1. Beadandó feladat dokumentáció

Feladat:

Készítsünk programot, amellyel a következő két személyes játékot játszhatjuk. Adott egy n × 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 × 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 × 4, 6 × 6, 8 × 8), valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött. Játék közben folyamatosan jelenítse meg a játékosok aktuális életerő számukat.

Elemzés:

• A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.

• Az ablakban elhelyezünk:

1. Egy menüt az ablak tetején a következő menüpontokkal: File (Új játék, Játék betöltése, Játék mentése, Kilépés).
2. Az ablak közepén találhatjuk, a választott nagyságnak megfelelő méretű játékasztalt.
    1. Amikor elindítjuk a programot, a bal felső sarokban tudunk új játékot indítani, vagy egy korábbi expliciten lementett állapotot betölteni.
    2. Egy új játék kezdésekor létrehozunk egy új játékasztalt, egy n x n-es Label-ekből álló rácsot. A malacok helyzetét és irányát a megfelelő Label-en szöveg segítségével reprezentáljuk. A két játékost, a malacok színei segítségével lehet megkülönböztetni.
    3. Amikor a malacok valamilyen támadást hajtanak végre, a táblán az érintett mezők a malac színével felvilágítanak. Így egyértelműen látszik a felhasználóknak, hogy ki, mikor és miért sebződött.
3. Az ablak alján található egy a felhasználók által használt panel:
    1. A jobb oldalon található az egy-egy játékos hátralévő életereje.
    2. A bal oldalon található egy felirat, amely jelzi, hogy melyik játékostól várjuk, hogy a parancsait beüsse. Alatta található a multiline-textbox amelyen ezt megteheti.
        1. A textbox mellett található „Rögzítés” feliratú gombbal tudják véglegesíteni a döntéseiket. Amennyiben bemenet hibás, egy új felugróablak értesítjük őt erről a hibáról és, hogy mit kell tennie, hogy érvényes parancsokat adjon.
        2. Ez a szekció csak akkor aktív, ha épp bemenetet várunk.
    3. A panel közepén található egy „Kezdés” feliratú gomb. Ezzel a gombbal lehet elindítani a kört miután mindkét játékos érvényes parancsokat adott meg.
        1. A gomb alatt található egy „Auto” feliratú checkbox, amelyet ha bekapcsolunk, akkor a „Kezdés” feliratú gomb megnyomása után másodpercenként automatikusan lejátszódnak a körök. Ezt bármikor ki vagy be lehet kapcsolni.
    4. Minden egyes lépéskor fontos fenntartani a reprezentáció helyességét. Ebben az esetben arra kell figyelni, hogy a malacok ne tudják elhagyni a játékteret.

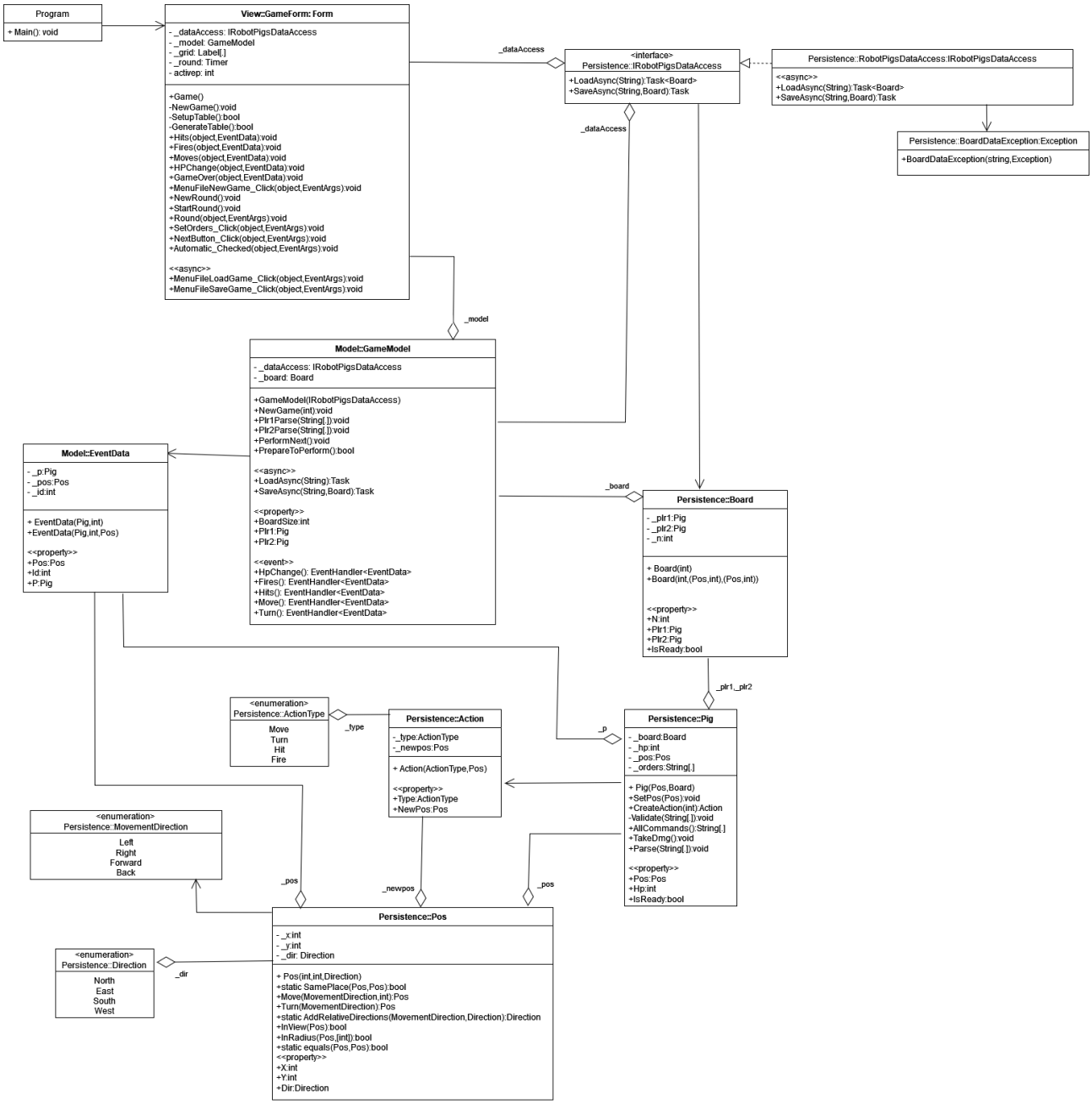
• A játék önműködően feldob egy dialógusablakot, amikor vége a játéknak, hogy melyik játékos nyert. Szintén dialógusablakokkal végezzük el a mentést, illetve betöltést, a fájlneveket a felhasználó adja meg. A választott állomány típus: „.dat”

Felhasználói esetek:

(userdiag.png)
Először az egyes, majd a kettes játékos rögzítheti a parancsait, a kezdés csak ezután válik elérhetővé.
A kezdés gomb megnyomása után a parancsok beütése illetve a játék mentése elérhetetlenné válik amíg a kör be nem fejeződik, a „Következő” gomb 5 darab megnyomása által.

Tervezés:

- Programszerkezet:
  - A programot háromrétegben valósítjuk meg. A megjelenítés a „WFA”, a modell a „Model”, míg a perzisztencia a „Persistence” névtérben helyezkedik el.
  - A programot két projektre osztjuk: a „Persistence „és „Model” csomagok a program felületfüggetlen projektjében, míg a „WFA” csomag a Windows Formstól függő projektjében kap helyet.
  - A program a „WFA” projektet indítja el, a „WFA” felhasználja mind a „Model „-t illetve „Persistence”-t a megjelenítéshez. A „Model” csomag működéséhez felhasználja a „Persistence” csomagot.
- Perszistencia:
  - A „Board” mindig egy érvényes táblát ábrázol, azaz az osztály illetve adattagjai, mindig ellenőrzik a beállított értékeket.
  - A tábla illetve adattagjai lehetőséget adnak az értékeik lekérdezésére, property-k által, illetve egyes értékek beállítására settereken keresztül.
  - A lemezre való mentés lehetőségét az „IRobotPigsDataAccess” interfész adja. A töltéshez „LoadAsync”, a mentéshez pedig „SaveAsync” műveletet lehet használni.
  - Ezt az interfészt a „RobotPigsDataAccess” osztály valósítja meg, a fellépő esetleges hibákat a „BoardDataException” kivétel jelzi.
  - A program az adatokat szöveges fájlban menti el, ahol az első sorban a tábla mérete található, a következő két sorban pedig a malacok tulajdonságait írja le számokkal reprezentálva: X Y koordináta, majd az irány és végül az életereje.
- Nézet:
  - A nézetet a Game osztály valósítja meg. Amely tárolja a modellt illetve az adatelérés módját.
  - A játéktáblát dinamikusan hozzuk létre egy játék kezdésekor, n-szer n-es Label ráccsal megvalósítva. Ezen felül létrehozzuk a megfelelő menüpontokat illetve a felhasználók által használt bemeneteket illetve az ezekhez tartozó eseménykezelőket.
  - Az input sok lehetséges állapotnak a logikájára esetleges átdolgozáskor nagyon fontos odafigyelni!
- Model:
  - A „GameModel” osztály valósítja meg, a nézet ezen keresztül tudja módosítani a táblát, illetve ezen keresztül tudja elérni tábla adatait is.
  - Lehetőséget ad egy új játék kezdésére, avagy annak mentésére, betöltésére.
  - A játék állapotának változásáról öt darab különböző cselekvéshez kapcsolt eseménnyel értesíti a nézetet. Minden esetben egy „EventData” osztályban küldi el az új, illetve az esemény szempontjából hasznos adatokat.
- A teljes statikus szerkezet, illetve az egyes műveletek pontos azonosítója, szignatúrája a következő oldalon látható:



(uml.drawio.png)

Tesztelés:
A modell működőképességét egységtesztek ellenőrzik. Ezek a következő osztályokban és funkciók csoportjaiban teljesülnek.

BoardTest
├ PlrPosTests: Egy új játék indulásakor a malacok elhelyezkedése és életereje.
├ PerformTests: Aktív játék során bemenetek feldolgozása.
PosTest
├ AddRelativeDirectionTests: Relatív irányok összeadása
├ MoveTests: Mozgatások
├ MoveTestHitWallX: olyan mozgás amikor „X” falnak nekimegy egy malac
├ InviewTest: Eltalál-e egy malac a másik malacot a helyzeteik alapján.
├ InRadiusTest: Megtud-e ütni egy malac a másik malacot a helyzeteik alapján.
PigTest
├ ValidateTests: Lehetséges parancssorozatok validációja.
RobotDataAccessTest
├ Test1: Egy új játék elindítása, mentése betöltése.
└ Test2: Egy már folyó játék elindítása.