**1. Beadandó feladat dokumentáció**

**Készítette:**

Ramel Dániel

ZC8YR8

**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 pályán játszhatjuk: 11x11, 15x15, 21 x 21.
* A program indításkor 11x11 pályát állít be.
* A feladatot .NET MAUI alkalmazásként, elsődlegesen Windows és Android
* platformon valósítjuk meg. Az alkalmazás négy lapból fog állni. Az alkalmazás
* portré tájolást támogat.

• A játék négy képernyőn fog megjelenni.

• Az első képernyő (Játék) tartalmazza a játéktáblát, a játék állását (fennmaradó idő) a lap alján, az új játék, valamint a beállítások gombjait a lap tetején.

• A második képernyőn van lehetőség betöltésre, illetve mentésre.

• A további két képernyő a betöltésnél, illetve mentésnél megjelenő lista, ahol a játékok elnevezése mellett a mentés dátuma is látható.

Mentés esetén ezen felül lehetőség van új név megadására is.

* A játék automatikusan feldob egy dialógusablakot, amikor vége a játéknak . Szintén dialógusablakokkal végezzük el a mentést, illetve betöltést, a fájlneveket a felhasználó adja meg.
* A felhasználói esetek az 1. ábrán láthatóak

A diagram of a diagram

Description automatically generated

**1. ábra: Felhasználói esetek diagramja**

**Tervezés:**

**Programszerkezet:**

• A szoftvert két projektből építjük fel: a modellt és a perzisztenciát

tartalmazó osztálykönyvtárból (.NET Standard Class Library), valamint a

.NET MAUI többplatformos projektből, amelyet Windows és Android

operációs rendszerre is le tudunk fordítani.

A programot MVVM architektúrában valósítjuk meg, ennek megfelelően

View, Model, ViewModel és Persistence névtereket valósítunk meg az

alkalmazáson belül.

• A megvalósításból külön építjük fel a játék, illetve a betöltés és mentés

funkciót, valamennyi rétegben. Utóbbi funkcionalitást újrahasznosítjuk

egy korábbi projektből, így nem igényel újabb megvalósítást.

• A program vezérlését az alkalmazás osztály (App) végzi, amely

példányosítja a modellt, a nézetmodell és a nézetet, biztosítja a

kommunikációt, valamint felügyeli az adatkezelést.

• A program csomagdiagramja a 2. ábrán látható.

A diagram of application software

Description automatically generated

**2. ábra: Az alkalmazás csomagdiagramja**

Perzisztencia:

* Az adatkezelés feladata az Escape táblával kapcsolatos információk tárolása, valamint a betöltés/mentés biztosítása.
* A **EscapeTable** osztály egy érvényes Escape táblát biztosít.
* A hosszú távú adattárolás lehetőségeit az **IEascapeDataAccess** 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 az **EscapeFileDataAccess** osztály valósítja meg. A fájlkezelés során fellépő hibákat a

**EscapeDataException** kivétel jelzi.

* A program az adatokat szöveges fájlként tudja eltárolni, melyek a **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 fájl többi része izomorf leképezése a játéktáblának.

Modell:

* A modell lényegi részét a **EscapeGameModel** osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék egyéb paramétereit, úgymint az idő (**\_gameTime**). A típus lehetőséget ad új játék kezdésére (NewGame), valamint lépésre (**movePlayer**). Új játéknál automatikusan generálódnak kezdő mezők. Az idő előreléptetését időbeli lépések végzéséve (**AdvanceTime**) tehetjük meg.
* Az üldözők állapotváltozásáról, a játékállapot megváltozásáról (lépések száma, hátra lévő idő) a **GameAdvanced** esemény tájékoztat. Az esemény argumentuma (**EscapeEventArgs**) tárolja a megváltozott mező pozícióját.
* A játék végéről a **GameOver** esemény tájékoztat. Az események argumentuma (**EscapeEventArgs**) tárolja a győzelem állapotát és a játékidőt.
* A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (**LoadGameAsync**) és mentésre

(**SaveGameAsync**)

Nézetmodell:

* A nézetmodell megvalósításához felhasználunk egy általános utasítás

(**DelegateCommand**), valamint egy ős változásjelző (**ViewModelBase**) osztályt.

* A nézetmodell feladatait a **EscapeViewModel** osztály látja el, amely parancsokat biztosít az új játék kezdéséhez, játék betöltéséhez, mentéséhez, valamint a kilépéshez. A parancsokhoz eseményeket kötünk, amelyek a parancs lefutását jelzik a vezérlőnek. A nézetmodell tárolja a modell egy hivatkozását (**\_model**), de csupán információkat kér le tőle, illetve a játéknehézséget szabályozza. Direkt nem avatkozik a játék futtatásába.
* A játékmező számára egy külön mezőt biztosítunk (**EscapeField**), amely eltárolja a pozíciót, szöveget, valamint engedélyezettséget. A mezőket egy felügyelt gy űjteménybe helyezzük a nézetmodellbe (**Fields**).
* A tárolt játékállapotok egy-egy **StoredGameViewMode**l példánnyal írhatóak le. Ezek kollekcióját nem ágyazzuk be a fő nézetmodellbe

(**EscapeViewModel**), hanem a betöltéskor és mentéskor dinamikusan állítjuk elő és adjuk át a nézet számára.

A screenshot of a computer program

Description automatically generated

**5. ábra: Az applikáció osztálydiagramja**

Nézet:

* A nézetet navigációs lapok segítségével építjük fel.
* A **GamePage** osztály tartalmazza a játéktáblát, amelyet egy **Grid** segítségével valósítunk meg, amelyben **Button** elemeket helyezünk el.
* A **SettingsPage** osztály tartalmazza a betöltés, mentés gombjait, illetve egy **Picker** példányt a nehézség állítására.
* A **LoadPage** és a **SavePage** szolgál egy létező játékállapot betöltésére, illetve egy új mentésére.
* Vezérlés (6. ábra):
* Az **App** osztály feladata az alkalmazás vezérlése, a rétegek példányosítása és az események feldolgozása.
* A **CreateWindow** metódus felüldefiniálásával kezeljük az alkalmazás életciklusát a megfelelő eseményekre történő feliratkozással. Így az alkalmazás felfüggesztéskor (**Stopped**) elmentjük az aktuális játékállást

(**SuspendedGame**), míg folytatáskor vagy újraindításkor (**Activated**) pedig folytatjuk, amennyiben történt mentés.

* Az alkalmazás lapjait egy **AppShell** keretben helyezzük el. Ez az osztály felelős a lapok közötti navigációk megvalósításáért.

A screenshot of a black screen

Description automatically generated

**6. ábra: A vezérlés osztálydiagramja**

**Tesztelés:**

* A modell funkcionalitása egységtesztek segítségével lett ellenőrizve az **EscapeGameModelTest** osztályban.
* Az alábbi tesztesetek kerültek megvalósításra:
* **EscapeGameModelNewGameEasyTest**,

**EscapeGameModelNewGameMediumTest**,

**EscapeGameModelNewGameHardTest**: Új játék indítása, mezők nagyságának ellenőrzése, üldözők számőának ellenőrzése.

* **EscapeGameModelTimeTest**: Játékbeli időeltelés hatásainak ellenőrzése, játék megkezdése előtt, valamint után, újraindítás esetén.
* **PlayerMovementTest**: A játékos mozgásának ellenőrzése.
* **EscapeGameModelLoadTest**: A játék modell betöltésének tesztelése mockolt perzisztencia réteggel.