1. Beadandó feladat dokumentáció

Készítette:
Czigány Kristóf Péter
LGLFAT

Feladat (5):

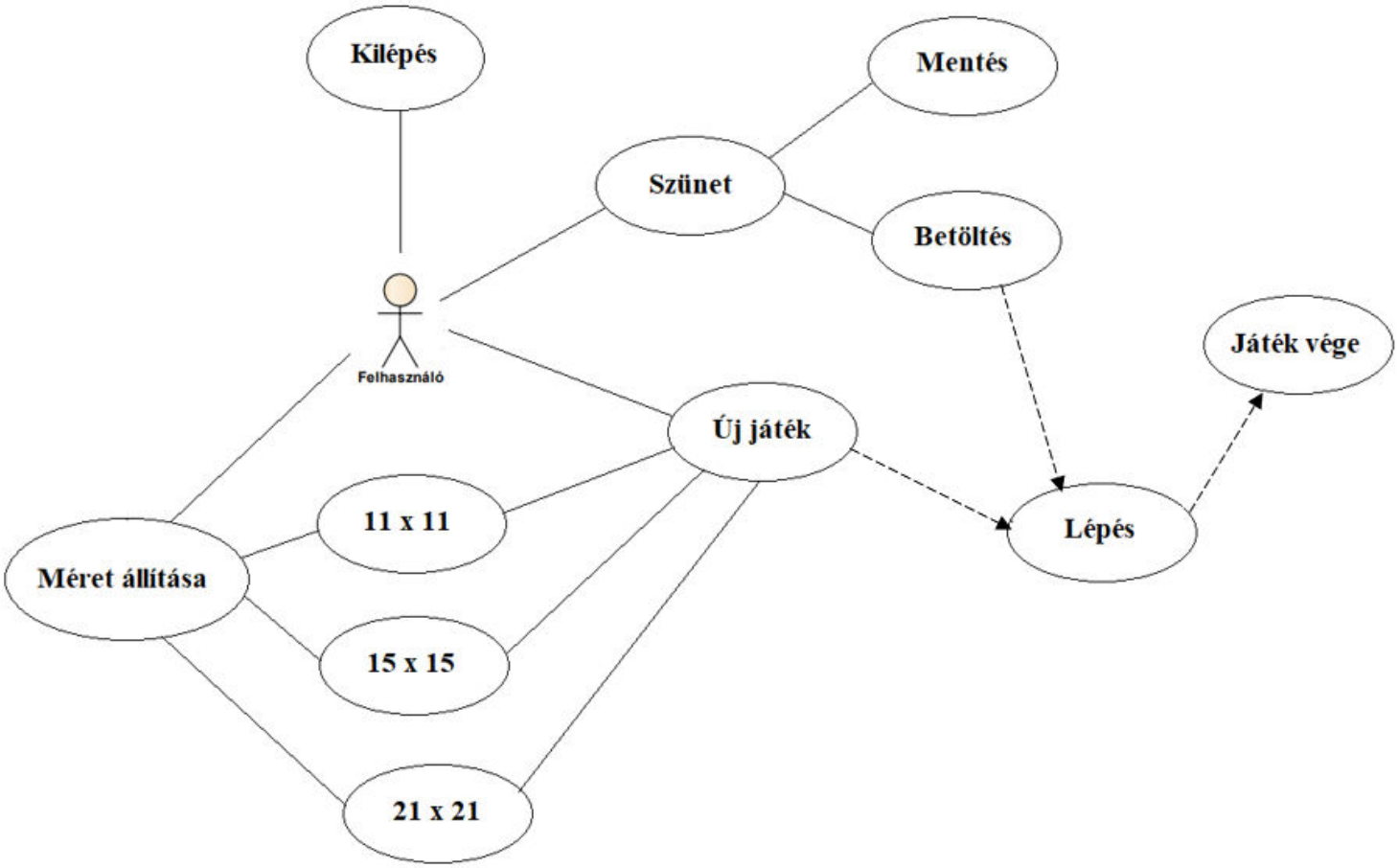
Menekülj

Készítsünk programot, amellyel a következő játékot játszhatjuk.
Adott egy $n \times n$ elemből álló játékpálya, ahol a játékos két üldöző elől próbál menekülni, illetve próbálja őket aknára csalni.
Kezdetben a játékos játékpálya felső sorának közepén helyezkedik el, a két üldöző pedig az alsó két sarokban. Az ellenfelek adott időközönként lépnek egy mezőt a játékos felé haladva úgy, hogy ha a függőleges távolság a nagyobb, akkor függőlegesen, ellenkező esetben vízszintesen mozognak a játékos felé.
A pályán véletlenszerű pozíciókban aknák is elhelyezkednek, amelyekbe az ellenfelek könnyen beleléphetnek, ekkor eltűnnek (az akna megmarad).
A játékos vízszintesen, illetve függőlegesen mozoghat (egyesével) a pályán, és célja, hogy az ellenfeleket aknára csalja, miközben ő nem lép aknára. Ha sikerül minden üldözőt aknára csalnia, akkor győzött, ha valamely ellenfél elkapja (egy pozíciót foglal el vele), vagy aknára lép, akkor veszített.
A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával (11×11, 15×15, 21×21), valamint játék szüneteltetésére (ekkor nem telik az idő, és nem léphet senki). Ismerje fel, ha vége a játéknak, és jelenítse meg, hogy győzött, vagy veszített-e a játékos. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére. A program játék közben folyamatosan jelezze ki a játékidőt.

Elemzés:

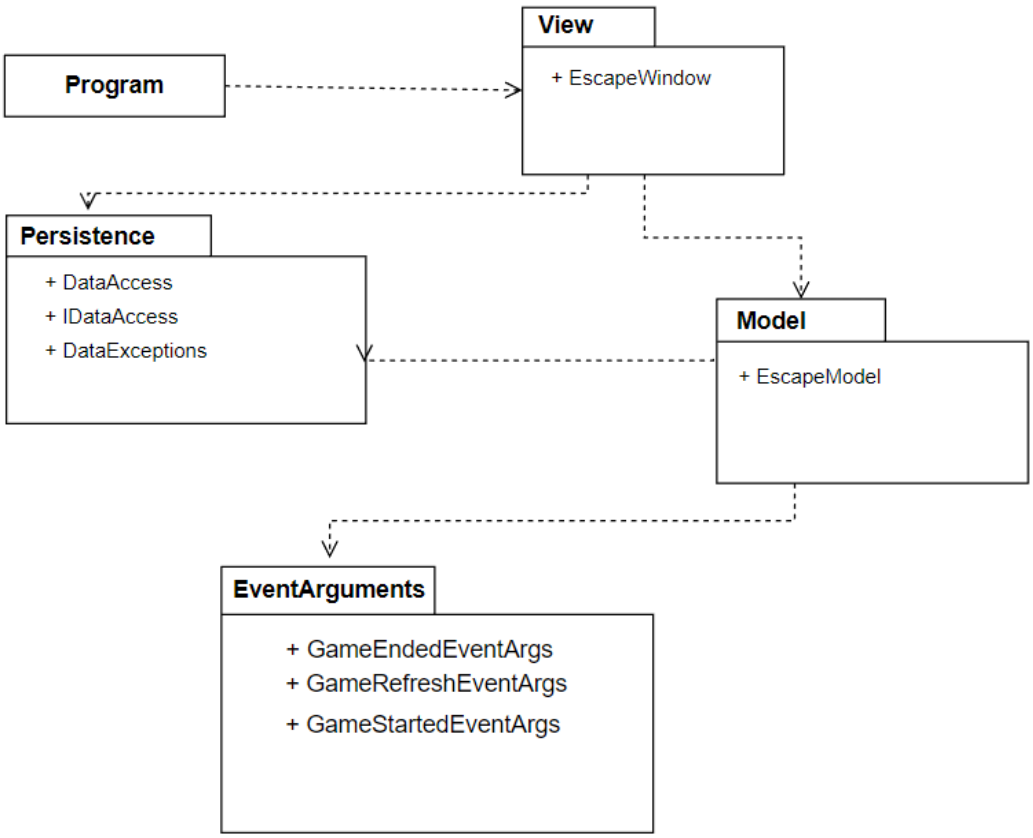
- A játékot 3 különböző méretű pályán lehet játszani, ami alapértelmezetten 15 x 15 méretű.
- A feladatot egyablakos asztali Windows Forms grafikus felülettel valósítjuk meg.
- Az ablakban elhelyezünk egy menüt a következő menüpontokkal: File (Save Game, Load Game), New Game (11 x 11, 15 x 15, 21 x 21), Start Game, Continue, Pause. Az ablak alján megjelenítünk egy státuszsort, amely az eltelt időt mutatja.
- A játékban egy játékos szerepel, amelyet a zöld szín valósít meg (1 bábú), s kezdetben az felső sor közepén áll. A játékossal a nyilakkal lehet lépni egyesével (fel, le, jobbra, balra), 2 másodpercenként.
- Az ellenfél (számítógép), amelyet a piros szín valósít meg (2 bábú). Az ellenfél 3-3 másodpercenként lépked mindkét bábújával, 1 másodperc különbséggel. A lépés iránya a játékos felé attól függően történik, hogy vízszintes irányban vagy függőleges irányban vannak távolabb a játékostól.
- A játéktéren véletlenszerűen megjelenítünk bombákat, melyeket a fekete szín jelez. Amennyiben a játékos erre a mezőre lép, veszít. Abban az esetben, ha mindkét ellenfél bombára lép, a játékos nyer.

- A játék automatikusan feldob egy dialógusablakot, amikor vége a játéknak és tájékoztatja a játékost, hogy nyert vagy veszített, valamint a játékidőről is.
- Felhasználói esetek:



Tervezés:

- Programszerkezet:
  - A programot háromrétegű architektúrában valósítjuk meg. A megjelenítés a View, a modell a Model, míg a perzisztencia Persistence névtérben helyezkedik el. A program csomagszerkezete a 2. ábrán látható.



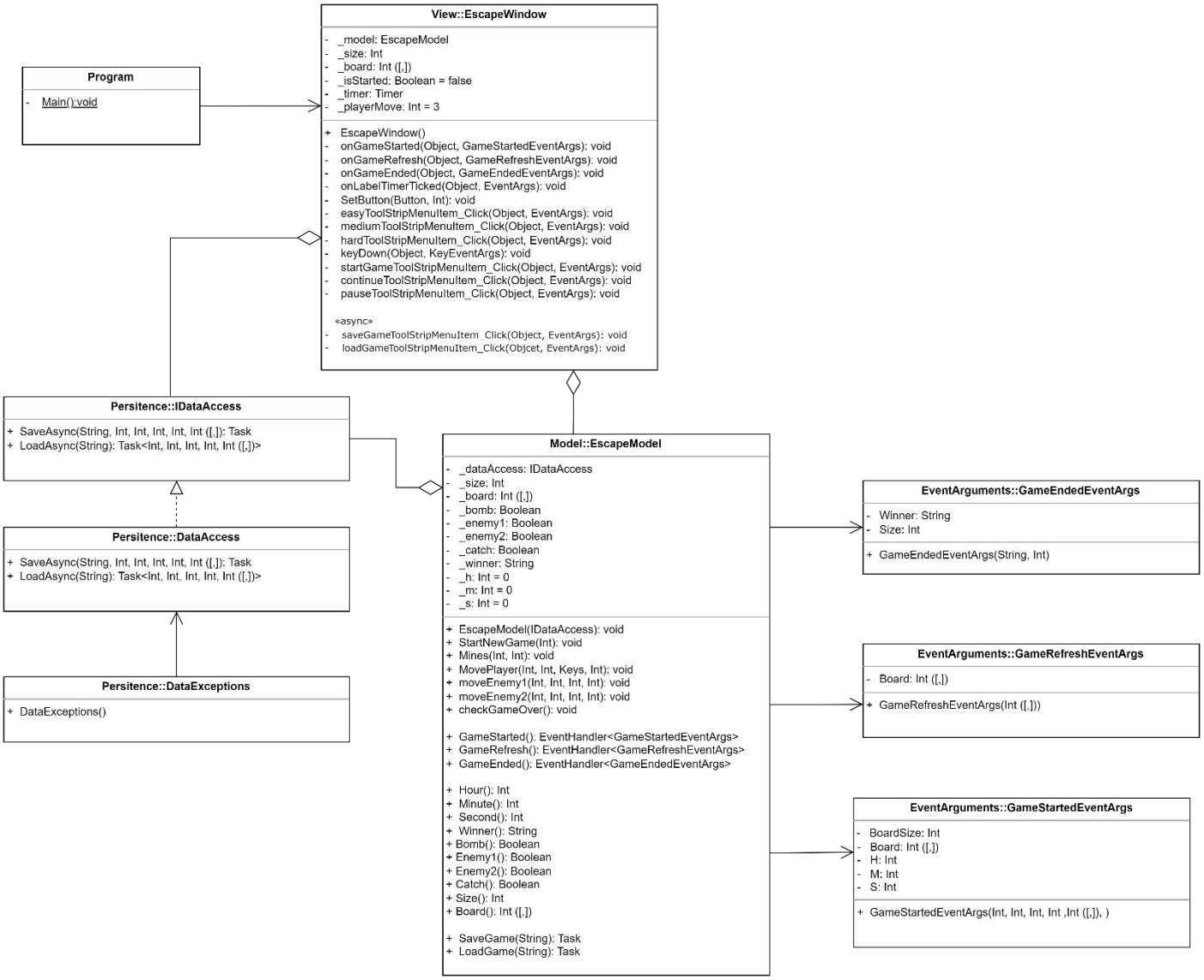
  - A program szerkezetét két projektre osztjuk implementációs megfontolásból: a Persistence és Model csomagok a program felületfüggetlen projektjében, míg a View csomag a Windows Formstól függő projektjében kap helyet.

- Perzisztencia:
  - Az adatkezelés feladata az Escape táblával kapcsolatos információk tárolása, valamint a betöltés és a mentés biztosítása.
  - A hosszú távú adattárolás érdekében létrehoztuk az IDataAccess interfészt amely lehetőséget ad az adott játék elmentésére(SaveGame) és előzetes mentések betöltésére(LoadGame).
  - Az interfész szöveges alapú adatkezelésére a DataAccess osztály szolgál.
  - A fájlkezelés során fellépő hibákat a DataExceptions kivétel jelzi.

- A program az adatokat szöveges fájlként tudja eltárolni, melyek az txt 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, a második sor a játékban eltelt órát, a harmadik sor a játékban eltelt percet, a negyedik sor a játékban eltelt időt, a fájl többi része pedig a pálya tartalmát, mely 0 és 4 közötti számok lehetnek, ahol 0 a szabad mezőt, 1 a játékos mezőjét, 2 az ellenfél 1. bábúját, 3 az ellenfél 2. bábúját, 4 pedig a bombákat reprezentálja.

- Modell:
  - A modell lényegi részét az *EscapeModel* osztály valósítja meg, amely szabályozza a játéktábla tevékenységeit, valamint a játék egyéb paramétereit. Ez az osztály ad lehetőséget új játék kezdésére (*StartNewGame*). A játék végét a *checkGameOver* ellenőrzi. A játékos mozgatásáért a *movePlayer*, az ellenfelek mozgásáért pedig a *moveEnemy1* és a *moveEnemy2* felel. A bombák elhelyezéséért a *Mines* felel.
  - A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (*LoadGameAsync*) és mentésre (*SaveGameAsync*).
  - A játék állapotát a *ViewRefresh* frissíti.
- Nézet:
  - A néztet az EscapeWindow osztály biztosítja, amely tárolja a modell egy példányát (*_model*), valamint az adatelérés konkrét példányát (*_dataAccess*).
  - A játéktáblát egy dinamikusan létrehozott gombmező reprezentálja. A felületen létrehozzuk a megfelelő menüpontokat, illetve státuszsort, valamint dialógusablakokat, és a hozzájuk tartozó eseménykezelőket.
  - A játék időbeli kezelését egy időzítő végzi (*_timer*), amelyet mindig aktiválunk játék során, illetve inaktiválunk, amennyiben bizonyos menüfunkciók futnak.

A program teljes szerkezete:



Tesztelés:

A modell funkcionalitása egységtesztek segítségével lett elllenőrizve az EscapeTest osztályban.

Az alábbi tesztesetek kerültek megvalósításra:

- StartNewGameBoardSize
- StartNewGameBoardFields
- CheckGameOverIfPlayerWin
- CheckGameOverIfCatched
- CheckGameOverIfExplosed
- PlayerMoveDown
- PlayerMoveUp
- MovePLayerLeft
- MovePLayerRight
- Enemy1Move
- Enemy2Move