatkezelést.

### Perzisztencia

Az adatkezelés feladata a játéktábla és a játékállapot (idő, pontszám, alakzatok helyzete) tartós tárolása.

- **ITetrisDataAccess interfész:** tábla betöltése és mentése
- **TetrisFileDataAccess:** Az interfész implementációja szöveges fájl alapú tároláshoz. A fájlban soronként tároljuk a tábla méretét, a játékidőt és a rács tartalmát.
- **TetrisShape:** Az egyes alakzatok (I, J, L, O, S, T, Z) logikáját (forgatási fázisok, koordináták) tartalmazzák.



### Modell (Model)

A játék logikája és alap felépítése

- **TetrisTable:** Ez az osztály reprezentálja a játéktáblát adat szinten, tárolja a rács méretét és az egyes mezők állapotát
- **Timer:** időzítő
- **TetrisGameModel:**
  - Ellenőrzi az ütközéseket és a teli sorokat.
  - Eseményeket vált ki: GameAdvanced (idő telik/mozgás történt), GameOver (játék vége), GameCreated (új játék indult).

## Nézetmodell (ViewModel)

A ViewModel köti össze a Modellt a View-val. Állapotot tart nyilván a felület számára, és parancsokat biztosít.

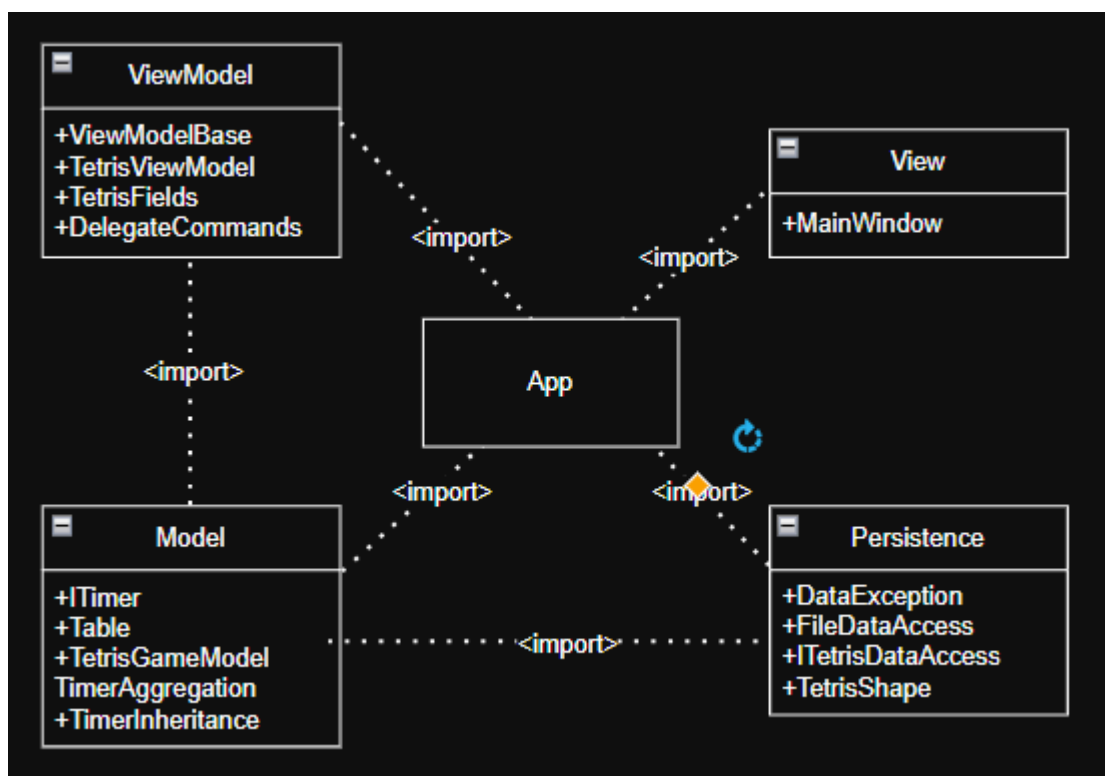
- **TetrisViewModel:**
  - **Properties:** gameTime (kijelzett idő), Score (pontszám), Fields (a mezők gyűjteménye).
  - **Parancsok (DelegateCommand):** NewGameCommand, LoadGameCommand, SaveGameCommand, MoveLeftCommand, MoveRightCommand, RotateCommand, DropCommand. Ezeket kötjük a billentyűzethez és a menükhöz.
  - **Mezők:** A Fields gyűjtemény TetrisField objektumokat tartalmaz, amelyek tulajdonságaihoz (pl. Color) a View gombjai vannak kötve. Ha a Modellben változik a tábla, a ViewModel frissíti a TetrisField színeit, ami automatikusan megjelenik a felületen.

## Nézet (View)

- **MainWindow:** A WPF ablak, amely XAML-ben definiálja a felületet.
  - Tartalmaz egy ItemsControl-t, amely UniformGrid-ben jeleníti meg a TetrisField-eket.
  - A billentyűleütéseket (InputBindings) a ViewModel parancsaihoz rendeli.
  - Nem tartalmaz játékkodot, csak megjelenítést.

## Alkalmazás (App)

- **App osztály:** A program belépési pontja. Itt példányosítjuk a rétegeket (DataAccess -> Model -> ViewModel -> View) és kötjük össze az eseménykezelőket (pl. ha a Model jelzi a játék végét, az App megjelenít egy üzenetablakot).



## Tesztelés

A modell működését egységtesztetekkel ellenőrizzük, különválasztva a felülettől és a fájlrendszerétől. A tesztekhez a MSTest keretrendszert és a Moq könyvtárat használjuk

Megvalósított tesztesetek:

- **TetrisGameModelNewGameTest:** Ellenőrzi, hogy új játék indításakor a tábla üres-e, a méret megfelelő-e, és az időzítő elindul-e.
- **TetrisGameModelMoveTest:** Teszteli az alakzatok mozgását és azt, hogy a falaknál megállnak-e.
- **TetrisGameModelRotateTest:** Ellenőrzi, hogy az alakzatok forgatása helyes-e, és nem fordulnak-e bele a falba vagy más alakzatba.
- **TetrisGameModelGravityTest:** Ellenőrzi, hogy az alakzatok rendesen esnek-e lefelé
- **TetrisGameModelFullRowTest:** Teszteli, hogy ha egy sor megtelik, akkor az törlődik, és a pontszám növekszik.
- **TetrisGameModelGameOverTest:** Ellenőrzi, hogy ha az alakzatok elérik a tábla tetejét, a játék véget ér-e
- **TetrisGameModelLoadTest:** Teszteli, hogy a játékállapot betöltése jól működik-e