

Eseményvezérelt alkalmazások fejlesztése 2

8. beadandó - Lopakodó

Feladat

Készítsünk programot, amellyel a következő játékot játszhatjuk. Adott egy $n \times n$ elemből álló játékpálya, amely falakból és padlóból áll, valamint örök járőröknek rajta. A játékos feladata, hogy a kiindulási pontból eljusson a kijáratig úgy, hogy közben az örök nem látják meg. Természetesen a játékos, illetve az örök csak a padlón tudnak járni.

Az örök adott időközönként lépnek egy mezőt (vízszintesen, vagy függőlegesen) úgy, hogy folyamatosan előre haladnak egészen addig, amíg falba nem ütköznek. Ekkor véletlenszerűen választanak egy új irányt, és arra haladnak tovább. Az ör járőrözés közben egy 2 sugarú körben lát (azaz egy 5×5 -ös négyzetet), ám a falon nem képes átlátni.

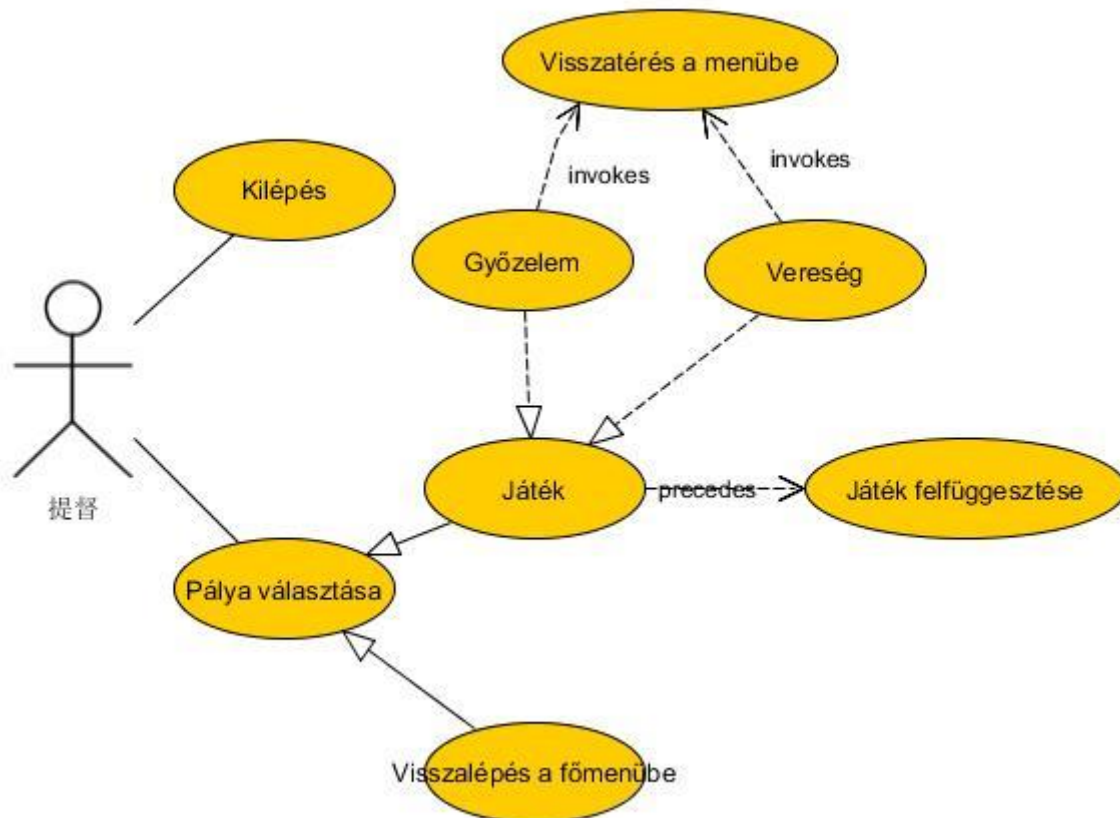
A játékos a pálya előre megadott pontján kezd, és vízszintesen, illetve függőlegesen mozoghat (egyesével) a pályán. A pályák méretét, illetve felépítését (falak és kijárat helyzete, játékos és örök kezdőpozíciója) tároljuk fájlban. A program legalább 3 különböző méretű pályát tartalmazzon.

A program biztosítson lehetőséget új játék kezdésére a pálya kiválasztásával, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem léphet a játékos). Továbbá ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, hogy győzött, vagy veszített-e a játékos.

Elemzés

- A játékot Windows Presentation Foundations (WPF) keretrendszerrel valósítjuk meg, egyablakos alkalmazásként
- A felhasználó a program indításakor a főmenüvel találkozik, ahol kiléphet, vagy választhat pályát.
- Pálya választásnál láthatóak a megfelelő mappában elhelyezett és megfelelő formátumú pályák. Itt egy pályát kiválasztva indítható a játék, vagy visszaléphetünk a menübe
- A játéklemezőn egy tengeralattjáró reprezentálja a játékos pozícióját, melyet a W, A, S, D gombokkal irányíthatunk, illetve az ESC gombbal szüneteltethető a játék.
- A játék végén az eredményt a program egy párbeszédablakkal kommunikálja a játékos felé, melynek nyugtázása után a játék visszatér a főmenübe.

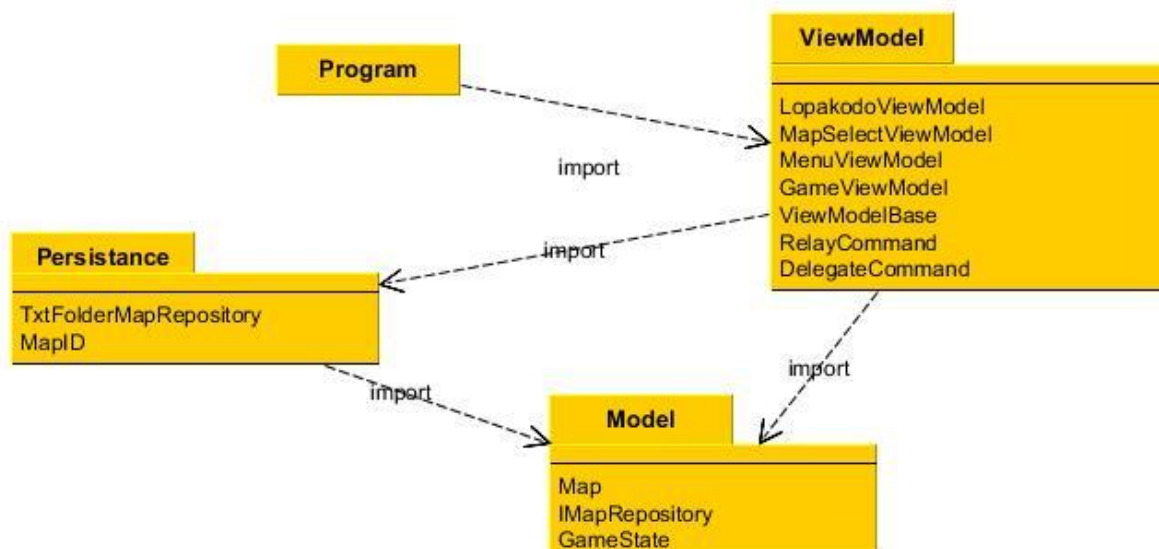
Használati esetek



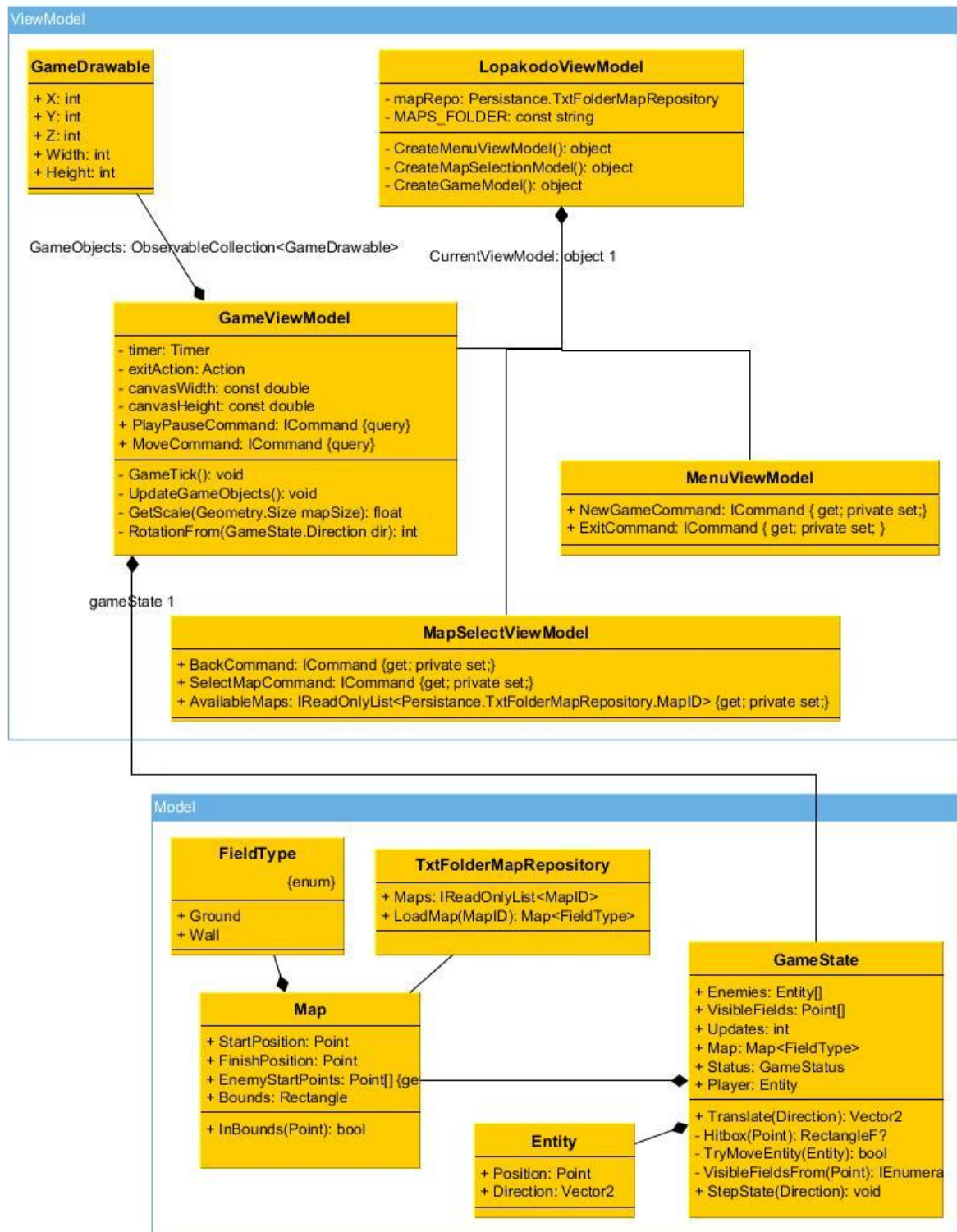
Tervezés

Az alkalmazást modell-nézet-nézetmodell-perzisztencia architektúrával valósítjuk meg.

Perzisztencia: a pálya fájlok egy előre rögzített formátumban tárolódnak. A „map” kiterjesztésű szöveges fájlok első sorában a pálya neve található, második sorban a pálya méretei egy „NxM” formában, utána pedig N sorban M karakter. A sorok végén lehetnek az M databan felüli karakterek, melyek az adott sorban kommentként szolgálhatnak.



Osztályszerkezet



Tesztek

- Mivel a modellt reprezentáló `Lopakodo.Mechanics` osztály nagyon magas absztrakciós szinten van, melyet a `RipSeiko.Geometry` típusai és függvényei biztosítanak, ezért a tesztesetek ezekre a működésekre fókuszálnak és a `RipSeiko.Geometry.Tests` projektben találhatóak, hiszen ezekre építve a modell helyessége garantált.

- A modellben a legfontosabb elem egy pont láthatósága falakon keresztül, így a legtöbb teszt e téglalap-szakasz metszés ellenőrzésének helyességét vizsgálja.