EVA 3.beadandó

Hartyányi Kevin
hartyanyi.kevin@gmail.com
Neptun kód: C0S0RJ

2018.11.27

Contents

1 Feladat

Készítsünk programot, amellyel a következő két személyes játékot játszhatjuk. Adott egy n x n elemből álló játékpálya, ahol két harcos robotmalac helyezkedik el, kezdetben a két ellentétes oldalon, a középvonaltól eggyel jobbra, és mindkettő előre néz. A malacok lézerágyúval és egy támadóököllel vannak felszerelve. A játék körökből áll, minden körben a játékosok egy programot futtathatnak a malacokon, amely öt utasításból állhat (csak ennyi fér a malac memóriájába). A két játékos először leírja a programot (úgy, hogy azt a másik játékos ne lássa), majd egyszerre futtatják le őket, azaz a robotok szimultán teszik meg a programjuk által előírt 5 lépést. A program az alábbi utasításokat tartalmazhatja: előre, hátra, balra, jobbra: egy mezőnyi lépés a megadott irányba, közben a robot iránya nem változik. fordulás balra, jobbra: a robot nem vált mezőt, de a megadott irányba fordul. tűz: támadás előre a lézerágyúval. ütés: támadás a támadóököllel. Amennyiben a robot olyan mezőre akar lépni, ahol a másik robot helyezkedik, akkor nem léphet (átugorja az utasítást), amennyiben a két robot ugyanoda akar lépni, akkor egyikük se lép (mindkettő átugorja az utasítást). A két malac a lézerrel és az ököllel támadhatja egymást. A lézer előre lő, és függetlenül a távolságtól eltalálja a másikat. Az ütés pedig valamennyi szomszédos mezőn (azaz egy 3 x 3-as négyzetben) eltalálja a másikat. A csatának akkor van vége, ha egy robotot háromszor eltaláltak. A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával (4 x 4, 6 x 6, 8 x 8), valamint az aktuális játék mentésére és egy korábban elmentett játék betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött. Játék közben folyamatosan jelenítse meg a játékosok aktuális sérülésszámait.

2 Tervezés

2.1 Elemek

Elössző egy grafikus felületen bekérjük a felhasználótól a tábla méreteit, ha új játékot kíván kezdeni. Továbbá lehetővé tesszük egy korábban elmentett játékállás visszaállítását is.

A méret bekérésénél figyelünk rá, hogy csak megfelelő értékeket fogadjunk el. Ezután egy új felületen (fő felület) létrehozzuk a kért n x n-es nyomógombokból álló táblát.

A nyomógombot megfelelő színűre szinezzük. És elhelyezzük a két játékos harcosát a táblán. A fő felületen helyet kap egy menü sor, ahonnan új játékot indíthatunk, vagy kiléphetünk az alkalmazásból, és elhelyezünk egy pause gombot, amivel a játék felfüggeszthető, továbbá egy másik gombbal lehetőséget adunk az adott játék elmentésére, valamint kijelezzük a két játékos aktuális életpontjait.

3 Megvalósítás

3.1 Fontosabb osztályok és változók

Az alkalmazás indításakor a display osztályt hozzuk létre melyet a QMainWindow-ből származtatunk, mely a játék bezárásáig aktív. Itt a bekért adatok után létrehozzuk a játék model-jét a model osztályt, ez végez minden logikát. A model a két játékos életerejét és koordinátáit egy robotpig osztályon belül tárolja el. Amennyiben a játék véget ért, egy új ablakban értesítjük erről a játékost, hogy nyert vagy vesztett, továbbá lehetőséget biztosítunk új játék kezdésére vagy a játékból való kilépésre.

Amennyiben játék mentésére vagy betöltésére lenne szükség a model kommunikál a filecontroll osztállyal, hogy elérje a kívánt eredményt.

3.2 Eseményvezérlés

Osztályok

1. *display*
   - Slots
     (a) GameOver
         – **Signal** model GameOver

Table 1: Használati esetek

Felhasználói eset	Leírás
Alkalmazás indítása	GIVEN: az alkalmazás telepítve van WHEN: alkalmazás indítása THEN: *display* és Size osztály létrehozása és a felületének megjelenítése
Kilépés	GIVEN: *display* felület WHEN: a felület ablakának lezáró ikonjára kattint THEN: alkalmazás befejezése
Go	GIVEN: *Size* felület WHEN: a **Go** gombra kattint THEN: a *Size* eltűnik és megjelenik a *display* osztály felülete a kért méretű táblával
	GIVEN: *Size* felület WHEN: a **Load** checkbox bepipálása THEN: a *Size* eltűnik és megjelenik a *display* osztály felülete a kért játékállás betöltésév
Pause	GIVEN: *display* felület WHEN: a **Pause** gombra kattint THEN: a játék szünetel, és egy új ablak jelenik meg
	GIVEN: *display* felület WHEN: a **Save** gombra kattint THEN: savedialog megjelenítése ahol bekérjük a játék nevét
	GIVEN: *savedialog* felület WHEN: a **Save** gombra kattint THEN: a játékállás elmentése
Resume	GIVEN: *Pause* felület WHEN: a **Resume** gombra kattint THEN: a játék folytatódik
Új játék	GIVEN: *display* felület WHEN: az új játék-ra kattint a menüsorban THEN: *Size* megjelenítése
Kilépés	GIVEN: *display* felület WHEN: a felület ablakának lezáró ikonjára kattint THEN: alkalmazás befejezése
Lépés	GIVEN: *display* felület WHEN: a next turn gomb lenyomása THEN: megjelenik a lépést bekérő dialógus
	GIVEN: **Next turn** felület WHEN: a command gombra kattintás THEN: a játék elvégzi a kért lépéseket
GameOver	GIVEN: *display* felület WHEN: valamelyik játékos életereje eléri a 0-át THEN: a *gameOver* felület megjelenítése
Kilépés	GIVEN: *gameOver* felület WHEN: a felület ablakának lezáró ikonjára kattint THEN: alkalmazás befejezése
Új játék	GIVEN: *gamaOver* felület WHEN: az **új játék** gombra kattint THEN: a gameOver eltűntetése és a *Size* megjelenítése

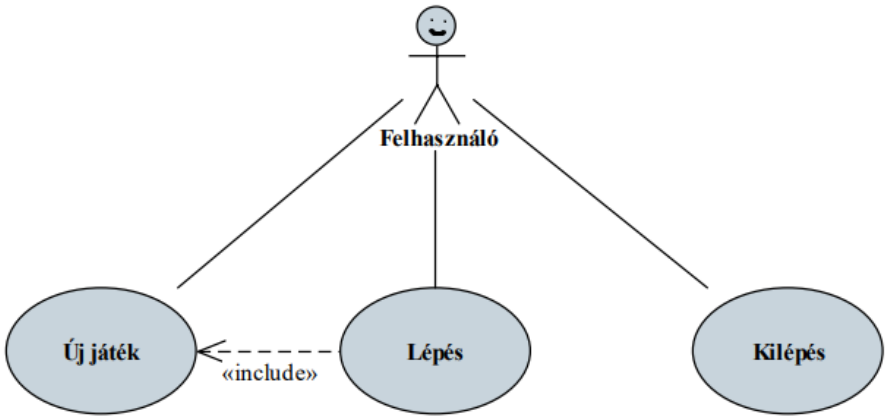

Figure 1: Mr. UML

  - **Action** beállítja a táblát grafikus megjelenítésre
- (b) pause
  - **Signal** *pausebutton* clicked()
  - **Action** a játék szünetelése
- (c) Moved
  - **Signal** model Moved
  - **Action** karakter mozgatása
- (d) setSize
  - **Signal** Size sendSize()
  - **Action** beállítja a méretét a pályának
- (e) changeDirection
  - **Signal** model changeDirection
  - **Action** karakter irányának változtatása
- (f) loadGame
  - **Signal** Size sendLoad()
  - **Action** korábbi játékot tölt be
- (g) Laser
  - **Signal** model LaserAttack()
  - **Action** a karakter laser-el támad
- (h) Punch
  - **Signal** model Punchattack(int,int)
  - **Action** a karakter ütés-el támad
- (i) LoadTheGame
  - **Signal** model SendLoad()
  - **Action** a játék betöltése

- Signals **nincs**

2. *model*

- Slots **nincs**
- Signals
  - Moved
  - ChangeDirection
  - LaserAttack
  - PunchAttack
  - GameOver

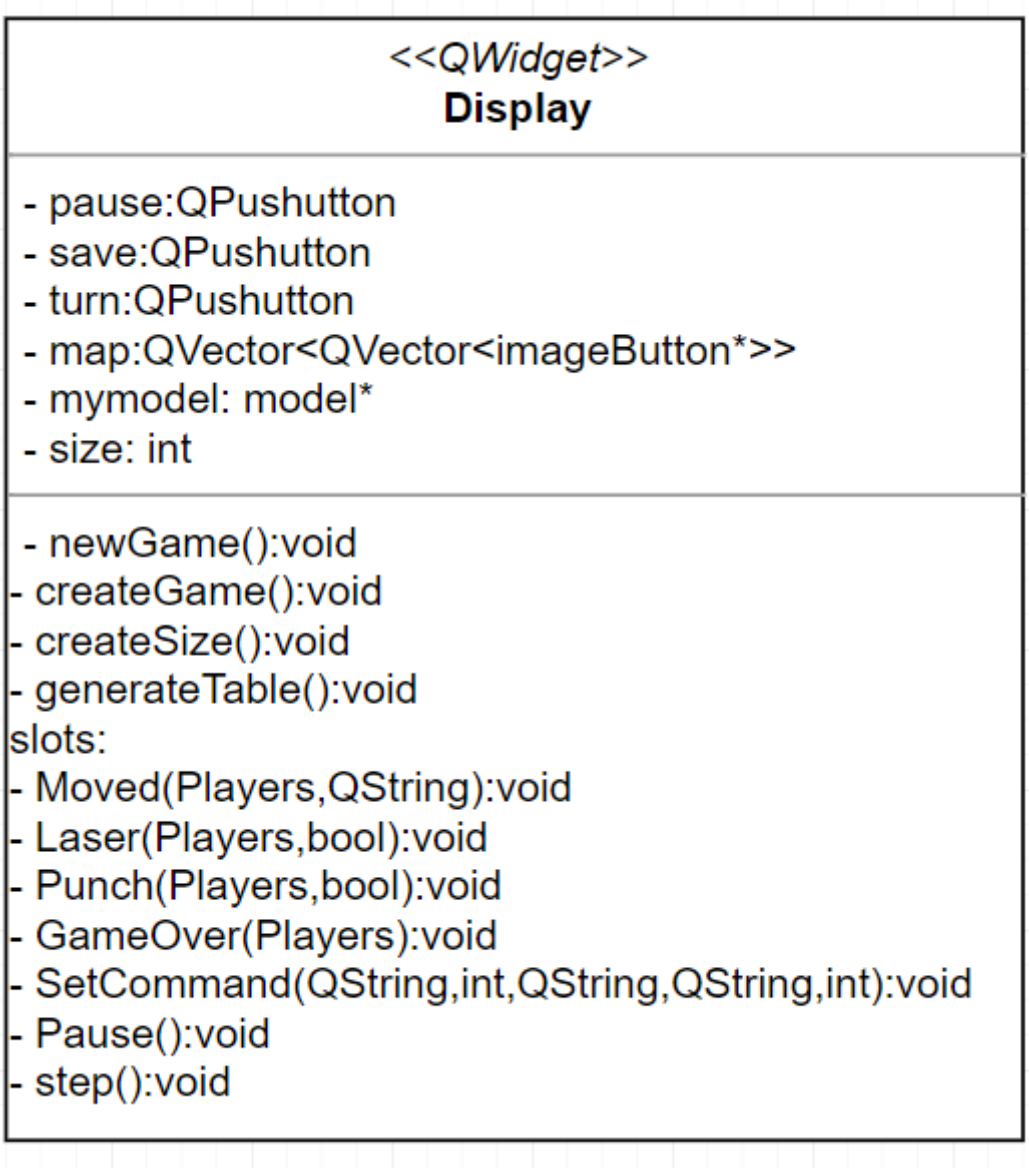

Figure 2: *Display* UML diagram

4 Végfelhasználói tesztesetek

Teszt eset	Elvárt hatás
Alkalmazás indítás hatása	Megjelenik a kért méretű tábla a két karakterrel a pályán a középvonaltol eggyel távolabb mindkettő a saját oldalán
Alkalmazás indítása betöltéssel	Korábban elmentett játékállás visszaállítása
Kilépés játékból	Az alkalmazás leáll
Szüneteltetés	Új ablak megjelenítése és a játék szüneteltetése
Folytatás	A játék folytatása
Játék kezdésnél rossz érték megadása méretnek	A pálya a beállított legkisebb mérettel jön létre
A Size ablakon nem a Go gombra, hanem a bezárásra kattint	A játék a kiválasztott mérettel jön létre

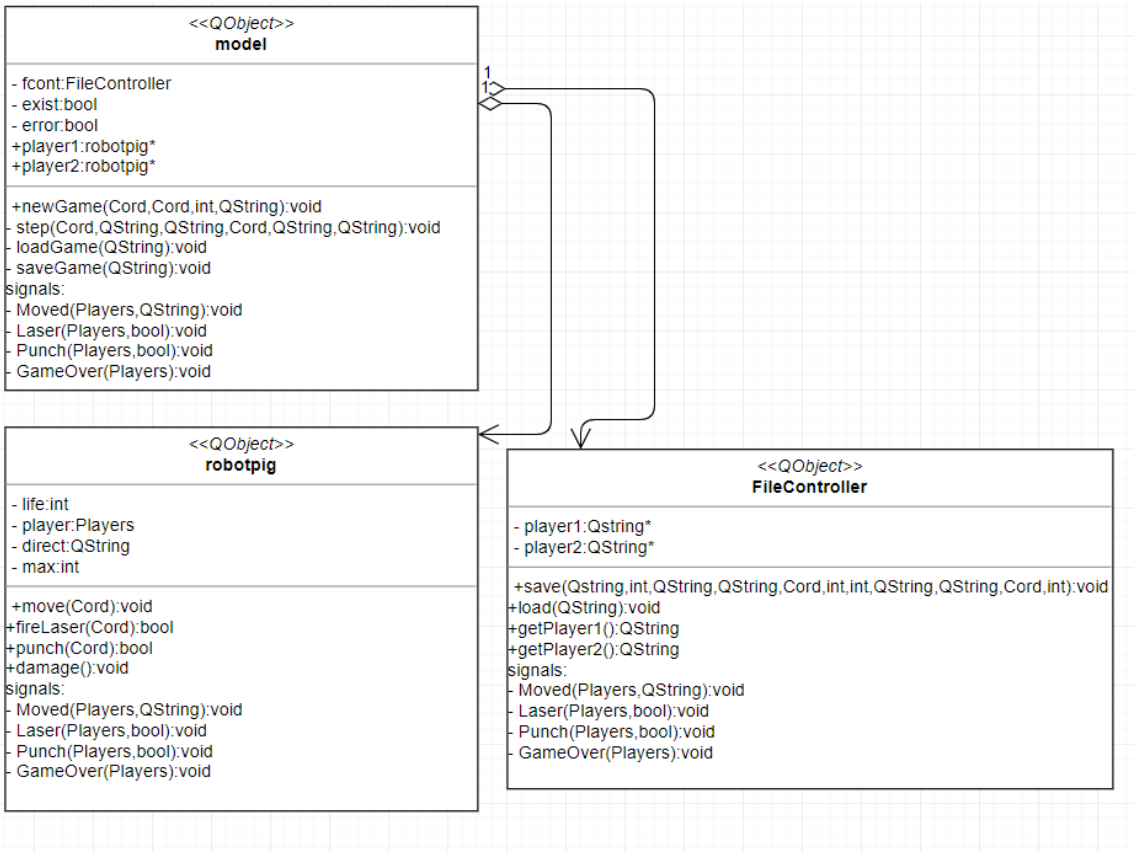

Figure 3: *model* UML diagram