Lombosi Balázs
D3BA80
Eseményvezérelt alkalmazások I.
2016.04.11.

2. Beadandó feladat

Feladat:

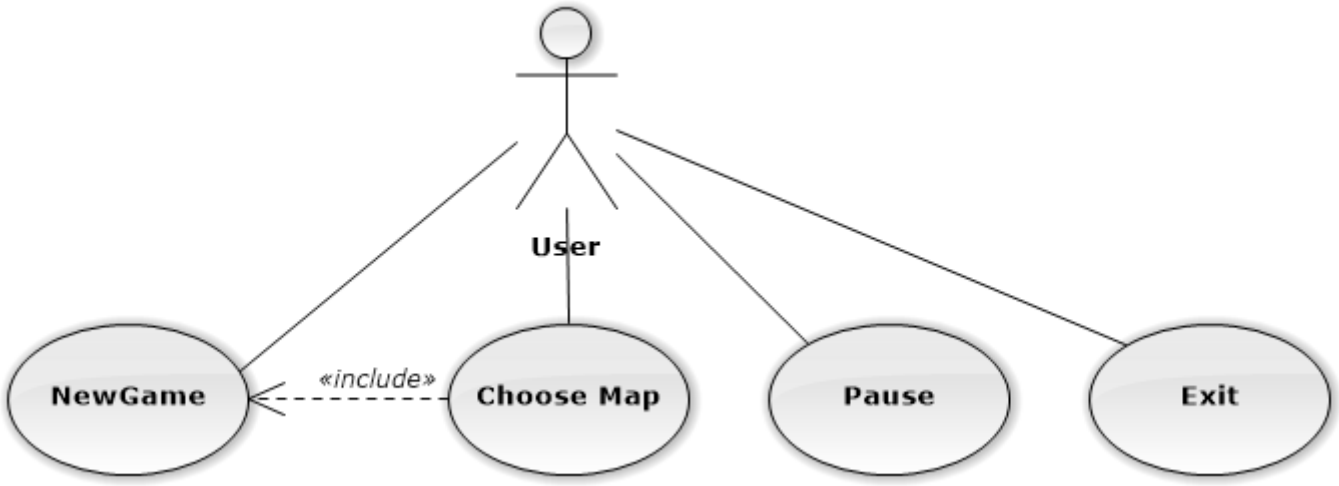
Készítsünk programot, amellyel a következő játékot játszhatjuk. Adott egy $n \times n$ mezőből álló erdő, amelyben Maci Lacival kell piknikkosarakra vadásznunk, amelyek a játékpályán helyezkednek el. A játék célja, hogy a piknikkosarakat minél gyorsabban begyűjtsük. A játékpályán a piknikkosarak mellett akadályok (pl. fa) is elhelyezkedhetnek, amelyekre nem léphetünk. Apályán emellett vadőrök is járőröznek, akik adott időközönként lépnek egy mezőt (vízszintesen, vagy függőlegesen). A járőrözés során egy megadott irányba haladnak egészen addig, amíg akadályba (vagy az pálya szélébe) nem ütköznek, ekkor megfordulnak, és visszafelé haladnak (tehát folyamatosan egy vonalban járőröznek). A vadőr járőrözés közben a vele szomszédos mezőket látja (átlósan is, azaz egy 3 × 3-as négyzetet). A játékos kezdetben a bal felső sarokban helyezkedik el, és vízszintesen, illetve függőlegesen mozoghat (egyesével) a pályán, a piknikkosárra való rálépéssel pedig felveheti azt. Ha Maci Lacit meglátja valamelyik vadőr, akkor a játékos veszít. A pályák méretét, illetve felépítését (piknikkosarak, akadályok, vadőrök kezdőpozíciója) tárolhatjuk fájlban, vagy létrehozhatjuk véletlenszerűen (előre rögzített paraméterek mellett). 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). Ismerje fel, ha vége a játéknak, és jelezze, győzött, vagy veszített a játékos. A program játék közben folyamatosan jelezze ki a játékidőt, valamint a megszerzett piknikkosarak számát.

Lombosi Balázs
D3BA80
Eseményvezérelt alkalmazások I.
2016.04.11.

Elemzés:

- A Játékot egy grafikus felületen jelenítjük meg, ahol nyomógombokat használhat a felhasználó. Egy nyomógomb az új játék kezdéséért felel egy pedig a Játék szüneteltetéséért, ezen kívül található egy gomb ami megnyomása utána játékos kiválaszthat egy pályát (txt file) és egy kilépés gomb. A játék mezőt egy címkékből álló n × n es *QGridSize* jeleníti meg,
- A felületen ezen felül elhelyezünk két címkét amelyek a játékidőt illetve a játékos pontszámát mutatják.
- A játék végén egy előugró ablakkal jelezzük a játék végkimenetelét.

Használati esetek:



Tervezés:

A program szerkezetét két rétegre bontjuk a modell/nézet architektúrának megfelelően. A modell eseményeken keresztül kommunikált a nézettel.
A modellt a Game osztály valósítja meg amely a játéklogikát biztosítja.

- Az esemény kezelés megvalósítása érdekében az osztályt a QObject-ből származtatjuk.
- A játékban eltelt időt egy időzítő segítségével számoljuk **(_gameTimer)** és időzítőt használunk a vadőrök léptetéséhez is **(_timer)**
- A game osztálynak három eseménye a játéktábla változás **(tableChanged()),** a játék vége **(gameOver (<nyert?> , <pont> ,<idő>))** , illetve a játékban eltelt idő megváltozása **(gameTimeChanged(<idő>)),** amelyek paraméterek segítségével adják meg a módosításokat.
- A játék állását egy mátrixban tároljuk **(gametable)**.

A megjelenítést a UserInterface biztosítja amely a QWidget leszármazottja.

- A megjelenítéshez QLabel példányokat használunk amelyre felírjuk, az objektumokat(maci laci, fa, piknikkosár , vadőr). Illetve az időt és a pontokat. A címkéket elrendezéssel **(_table)** helyezzük el az ablakban.
- Ezen felül a felületen helyezünk el négy gombot**(_newGame, _chooseMap, _pause,_exit)** amellyel a különböző funkciókat tudjuk meghívni.
- Lekezeljük a játéklogika három eseményét **(Game::gameOver, Game::gameTimeChanged, Game::tableChanged())**

Lombosi Balázs
D3BA80
Eseményvezérelt alkalmazások I.
2016.04.11.

Osztály szerkezet: