Eva Beadandó 1

Hartyányi Kevin
hartyanyi.kevin@gmail.com
C0S0RJ

November 2019

Contents

1 Feladat

Készítsünk programot a következő játékra.A játékban egy tengeralattjárót kell irányítanunk a képernyőn (balra, jobbra, fel, illetve le), amely felett ellenséges hajók köröznek, és folyamatosan aknákat dobnak a tengerbe. Az aknáknak három típusa van (könnyű, közepes, nehéz), amely meghatározza, hogy milyen gyorsan süllyednek a vízben (minél nehezebb, annál gyorsabban). Az aknákat véletlenszerűen dobják a tengerbe, ám mivel a hajóskapitányok egyre türelmetlenebbek, egyre gyorsabban kerül egyre több akna a vízbe. A játékos célja az, hogy minél tovább elkerülje az aknákat. A játék addig tart, ameddig a tengeralattjárót el nem találta egy akna.A program biztosítson lehetőséget új játék kezdésére, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozog semmi a játékban). Ismerje fel, ha vége a játéknak, és jelenítse meg, mennyi volt a játékidő.Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére.

2 Elemzés

- A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.
- Az ablakban elhelyezünk egy menüt a következő menüpontokkal: Start Game, Stop Game, Save Game. Figyelni kell, hogy a felhasználó csak akkor tudjon menteni, ha előtte a játékot megállította a Stop Game megnyomásával.
- A játéktáblát egy nxn nyomógombokból álló rács reprezentálja. A nyomógombokal a w,a,s,d gombok használatával tudunk kommunikálni (mozgatni a játékost).
- A játék automatikusan feldob egy dialógusablakot, amikor kezdődik/vége a játéknak, ahol bekérünk adatokat a játék kezdésével kapcsolatban. Szintén dialógusablakokkal végezzük el a mentést, illetve betöltést, a fájlneveket a felhasználó adja meg.
- A felhasználói esetek a figure 2 ábrán láthatóak.

3 Tervezés

- Programszerkezet:
  - A programotháromrétegűarchitektúrában valósíjuk meg. A megjelenítés a View, a modell a Model, míg a perzisztencia a Data.A program csomagszerkezete a figure 1 ábrán látható.
- Perzisztencia

  - Az adatkezelés feladata a Model-el kapcsolatos információk, betöltés/mentés.
  - Az adatokat egy speciális osztály segítségével adjuk át a perzisztencia és a Model között, betöltés esetén.
  - A hosszú távú adattá rolás lehetőségeit az IData interfész adja meg, amely lehetőséget ad a tábla betöltésére (Load), valamint mentésére (Save).
  - Az interfészt szöveges fájl alapú adatkezelésre a Data osztály valósítja meg.
  - A program az adatokat szöveges fájlként tudja eltárolni, melyek a saveGame kiterjesztést kapják. Ezeket az adatokat a programban bármikor be lehet tölteni két játék között, illetve ki lehet menteni az aktuális állást, amenyiben a játékot szüneteltettük előtte.

- Modell
  - A modell lényegi részét a GameControlModell osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék egyéb paramétereit, úgy mint az idő (gameTime) és a nehézség (difficultyTime). A típus lehetőséget ad új játék kezdésére (NewGame), szüneteltetésére(StopGame), továbbá folytatására (StartGame). Új játéknál megadható a kiinduló játéktábla is. Az idő előreléptetését Timer segítségével valósítjuk meg.
  - A játék állapot változásáról különböző események tájékoztatják a nézetett (ShipMoveEvent, BombMoveEvent, BombCreateEvent, ...) a játék végéről a GameOverEvent esemény tájékoztat, ami az aktuális játékídőt is továbbadja.
  - A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (LoadGame) és mentésre (SaveGame)
  - A játék nehézsége folyamatosan növekszik az idő elörehaladtával, ameddig a játékos meg nem hal.

- Nézet
  - A nézetet a Form1 osztály biztosítja, amely tárolja a modell egy példányát (model) , valamint létrehozza az adatelérés példányát, de azt nem tárolja.
  - A játéktáblát egy dinamikusan létrehozott gombmező (table) reprezentálja. A felületen létrehozzuk a megfelelő menüpontokat és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását (a StartGame-ben végezzük miután a pálya méretétt bekértük a felhasználótól.

- Az osztálydiagram a figure 3 ábrán látható.

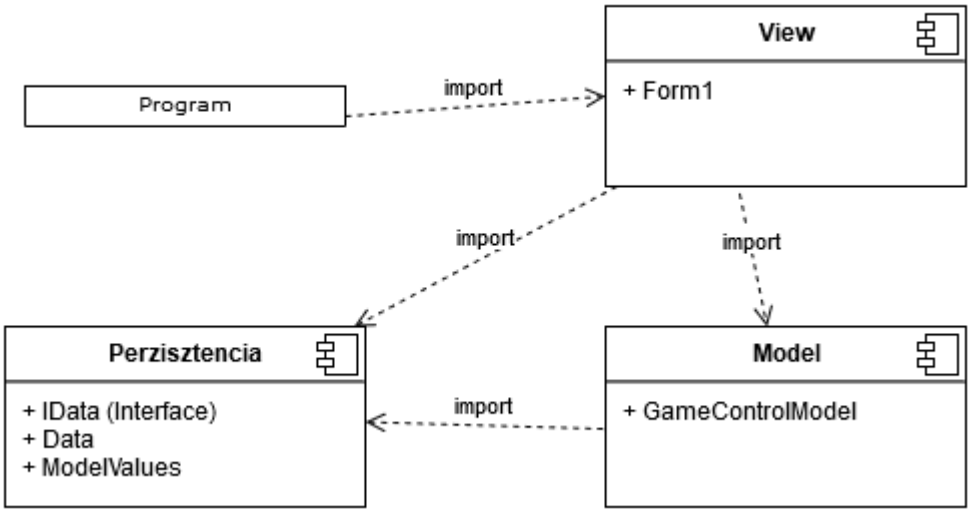

Figure 1: Az alkalmazás csomagdiagramja

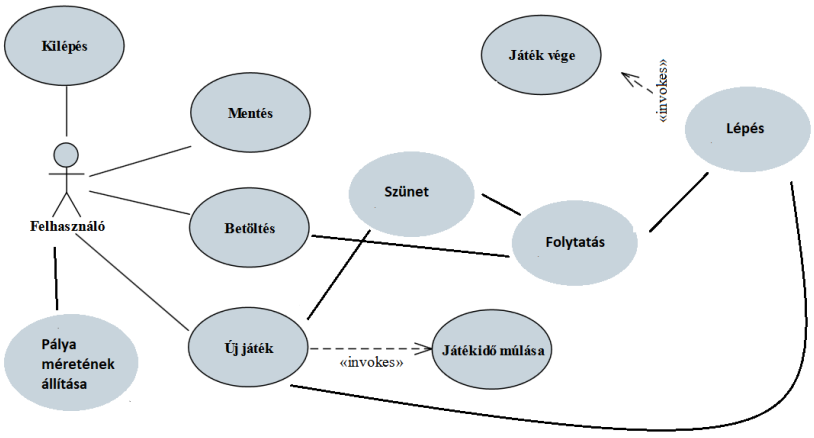

Figure 2: Felhasználói esetek

4 Tesztelés

- A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a Test osztályban.

- Az alábbi tesztesetek kerültek megvalósításra:
  - LoadTest: Teszteljük a játék betöltését
  - SaveTest: Teszteljük a játék mentését, és, hogy az adatok nem változnak a modelben ennek hatására
  - NewGame: Az új játék indítását teszteljük, megfelelően töltödik-e fel a model értékekkel.
  - GameOverTest: A játék végét teszteljük
  - BombCreateTest: Teszteljük, hogy megfelelően létrejönnek-e a bombák a modelben és erről eseményt küld e a form-nak.
  - PlayerMove: Teszteljük a játékos mozgatását mind a 4 irányba.

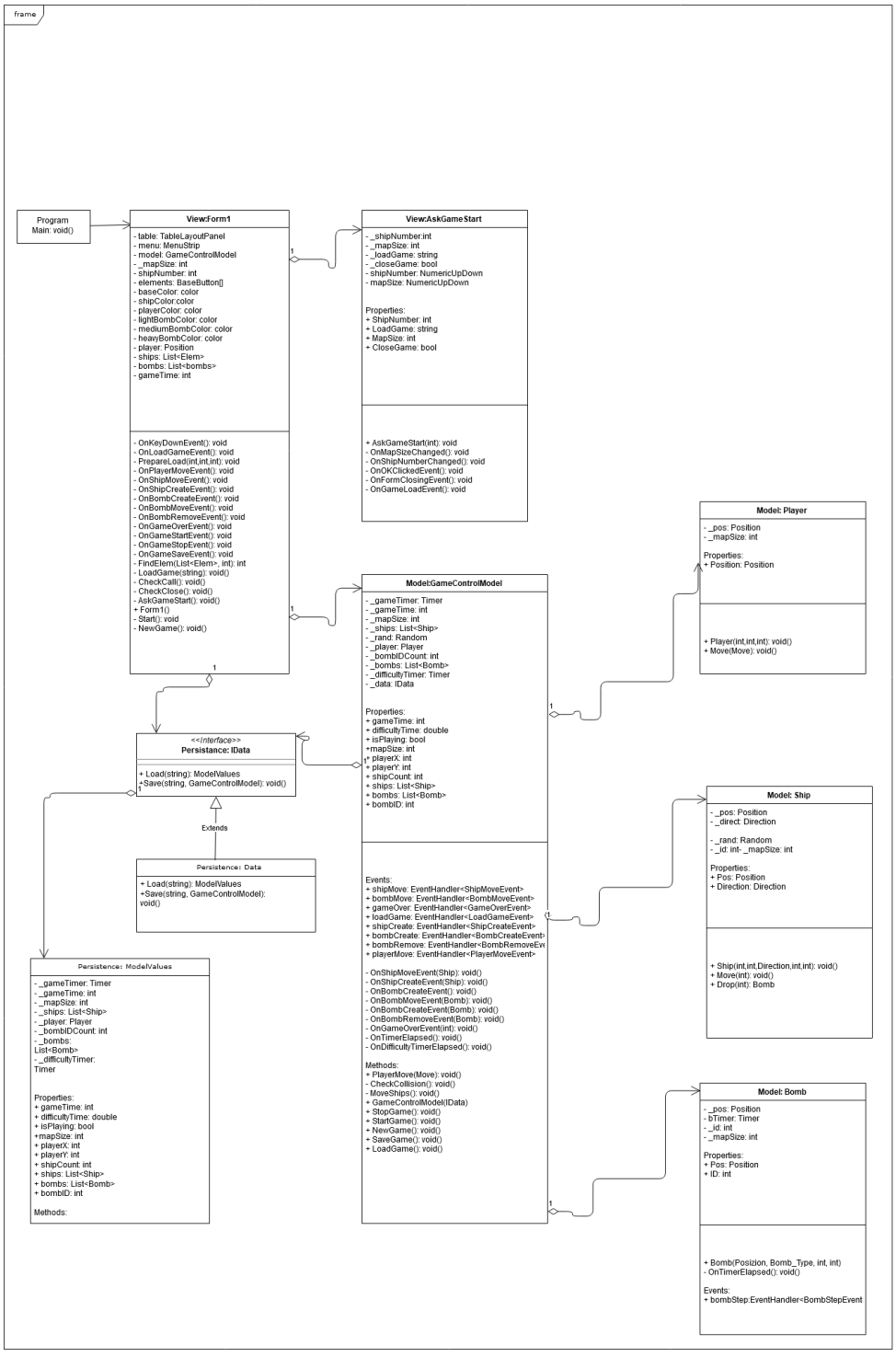

Figure 3: Az alkalmazás osztálydiagramja