8. Részletes tervek

40 – *[scrum_that]*

Konzulens:
Szabó Ádám Imre

Csapattagok:

Kovács Levente Ákos	CM6UKU	vazul250@gmail.com
Lovász Attila Bence	INCMI7	attonet2@gmail.com
Graics Vince	HY9XQ6	wince17@gmail.com
Magyar Milán Bertalan	MCDNQL	milangfx@gmail.com
Tóth Krisztián Dávid	J38GIK	tht.krisztian@gmail.com

2015. április 6.

Tartalomjegyzék

0. Módosítások

0.0.1. Bemeneti nyelv

- Cleaner <x> <y>
  - Leírás: Cleaner létrehozása a pályán megadott koordinátákon.
  - Opciók: Két koordináta.
- cycles_elapsed <amount>
  - Leírás: Szimulálja, hogy hányat léptek a robotok a játékon belül.
  - Opciók: Lépésszám megadása.
- Glue <x> <y>
  - Leírás: Ragacs létrehozása a pályán.
  - Opciók: A ragacs (x,y) koordinátái.
- keyPressed <akadály_gomb>
  - Leírás: Akadály lerakása a robot koordinátáira.
  - Opciók:
    - akadály_gomb
      - Up: Robot1 olaj lerakás
      - Down: Robot1 ragacs lerakás
      - W: Robot2 olaj lerakás
      - S: Robot2 ragacs lerakás
- keyPressed <irány_gomb> <forgás értéke fokban>
  - Leírás: A Robot következő irányának változtatása.
  - Opciók:
    - irány_gomb
      - Left: Robot1 balra forgatása
      - Right: Robot1 jobbra forgatása
      - A: Robot2 balra forgatása
      - D: Robot2 jobbra forgatása
    - forgás érték = (0,180)
- listCleaners
  - Leírás: A palyán lévő takarító kisrobotok kilistázása.
  - Opciók: -
- listObstacles
  - Leírás: Kilistázza a pályán lévő akadályokat.
  - Opciók: -

- listRobots
  - Leírás: Kilistázza a robotokat.
  - Opciók: -
- MapBuilder
  - Leírás: Pálya betöltése a játékba.
  - Opciók: Pálya kiválasztása.
- move
  - Leírás: Robotok, Cleanerek mozgatása.
  - Opciók: -
- Oil <x> <y>
  - Leírás: Olajfolt létrehozása a pályán.
  - Opciók: Az olajfolt (x,y) koordinátái.
- Phoebe <mode> <count>
  - Leírás:
    Egy teljes értékű játék indítására való parancs. Inicializál két robotot és pár Obstaclet. A játék indulásához szükséges feltételek teszteléséhez szolgáló parancs.
  - Opciók:
    - mode
      - time: Időjátékmódú menet beállítás
      - lap: Körjátékmódú menet beállítása
    - count:
      - Időjátékmód esetén a rendelkezésre álló időt kell megadni másodpercben.
      - Körjátékmód esetén a teljesítendő körök számát kell megadni.
- Robot <x> <y>
  - Leírás: Robot létrehozása a pályán megadott koordinátákon.
  - Opciók: Két koordináta.
- setCheckpoints
  - Leírás: A pálya checkpointjainak a beállítása.
  - Opciók: -
- time_spend <count>
  - Leírás: A játékban nagy szerepet játszik az idő múlása. Ezzel a paranccsal a tesztelő szimulálhatja, hogy bizonyos másodperccel előrelépjen a játék a működésében. Ekkor nem gyorsul fel a játék, csak az órát pörgetjük tovább.
  - Opciók: Az eltelendő idő megadása másodpercben.

1. Részletes tervek

1.1. Osztályok és metódusok tervei

1.2. Objektum katalógus

1.2.1. Glue

A „Glue” objektum megvalósít egy adott tulajdonságú akadályt. Amely robot belemegy, annak a sebességét megfelezi.

1.2.2. GUI

A grafikus felületet megvalósító objektum. Ez az objektum maga a menü, ami a játék indítása után ugrik fel. Itt találhatóak a beállítások (mint például a gondolkodás idő és a maximális játék idő vagy a körök száma) és a játékmódok. Gombnyomásra fogja elindítani a játék működési szálát. Ez az objektum kezeli az ablak eseményeit és a játék bezárását.

1.2.3. HUD

Ez az objektum követi és nyilvántartja, hogy a robotok hány checkpoint-on mentek át, illetve kiírja a képernyőre a hátramaradó időt és a megtett körök számát. Feladata, hogy minden körben megvizsgálja, hogy a robotok elérték-e a következő checkpointot.

1.2.4. MapBuilder

Fájlból beolvassa és létrehozza a memóriában a pályát, a kezdő pozíciókat és a checkpointokat reprezentáló objektumokat. Mivel a MapBuilder objektum tárolja a pályát így feladat, hogy vizsgálja a robotok, akadályok azon belül tartózkodását.

1.2.5. Oil

Ez az objektum az Obstacle osztály leszármazottja. Hasonlóan a Glue objektumhoz, egy adott hatást valósít meg, ami letiltja a következő körben történő irányítását a robotnak, ami belelépett.

1.2.6. Phoebe

A játék logikát megvalósító objektum. Listában tárolja a pályán tartózkodó robotokat, akadályokat és figyeli, hogy mikor ér véget a játék. A „Phoebe” objektum rajzolja ki az objektumokat a pályán és szálként indítható osztályt, melyben maga a játék fut. Játékindításkor berakja a pályára a robotokat és az akadályokat a kezdő pozíciókba. Ebben az objektumban történnek az ellenőrzések (akadályba vagy robotba ütközések, pályáról leesés).

1.2.7. Robot

Olyan objektum, mely a pályán található robotokat valósítja meg. Leírja a viselkedésüket és a kezelésüket. A „Robot” osztály a Unit-ból származik le, ezáltal van pozíciója és az ütközés is le van kezelve. Felelős a mozgásért, megállapítja egy adott akadállyal vagy robottal ütközött-e és kezeli a robot által felhasználható akadálykészleteket, illetve tartalmaz gombnyomást lekezelő metódusokat is.

1.2.8. MyTimer

Az eltelt időt és a fennmaradt idő nyilvántartásáért felelős. Ilyen például a játék elején a három másodperces visszaszámlálás vagy az időlimites játékmód esetén, amikor a maximális időtől számol visszafelé.

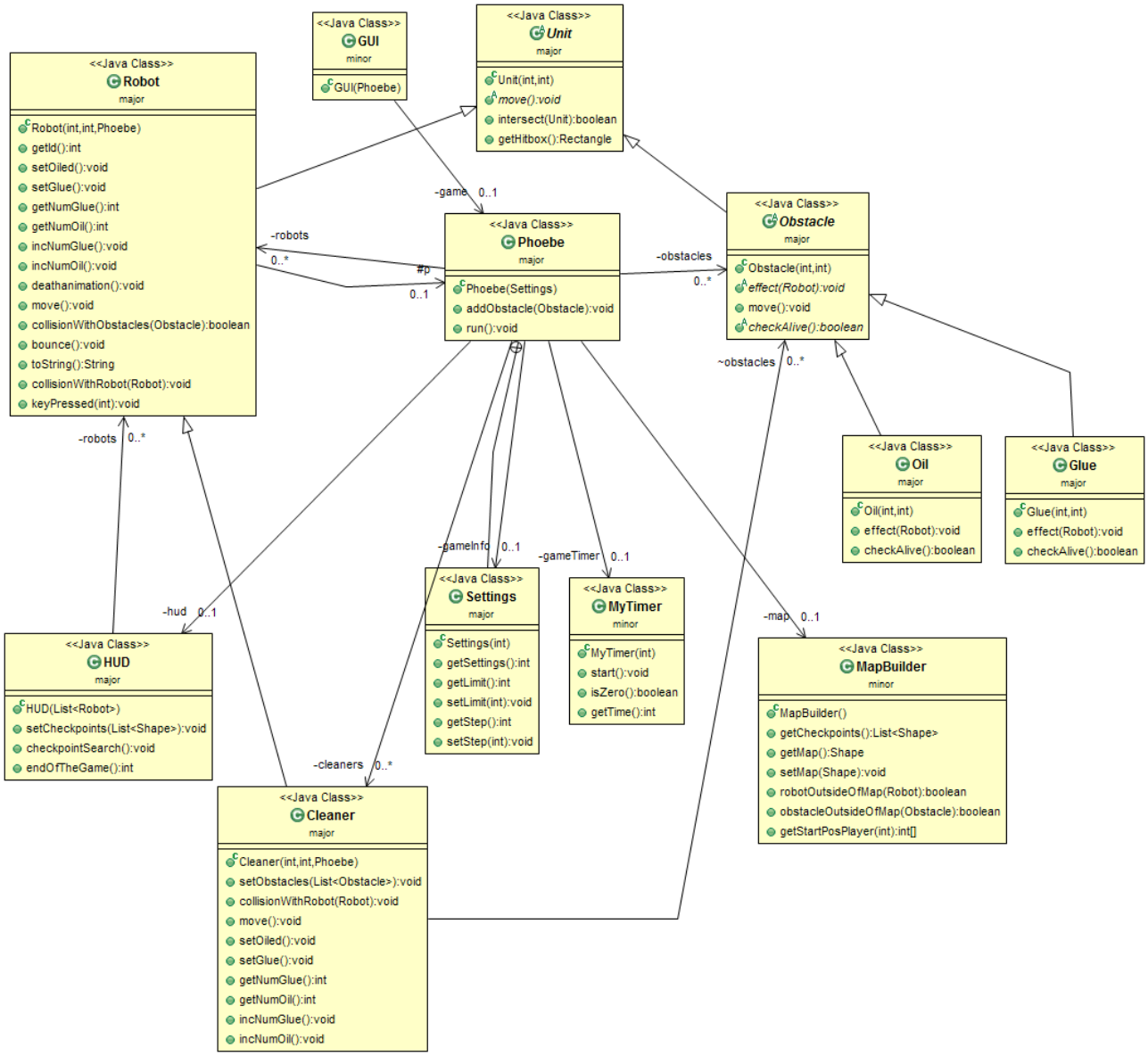
1.2.9. MyListener

A játék keylistener-jét megvalosító osztály. Külön szálon fut, hogy az egyszerre lenyomott gombok ne okozhasanak problémát. A játékban részvevő robotok KeyPressed függvényét hivogatja, a megfelelő KeyEvent paraméterrel.

1.2.10. Cleaner

Takarító robotk melyek a pályán olaj és ragacsfoltokat szednek fel. A robotkkal való ütközés során megsemmisülhetnek.

1.3. Statikus struktúra diagramok



1.1. ábra. Osztály diagram

1.4. Osztályok leírása

1.4.1. Cleaner

- Felelősség
  A takarító robotot reprezentáló osztály.
- Ősosztályok
  Unit→Robot
- Attribútumok
  - List obstacles:

- Metódusok
  - **Cleaner**(int x, int y, Phoebe p):
  - void **setObstacles**(List obsts):
  - boolean **collisionWithRobot**():
  - void **collisionWithCleaner**():
  - void **move**():

1.4.2. GUI

- Felelősség
  A grafikus felületért felelős osztály, amely a menüt és a játékot jeleníti meg.
- Attribútumok
  - **Phoebe** game: referencia a játékra
- Metódusok
  - **GUI**(): Konstruktor. Beállítja az ablak nevét, létrehozza az ablak elemeit, elrendezi őket és beállítja a figyelőket(ActionListener).

1.4.3. HUD

- Felelősség
  A robotok megtett köreit és checkpontjait tartja számon. Megvalósítja a checkpoint ellenőrzést.
- Attribútumok
  - **int[]** checkpointReached: Minden robothoz külön tárolja a legutoljára érintett checkpoint sorszámát.
  - **int[]** lap: Minden robothoz tárolja a megtett körök számát.
  - **List** checkpoints: Tárolja a checkpointokat reprezentáló objektumokat List adatszerkezetben. A checkpointSearch függvény kérdezi le ebből a következő checkpoint helyzetét.
  - **List<Robot>** robots: A robotokat tároló List adatszerkezet. A checkpointSearch függvény kérdezi le ebből a robotokat, majd azok helyzetét.
- Metódusok
  - **HUD**(List<Robot> robs): Robot objektumokat tároló ArrayList. Célja, hogy a checkpointsearch() függvényben minden robotra elvégezzük a keresést.
  - void **checkpointSearch**(): Minden híváskor ellenőrzi, hogy a robot és a checkpoint metszete üres-e.
  - int **endOfTheGame**(): A játék végén eldönti, hogy melyik játékos nyert. Visszatér egy számmal, amiből egyértelműen eldönthető, hogy ki nyert. Ha negatív akkor az 1-es számú játékos nyert, ha nulla akkor döntetlen, ha pozitív akkor a 2-es számú játékos nyert.
  - void **setCheckpointReached**(Robot r): Ha a paraméterként átadott robot következő checkpointja a célvonal (utolsó checkpoint) akkor lenullázza a checkpointReached-et és növeli a megtett körök számát, illetve ha nem akkor növeli az érintett checkpointok számát.
  - void **setCheckpoints**(List checkObj): Checkpointokat reprezentáló adatszerkezet betöltése.CheckpointReached inicializálása a checkpointok számától függően.

1.4.4. IVisible

- Felelősség
  A grafikus motorhoz szükséges interfész. Olyan osztályok, melyek kirajzolható elemeket tartalmaznak megvalósítják ezt az interfészt.
- Metódusok
  - void **paint**(Graphics2D g): Rajzolást elvégző metódus.

1.4.5. List

- Felelősség
  - Objektumok tárolása, ezt az interfészt megvalósító osztályban.
  - `https://docs.oracle.com/javase/6/docs/api/java/util/List.html`

1.4.6. MapBuilder

- Felelősség
  A pálya felépítéséért, a checkpointok tárolásáért és a robot pályán tartózkodásának vizsgálatáért felelős osztály.
- Attribútumok
  - **List** checkpoints: Tárolja a checkpointokat reprezentáló objektumokat List adatszerkezetben.
  - **Object** map: A pályát reprezentáló objektum.
  - **int[]** startPosPlayerOne: Meghatároz egy (x,y) koordinátát, ahol az első játékos kezd.
  - **int[]** startPosPlayerTwo: Meghatároz egy (x,y) koordinátát, ahol az második játékos kezd.
- Metódusok
  - **MapBuilder**(): Konstruktor, a pálya beolvasása fájlból, majd létrehozása.
  - int[] **getStartPosPlayer**(int id): Paraméterül kap egy Robot id-t, majd visszatér egy int tömbbel, melyben található a robot kezdőpozíciója a pályán.
  - boolean **obstacleOutsideOfMap**(Obstacle obs): Egy akadályt vizsgál, hogy a pályán van-e.
  - boolean **robotOutsideOfMap**(Robot r): Igaz értéket ad vissza, ha a robot leesett a pályáról, hamisat ha még rajta van.

1.4.7. MyListener

- Felelősség
  A külön szálon futó KeyListener-t megvalósító osztály.
- Interface-ek:
  - KeyListener
  - Runnable
- Attribútumok
  - -**boolean** isUp: Azt tárolja, hogy le van-e nyomva a felfele nyil.
  - -**boolean** isDown: Azt tárolja, hogy le van-e nyomva a lefele nyil.
  - -**boolean** isRight:Azt tárolja, hogy le van-e nyomva a jobbra nyil.

  - -**boolean** isLeft:Azt tárolja, hogy le van-e nyomva a balra nyil.
  - -**boolean** isW:Azt tárolja, hogy le van-e nyomva a W a billentyűzeten.
  - -**boolean** isD:Azt tárolja, hogy le van-e nyomva a D a billentyűzeten.
  - -**boolean** isS:Azt tárolja, hogy le van-e nyomva az S a billentyűzeten.
  - -**boolean** isA:Azt tárolja, hogy le van-e nyomva az A a billentyűzeten.
  - -**List<Robot>** robots: Azon robotok listája akiknek a keyPressed függvényét kell hívnia.

- Metódusok
  - + **MyListener**(List<Robot> r): beállítja a lenyomott gombokat figyelő változókat false-ra , továbbá beállítja a robots változó referenciáját a paraméterként kapotra.
  - + void**keyPressed**(KeyEvent e): Beállítja a lenyomot gombokat figyelő változók küzül a Keyeventnek megfelelőt true-ra , ha meg nyomták valamelyiket a figyelt gombok közül.
  - + void**keyReleased**(KeyEvent e): Beállítja a lenyomot gombokat figyelő változók küzül a Keyeventnek megfelelőt false-ra , ha fel engedték valamelyiket a figyelt, lenyomott gombok közül.
  - + void**run**(): Végtelen ciklust futtat. Megvizsgálja hogy melyik gombok vannak lenyomva, majd a nekik megfelelő KeyEvent-tel paraméterezve meghívja a hozzátartozó robotoknak a Keypressed függvényét. Majd alszik a szál 30 mili secundomig.

1.4.8. MyTimer

- Felelősség
  A játék elején a kezdésig visszaszámol három másodpercet, utána indulhat a játék. Játéktípustól függően felfelé(kör játékmód) vagy visszafelé(idő játékmód) számol. Ez az osztály felelős, azért ha lejár az idő vége legyen a játéknak,

- Attribútumok
  - - enum DIR: Az óra számolási irányának enumerizácciója.
  - - long T_start: Az óra indításának időpontja millisec pontossággal.
  - - int duration: Ha az óra visszafelé számol, akkor tárolja, hogy mennyi volt a kezdő érték, ha felfelé számol, akkor értéke 0. Mértékegysége millisekundum.
  - - DIR direction: Az óra számolási irányának eltárolásáért felelős enum.

- Metódusok
  - + **MyTimer**(int i): Konstruktor. Ha 0-val vagy negatívval inicializálják felfele számol, ha pozitív számmal inicializálják akkor lefelé számol.
  - + boolean **isZero**(): Az idő lejárását ellenörző függvény, megadja, hogy az indítás plusz a megadott időtartam kisebb-e a pillanatnyi időnél.
  - + int **getTime**(): Ha pozitív száámal inicializálódott az objektum, akkor megadja mennyi idő van még hátra a visszaszámlálásból vagy, ha nullával, akkor a start() hívás óta eltelt idővel tér vissza.
  - + void **start**(): Az óra indításakor vagy újraindításakor meghívott függvény. Csak akkor indul újra (visszaszámláló üzemmódban), ha elérte a 0-t. A Phoebe run() metódusa hívja meg, mikor vissza kell számolni a játék kezdete előtt három másodpercet. Illetve, a játék kezdetekor.

1.4.9. Obstacle

- Felelősség
  A pályán/játékosoknál lévő különböző akadályokat (ragacs,olaj) összefogó ősosztály.
- Ősosztályok
  Unit
- Attribútumok
  - # **int** WIDTH: Az akadályokat jellemző szélesség. Szükség van rá, hogy létrehozzuk a leszármazottak hitbox-át(sokszög pályaelem).
  - # **int** HEIGHT: Az akadályokat jellemző hosszúság. Szükség van rá, hogy létrehozzuk a leszármazottak hitbox-át(sokszög pályaelem).
  - # **int** lifetime: Generikussan az akadályok életben maradásának ellenörzésére szolgál. Olaj esetében megadja, hogy hány kör telt el az akadály lerakása óta. Ragacs esetében pedig, hogy hányan léptek rá mióta lekerült.
- Metódusok
  - +**Obstacle**(int x, int y): meghívja a Unit konstruktorát a megadott adatokkal és létrehoz egy négyzet elemet ami reprezentálja a pályán majd.
  - + void **effect**(Robot r): Meghatározza, milyen hatással van a robotra, ha érintkezik egy Obstacle-lel. Absztrakt.
  - +boolean **checkAlive**():Az akadályok vizsgálata amit a játékmotor minden körben meghív minden akadályra. Absztrakt.

1.4.10. Glue

- Felelősség
  A játékban szereplő Ragacs foltok viselkedését leíró osztály
- Ősosztályok
  Unit→Obstacle
- Attribútumok
  - -**BufferedImage** img: Ez a statikus atribútum a ragacs képét tárolja, a megjelenítésben van szerepe.
- Metódusok
  - **Glue**(int x,int y):A ragacs konstruktora, meghívja az őse (obstacles) konstruktorát x,y paraméterrel, továbbá beállítja a lifetime-ot 4-re.
  - + void **effect**(Robot r): Ütközéskor hívja meg az ütközést vizsgáló függvénye a Robot osztálynak. Módosítja a robot slowed értékét a 50%-ra a robot slowed attribútum setterének meghívásával. Továbbá csökkenti a ragacs élettartalmát egyel.
  - + boolean **checkAlive**():A játékmotor hívja meg minden kör végén, ha a lifetime értéke>0 true-val tér vissza, különben false-al.
  - + void **paint**(Graphics2D g):Kirajzolja a ragacs képét az x,y koordinátákon.
  - + void **setUnitImage**():Beállítja a ragacs osztályhoz tartozó képet a user directoryban található glue.jpg-re
  - + String **toString**()Vissza ad egy stringet a ragacs legfontosabb értékeivel.(x, y, WIDTH, HEIGHT, lifetime)

1.4.11. Oil

- Felelősség
  A pályára lerakható olaj megvalósítása. Ha belelép egy játékos egy ilyen olajfoltba az effect függvény letiltja a mozgatást az adott roboton a következő ugrásig.
- Ősosztályok
  Unit → Obstacle
- Attribútumok
  - - **BufferedImage** img: Ez a statikus atribútum az olaj képét tárolja, a megjelenítésben van szerepe.
- Metódusok
  - + **Oil**(int x, int y): Egy Oil elem létrehozásáért felelős. Meghívja az ős konstruktorát x,y-paraméterrel, továbbá beállítja a lifetime-ot default értékre.
  - + void **effect**(Robot r): Meghatározza, milyen hatással van a robotra, ha beleugrik egy olajfoltba. Ebben az esetben letiltja a játékost, hogy irányt váltson.
  - + void **setUnitImage**():Beállítja az olaj osztályhoz tartozó képet a user directoryban található oil.jpg-re
  - + boolean **checkAlive**():A játékmotor hívja meg minden kör végén, lifetime értékét csökkenti 1 el, ha a lifetime értéke>0 (csökkentés után)true-val tér vissza, különben false-al.
  - + String **toString**()Vissza ad egy stringet az olaj legfontosabb értékeivel.(x, y, WIDTH, HEIGHT, lifetime)

1.4.12. Phoebe

- Felelősség
  A játék motorját megvalósító objektum. Listában tárolja a pályán tartózkodó robotokat,kisrobotokat, akadályokat és figyeli, hogy mikor ér véget a játék. A „Phoebe” objektum rajzolja ki az objektumokat a pályán és szálként indítható osztályt, melyben maga a játék fut. Játékindításkor berakja a pályára a robotokat és az akadályokat a kezdő pozíciókba. Ebben az objektumban történnek az ellenőrzések (akadályba vagy robotba ütközések, pályáról leesés)
- Interface:
  - Runnable
- Attribútumok
  - - **boolean** ended: Állapot változó, ha vége a játéknak, akkor true. Ha beteljesül egy játék végét jelentő esemény, akkor ezen a változón keresztül leáll a játék és megállapítódik a nyertes.
  - + **BufferedImage** background: A játék hátterét adó kép.
  - - **List<Robot>** robots: A játékban szereplő robotok listája.
  - - **List<Obstacle>** obstacles: A játékban szereplő akadályok listája.
  - - **HUD** hud: A játékosok előrehaladását, ragacs és olajkészleteit tartja számon
  - - **MapBuilder** map: TODO
  - - **Settings** gameInfo:A játék beállításait tartalmazza
  - - **List<Cleaner>** cleaners: A játékban lévő aktív kis tisztogató robotokat tartja számon.
  - - **MyTimer** gameTimer: A játékban futó óra, ami visszaszámlálásoknál és a játék végének meghatározásánál játszik szerepet.

- Metódusok
  - + **Phoebe**(Setting set): A játék felépítése, a robotok,a tisztogató kisrobotok és az akadályok listáinak létrehozása.Az ended inicializálása ,az init függvény meghívása és a grafikus felület felépítése történik itt.
  - + void **run**(): Ez a metódus futtatja a főciklust, amelyben maga a játék működik.
  - + void **Paint**(Graphics2D g2d): Kirajzolja a játék aktuális állását.
  - + void **addObstacle**(Obstacle item): Hozzá ad egy Obstacle-t a játékban lévők listájához.
  - - void **init**(): Inicializálja a játékot a kezdeti beállításokra. Létrehozza az időzítőt, a mapot, a robotokat a kezdőpozíciók szerint,a hudot és beállítja a különböző osztályokhoz tartozó statikus képeket, továbbá a pálya alap ragacsait és olajait is szétszórja.

1.4.13. Robot

- Felelősség
  A játékban résztvevő ugráló robotok viselkedését és kezelését leíró osztály, tárolja és kezeli a felhasználható akadályok számát. Olyan objektum, mely a pályán található robotokat valósítja meg. Leírja a viselkedésüket és a kezelésüket. A „Robot" osztály a Unit-ból származik le, ezáltal van pozíciója és az ütközés is le van kezelve. Felelős a mozgásért, megállapítja egy adott akadállyal vagy robottal ütközött-e és kezeli a felhasználó által leütött gombokat.
- Ősosztályok
  Unit
- Attribútumok
  - # **int** staticID: Az osztályhoz tartozó statikus azonosító, a példány azonosítójának(id) meghatározásához szükséges.
  - - **static final int** r: Az ugrás számításához tartozó sugár.
  - - **static final int** ANIMATIONSPEED: Az ugrás animálásának részletessége(hányszor hívja meg a paintet).
  - # **static int** HEIGHT: A robot képének magassága, collision detektálásnál, továbbá az irányítást segítő nyíl kezdő koordinátájának meghatározásánál szükséges.
  - # **static int** WIDTH:A robot képének szélessége, funkcionalitásban hasonló a WIDTH-hez.
  - - **int** ID: A robot példányának egyedi azonosítója, a keyconfig sorának indexelésére és a collison detektálásnál az önmagával való ütközés kivédésére szükséges.
  - - **int** numGlue:A robotnál lévő ragacskészletet tárolja.
  - - **int** numOil:A robotnál lévő olajkészletet tárolja.
  - - **boolean** leftobstacle:Megmondja hogy raktunk-e már le ebben a körbe olajat vagy ragacsot kezdő érték false, minden lépés után vissza áll false ra és minden obstacle lerakásnál true ra .
  - # **BufferedImage**img[]:A robotok képeit tartalmazza,az animáció miatt többet.
  - - **double** slowed: A sebesség modosításáért felel, default értéje 1.0, amennyiben ragacsba lép a robot ez 0.5-re módosul és minden ugrás végén visszaáll az eredeti értékére, ugrásnál ezzel szorozzuk be a végkordinátát kiszámító sugár hosszát.
  - - **boolean** oiled: Azt jelzi, hogy olajba lépett-e, ennek hatására a mozgás iránya módosíthatatlanná válik egy kis időre.
  - # **int** arrowendx: A robot irányítását segítő nyilnak az x koordinátája, a nyíl kirajzolásánál van szerepe.

  - # **int** arrowendy: A robot irányítását segítő nyilnak az y koordinátája, a nyíl kirajzolásánál van szerepe.
  - - **double** alpha: A robot irányítását segítő nyíl vízszintessel bezárt szöge. A nyil kirajzolásánál, az ugrás végpontjának meghatározánál van szerepe.
  - # **boolean** moved: Azt jelöli, hogy lépett-e már a robot az aktuális körben. A megjelenítésnél(nyilat ugrás közben nem jelenítjük meg),illetve az irányítás letiltásánál van szerepe(olajba lépés esetén).

- Metódusok

  - + **Robot**(int x,int y,Phoebe p): Létrehoz egy robotot a megadott x,y kordinátákon, inicializálja a tagváltozóit és eltárolja a játékmotor referenciáját.
  - + void**deathanimation**():A Robot halálának gafikus megjelenítéséért felelős függvény.
  - + void**setOiled**():Az oiled értékét true-ra állítja.
  - + void**setGlued**():A slowed értékét 0.5-re állítja.
  - + int**getId**():A Robot id-ét adja vissza.
  - + int**getNumGlue**():Visszatér a felhasználható ragcsok számával.
  - + int**getNumOil**():Visszatér a felhasználható olajok számával.
  - + void**incNumOil**():Növeli a robotnál tárolt olajok számát, ha az nem haladja meg a 3 at.
  - + void**incNumGlue**():Növeli a robotnál tárolt ragacsok számát, ha az nem haladja meg a 3 at.
  - + void**paint**(Graphics2D g):kirajzolja a robotot a saját koordinátáin, ha nem lép éppen akkor az irányítást segítő nyilat is.
  - + void**setUnitImage**():beállítja a robot osztályhoz tartozó képeket.
  - + void**bounce**():A robotok ütközésekor a lepattanás céjának koordinátáinak számolása történik itt.
  - + void**move**():A robot mozgatásáért és annak leanimálásáért felelős függvény.Kiszámolja az új koordinátát és oda ugrasztja a robotot, majd frissíti a hitboxot.
  - + boolean **collisionWithObstacle**(Obstacle o): Ellenőrzi hogy a robot ütközött-e az akadállyal, igazzal tér vissza ha igen, hamissal ha nem.
  - + boolean **collisionWithRobot**(Robot r): Ellenőrzi hogy a robot ütközött-e másik robottal , referencia alapján kiszüri ha önmagára hívják meg.Ha volt ütközés meghívja a bounce függvényt önmagára.Igazzal tér vissza ha volt ütközés és hamissal ha nem.
  - + void **keyPressed**(int e):A robot irányítását megvalósító függvény, a játékmotor keylistener-e által hívódik meg, a lenyomott billentyű azonosítójával. A következő ugrás beállítása, a ragacs/olaj lerakása történhet itt. A Settings.keyconfig változó felhasználásával.
  - + String **toString**()Vissza ad egy stringet a robot legfontosabb értékeivel.(id, slowed, oiled, x, y, nextx, nexty, alpha, WIDTH, HEIGHT, numGlue, numOil)

1.4.14. Unit

- Felelősség
  A pályán található objektumokért felel és azok viszonyáról (például ütközésükről).

- Attribútumok
  - # **Object** hitbox: Az egységet a pályán reprezentáló sokszög.
  - # **int** x: Az egység x koordinátája

  - # **int** y: Az egység y koordinátája

- Metódusok
  - + **Unit**(): A Unit osztály konstruktora. Feladata, hogy eltárolja az x,y koordinátát.
  - + boolean **intersect**(Unit u): Két egység ütközését meghatározó függvény.
  - + void **move**() : Absztrakt függvény, mely a leszármazottakban fog megvalósulni. Az egységek mozgásáért felelős.

1.5. A tesztek részletes tervei, leírásuk a teszt nyelvén

1.5.1. CollisionWithRobot_VOLTÜTKÖZÉS_TESZT

- Leírás: Ez a teszt a collisionWithRobot(Robot r) függvény teszteléséért felelős. Létrehoz 2 robotot majd beállítja az őket, úgy hogy egymásra ugorjanak . Kiirja az adataikat és meghivja a move függvényt ezt követően pedig az egyik collisionWithRobot függvényét a másikra és újból ki listáza őket.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük hogy sikeressen össze tudta-e vetni a hitboxokat a függvény az ugrás végeztével . Várható hiba ha hamissal tér vissza a függvény, amiből látjuk, hogy rossz a hitbox létrehozása az új helyen az ugrást követően.

- Bemeneti nyelv :
  - Robot(500,500)
  - Robot(600,400)
  - Keypressed(Keyevent.VK_A,90)
  - listRobots
  - move
  - listRobots

- Elvárt kimenet:
  "Robot [id=0, slowed=1,oiled=false, x=500,y=500,
  numGlue=3,numOil=3,nextx=500,
  nexty=400,alpha=1.57,width=40,height=40]"
  "Robot [id=1, slowed=1,oiled=false, x=600,y=400,
  numGlue=3,numOil=3,nextx=600,
  nexty=300,alpha=1.57,width=40,height=40]"

  "nextx ,nexty modified to:500,400"

  "Robot [id=0, slowed=1,oiled=false, x=500,y=500,
  numGlue=3,numOil=3,nextx=500,
  nexty=400,alpha=1.57,width=40,height=40]"
  "Robot [id=1, slowed=1,oiled=false, x=600,y=400,
  numGlue=3,numOil=3,nextx=500,
  nexty=400,alpha=1.57,width=40,height=40]"

  "there was a collision between this: "Robot [id=0, slowed=1,oiled=false, x=500,y=400,
  numGlue=3,numOil=3,nextx=500,
  nexty=400,alpha=1.57,width=40,height=40]"
  and this: "Robot [id=1, slowed=1,oiled=false, x=500,y=400,
  numGlue=3,numOil=3,nextx=500,
  nexty=150,alpha=3.04,width=40,height=40]"

  "Robot [id=0, slowed=1,oiled=false, x=500,y=400,
  numGlue=3,numOil=3,nextx=500,
  nexty=400,alpha=1.57,width=40,height=40]"

  "Robot [id=1, slowed=1,oiled=false, x=500,y=400,
  numGlue=3,numOil=3,nextx=500,
  nexty=150,alpha=3.04,width=40,height=40]"

1.5.2. CollisionWithRobot_NEMVOLTÜTKÖZÉS_TESZT

- Leírás: Ez a teszt a collisionWithRobot(Robot r) függvény teszteléséért felelős. Létrehoz 2 robotot . Kiirja az adataikat és meghivja a move függvényt ezt követően pedig az egyik collisionWithRobot függvényét a másikra és újból ki listáza őket.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük hogy sikeressen össze tudta-e vetni a hitboxokat a függvény az ugrás végeztével . Várható hiba ha loggol collisiont a függvény, amiből látjuk, hogy rossz a hitboxok össze vetése , hisz nem lehet ütközés.

- Bemeneti nyelv :
  - Robot(500,500)
  - Robot(600,400)
  - listRobots
  - move
  - listRobots

- Elvárt kimenet:
  "Robot [id=0, slowed=1,oiled=false, x=500,y=500,
  numGlue=3,numOil=3,nextx=500,
  nexty=400,alpha=1.57,width=40,height=40]"
  "Robot [id=1, slowed=1,oiled=false, x=600,y=400,
  numGlue=3,numOil=3,nextx=600,
  nexty=300,alpha=1.57,width=40,height=40]"

  "Robot [id=0, slowed=1,oiled=false, x=500,y=400,
  numGlue=3,numOil=3,nextx=500,
  nexty=400,alpha=1.57,width=40,height=40]"
  "Robot [id=1, slowed=1,oiled=false, x=600,y=300,
  numGlue=3,numOil=3,nextx=600,
  nexty=300,alpha=1.57,width=40,height=40]"

1.5.3. CollisionWithRobot_IRÁNYVÁLTOZTATÁS_TESZT

- Leírás: Ez a teszt a KeyPressed(int e) függvény teszteléséért felelős. Létrehoz 2 robotot. Kiirja az adataikat és meghivja a KeyPressed(int e) függvényt VK_D és VK_LEFT paraméterekkel majd meghívjuk a move-ot és újból ki listáza őket. A várt eredmény, hogy 180 fokban elfordultak és úgy léptek.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük hogy sikeressen ki tudta e számolni a függvény az új koordinátákat. Várható hiba hogy rosszul modosítja a cél koordinátákat vagy az alpha szöget. Ezt onnan látjuk, hogy nem a várt koordinátákat látjuk a nextx, nexty, alpha tagváltozókban miután másodjára is kilistáztuk őket vagy mikor a keypressed kiirja.

- Bemeneti nyelv :
  - Robot(500,500)
  - Robot(600,400)
  - listRobots
  - Keypressed(Keyevent.VK_A,180)

  - Keypressed(Keyevent.VK_LEFT,180)
  - move
  - listRobots

- Elvárt kimenet:
  "Robot [id=0, slowed=1,oiled=false, x=500,y=500,
  numGlue=3,numOil=3,nextx=500,
  nexty=600,alpha=1.57,width=40,height=40]"
  "Robot [id=1, slowed=1,oiled=false, x=600,y=400,
  numGlue=3,numOil=3,nextx=600,
  nexty=500,alpha=1.57,width=40,height=40]"

  "nextx ,nexty modified to:500,400"
  "nextx ,nexty modified to:600,300"

  "Robot [id=0, slowed=1,oiled=false, x=500,y=400,
  numGlue=3,numOil=3,nextx=500,
  nexty=400,alpha=4.71238898,width=40,height=40]"
  "Robot [id=1, slowed=1,oiled=false, x=600,y=300,
  numGlue=3,numOil=3,nextx=600,
  nexty=300,alpha=4.71238898,width=40,height=40]"

1.5.4. CollisionWithObstacles_OLAJBA.UGRÁS_TESZT

- Leírás: Ez a teszt a robot és az olajfolt ütközésének érzékeléséért felelős. Létrehoz egy robotot és egy olajfoltot a megfelelő koordinátákon, kiírja az adatokat, majd meghívja a move-ot, és újból kilistáza őket. A várt eredmény, hogy volt ütközés.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük, hogy a robot a megfelelően lerakott olajba ugrik-e, és ezt érzékeli is. Várható hiba, hogy a robot nem a megfelelő helyre ugrik, vagy nem érzékeli, hogy olajfoltba ugrott.

- Bemeneti nyelv :
  - Robot(0,0)
  - Oil(0,100)
  - listRobots
  - listObstacles
  - move

- Elvárt kimenet:
  "Robot [id=0, slowed=1,oiled=false, x=0,y=0,
  numGlue=3,numOil=3,nextx=0,
  nexty=100,alpha=1.57,width=40,height=40]"
  "Oil[x=0, y=100, Width=40, Height=40, remainingturns=15"

  "there was a collision between this: "Robot [id=0, slowed=1,oiled=false, x=0,y=100,
  numGlue=3,numOil=3,nextx=0,

nexty=100,alpha=1.57,width=40,height=40]"
and this: "Oil[x=0, y=100, Width=40, Height=40, remainingturns=15"

1.5.5. CollisionWithObstacles_RAGACSBA.UGRÁS_TESZT

- Leírás: Ez a teszt a robot és a ragacs ütközésének érzékeléséért felelős. Létrehoz egy robotot és egy ragacsot a megfelelő koordinátákon, kiírja az adatokat, majd meghívja a move-ot, és újból kilistáza őket. A várt eredmény, hogy volt ütközés.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük, hogy a robot a megfelelően lerakott ragacsba ugrik-e, és ezt érzékeli is. Várható hiba, hogy a robot nem a megfelelő helyre ugrik, vagy nem érzékeli, hogy ragacsba ugrott.

- Bemeneti nyelv :
  - Robot(0,0)
  - Glue(0,100)
  - listRobots
  - listObstacles
  - move

- Elvárt kimenet:
  "Robot [id=0, slowed=1,oiled=false, x=0,y=0,
  numGlue=3,numOil=3,nextx=0,
  nexty=100,alpha=1.57,width=40,height=40]"
  "Glue[x=0, y=100, Width=40, Height=40, remainingturns=15"

  "there was a collision between this: "Robot [id=0, slowed=1,oiled=false, x=0,y=100,
  numGlue=3,numOil=3,nextx=0,
  nexty=100,alpha=1.57,width=40,height=40]"
  and this: "Glue[x=0, y=100, Width=40, Height=40, remainingturns=15"

1.5.6. CollisionWithObstacles_OLAJ.HATÁSA_TESZT

- Leírás: Ez a teszt az olajfolt hatásának érvényességéért felelős. Létrehoz egy robotot és egy olajfoltot a megfelelő koordinátákon, kiírja az adatokat, meghívja az effect-et, majd megpróbál irányt változtatni, hogy beleugorjon a következő olajba. A várt eredmény, hogy lesz ütközés.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük, hogy az olaj hatássa kihat-e a robotra. Várható hiba, hogy a robot irányát tudjuk változtatni.

- Bemeneti nyelv :
  - Robot(0,0)
  - Oil(0,0)
  - Oil(0,100)
  - listRobots
  - listObstacles
  - effect
  - Keypressed(Keyevent.VK_A,90)
  - move

- Elvárt kimenet:
  "Robot [id=0, slowed=1,oiled=false, x=0,y=0,
  numGlue=3,numOil=3,nextx=0,
  nexty=100,alpha=1.57,width=40,height=40]"
  "Oil[x=0, y=0, Width=40, Height=40, remainingturns=15"
  "Oil[x=0, y=100, Width=40, Height=40, remainingturns=15"

  "you jumped into oil"

  "there was a collision between this: "Robot [id=0, slowed=1,oiled=true, x=0,y=100,
  numGlue=3,numOil=3,nextx=0,
  nexty=100,alpha=1.57,width=40,height=40]"
  and this: "Oil[x=0, y=100, Width=40, Height=40, remainingturns=15"

1.5.7. CollisionWithObstacles_RAGACS.HATÁSA_TESZT

- Leírás: Ez a teszt a ragacs hatásának érvényességéért felelős. Létrehoz egy robotot és egy ragacsot a megfelelő koordinátákon, kiírja az adatokat, meghívja az effect-et, majd a move-ot, hogy beleugorjon a következő ragacsba. A várt eredmény, hogy nem lesz ütközés.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük, hogy a ragacs hatássa kihat-e a robotra. Várható hiba, hogy a robot elugrik a ragacsig.

- Bemeneti nyelv :
  - Robot(0,0)
  - Glue(0,0)
  - Glue(0,100)

- listRobots
- listObstacles
- effect
- move
- listRobots

"Robot [id=0, slowed=1,oiled=false, x=0,y=0,
numGlue=3,numOil=3,nextx=0,
nexty=100,alpha=1.57,width=40,height=40]"
"Glue[x=0, y=0, Width=40, Height=40, remainingturns=4"
"Glue[x=0, y=100, Width=40, Height=40, remainingturns=4"

"you jumped into glue"
"Robot [id=0, slowed=0.5,oiled=false, x=0,y=0,
numGlue=3,numOil=3,nextx=0,
nexty=100,alpha=1.57,width=40,height=40]"

1.5.8. robotOutsideOfMap_A.PÁLYÁRÓL.LEESETT_TESZT

- Leírás:
  Ez a teszt a robotOutsideOfMap(Robot r) függvény teszteléséért felelős. Létrehoz 1 robotot majd beállítja, úgy hogy kiugorjon a pályáról . Kiirja az adatait és meghivja a move függvényt ezt követően pedig a robotOutsideOfMap függvényt és leellenőrizzük hogy a listában maradt-e a robot.

- Ellenőrzött funkcionalitás, várható hibahelyek:
  Azt ellenörizzük hogy sikeressen össze tudta-e vetni a hitboxokat a függvény az ugrás végeztével . Várható hiba ha hamissal tér vissza a függvény, amiből látjuk, hogy rossz a hitbox létrehozása az ugrást követően.

- Bemeneti nyelv :
  - MapBuilder map
  - Robot(400,500)
  - listRobots
  - Keypressed(Keyevent.VK_A,90)
  - move
  - listRobots

- Elvárt kimenet:
  "Robot [id=0, slowed=1,oiled=false, x=400,y=500,
  numGlue=3,numOil=3,nextx=400,
  nexty=400,alpha=1.57,width=40,height=40]"

  "nextx ,nexty modified to:400,400"

  "Robot [id=0, slowed=1,oiled=false, x=400,y=500,
  numGlue=3,numOil=3,nextx=400,

nexty=400,alpha=1.57,width=40,height=40]"

"there was a collision between this: "Robot [id=0, slowed=1,oiled=false, x=500,y=400,
numGlue=3,numOil=3,nextx=500,
nexty=400,alpha=1.57,width=40,height=40]"
and this: "Map"

1.5.9. robotOutsideOfMap_NEM.ESETT.LE.PÁLYÁRÓL_TESZT

- Leírás: Ez a teszt a robotOutsideOfMap(Robot r) függvény teszteléséért felelős. Létrehoz 1 robotot majd beállítja, úgy hogy ne ugorjon ki a pályáról . Kiirja az adatait és meghivja a move függvényt ezt követően pedig a robotOutsideOfMap függvényt és leellenőrizzük hogy a listában maradt-e a robot.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük hogy sikeressen össze tudta-e vetni a hitboxokat a függvény az ugrás végeztével . Várható hiba ha igazzal tér vissza a függvény, amiből látjuk, hogy rossz a hitbox létrehozása az ugrást követően.

- Bemeneti nyelv :
  - MapBuilder map
  - Robot(400,500)
  - listRobots
  - Keypressed(Keyevent.VK_D,90)
  - move
  - listRobots

- Elvárt kimenet:
  "Robot [id=0, slowed=1,oiled=false, x=400,y=500,
  numGlue=3,numOil=3,nextx=400,
  nexty=600,alpha=1.57,width=40,height=40]"

  "nextx ,nexty modified to:400,600"

  "Robot [id=0, slowed=1,oiled=false, x=400,y=500,
  numGlue=3,numOil=3,nextx=400,
  nexty=600,alpha=1.57,width=40,height=40]"

1.5.10. checkpointSearch_CHEICKPOINTONBA.UGRÁS_TESZT

- Leírás: Ez a teszt a checkpointSearch() függvény teszteléséért felelős. Létrehoz 1 robotot majd beállítja, úgy hogy ugorjon bele a checkpointba. Kiirja az adatait és meghivja a move függvényt ezt követően pedig a checkpointSearch függvényt és leellenőrizzük hogy van-e a két hitboxnak metszette.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük hogy sikeressen össze tudta-e vetni a hitboxokat a függvény az ugrás végeztével . Várható hiba ha hamissal tér vissza a függvény, amiből látjuk, hogy rossz a hitbox létrehozása az ugrást követően.

- Bemeneti nyelv :
  - MapBuilder map
  - MapBuilder Map
  - setCheckpoints 100,180
  - Robot(100,100)
  - listCheckpoints
  - listRobots
  - Keypressed(Keyevent.VK_D,90)
  - move
  - listRobots

- Elvárt kimenet:
  "Checkpoint: 100,180"

  "Robot [id=0, slowed=1,oiled=false, x=100,y=100,
  numGlue=3,numOil=3,nextx=100,
  nexty=200,alpha=1.57,width=40,height=40]"

  "nextx ,nexty modified to:100,200"

  "there was a collision between this: Robot [id=0, slowed=1,oiled=false, x=100,y=200,
  numGlue=3,numOil=3,nextx=100,
  nexty=200,alpha=1.57,width=40,height=40]
  and this: Checkpoint: 100,180"

  "Robot [id=0, slowed=1,oiled=false, x=100,y=200,
  numGlue=3,numOil=3,nextx=100,
  nexty=200,alpha=1.57,width=40,height=40]"

1.5.11. checkpointSearch_CHEICKPOINTONBA.NEM.UGRÁS_TESZT

- Leírás: Ez a teszt a checkpointSearch() függvény teszteléséért felelős. Létrehoz 1 robotot majd beállítja, úgy hogy ne ugorjon bele a checkpointba. Kiirja az adatait és meghivja a move függvényt ezt követően pedig a checkpointSearch függvényt és leellenőrizzük hogy van-e a két hitboxnak metszette.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenörizzük hogy sikeressen össze tudta-e vetni a hitboxokat a függvény az ugrás végeztével . Várható hiba ha igazzal tér vissza a függvény, amiből látjuk, hogy rossz a hitbox létrehozása az ugrást követően.

- Bemeneti nyelv :
  - MapBuilder map
  - MapBuilder Map
  - setCheckpoints 100,180
  - Robot(100,300)
  - listCheckpoints
  - listRobots
  - Keypressed(Keyevent.VK_D,90)
  - move
  - listRobots

- Elvárt kimenet:
  "Checkpoint: 100,180"

  "Robot [id=0, slowed=1,oiled=false, x=100,y=300,
  numGlue=3,numOil=3,nextx=100,
  nexty=400,alpha=1.57,width=40,height=40]"

  "nextx ,nexty modified to:100,400"

  "Robot [id=0, slowed=1,oiled=false, x=100,y=400,
  numGlue=3,numOil=3,nextx=100,
  nexty=400,alpha=1.57,width=40,height=40]"

1.5.12. RobotCollisionWithCleaner_TESZT

- Leírás
  A teszt azt az esetet akarja szimulálni, amikor egy robot egy takarító robot felé halad, miközben az éppen dolgozik. Az a kimenetel lesz tesztelve, hogy a takarító megsemmisül és a robot zavartalanul tovább halad.

- Ellenőrzött funkcionalitás, várható hibahelyek
  Várhatóan a takarító megsemmisül, olajfoltot hátrahagyva maga után. A robotra nincs hatással az ütközés, de a takarító maga után hagyott olaj ellehetetleníti a robot irányváltoztatását. Lehetséges hibák a következőek: a takarító nem takarítja fel a foltot vagy feltakarítja, de nem tűnik el a pályáról, emellett lehetséges, hogy nem semmisül meg a takarító robot.

- Bemenet
  Glue 100,300
  Robot 100,100
  Cleaner 200,300

  listObstacles
  listRobots
  listCleaners

  move

  listRobots
  listCleaners

  move

  listObstacles
  listRobots
  listCleaners

- Elvárt kimenet

  - Glue [x=100, y=300, Width=30, Height=30, Lifetime=4]
  - Robot [id=0, slowed=1, oiled=false, x=100, y=100, nextx=100, nexty=200, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3]
  - Cleaner[x=200, y=300, Width=30, Height=30, State=MOVING]
  - there was a collision between this: Cleaner[x=100, y=300, Width=30, Height=30, State=MOVING] and this: Glue [x=100, y=300, Width=30, Height=30, Lifetime=4]

  - Robot [id=0, slowed=1, oiled=false, x=100, y=200, nextx=100, nexty=200, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3]
  - Cleaner[x=100, y=300, Width=30, Height=30, State=WORKING]
  - there was a collision between this: Cleaner[x=100, y=300, Width=30, Height=30, State=WORKING] and this: Robot [id=0, slowed=0.5, oiled=false, x=100, y=300, nextx=100, nexty=300, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3]
  - This: Cleaner[x=100, y=300, Width=30, Height=30, State=WORKING] died
  - Oil[x=100, y=300, Width=60, Height=60, remainingturns=15]
  - there was a collision between this: Robot [id=0, slowed=1, oiled=false, x=100, y=300, nextx=100, nexty=300, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3] and this: Glue [x=100, y=300, Width=30, Height=30, Lifetime=4]
  - you jumped into glue
  - there was a collision between this: Robot [id=0, slowed=0.5, oiled=false, x=100, y=300, nextx=100, nexty=300, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3] and this:
  - Oil[x=100, y=300, Width=60, Height=60, remainingturns=15]
  - you jumped into oil

  - Glue [x=100, y=300, Width=30, Height=30, Lifetime=3]
  - Oil[x=100, y=300, Width=60, Height=60, remainingturns=15]
  - Robot [id=0, slowed=0.5, oiled=true, x=100, y=300, nextx=100, nexty=300, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3]

1.5.13. Initialisation_Test

- Leírás
  A játék indulását hivatott tesztelni. A pálya betöltésétól kezdve az egyes elemek létrehozását(Robotok, akadályok,...) hivatott tesztelni.
- Ellenőrzött funkcionalitás, várható hibahelyek Az elemek létrehozása, esetleges kivételek kezelése. Lehetséges probléma, hogy nem tudja betölteni a pályát vagy a checkpointokat vagy esetleges nem várt kivételekkel terminálódik a futás.
- Bemenet
  Phoebe time 300

  listRobots
  listObstacles
  listCleaners

  move

- Elvárt kimenet

  - Robot [id=0, slowed=1, oiled=false, x=100, y=100, nextx=100, nexty=200, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3]
  -
  - Robot [id=1, slowed=1, oiled=false, x=300, y=100, nextx=300, nexty=200, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3]
  - Glue [x=<random>, y=<random>, Width=30, Height=30, Lifetime=4]
  - Glue [x=<random>, y=<random>, Width=30, Height=30, Lifetime=4]
  - Glue [x=<random>, y=<random>, Width=30, Height=30, Lifetime=4]
  - Oil[x=<random>, y=<random>, Width=60, Height=60, remainingturns=15]
  - Oil[x=<random>, y=<random>, Width=60, Height=60, remainingturns=15]
  - Oil[x=<random>, y=<random>, Width=60, Height=60, remainingturns=15]

1.5.14. GameEndWithTimeElapsing_Test

- Leírás
  A játék végét szimuláló teszt. Azt az esetet szimulálja, hogy letelik az idő játék futása közben.
- Ellenőrzött funkcionalitás, várható hibahelyek Egy játék inicializálás után átugrunk időben a játék végére és szimuláljuk, hogy egy utolsó ugrást tesznek a robotok. Várható hibalehetőségek: nem várt kivétel, a győztes kihirdetése nem valósul meg.
- Bemenet
  Phoebe time 60

  listRobots
  listObstacle
  listCleaners

  time_spend 60
  move
- Elvárt kimenet
  - Robot [id=0, slowed=1, oiled=false, x=100, y=100, nextx=100, nexty=200, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3]
  - Robot [id=1, slowed=1, oiled=false, x=300, y=100, nextx=300, nexty=200, alpha=1.5708, width=40, height=40, numGlue=3, numOil=3]
  - Glue [x=<random>, y=<random>, Width=30, Height=30, Lifetime=4]
  - Glue [x=<random>, y=<random>, Width=30, Height=30, Lifetime=4]
  - Glue [x=<random>, y=<random>, Width=30, Height=30, Lifetime=4]
  - Oil[x=<random>, y=<random>, Width=60, Height=60, remainingturns=15]
  - Oil[x=<random>, y=<random>, Width=60, Height=60, remainingturns=15]
  - Oil[x=<random>, y=<random>, Width=60, Height=60, remainingturns=15]
  - EndOfTheGame

1.5.15. AddObstacle_AKADÁLY_LERAKÁS_TESZT

- Leírás: Ez a teszt az akadály lerakás teszteléséért felelős. Létrehoz egy robotot, majd négyszer meghívja a KeyPressed(int e) függvényt VK_UP paraméterrel. A várt eredmény, hogy az első három esetben sikerül az olaj lerakása, míg a negyedikben nem, hiszen a robot kezdetben három olajjal rendelkezik.
- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenőrizzük, hogy a megfelelő gomb megnyomásakor az akadály tényleges létrejön-e, illetve hogy egy akadály lerakása után csökken-e a robotnál található akadályok száma. Várható hiba, hogy az akadály létrejöttéről nincs visszajelzés. Emellett az is hiba, ha az akadály a negyedik alkalommal is létrejön, vagyis nincs "Not enough oil" jelzés, ami azt jelenti, hogy nem csökkent a robot akadályainak száma.
- Bemeneti nyelv :
  Robot 100,200
  keyPressed UP
  keyPressed UP

keyPressed UP
keyPressed UP

- Elvárt kimenet:
  new oil created at:100,200
  new oil created at:100,200
  new oil created at:100,200
  Not enough oil

1.5.16. ObstacleLife_AKADÁLY_ÉLETTARTAM_TESZT

- Leírás: Ez a teszt az akadályok élettartamát vizsgálja, hogy eltűnnek-e bizonyos kör lefutása után. A várt eredmény, hogy az akadályok eltűnnek.

- Ellenőrzött funkcionalitás, várható hibahelyek: Az akadályok élettartamát ellenőrizzük. Várható hiba, hogy az akadályok megmaradnak.
- Bemeneti nyelv :
  - Glue(0,0)
  - Oil(0,100)
  - listObstacles
  - cycles_elapsed(4)
  - listObstacles
  - cycles_elapsed(11)
  - listObstacles
- Elvárt kimenet:
  "Glue[x=0, y=0, Width=40, Height=40, remainingturns=4"
  "Oil[x=0, y=100, Width=40, Height=40, remainingturns=15"
  "Oil[x=0, y=100, Width=40, Height=40, remainingturns=11"

1.5.17. Cleaner_TAKARÍTÓ_KISROBOT_MOZGÁS_TAKARÍTÁS_TESZT

- Leírás: Ez a teszt a takarító robot mozgását és takarítását ellenőrzi. Létrehozunk egy takarító robotot, majd két akadályt különböző koordinátákon. Meghívjuk a move metódust és megnézzük, hogy a közelebbi akadályhoz ment-e a takarító robot. Mivel egy foltot három lépés alatt takarít fel, ezért meghívjuk háromszor a move metódust és megnézzük, hogy az akadály fel lett-e takarítva.

- Ellenőrzött funkcionalitás, várható hibahelyek: Azt ellenőrizzük, hogy a takarító robot odatalál-e a hozzá legközelebbi akadályhoz és hogy a takarítással törli-e az akadályok listájából. Lehetséges hiba, hogy a takarító robot rossz akadályhoz megy, ami abból látszik, hogy a move meghívása után a robot a tőle eredetileg távolabbi olajfolt koordinátáin van. Egy másik hiba, ha a háromszori move hívás után a kiválasztott olajfolt nincs feltakarítva, azaz megjelenik az akadályok listázása után.

- Bemeneti nyelv :
  Cleaner 100,200
  Oil 100,300
  Oil 100,400
  listCleaners
  listObstacles
  move
  listCleaners
  move
  move
  move
  listObstacles

- Elvárt kimenet:
  "Cleaner [x=100,y=200,nextx=100,nexty=200,alpha=1.57,width=30,height=30]"
  "Oil [x=100, y=300, Width=40, Height=40, lifetime:15]"
  "Oil [x=100, y=400, Width=40, Height=40, lifetime:15]"
  "Cleaner [x=100,y=300,nextx=100,nexty=300,alpha=1.57,width=30,height=30]"
  "Oil [x=100, y=400, Width=40, Height=40, lifetime:12]"

1.6. A tesztelést támogató programok tervei

A tesztelő elindítja a teszteket, amik fájlba írják kimenetüket. Ezután a TotalCommander program segítségével (Fájl->Összehasonlítás tartalomra...) összehasonlítja a produkált és elvárt kimenetet. Ha megegyezik a kettő, akkor a teszt sikeres.

1.7. Napló

Kezdet	Időtartam	Résztvevők	Leírás
2015.03.31. 13:00	0,5 óra	**Kovács** **Lovász** **Tóth** **Graics** **Magyar**	Konzultáció, részfeladatok kiosztása
2015.03.31. 18:00	1 óra	**Tóth**	Tevékenység: MyTimer implementálás, dokumentálás
2015.04.01. 08:15	1 óra	**Tóth**	Konzultáció
2015.04.01. 16:00	1 óra	**Graics**	Tesztelés előkészítése
2015.04.01. 17:00	2 óra	**Kovács**	"Osztályok Leirása" frissítése, kód apró módosítása
2015.04.06. 18:00	2 óra	**Kovács** **Lovász** **Tóth** **Graics** **Magyar**	Tesztesetek kidolgozása