

### 3.Beadandó feladat dokumentáció

#### Készítette:

Li Dominik

EF9XHW

E-mail: [lidominik02@gmail.com](mailto:lidominik02@gmail.com)

#### Feladat:

Készítsünk programot, amellyel a következő két személyes játékot játszhatjuk. Adott egy  $n \times n$  mezőből álló tábla, amelyen a mezők két szint vehetnek fel spirális alakban (tradicionálisan pirosat, illetve zöldet), továbbá a középső mező szürke. Minden mezőn, kivéve a középsőn egy kaméleon helyezkedik el, amelynek színe megegyezik a mezővel, így minden játékos  $(n^2 - 1) / 2$  kaméleonnal rendelkezik. A játékosok felváltva léphetnek. Egy kaméleonnal léphetünk egy szomszédos üres mezőre (vízszintesen, illetve függőlegesen), illetve átugorhatjuk az ellenfél kaméleonját (vízszintesen, illetve függőlegesen), amennyiben a rákövetkező mező üres. Az átugrott kaméleon lekerül a tábláról. A játék célja, hogy a másik játékos elveszítse az összes kaméleonját. A játékban a csavar, hogy a kaméleonok alkalmazkodnak a környezetükhöz. Amennyiben egy kaméleon egy másik színű mezőre ugrott, vagy lépett, akkor további 1 kör elteltével átszíneződik a másik színre (tehát a másik játékosé lesz). Ez alól kivétel a középső mező.

A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával ( $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ), valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött.

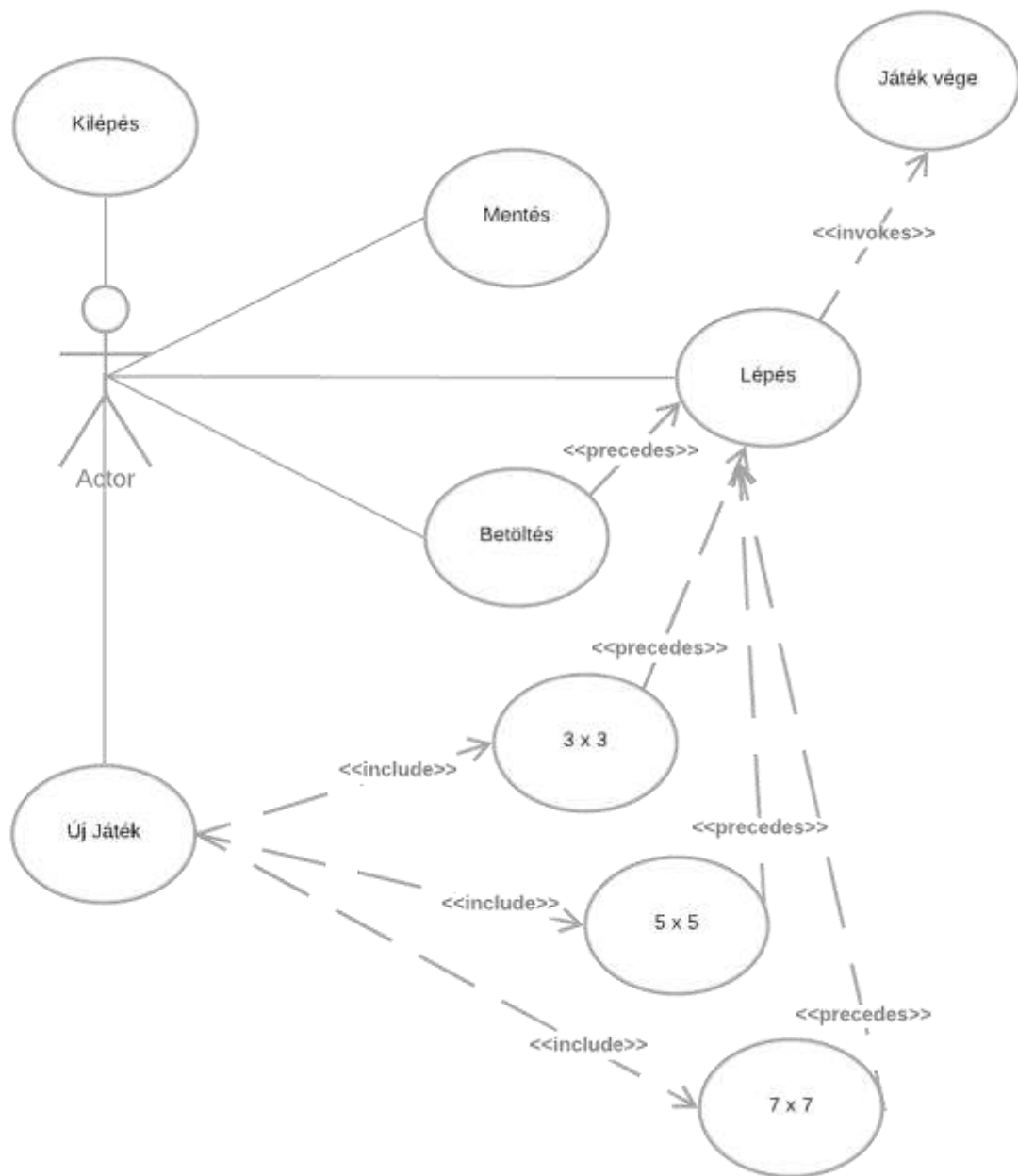
#### Elemzés:

- A játékot 3 különböző pályán játszhatjuk:  $3 \times 3$ ,  $5 \times 5$ , illetve  $7 \times 7$  pályán. A program indításkor új játékot indít egy  $5 \times 5$ -ös pályán.
- A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.
- Az ablakban elhelyezünk egy menüt a következő menüponttal: Opciók, ezen belül: Játék mentése, Játék betöltése, Új játék indítása. Az ablak alján megjelenítünk egy státuszsort, amely megmutatja, hogy melyik játékos léphet.
- A játék táblát egy  $3 \times 3$ -s,  $5 \times 5$ -s, vagy pedig egy  $7 \times 7$ -es nyomógombokból álló rács reprezentálja, attól függően, hogy milyen pályát választottunk. Ahhoz, hogy egy kaméleonnal lépni tudjunk, először a mozgatni kívánt kaméleon nyomógombjára kell kattintanunk, ezt követően annak az üres mezőnek a nyomógombjára, amelyikre lépni

### 3.Beadandó feladat dokumentáció

szeretnénk. A program, az érvénytelen lépések kor egy dialógus ablakot dob fel, ezzel jelezve a hibás lépést.

- A játék automatikusan feldob egy dialógus ablakot akkor is, ha a játék véget ért. Szintén dialógusablakkal végezzük el a mentést, illetve a betöltést, a fájlneveket a felhasználó adja meg.
- A felhasználói esetek az 1.ábrán láthatóak.



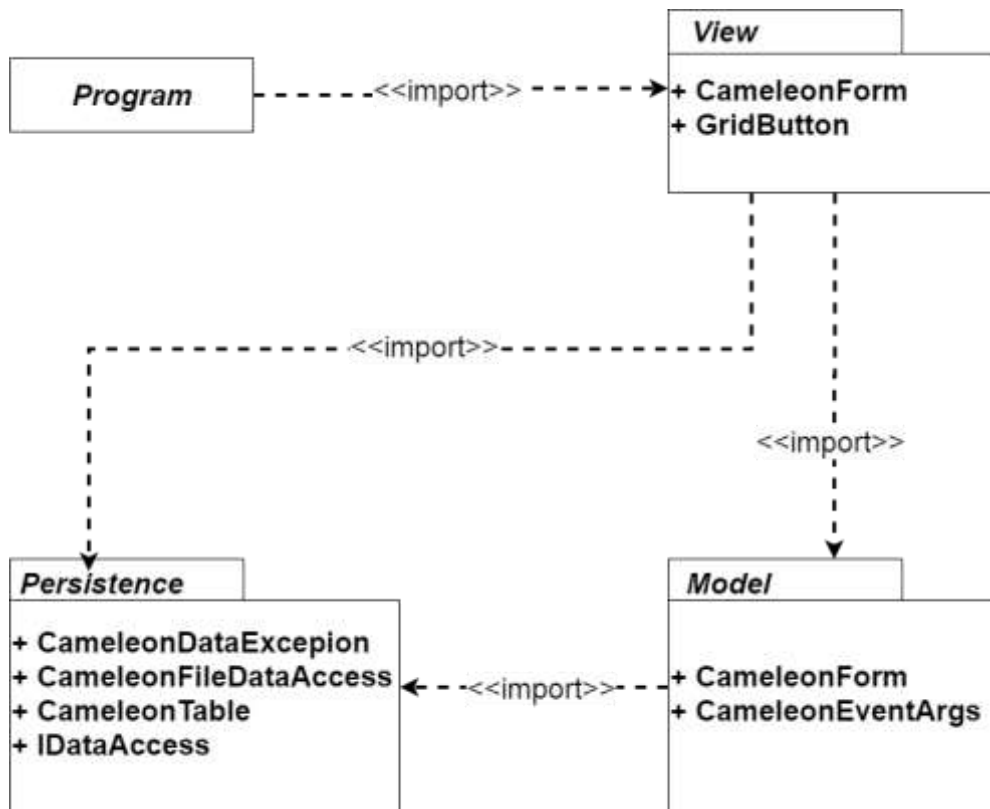
1. ábra: Felhasználói esetek diagramja

### 3.Beadandó feladat dokumentáció

#### 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 játék táblával kapcsolatos információk tárolása, valamint a betöltés / mentés biztosítása.
  - A CameleonTable osztály egy érvényes táblát biztosít ehhez a játékhoz (azaz mindig ellenőrzi a beállított értékeket), ahol minden mezőre ismert színe (\_colorTable), illetve az, hogy melyik mezőn épp milyen kaméleon van (\_cameleonTable). A tábla alapértelmezés szerint 3 x 3 -as de ez a konstruktorban paraméterezhető. A tábla lehetőséget ad az állapotok lekérdezésére (GetColor, GetValue,Size,CurrentCameleon,WinnerCameleon), valamint szabályos lépésre (Step), illetve direkt beállítás (SetValue) elvégzésére.
  - A hosszútávú adattárolás lehetőségeit a CameleonFileDataAccess osztály valósítja meg. A fájlkezelés során fellépő hibákat a CameleonDataException kivétel jelzi.
  - A program az adatokat szöveges fájlként tudja eltárolni. 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 soron következő játékost tartalmazza. A fájl többi része izomorf leképezése a játéktáblának, azaz a fájl első sorában leírt számnyi sor következik, és minden sor ugyanennyi számot tartalmaz, szóközzel elválasztva. A számok 0,1, illetve 2, lehet ahol 0 az üresmezőt-, az 1 a piros kaméleont-, a 2 pedig a zöld kaméleon reprezentálja.

### 3.Beadandó feladat dokumentáció

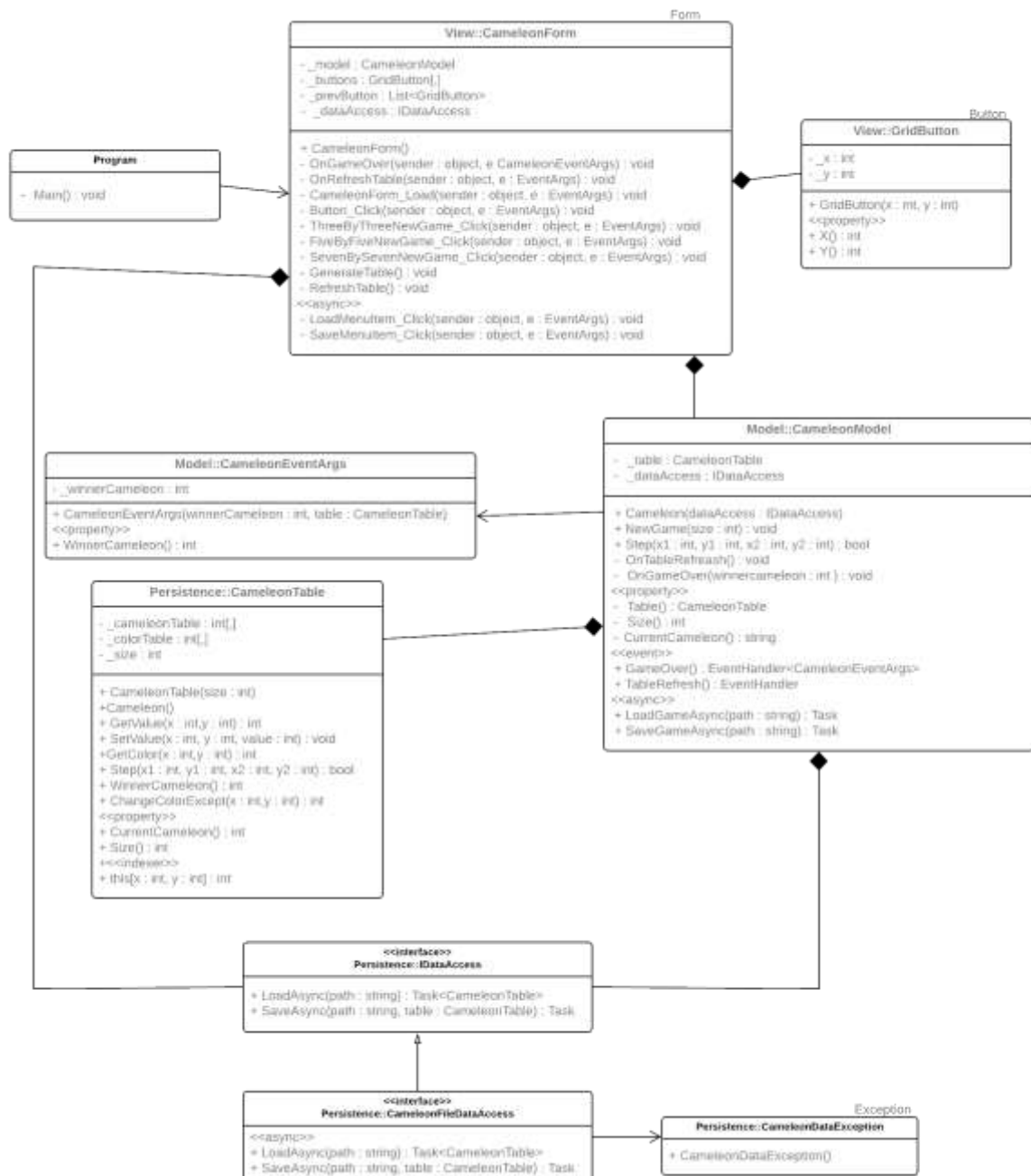


2. ábra: Az alkalmazás csomagdiagramja

- **Modell:**
  - A modell lényegi részét a **CameleonModel** osztály valósítja meg, amely szabályozza a tábla tevékenységeit. A típus lehetőséget ad az új játék kezdésére (**NewGame**), valamint lépésre (**Step**). Új játéknál megadható a játéktábla mérete.
  - A játék állapot változásáról a **TableRefresh** esemény, míg a játék végéről a **GameOver** esemény tájékoztat. A játék vége esemény argumentuma (**CameleonEventArgs**) tárolja a győztes kaméleont.
  - A modell példányosításakor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (**LoadGameAsync**) és mentésre (**SaveGameAsync**),
- **Nézet:**
  - A nézetet a **CameleonForm** osztály biztosítja, amely tárolja a modell egy példányát (**\_model**), valamint az adatkezelés konkrét példányát (**\_dataAccess**).

### 3. Beadandó feladat dokumentáció

- A játéktáblát egy dinamikusan létrehozott gombmező (\_buttonGrid) reprezentálja. A nyomógombok, a Button osztályból származtatott, GridButton típusúak, melyek tartalmaznak egy x és egy y adatot is. A felületen létrehozuk a megfelelő menüpontokat, illetve státuszsort, valamint dialógusablakokat, és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását (GenerateTable), illetve a játék állapotának frissítését (RefreshTable) külön metódusok végzik.
- A program statikus szerkezete a 3. ábrán látható.



3. ábra: Az alkalmazás osztály diagramja

### 3.Beadandó feladat dokumentáció

#### Tesztelés:

- A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a CameleonTest osztályban.
- Az alábbi tesztesetek kerültek megvalósításra:
  - **CameleonNewGameTest:** Az új játék indítása, a mezők méretének ellenőrzése a játékpálya méretének függvényében.
  - **CameleoLoadTest:** A játék modell betöltésének tesztelése mockolt perzesztencia réteggel.
  - **Model\_GameOver:** A játék vége esemény kiváltódásának, valamint a nyertes játékos helyességének ellenőrzése.
  - **CameleonStepTest:** A játék lépés hatásainak ellenőrzése, helyes lépés, illetve érvénytelen lépés esetén. Több lépés végrehajtása azonos játékmezőn, esemény kiváltásának ellenőrzése.