1. Beadandó feladat dokumentáció

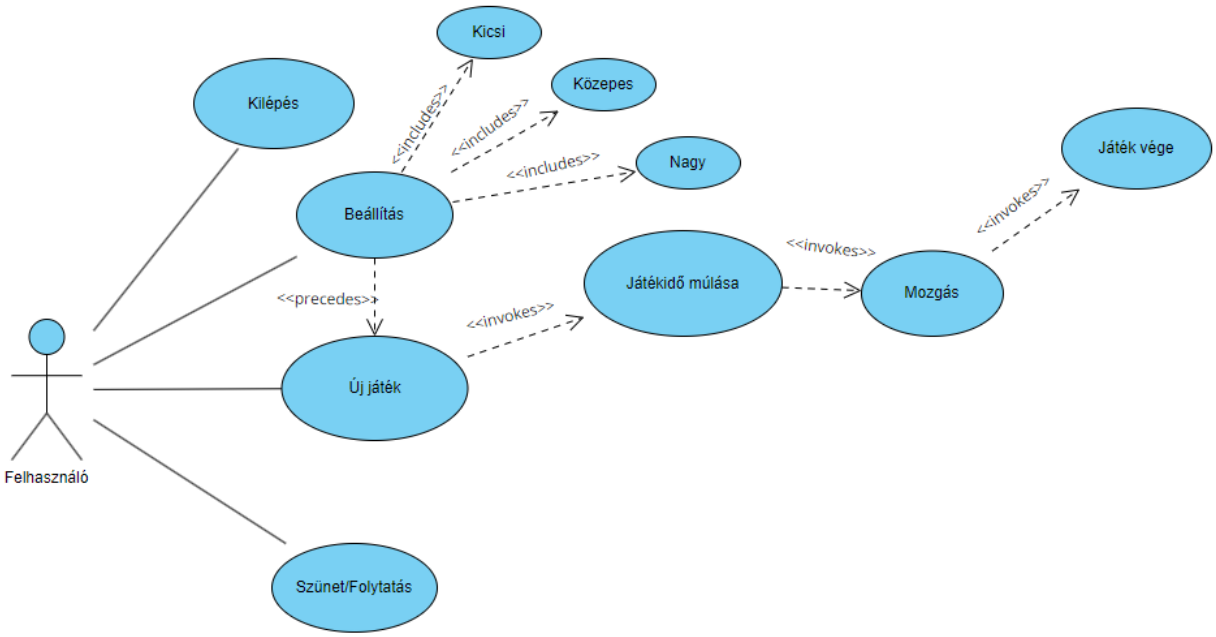
Készítette: Márton Milán
Email: bwaf4v@inf.elte.hu

Feladat: 11. Snake

Készítsük programot, amellyel a klasszikus kígyó játékot játszhatjuk. Adott egy $n \times n$ elemből álló játékpálya, amelyben akadályok (falak) találhatóak. A játékos egy kezdetben 5 hosszú kígyóval indul a képernyő közepén, amely vízszintesen, illetve függőlegesen halad rögzített időközönként a legutoljára beállított irányba. A kígyóval elfordulhatunk balra, illetve jobbra. A pályán véletlenszerű pozícióban mindig megjelenik egy tojás, amelyet a kígyóval meg kell etetni. Minden etetéssel eggyel nagyobb lesz a kígyó. A játék célja, hogy a kígyó minél tovább elkerülje az ütközést az akadályokkal, a pálya szélével, illetve saját magával. A pályák méretét, illetve felépítését (falak helyzete) tároljuk fájlban. A program legalább 3 különböző méretű pályát tartalmazzon. A program biztosítson lehetőséget új játék kezdésére a pálya kiválasztásával, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozog a kígyó). Továbbá ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, hány tojást sikerült elfogyasztania a játékosnak.

Elemzés:

- A játékot három különböző méretű pályán játszhatjuk: kicsi (15x15), közepes (20x20) és nagy (25x25). A program indításkor a kicsi pályával inicializál, amit a játékos a space megnyomásával kezdhet el.
- A feladatot egyablakos asztali alkalmazásként valósítjuk meg.
- Az ablakban elhelyezünk egy menüt a következő opciókkal: egy Játék menüpont, amiben a játékos új játékot indíthat, vagy kiléphet az alkalmazásból. Valamint egy Beállítások menüpont, amelyben a játékos beállíthatja a következő játék pályaméretét.
- Az ablak alján megjelítünk egy státuszsort, ami egyrészt kiírja a Score-t, ami a megvett tojások számát jelenti, valamint kiírja, hogy space-el lehet a játékot elindítani, valamint megállítani.
- A játéktábla n x n-es dobozkákból áll, melyek, ha üresek, akkor világosszürke színűek, ha a kígyó az, akkor zöld színű, ha pedig egy dobozka a tojás, akkor piros. A játékos a WASD gombokkal irányíthatja, hogy a kígyó melyik irányba mozogjon és célja minél több tojás megevése.
- A játék automatikusan feldob egy dialógusablakot, ha a játéknak vége van (vesztés/nyerés), amin kiírja az elért pontszámat, valamint új játékot indíthatunk vele, vagy bezárhatjuk a programot.
- A felhasználói esetek az 1. Ábrán láthatóak.



1. Ábra

Tervezés:

Programszerkezet:

- A programot háromrétegű architektúrában valósítjuk meg. A megjelenítés a View, a modell a Model, a perzisztencia pedig a Persistence névtérben helyezkedik el. A program csomagszerkezete a 2. Ábrán látható.
- A program szerkezetét két projektre osztjuk: Snake (Model + Persistence), SnakeGame_WPF (View/ViewModel).

Perzisztencia:

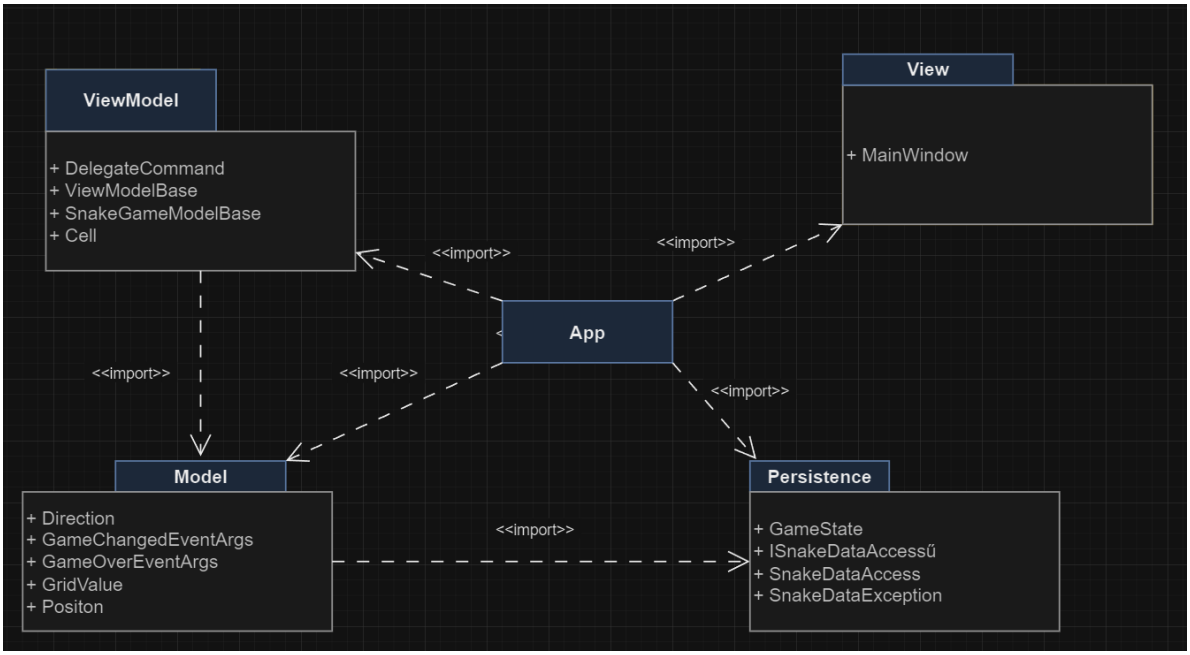
- Az adatkezelés feladata a Snake pályáival kapcsolatos információk tárolása, betöltése.
- A GridValue osztály lehetővé teszi, hogy n x n-es matrix típusaként megadva reprezentáljuk a pályát és annak összes lehetséges tagját (Empty, Snake, Egg, Outside)
- A hosszútávú adattárolás lehetőségeit az ISnakeDataAccess interfész adja meg, amely lehetőséget ad a pályák betöltésére (LoadAsync).
- Az interfészt szöveges fájl alapú adatkezelésre a SnakeDataAccess osztály valósítja meg. A fájlkezelés során fellépő hibákat a SnakeDataException kivétel jelzi.
- A program az adatokat szöveges fájlként tárolja, melyek a .txt kiterjesztést kapják. Ezeket az adatokat a program új játék kezdésekor tölti be.
- A fájl n + 1 sorból áll, ahol az első sor maga az n, utána pedig n sornyi n elemből álló sor van, melyben a pálya mezőinek az elfoglalóját reprezentáló számok vannak (0, 1, 2) egy-egy space-el elválasztva egymástól.

Modell:

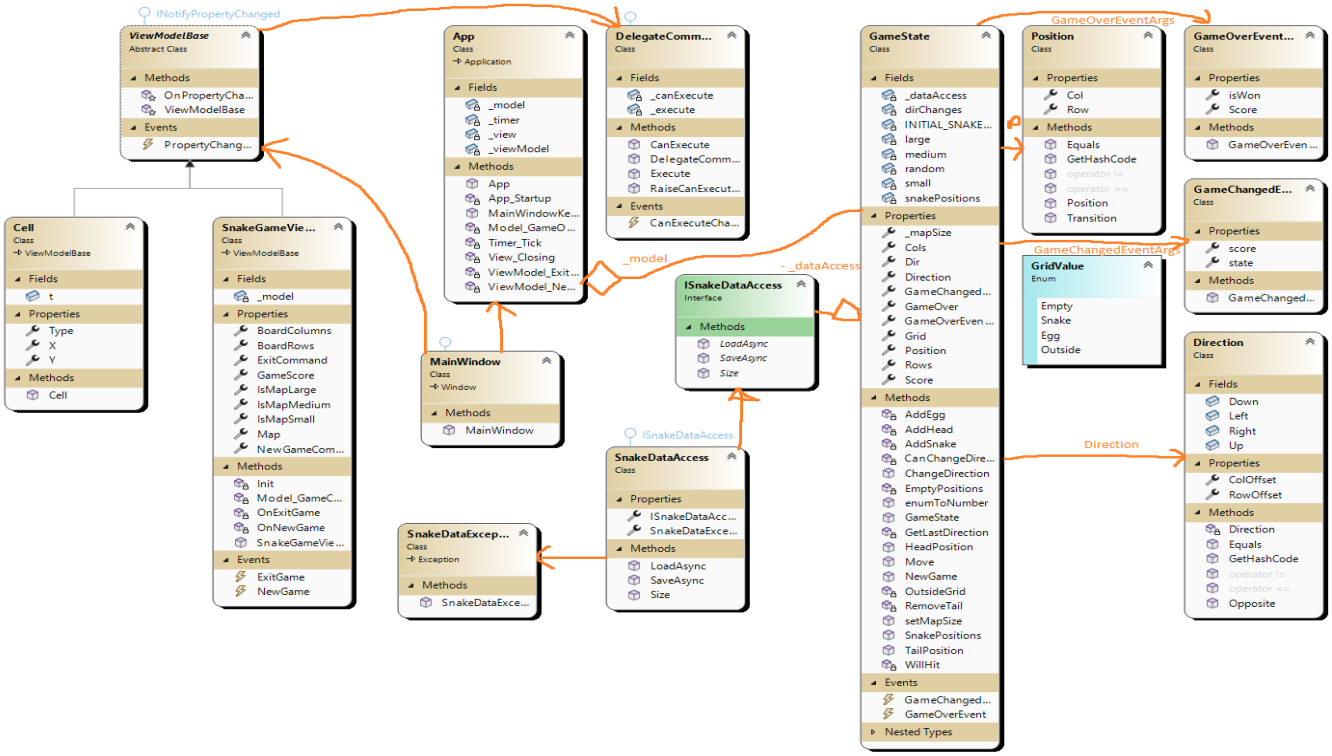
- A modell lényegi részét a GameState osztály valósítja meg, amely szabályozza a pálya tevékenységeit, valamint a játék egyéb paramétereit, mint például a pálya méretét (_mapSize) és az összegyűjtött pontokat (Score). A típus lehetőséget ad új játék kezdésére (NewGame), valamint a kígyó mozgatására (Move). Az új játék új pályát general a kiválasztott pályamérethez.
- A mezők állapotváltozásáról (kígyó mozgás) a GameChangedEvent esemény, míg a játék végéről a GameOverEvent esemény jelez.
- A modell példányosításkor megkapja az adatkezelés felületét, amelynek a segítségével lehetőséget ad betöltésre (LoadAsync). A szövegesfájlból egy mátrixot kiolvasva átadjuk a Model GridValue mátrixának azt.
- A WPF Appban jelenik meg az időzítő (_timer), valamint a különöző pályák elérhetőségi útvonalai.

Nézet:

- A nézetet a SnakeGameViewModel osztály biztosítja, amely tárolja a model egy példányát (_model).
- A játékpályát egy dinamikusan létrehozott buttonGrid reprezentálja. A felületen létrehozzuk a megfelelő menüpontokat, illetve a státuszsort, és hozzájuk tartozó eseménykezelőket. A játékpálya generálását az Init valamint az Model_GameChanged metódusok végzik.
- A játékidő múlását a már említett _timeridőzítő méri, amely 120 miliszekundumonként magától aktiválódik, hogy frissüljön a játék.
- A program teljes statikus szerkezete a 3. Ábrán látható.



2. Ábra



3. Ábra

Tesztelés:

- A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a SnakeModelTest osztályban.
- Az alábbi tesztesetek kerültek megvalósításra:
- GameStateInitializationTest : a játék inicializálásának tesztelése
- AddSnakePositionsSnakeCorrectly : annak tesztelése, hogy a kígyó jó helyre kerül-e az AddSnake metódus által
- AddEggPlacesEggOnGrid : megnézi hogy az AddEgg metódus tényleg lerak-e egy tojást a pályára
- MoveUpdatesSnakePosition : megnézi, hogy a kígyó feje tényleg mozog a Move metódus meghívása után
- ChangeDirectionUpdatesDirection : megnézi, hogy tényleg megváltozik-e az irány, ha meghívjuk a ChangeDirection metódust
- ChangeDirectionUpdatesDirectionOpposite : megnézi, hogy nem változik meg az irány, ha a ChangeDirection metódust a jelenlegivel ellentétes iránnyal hívjuk meg
- GameOverMoveMethodTest : megnézi, hogy működik-e a GameOver metódus
- NewGameResetsGameState : megnézi, hogy a NewGame metódus alaphelyzetbe állítja-e az adattagokat
- SnakePositionsReturnsAllPositions : megnézi, hogy a SnakePositions metódus vissza adja-e a kígyót