1. **Analízis modell kidolgozása (Javítás)**
   1. ***Objektum katalógus***
      1. **EndOfField**

Ezen a mezőn nincsen talaj, amire landolni tudna a jármű. Így ha ilyenre lép a körében egy játékos, akkor az kiesik a játékból. Ez a mező is ismeri a saját szomszédjait.

* + 1. **Finish**

Ha a robot a köre végén egy ilyen típusú mezőre lép, akkor befejeződött a játék, és a robotot irányító játékos nyert. Ez a mező is ismeri a saját szomszédjait.

* + 1. **Game**

A játék azon részét kezeli, amikor nem történik felhasználói interakció. Kezdetben bekéri a maximálisan futtatott körök számát, és betölti a pályát. Későbbi feladata a körök léptetése. Feladata még, hogy új játék esetén az előző pálya eldobása, és az új betöltése.

* + 1. **Goo**

Ha a robot ilyen anyaggal ellátott mezőre lép, a sebessége megfeleződik. Ez a mező is ismeri a saját szomszédjait.

* + 1. **MapFactory**

Feladata a játék indulásakor a pálya fájlból való betöltése, felépítése, és tárolása. A pálya több különböző mezőből épül fel. Ezek a mezők mind külön-külön objektumok, amiket ez fog elrendezni, hogy egymással egységben megalkossák a kívánt pályát.

* + 1. **NormalTile**

Ez a fajta mező semmi különöset nem nyújt, ha valaki ráugrik, minden változás nélkül tovább haladhat. Ez a mező is ismeri a saját szomszédjait.

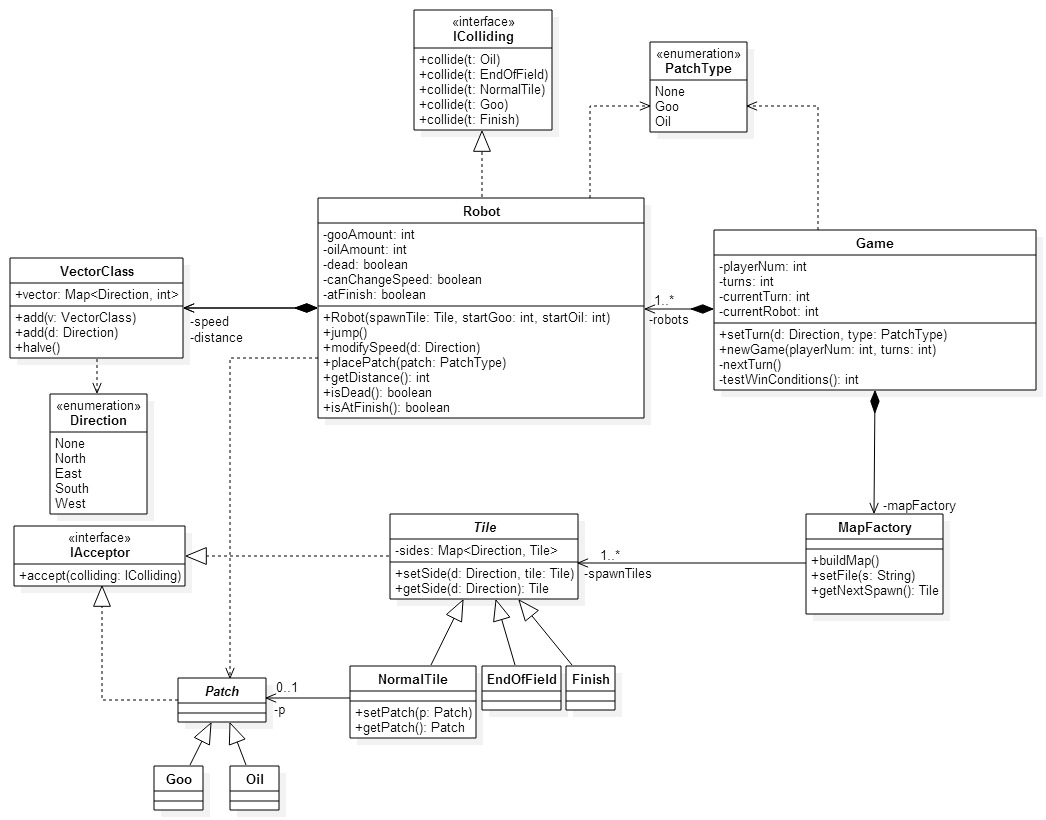
* + 1. **Oil**

Ha a köre végét egy ilyen adalékkal ellátott mezőn fejezi be a robot, akkor a következő körében nincs lehetősége módosítani a sebességét. Ez a mező is ismeri a saját szomszédjait.

* + 1. **Robot**

A játékos irányította robot. Ez az objektum felelős a robotért. Ismeri a saját pozícióját, állapotát, ragacs és olajkészletét, illetve a sebességét. A felhasználó vele tud minden körében kommunikálni arról, hogy miként szeretné módosítani a sebességét. Ez az objektum felelős még, a robot ugrásának következményeinek vizsgálatáért.

* 1. ***Statikus struktúra diagramok***



* 1. ***Osztályok leírása***
     1. **EndOfField**
        + **Felelősség**

Ezen a mezőn nincsen talaj, amire landolni tudna a jármű. Így ha ilyenre lép a körében egy játékos, akkor az kiesik a játékból.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**IAcceptor
      * **Attribútumok**
      * **Metódusok**
    1. **Finish** 
       - **Felelősség**

Ezen a mező annyit tud, hogy van rajta talaj rá lehet ugrani, és ha a robot ezen e mezőn fejezi be a körét akkor megnyerte a játékot.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**IAcceptor
      * **Attribútumok**
      * **Metódusok**
    1. **Game**
       - **Felelősség**

A program inicializálása, új játék indítása, valamint annak levezénylése. A játékosok számának, robotjainak nyilvántartása, valamint egyéb a játékmenethez köthető információk tárolása, kezelése.

* + - * **Ősosztályok**
      * **Interfészek**
      * **Attribútumok**
* **int playerNum**: A játékosok száma a legutóbb indított játékban. A pályára ennyi robot fog kerülni a játék elején.
* **int turns**: A maximálisan megengedett körök száma. Amennyiben ezt elérjük, a játéknak vége, és a legnagyobb távolságot megtett robot tulajdonosa nyer.
* **int currentTurn**: A jelenlegi kör száma. (Vagyis az eddig eltelt körök száma.)
* **int currentRobot**: Annak a robotnak a száma, amelynek éppen az irányítását végzik.
* **Vector<Robot> robots**: A játék alatt a játékosok által irányítható robotok listája.
* **MapFactory mapFactory**: A pálya betöltését végző objektum.
  + - * **Metódusok**
* **void setTurn(Direction d, PatchType type)**: A jelenleg soronlévő robot sebességének módosítása és a lerakandó folt típusának beállítása annak *modifySpeed* és *placePatch* nevű függvényeinek meghívásával.
* **void newGame(int playerNum, int turns)**: Új játék indítása. Beállítja a maximálisan megengedett körök számát (*turns*), játékosok számát (*playerNum*), létrehozza az azokhoz tartozó robotokat, valamint betölteti és inicializáltatja a pályát a *mapFactory* objektummal.
  + 1. **Goo** 
       - **Felelősség**

Ez a ragacs osztálya, ha ilyen mezőre érkezik a robot, akkor elveszti sebességének a felét.

* + - * **Ősosztályok**

Patch

* + - * **Interfészek**IAcceptor
      * **Attribútumok**
      * **Metódusok**
    1. **IAcceptor**
       - **Felelősség**Visitor mintán alapuló interfész biztosítása azoknak a mezőknek, és ragacsoknak.
       - **Ősosztályok**
       - **Interfészek**
       - **Attribútumok**
       - **Metódusok**

Az alábbi metódusokat a leszármazott osztályoknak kell megvalósítaniuk, és az itt megadott helyzetek kezelését hivatottak megoldani.

* **void accept(IColliding colliding)**: Feladata a kapott IColliding megfelelő collide metódusának meghívása.
  + 1. **IColliding (interfész)**
       - **Felelősség**

Visitor mintán alapuló interfész biztosítása azon osztályoknak, melyek interakcióba kerülhetnek a mezőkkel.

* + - * **Ősosztályok**
      * **Metódusok**

Az alábbi metódusokat a leszármazott osztályoknak kell megvalósítaniuk, és az itt megadott helyzetek kezelését hivatottak megoldani.

* **void collide(Oil t)**: Viselkedés leírása, amennyiben a robot olajfoltra lépett.
* **void collide(EndOfField t)**: Viselkedés leírása, amennyiben a robot olyan mezőre lépett, amely már nem tartozik azon részhez, amin érvényesen haladhat.
* **void collide(NormalTile t)**: Viselkedés leírása, amennyiben a robot szabad mezőre lépett.
* **void collide(Goo t)**: Viselkedés leírása, amennyiben a robot ragacsra lépett.
* **void collide(Finish t)**: Viselkedés leírása, amennyiben a robot a pálya végét (célt) jelző mezőre lépett.
  + 1. **MapFactory**
       - **Felelősség**

Az eltárolt fájlnév alapján a pálya betöltése és inicializálása, az abban megadott kezdőpozíciók eltárolása.

* + - * **Ősosztályok**
      * **Interfészek**
      * **Attribútumok**
* **Vector<Tile> spawnTiles**: Azon mezők listája, amelyekről a robotok a játék elején elindulnak.
  + - * **Metódusok**
* **void buildMap()**: A megnyitott fájlban található adatok alapján felépíti a pályát.
* **void setFile(String s)**: Megnyitja a megadott nevű pályafájlt, és bezárja az esetleg korábban megnyitottat.
* **Tile getNextSpawn()**: Visszaadja a pályához tartozó soronkövetkező kezdőpozíció mezejét.
  + 1. **NormalTile** 
       - **Felelősség**

Ezen az általános mező, amire ugorhat a robot, és gond nélkül tovább is haladhat. Később erre kerülhet rá a ragacs, vagy az olaj.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**IAcceptor
      * **Attribútumok**
* **Patch p:** A mezőn található folt.
  + - * **Metódusok**
* **void setPatch(Patch p):** Beállítja a mezőn lévő foltot a megadott foltra.
* **Patch getPatch():** Visszaadja a mezőn lévő foltot, vagy null-t.
  + 1. **Oil** 
       - **Felelősség**

Ezen a mezőre rakható anyag, a robot sebességmódosító képességét veszi el, ha rálép a robot.

* + - * **Ősosztályok**

Patch

* + - * **Interfészek**IAcceptor
      * **Attribútumok**
      * **Metódusok**
    1. **Patch (absztrakt)**
       - **Felelősség**

Feladata a foltok egységbe zárása.

* + - * **Ősosztályok**
      * **Interfészek**IAcceptor
      * **Attribútumok**
      * **Metódusok**
    1. **Robot**
       - **Felelősség**

A kezdőpozíciótól megtett távolságának, a sebességének, valamint ragacs- és olajkészletének nyilvántartása és módosítása szükség esetén. Az ugrás elvégzése és adott esetben a ragacs/olajfolt letevése eközben a jelenlegi pozícióra.

* + - * **Ősosztályok**
      * **Interfészek**

IColliding

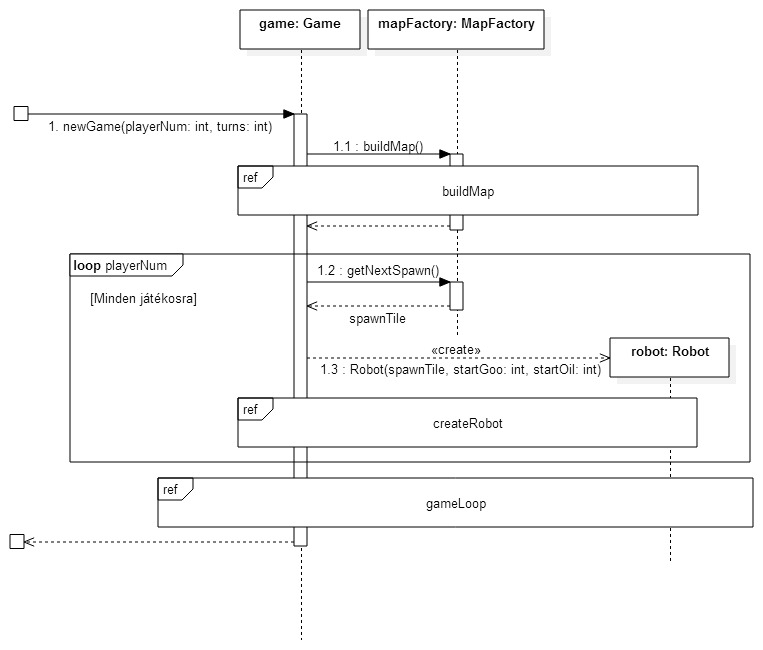
* + - * **Attribútumok**
* **int gooAmount**: A robotnál lévő ragacsok száma, melyket még a pályára letehet.
* **int oilAmount**: A robotnál lévő olajadagok száma, melyket még a pályára letehet.
* **Tile position**: Azon mező, amelyen a robot éppen tartózkodik.
* **VectorClass speed**: A robot jelenlegi sebessége.
* **VectorClass distance**: A robot által megtett össztávolság.
* **boolean dead**: Azt mutatja, hogy a robot halott-e. Amennyiben igen (értéke *true)*, a robot nem léphet többször a jelenlegi játék alatt.
* **boolean canChangeSpeed**: Megmutatja, hogy lehetőség van-e megváltoztatni a robot sebességét a jelenlegi körben. (Ha nem, az értéke *false*.)
* **boolean atFinish**: Ez jelzi, ha a robot egy olyan mezőn van, amely *Finish* típusú. Ha az értéke *true*, az adott robot beért a célba, tehát nyert.
  + - * **Metódusok**
* **Robot(Tile spawnTile, int startGoo, int startOil) (konstruktor)**: Inicializálja a robotot: Beállítja a pozícióját, valamint a ragacs- és olajkészletét a kapott értékek alapján.
* **void collide(Oil t)**: A robot *canChangeSpeed* változóját *false*-ra állítja, ezzel megakadályozva hogy ebben a körben sebességváltoztatás lépjen érvénybe.
* **void collide(EndOfField t)**: Ennek a függvénynek a meghívása azt jelenti, hogy a robot leesett. Beállítja a *dead* változó értékét *true*-ra.
* **void collide(NormalTile t)**: A robot üres mezőn van, nem történik semmi különös. (Nincs hatása)
* **void collide(Goo t)**: Lecsökkenti a robot sebességét a felére: Meghívja a robot *speed* változójának *halve* függvényét.
* **void collide(Finish t)**: Amennyiben a robot *Finish* típusú mezőre lépett, nyert. Beállítja az *atFinish* nevű változó értékét *true*-ra.
* **void jump()**: Ugrik, azaz elvégzi a pozíciójának és a megtett távolságának módosítását, ezután meghívja a saját *collide* függvényét azzal a mezővel paraméterként, amire érkezett.
* **void modifySpeed(Direction d)**: Módosítja a sebességét a paraméterében kapott irány szerinti egységvektorral, ha a *canChangeSpeed* változó értéke *false*.
* **void placePatch(PatchType patch)**: Lehelyez maga alá egy a paraméterében meghatározott típusú foltot, úgy hogy kicseréli a maga alatt lévő mezőt a megfelelő típusúra a mező *swapTile* nevű függvényének meghívásával. Ezután csökkenti a megfelelő típusú foltból hátralévő készletének számát.
* **int getDistance()**: Visszaadja a kezdőpozíciótól megtett távolságot. Visszatérési értéke az eltárolt *VectorClass distance* változó alapján kerül kiszámolásra.
* **boolean isDead()**: Visszaadja az *isDead* nevű változó értékét.
* **boolean isAtFinish()**: Visszaadja az *atFinish* nevű változó értékét.
  + 1. **Tile (absztrakt)**
       - **Felelősség**

A körülötte lévő szomszédos mezők nyilvántartása, saját cseréje esetén a pálya konzisztenciájának megtartása.

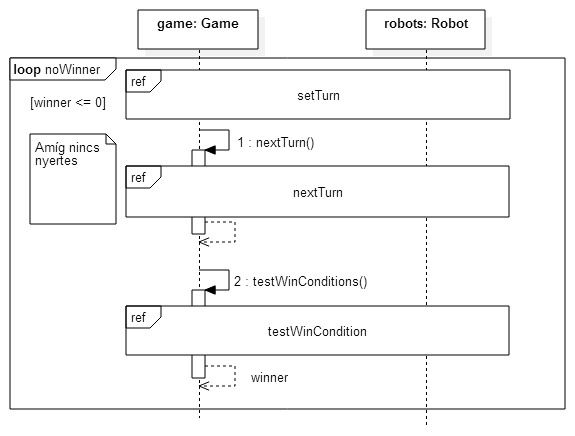
* + - * **Ősosztályok**
      * **Interfészek**IAcceptor
      * **Attribútumok**
* **Map<Direction, Tile> sides**: A szomszédos mezők, ahol minden lehetséges irányhoz egy darab szomszéd tartozik.
  + - * **Metódusok**
* **void setSide(Direction d, Tile t)**: Beállítja a megadott *d* irányban lévő szomszédját a *t* paraméterként kapott mezőre.
* **Tile getSide(Direction d)**: Visszaadja a mező paraméterként kapott irányban lévő szomszédját.
  + 1. **VectorClass**
       - **Felelősség**

Azon egységvektorok irányainak eltárolása, amelyeket egyenként felhasználva megkapunk egy a pályán navigációra használható vektort.

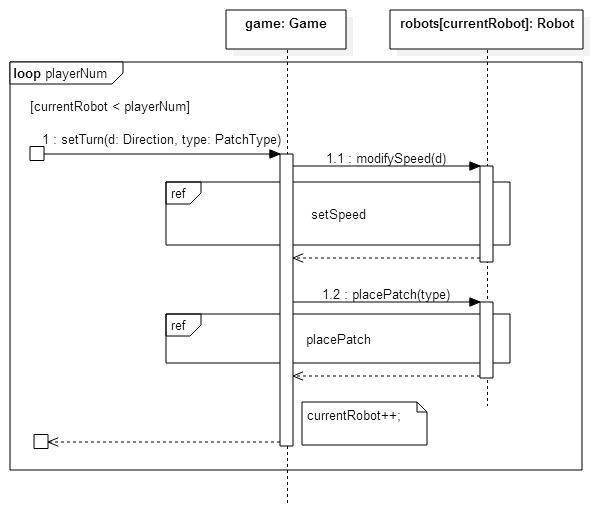
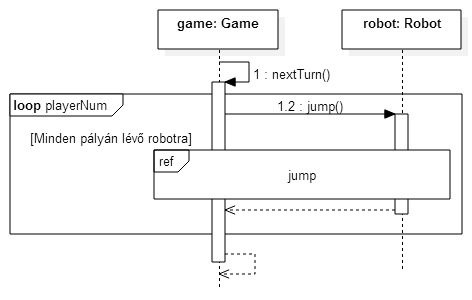
* + - * **Ősosztályok**
      * **Interfészek**
      * **Attribútumok**
* **Map<Direction, int> vector**: Azon egységvektorok iránya, melyekből a teljes vektor felépül. Az egyes irányokhoz egy szám van rendelve, ami megmutatja hogy hány egységet kell az adott irányba lépni.
  + - * **Metódusok**
* **void add(VectorClass v)**: Hozzáadja a paraméterül kapott vektort ehhez, úgy hogy az abban található tároló elemeit hozzáadja a saját tárolójához. (Ez a függvény használandó a távolságvektor megváltoztatására.)
* **void add(Direction d)**: Hozzáad egy paraméterül kapott irányú egységvektort a belső tárolójához, vagy amennyiben ez ellentétes irányú a tárolóban megtalálható vektorok egyikéhez képest, azt kiveszi a tárolóból. (Ez a függvény használandó a sebességvektor megváltoztatására.)
  + - * **void halve()**: Megfelezi a vektort: Az eltárolt egységvektorok számát lefelezi minden irányra nézve
  1. ***Szekvencia diagramok***



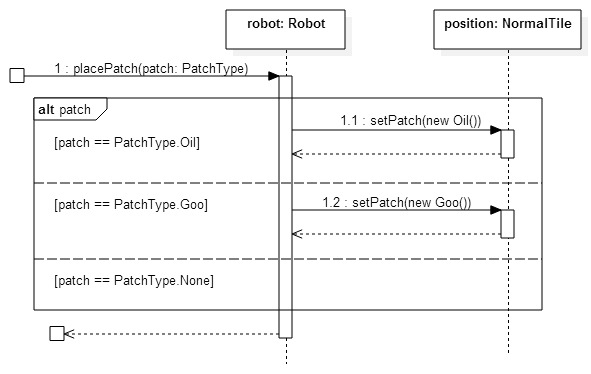
Error! No text of specified style in document.**‑1. diagram Új játék**

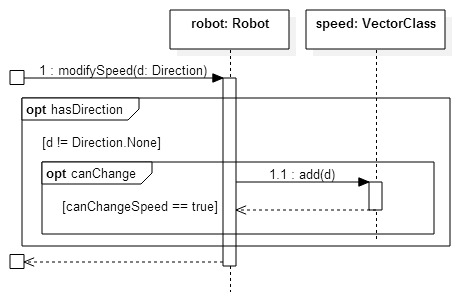


Error! No text of specified style in document.**‑2. diagram Főciklus (gameLoop)**

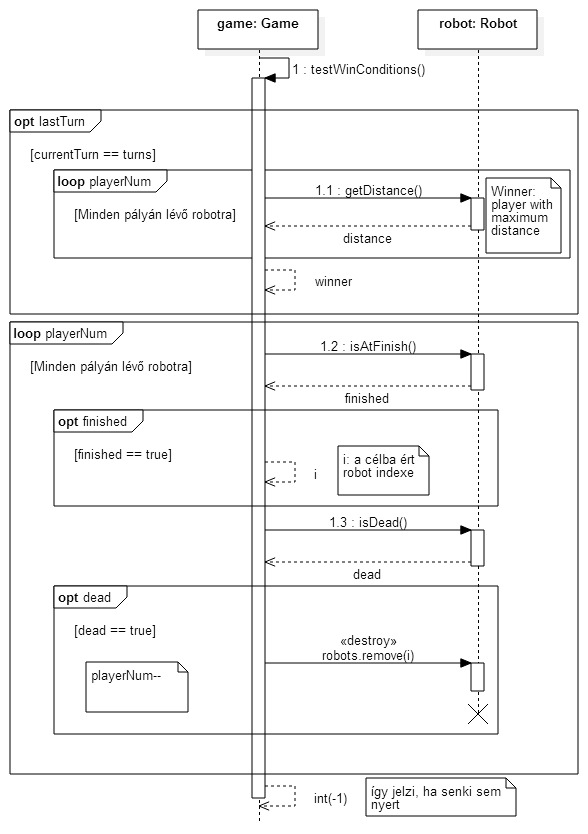
****Error! No text of specified style in document.**4‑3. diagram Robotok lépésének beállítása (setTurn) **

Error! No text of specified style in document.**‑4. diagram Kör léptetése (nextTurn)**

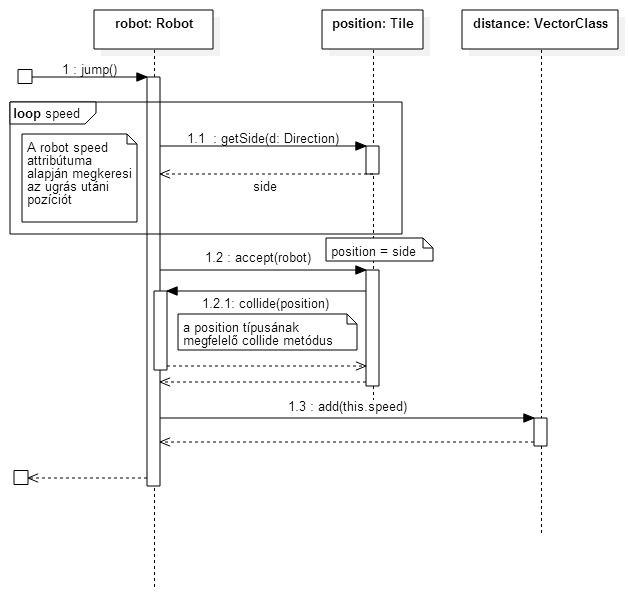


Error! No text of specified style in document.**‑5. diagram Folt elhelyezése (placePatch) **

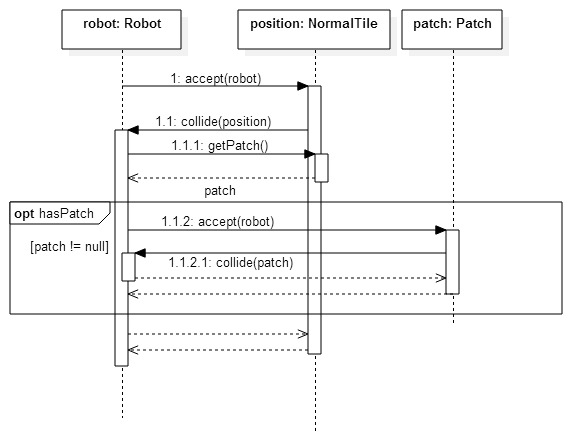
Error! No text of specified style in document.**‑6. diagram Sebesség módosítása (setSpeed)**



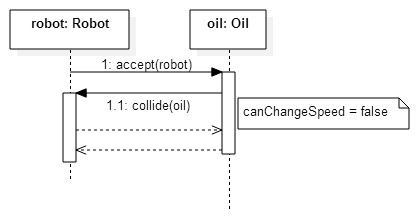
Error! No text of specified style in document.**‑7. diagram Győzelem tesztelése (testWinCondition)**



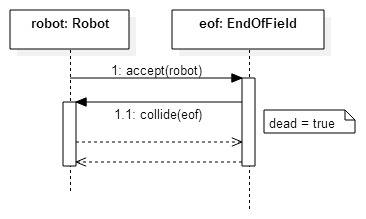
Error! No text of specified style in document.**‑8. diagram Robot ugrása (jump)**



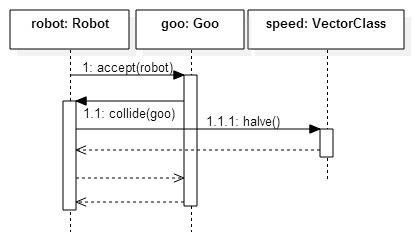
Error! No text of specified style in document.**‑9. diagram NormalTile mezőre lépés**



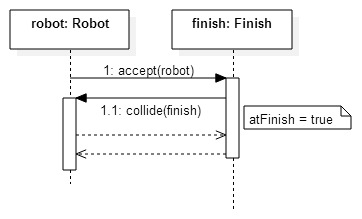
Error! No text of specified style in document.**‑10. diagram Olajfoltra lépés**



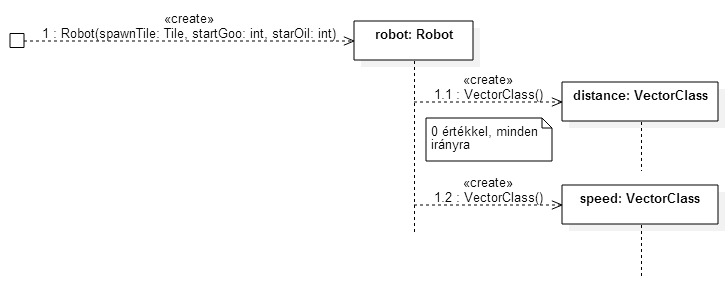
Error! No text of specified style in document.**‑11. diagram EndOfField-re lépés**

****

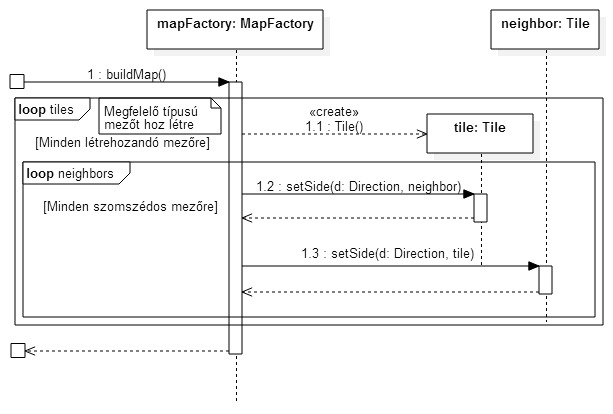
Error! No text of specified style in document.**‑12. diagram Ragacsfoltra lépés**

****

Error! No text of specified style in document.**‑13. diagram Cél mezőre lépés**



Error! No text of specified style in document.**‑14. diagram Robot létrehozása (createRobot)**



Error! No text of specified style in document.**‑15. diagram Pálya felépítése (buildMap)**

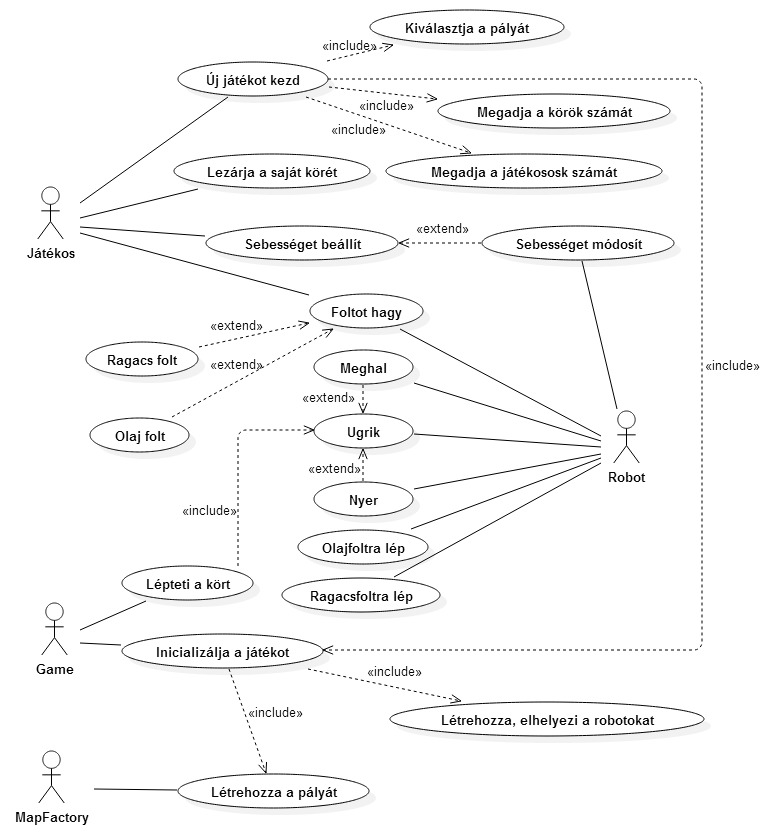
* 1. ***State-chartok***

Robot

## Szkeleton tervezése

## A szkeleton modell valóságos use-case-ei

### Use-case diagram



### Use-case leírások

|  |  |
| --- | --- |
| **Use-case neve** | Új játék kezdése |
| **Rövid leírás** | A felhasználó új játékot indít. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A program indítása után vagy egy másik játékmenet befejezése után a felhasználó a megjelenő opciók közül kiválasztja a játék indításáért felelőset. |

|  |  |
| --- | --- |
| **Use-case neve** | Létrehozza a pályát |
| **Rövid leírás** | A pálya betöltése, felépítése. |
| **Aktorok** | MapFactory |
| **Forgatókönyv** | Az objektum inicializálja a játék indításakor megadott pályát. |

|  |  |
| --- | --- |
| **Use-case neve** | Inicializálja a játékot |
| **Rövid leírás** | A játék kezdőállapotának beállítása. |
| **Aktorok** | Game |
| **Forgatókönyv** | A játék felhasználó általi indítása és a megfelelő paraméterek beállítása után, a pálya és a robotok inicializálása azok alapján. |

|  |  |
| --- | --- |
| **Use-case neve** | Létrehozza, elhelyezi a robotokat |
| **Rövid leírás** | A robotok létrehozása, inicializálása. |
| **Aktorok** | Game |
| **Forgatókönyv** | A beállított játékosszámnak megfelelő számú robotot hoz létre, és külön-külön beállítja azok pozícióját a pályán található kezdőpozíciók egyikére (amely még nem foglalt). |

|  |  |
| --- | --- |
| **Use-case neve** | Ugrik |
| **Rövid leírás** | A robot mozog. |
| **Aktorok** | Robot |
| **Forgatókönyv** | A robot a sebességvektorával megegyező irányba és távolságra ugrik el a jelenlegi pozíciójáról. |

|  |  |
| --- | --- |
| **Use-case neve** | Meghal |
| **Rövid leírás** | A robot meghal. |
| **Aktorok** | Robot |
| **Forgatókönyv** | Amennyiben a robot a pozíciójának megváltoztatása után nem a pályához tartozó mezőre kerül, leesik és meghal, az őt irányító játékos pedig ezzel kiesik a játékból, a robotot a jelenlegi játék alatt többször nem irányíthatja. |

|  |  |
| --- | --- |
| **Use-case neve** | Nyer |
| **Rövid leírás** | A robot beér a célba. |
| **Aktorok** | Robot |
| **Forgatókönyv** | Amennyiben a robot a pozíciójának módosítása után célmezőn tartózkodik az őt irányító játékos nyer, a játék pedig befejeződik. Akkor is ez történik, ha a körök száma elérte a maximum megengedhetőt. Ekkor a robot nyer, ha az összes robot közül ő tette meg a legnagyobb távolságot. |

|  |  |
| --- | --- |
| **Use-case neve** | Lezárja a saját körét |
| **Rövid leírás** | A játékos befejezi a körét. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A játékos a robotjának irányítása után lezárja a körét a megfelelő opció kiválasztásával, ezzel átadva a helyét a következő játékosnak (aki majd a saját robotját irányítja). |

|  |  |
| --- | --- |
| **Use-case neve** | Megadja a játékosok számát |
| **Rövid leírás** | A játékosok számának kiválasztása. ☺ |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A felhasználó megadja a játékosok számát, amely legfeljebb a kiválasztott pályához tartozó maximum lehet. |

|  |  |
| --- | --- |
| **Use-case neve** | Kiválasztja a pályát |
| **Rövid leírás** | A versenypálya kiválasztása. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A felhasználó kiválasztja egy listából azt a pályát, amelyen a többi játékossal szeretne versenyezni. |

|  |  |
| --- | --- |
| **Use-case neve** | Foltot hagy |
| **Rövid leírás** | Adott típusú folt lehelyezése. |
| **Aktorok** | Játékos, Robot |
| **Forgatókönyv** | A játékos a körében kiválasztja, hogy milyen foltot szeretne hagyni a robot alatt lévő mezőn, majd a robot a kör léptetésénél beállítja a mezőn lévő folt típusát ennek megfelelően. Abban az esetben, ha a robot készlete kiürült az adott típusú foltból, nem történik semmi. |

|  |  |
| --- | --- |
| **Use-case neve** | Ragacsfoltot hagy |
| **Rövid leírás** | Ragacs elhelyezése a pályán. |
| **Aktorok** | Játékos, Robot |
| **Forgatókönyv** | A játékos, amely épp a saját robotjának irányítását végzi kiválasztja a ragacs hagyásának megfelelő opciót, majd a robot a kör léptetésénél a robot a pozíciójának megfelelő mezőn lévő folt típusát ragacsra állítja, amennyiben a készlete ebből nem üres. |

|  |  |
| --- | --- |
| **Use-case neve** | Olajfoltot hagy |
| **Rövid leírás** | Olajfolt elhelyezése a pályán. |
| **Aktorok** | Játékos, Robot |
| **Forgatókönyv** | A játékos, amely épp a saját robotjának irányítását végzi kiválasztja a ragacs hagyásának megfelelő opciót, majd a robot a kör léptetésénél a robot a pozíciójának megfelelő mezőn lévő folt típusát olajfoltra állítja, amennyiben a készlete ebből nem üres. |

|  |  |
| --- | --- |
| **Use-case neve** | Lápteti a kört |
| **Rövid leírás** | A program a következő körbe lépteti a játékot. |
| **Aktorok** | Game |
| **Forgatókönyv** | Miután az összes játékos elvégezte a robotjának irányítását egy adott körben, a program minden robotnak utasítást ad az állapotuk és az alattuk lévő mezők frissítésére a játékosok által kiadott parancsoknak megfelelően. Ezután új kör indul, ahol ismét lehetőségük van a robotok irányítására azoknak a játékosoknak, akiknek robotja játékban (életben) maradt. |

|  |  |
| --- | --- |
| **Use-case neve** | Sebességet beállít |
| **Rövid leírás** | A robot sebességének módosítása. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A jelenlegi játékos kiválasztja azt az irányt, amely irány egységvektorával módosítani kívánja a robot sebességét. |

|  |  |
| --- | --- |
| **Use-case neve** | Sebességet módosít |
| **Rövid leírás** | A robot módosítja a saját sebességét. |
| **Aktorok** | Robot |
| **Forgatókönyv** | A robot módosítja a saját sebességét a felhasználó által korábban megadott értékkel, kivéve abban az esetben, ha olyan mezőn tartózkodik, amelyen olajfolt van. |

|  |  |
| --- | --- |
| **Use-case neve** | Olajfoltra lép |
| **Rövid leírás** | A robot olyan mezőre ugrik, amin olajfolt van |
| **Aktorok** | Robot |
| **Forgatókönyv** | Ha a robot pozícióját jelző mezőn olajfolt van a robot sebessége a körben nem változtatható |

|  |  |
| --- | --- |
| **Use-case neve** | Ragacsfoltra lép |
| **Rövid leírás** | A robot olyan mezőre ugrik, amin olajfolt van |
| **Aktorok** | Robot |
| **Forgatókönyv** | Ha a robot pozícióját jelző mezőn olajfolt van a robot sebessége a felére csökken |

|  |  |
| --- | --- |
| **Use-case neve** | Megadja a körök számát |
| **Rövid leírás** | Körök számának megadása |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A felhasználó megadja a játék végéhez szükséges körök számát, amely legfeljebb a kiválasztott pályához tartozó maximum lehet. |

## A szkeleton kezelői felületének terve, dialógusok

A szkeleton felülete parancssorból lesz működtethető. Eleinte egy felsorolás látható, amin a lehetséges bemenetek, és a hozzá tartozó esetek láthatóak. Ebből billentyűzet segítségével lehet kiválasztani a kívánt tesztesetet. Innentől az eset végigmegy, ahol lehet, ott a felhasználó beavatkozását kéri, közben a kimeneten látható, hogy mikor melyik osztály, mely függvénye hívódik meg, és milyen értékkel tér vissza.

Lehetséges tesztesetek:

1. Új Játék indítása

* Létezik-e ilyen pálya? I/N
* Érvényes a játékosok száma? I/N
* Lehetséges ennyi körös játék? I/N

1. Sebesség beállítása

* Volt sebességvektor megadva? I/N
* Módosíthat-e sebességet? I/N

1. Folt hagyása

* Milyen foltot hagyna? Olaj / Ragacs / Semmilyen
* Van-e ilyen folt a raktárban? I/N

1. Ugrás

* A pályán marad a robot? I/N
* Milyen típusú mezőre ugrik a robot? Ragacs / Olaj / JátékTérVége / Cél / Normál

1. Játék vége

* Utolsó kör volt? I/N
* Nyert a robot? I/N
* Meghalt a robot? I/N

1. Robotok létrehozása, elhelyezése

* Érvényes-e az éledő mező? I/N
* Érvényes a kezdő olajfolt szám? I/N
* Érvényes a kezdő ragacsfolt szám? I/N

Példa a második esetre, amikor volt megadva sebességvektor, és tudja módosítani a sebességén, mivel nem olajfoltban áll. A konzolon az alábbi üzenetváltás zajlik le:  
- Adja meg a parancsot:

2: Sebesség beállítása

* robot.modifySpeed(direction)

2.1 Volt sebességvektor megadva? I/N (Igen)

2.2 Módosíthatja a sebességét? I/N (Igen)

🡪speed.add(direction)

🡨speed.add()

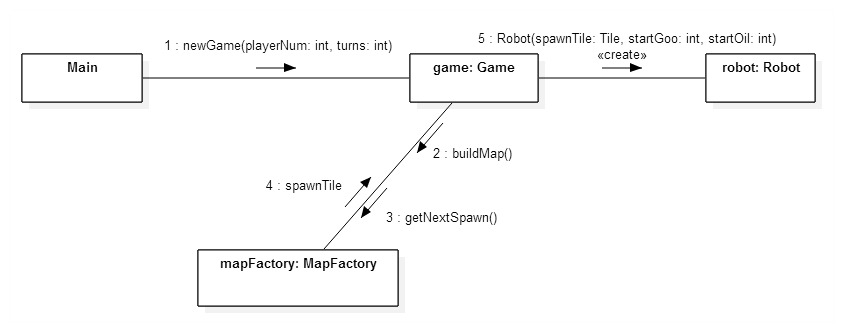
🡸robot.modifySpeed()

* Adja meg a parancsot:

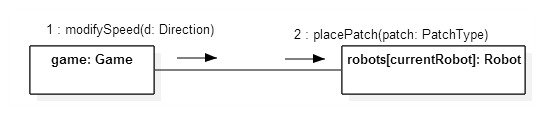
## Szekvencia diagramok a belső működésre

A szekvencia diagramok megegyeznek a 4.4 –ben lévő diagramokkal

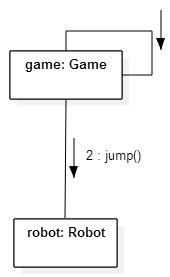
## Kommunikációs diagramok



Error! No text of specified style in document.‑. ábra Új játékot kezd



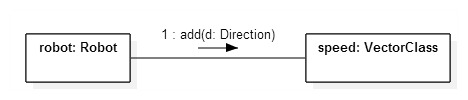
Error! No text of specified style in document.‑. ábra Sebességet beállít



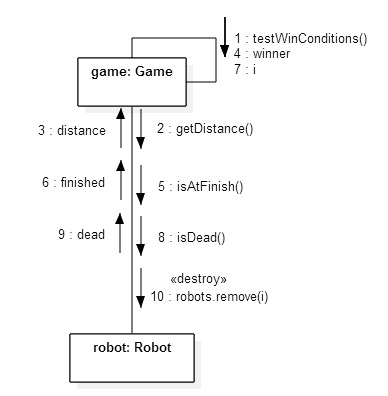
Error! No text of specified style in document.‑. ábra Lépteti a kört



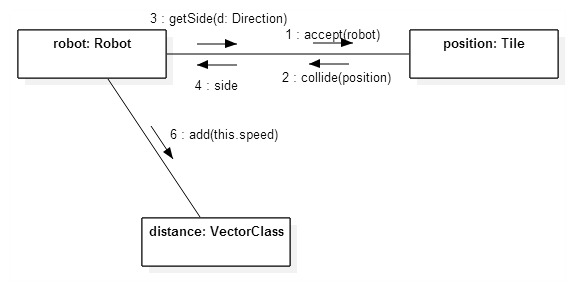
Error! No text of specified style in document.‑. ábra Foltot hagy



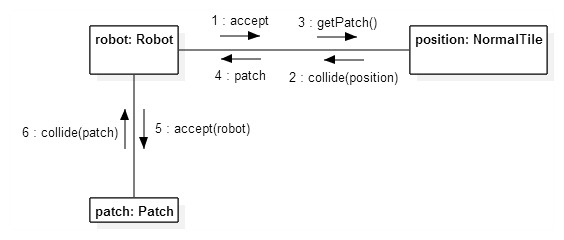
Error! No text of specified style in document.‑. ábra Sebességet módosít



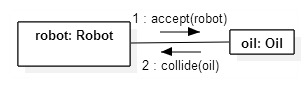
Error! No text of specified style in document.‑. ábra Győzelmi feltétel vizsgálata



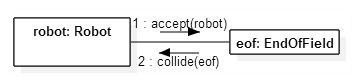
Error! No text of specified style in document.‑. ábra Ugrik



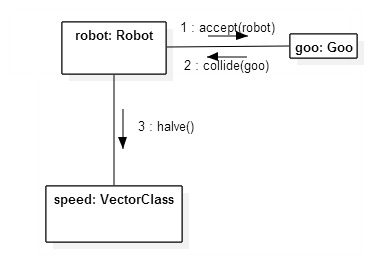
Error! No text of specified style in document.‑. ábra Ugrik 2



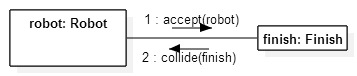
Error! No text of specified style in document.‑. ábra Olajfoltra lép



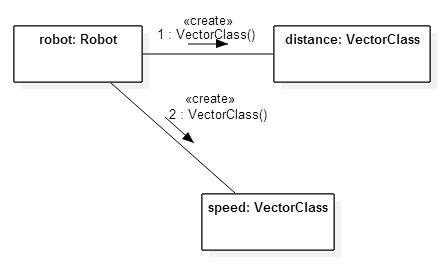
Error! No text of specified style in document.‑. ábra Leesik



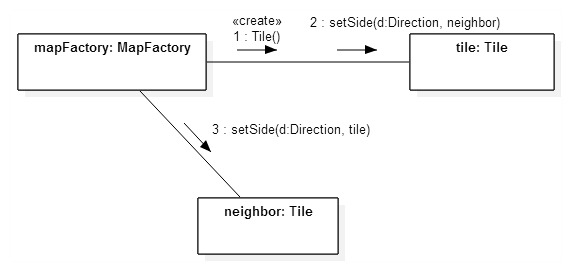
Error! No text of specified style in document.‑. ábra Ragacsfoltra lép



Error! No text of specified style in document.‑. ábra Nyer



Error! No text of specified style in document.‑. ábra Létrehozza a robotokat



Error! No text of specified style in document.‑. ábra Létrehozza a pályát

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2015.03.13. (péntek) 19:00 | 1 óra | Kovács Szabó Dobosy Dankó | Use-case-ek összeszedése, feladatok kiosztása |
| 2015.03.14. (szombat) 17:00 | 1 óra | Dankó | Use-case diagram elkészítése |
| 2015.03.14. (szombat) 20:00 | 2 óra | Szabó | Use-case leírások elkészítése |
| 2015.03.15. (vasárnap) 18:30 | 3 óra | Dobosy | Kommunikációs diagrammok |
| 2015.03.15. 18:30 (vasárnap) | 2 óra | Kovács | Szkeleton terv megírása |
| 2015.03.16. (hétfő) 9:15 | 1 óra | Dankó | Kész dokumentáció összeszerkesztése, hibák kijavítása |