**1. Beadandó feladat**

**Készítette:**

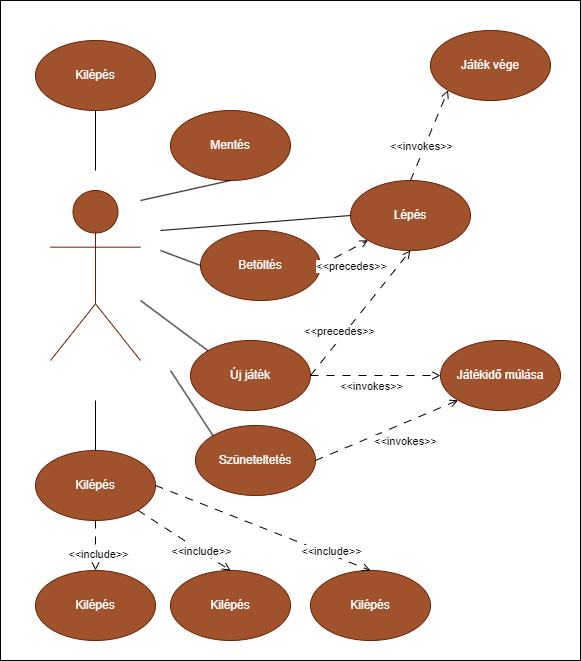
Kovács Zalán (D1ZQ36)

E-mail: d1zq36@inf.elte.hu

**Feladat:**

Készítsünk programot, amellyel a következő játékot játszhatjuk. Adott egy 𝑛 × 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 nehézségi szinttel lehet játszani: könnyű (11x11 tábla, 11 akna), közepes (15x15 tábla, 15 akna), illetve nehéz (21x21 tábla, 21 akna). Indításkor a program közepes nehézséget állít és így indít játékot.
* A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.
* Az ablakban elhelyezünk egy menüt a következő menüpontokkal: File (Új játék, Játék betöltése, Játék mentése, Kilépés), Settings (Könnyű játék, Közepes játék, Nehéz játék), illetve Pause. Az ablak alján megjelenítünk egy státuszsort, amely a lépések számát, illetve a hátralévő időt jelzi.
* A táblát egy a nehézségnek megfelelő N x N nyomógombból álló rács reprezentálja. A táblán a sárgászöld színű gomb a játékosunk, ezt irányíthatjuk a wasd billentyűk lenyomásával, és ennek hatására a két üldöző (piros és türkizkék) lép abba az irányba, amerre távolabb van a játékostól, lépni csak vízszintesen vagy függőlegesen lehet.
* A játéknak akkor van vége, ha a játékos nyer vagy veszít, erre egy felugró dialógusablak hívja fel a figyelmet. Ha a játékos aknára lép vagy elkapják az üldözők, akkor veszít, ha pedig mindkét üldözőt aknára csalta, akkor nyer. Szintén dialógusablakkal van megvalósítva a mentés, betöltés is, a fájlneveket a felhasználó adja meg, a kiterjesztés (.escp) lesz.
* A felhasználói esetek az 1. diagramon láthatók

**1. ábra: Felhasználói esetek diagramja**

**Tervezés:**

* Programszerkezet:
  + A programot háromrétegű architektúrában valósíjuk meg. A megjelenítés a **View**, a modell a **Model**, míg a perzisztencia a **Persistence** névtérben helyezkedik el.
  + 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/mentés biztosítása.
  + **EscapeTable** osztály egy érvényes Escape táblát biztosít (azaz mindig ellenőrzi a beállított értékek), ahol minden mezőre ismert az értéke (**\_fieldValues**). A tábla alapértelmezés szerint **medium** nehézségen indul, azaz 15x15-ös táblát hoz létre házakkal, de ez a konstruktorban paraméterezhető. A tábla lehetőséget az állapotok lekérdezésére (**IsFilled**, **IsEmpty**, **GetValue**), valamint szabályos lépésre (**Step**), illetve direkt beállítás (**SetValue**) elvégzésére.
  + A hosszú távú adattárolás lehetőségeit az **IEscapeAccess** interfész adja meg, amely lehetőséget ad a tábla betöltésére (**LoadAsync**), valamint mentésére (**SaveAsync**). A műveleteket hatékonysági okokból aszinkron módon valósítjuk meg.
  + Az interfészt szöveges fájl alapú adatkezelésre a **EscapeFileDataAccess** osztály valósítja meg.
  + A program az adatokat szöveges fájlként tudja eltárolni, melyek az **escp** 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 fájl többi része izomorf leképezése a játéktáblának, azaz összesen 11/15/21 sor következik, és minden sor 11/15/21 számot tartalmaz szóközökkel választva, a nehézségnek megfelelően. A számok 0, 2, 3, 4 vagy 5 értékűek lehetnek, ahol 0 reprezentálja a még üres mezőt, 2 az aknák helye, 3, a játékos, 4 és 5 pedig a 2 két üldöző, feltéve, hogy élnek még.
* Modell:
  + A modell lényegi részét a **EscapeGameModel** osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék egyéb paramétereit, úgymint az idő (**\_gameTime**), hogy vége van-e a játéknak (**\_isGameOver**), meg van-e állítva a játék(**\_isPaused**) vagy az életben lévő üldözőket (**Enemies**). A típus lehetőséget ad új játék kezdésére (**NewGame**), valamint lépésre (**Step**). Új játéknál automatikusan generálódnak kezdő mezők. Az idő előreléptetését (**AdvanceTime**) tehetjük meg.
  + A mezők állapotváltozásáról a **FieldChanged** esemény tájékoztat. Az esemény argumentuma (**EscapeFieldEventArgs**) tárolja a megváltozott mező pozícióját.
  + A játékállapot megváltozásáról (idő) a **GameAdvanced** esemény, míg a játék végéről a **GameOver** esemény tájékoztat. Az események argumentuma (**EscapeEventArgs**) tárolja a győzelem állapotát, valamint a játékidőt.
  + A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (**LoadGameAsync**) és mentésre (**SaveGameAsync**)
  + A játék nehézségét a **Difficulty** felsorolási típuson át kezeljük, és a **EscapeGame** osztályban konstansok segítségével tároljuk az egyes nehézségek paramétereit.
* Nézet:
  + A nézetet a **GameForm** 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 létrehozzuk a megfelelő menüpontokat, illetve státuszsort, valamint dialógusablakokat, és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását, illetve az értékek beállítását (**GenerateTable**) metódus végzi.
  + A játék időbeli kezelését egy időzítő végzi (**\_timer**), amelyet mindig aktiválunk játék során, illetve inaktiválunk, amennyiben bizonyos menüfunkciók futnak.
* A képen szöveg, képernyőkép, tervezés látható

  Automatikusan generált leírásA program teljes statikus szerkezete a 2. ábrán látható.

**2. ábra: Felhasználói esetek diagramja**