

21 - fantaziadus

Konzulens:
Hartung István

Csapattagok:

Kozaróczy Zsolt	A2VCE0	zsolt.kozaroczy@gmail.com
Molnár Tamara	DM5NDP	moflas@gmail.com
Papp Győző	EGMY7Z	pappgyozo@gmail.com
Tóth Ákos	Q7X72Y	dxmiamme@gmail.com
Varga Éva Árnika	RPMHZZ	arnikaeva@gmail.com

(2013.05.17)

1. Tartalomjegyzék

- 1. Tartalomjegyzék
- 2. Követelmény, projekt, funkcionalitás
 - 2.1. Bevezetés
 - 2.1.1 Cél
 - 2.1.2 Szakterület
 - 2.1.3 Definíciók, rövidítések
 - 2.1.3.1 Követelményazonosítók megnevezése
 - 2.1.4 Hivatkozások
 - 2.1.5 Összefoglalás
 - 2.2 Áttekintés
 - 2.2.1 Általános áttekintés
 - 2.2.2 Funkciók
 - 2.2.3 Felhasználók
 - 2.2.4 Korlátozások
 - 2.2.5 Feltételezések, kapcsolatok
 - 2.3 Követelmények
 - 2.3.1 Funkcionális követelmények
 - 2.3.3 Átadással kapcsolatos követelmények
 - 2.3.4 Egyéb nem funkcionális követelmények
 - 2.4 Lényeges use-case-ek
 - 2.4.1 Use-case leírások
 - 2.4.2 Use-case diagram
 - 2.5 Szótár
 - 2.6 Projekt terv

2.6.1 Lépések és Határidők

2.6.2 Felelősök

2.6.3 Használt eszközök

2.7 Napló

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Ant

3.1.2 AntHill

3.1.3 Antlion

3.1.4 AntScent

3.1.5 AntScentVanisherScent

3.1.7 Echidna

3.1.8 Food

3.1.9 FoodScent

3.1.10 Snail

3.1.11 Slime

3.2. Statikus struktúra diagramok

3.3. Osztályok leírása

3.3.1. Ant

3.3.2. AntHill

3.3.3. Antlion

3.3.4. AntScent

3.3.5. AntScentVanisherScent

3.3.6. AntScentVanisherScent

3.3.7. Barrier

3.3.8. Director

3.3.9. Echidna

3.3.10. Food

3.3.11. FoodScent

3.3.12. HexCell

3.3.13. IAcceptor

3.3.14. IInterActor

3.3.15. ITimed

3.3.16. Located

3.3.17. NonBlocking

3.3.18. Obstacle

3.3.19. Pebble

3.3.20. Poison

3.3.21. PoisonSpray

3.3.22. Puddle

3.3.23. Scent

3.3.24. Slime

3.3.25. Snail

3.3.26. Spray

3.3.27. Wanderer

3.4. Szekvencia diagramok

3.4.1. Ant::die

3.4.2. Ant::move

3.4.3. Ant::tick

3.4.4. Ant::interactWith(Food)

3.4.5. Director::initialize

3.4.6. PoisonSpray::doSpraying

3.4.7. Puddle Slows

3.4.8. ScentVasisherSpray::doSpraying

3.4.9. Wanderer::move

3.4.10. Director::tick()

3.4.11. AntScentVanisherScent::tick()

*

3.4.12. Ant::interactWith(Poison)

3.4.13. Ant::interactWith(Pebble)

3.5. State-chartok

3.5.1 Ant életciklusa

3.5.2 Echidna életciklusa

3.6. Napló

4. Analízis modell kidolgozása II

4.1. Napló

5. Szkeleton tervezése

5.1. A szkeleton modell valóságos use-case-ei

5.1.1. Use-case diagram

5.1.2. Use-case leírások

5.2. A szkeleton kezelői felületének terve, dialógusok

5.2.1 Szkeleton bemenetek

5.2.2 Kimenetek a megfelelő bemenetekre

5.3. Szekvencia diagramok a belső működésre

5.4. Kommunikációs diagramok

5.5 Napló

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

6.1.2 Fordítás

6.1.3 Futtatás

6.2 Értékelés

6.3 Napló

7. Prototípus koncepciója

7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

7.1.2 Bemeneti nyelv

7.1.3 Kimeneti nyelv

7.2 Összes részletes use-case

7.3 Tesztelési terv

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

7.4.1 logView

7.4.2 Cél-ellenőrző

7.5 Napló

8. Részletes tervek

8.1 Osztályok és metódusok tervei

8.1.1 Ant

8.1.2 AntHill

8.1.3 Antilon

8.1.4 AntScent

8.1.5 AntScentVanisherScent

8.1.6 AntScentVanisherSpray

8.1.7 Barrier

8.1.8 Director

- 8.1.9 Echidna
- 8.1.10 Food
- 8.1.11 FoodScent
- 8.1.12 HexCell
- 8.1.13 IAcceptor
- 8.1.14 IInterActor
- 8.1.15 ITimed
- 8.1.16 LoaderSaver
- 8.1.17 Located
- 8.1.18 Logger
- 8.1.19 Main
- 8.1.20 NonBlocking
- 8.1.21 Obstacle
- 8.1.22 Pebble
- 8.1.23 Poison
- 8.1.24 PoisonSpray
- 8.1.25 Puddle
- 8.1.26 Scent
- 8.1.27 SingletonException
- 8.1.28 Slime
- 8.1.29 Snail
- 8.1.30 Spray
- 8.1.31 Wanderer

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

- 8.2.1 Étél felvétele és leszállítása
- 8.2.2 Hangyalesőbe lépés

8.2.3 Hangyászsün hangyaevése

8.2.4 Hangyászsün hangyaevése II

8.2.5 Kavicsok mozgása

8.2.5 Kavicsok mozgása II

8.2.6 Méreg terjedése

8.2.7 Hangyaszag semlegesítése

8.3 A tesztelést támogató programok tervei

8.3.1 logView

8.3.2 ResultChecker

8.4 Napló

10. Prototípus

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

10.1.2 Fordítás

10.1.3 Futtatás

10.2 Tesztek jegyzőkönyvei

10.2.1 Beolvasás

10.2.2 Étél felvétele és leszállítása

10.2.2 Étél felvétele és leszállítása

10.2.3 Étél felvétele és leszállítása

10.2.4 Hangyászsün hangyaevése

10.2.5 Hangyászsün hangyaevése

10.2.6 Hangyászsün hangyaevése II

10.2.7 Kavicsok mozgása

10.2.8 Kavicsok mozgása II

10.2.9 Kavicsok mozgása II

- 10.2.10 Méreg terjedése
- 10.2.11 Méreg terjedése
- 10.2.12 Méreg terjedése
- 10.2.13 Hangyaszag semlegesítése
- 10.2.14 Hangyalesőbe lépés
- 10.2.15 Hangyalesőbe lépés

10.3 Értékelés

10.4 Napló

11. Grafikus felület specifikációja

11.1 A grafikus interfész

- 11.1.1 Start Screen
- 11.1.2 Field Screen
- 11.1.3 End Screen
- 11.1.4 Felhasznált képek

11.2 A grafikus rendszer architektúrája

- 11.2.1 A felület működési elve
- 11.2.2 A felület osztály-struktúrája

11.3 A grafikus objektumok felsorolása

- 11.3.1 Controller
- 11.3.4 Visualizer
- 11.3.5 Visualizer.Images
- 11.3.6 Visualizer.MapCell

11.5 Napló

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

- 13.1.1 Fájllista

13.1.2 Fordítás és telepítés

13.1.3 Futtatás

13.2 Értékelés

13.3 Napló

2. Követelmény, projekt, funkcionalitás

2.1. Bevezetés

2.1.1 Cél

A játék teljes specifikációjának elkészítése a megrendelőnek.

2.1.2 Szakterület

A Szoftver laboratórium 4. BME-VIK-es tantárgy feladatának megoldását adja a szoftver.

2.1.3 Definíciók, rövidítések

BME: Budapesti Műszaki és Gazdaságtudományi Egyetem

HSZK: Hallgatói Számítógép Központ

IIT: Irányítástechnikai és Informatika Tanszék

JDK: Java Development Kit, a Java fejlesztői környezete és API-ja.

JRE: Java Runtime Environment, a Java futtatókörnyezet.

m²: SI mértékegység, az 1 méter x 1 méter oldalhosszú négyzet területének megfelelő egység.

SzoftLab4: Szoftver Laboratórium 4.

2.1.3.1 Követelményazonosítók megnevezése

AAG: Átadás Alapvető General

EAG: Erőforrás Alapvető General

ENK: Egyéb Nem funkcionális Követelmény

FAT: Funkcionális Alapvető Task

FFT: Funkcionális Fontos Task

FOF: Funkcionális Opcionális Feature.

2.1.4 Hivatkozások

<http://hu.wikipedia.org/wiki/N%C3%A9gyzetm%C3%A9ter>

2.1.5 Összefoglalás

A dokumentum célja a játék teljes specifikációjának elkészítése a megrendelőnek. A kapcsolódó szakterület, hogy a program játékként lesz használható az örülés/m² növelésére.

A dokumentum további részeiben a következő fejezetek találhatóak:

- 2.2 Áttekintés: A project terveinek, funkcióinak és korlátozásainak áttekintése, felhasználók lehetőségeinek meghatározása
- 2.3 Követelmények: A project során elvárt követelmények kidolgozása, külön részletezve a funkcionális, erőforrás, átadással kapcsolatos és egyéb nem funkcionális követelményeket.
- 2.4 Lényeges use-case-ek: A lényeges use-case-ek felsorolása és ábrázolása
- 2.5 Szótár: A project során bevezetett fogalmak részletesebb körülírása.
- 2.6 Project terv: A project megvalósításával kapcsolatos feladatkörök és határidők részletes kifejtése
- 2.7 Napló: Az elvégzett feladatok és ráfordított idők felsorolása

2.2 Áttekintés

2.2.1 Általános áttekintés

A fontosabb alrendszerek:

- felhasználói felület
- játék menü
- pályageneráló
- pályatároló (tárolja a rajta lévő élőlényeket és foltokat)
- spray rendszer
- az idő léptetéséért és az élőlények mozgatásáért felelős rendszer (director)
- étel, idő és spray szint tároló és visszajelző rendszer (HUD)

A felhasználó közvetlenül a grafikus felhasználói felülettel áll kapcsolatban, ezen keresztül fér hozzá a játék menühöz és a spray rendszerhez.

A pályageneráló a pályatárolót inicializálja, közvetlen hozzáféréssel rendelkezik hozzá.

A spray rendszer a pályatárolóhoz foltot tud hozzáadni és eltávolítani onnan.

A felhasználói felület hozzáfér a játék menühöz, a pályatárolóhoz, a HUD-hoz, hiszen ki kell jeleznie azokat.

A director a pályatárolót módosítja a játék szabályainak függvényében.

A program nem igényel sem hálózati kapcsolatot sem kitüntetett adatbázist.

2.2.2 Funkciók

A feladat egy olyan számítógépes játék elkészítése, amiben a játékos célja az ételraktárainak megvédése a hangyáktól. A játékpályán található egy hangyaboly, a játék nehézsége alapján előre meghatározott számú ételraktár, a hangyák útját nehezítő különböző akadályok, illetve a hangyászsün és a hangyalesők, amik a játékost segítik.

A méreteket a következőképpen határoztuk meg, ha a hangya 1 egység méretű: az ételeket 3 egységről érzi először, a szagnyomokat 1 egység távrolól érzi. A szagok intenzitása növekszik a forráshoz közeledve. A hangyaboly 1 egység nagyságú, a hangyaleső tölcsére pedig 1 egység átmérőjű. Az akadályok változó formájúak és nagyságúak. A hangyászsün 1 egység nagyságú, de a megjelenített képe attól függ, mennyit evett már.

A hangyák célja, hogy a hangyabolyból kiindulva megszerezzék az összes élelmet, ami a játékpályán található, majd ezt visszajuttassák a bolyba. A hangyák az idő előrehaladtával egyre rövidebb időközönként jönnek elő a bolyból, egyesével.

Az ételraktárakban előre meghatározott mennyiségű egységnyi étel van. Az ételegységeket a hangyák, ha elég közel vannak hozzá, megérik. Egy hangya egyszerre csak egy ételt tud magával vinni. Ha egy hangya a halála pillanatában ételt vitt, az élelem ott marad a pályán, így egy másik hangya később felveheti és viheti tovább.

A hangyák alapvetően egyenesen haladnak, azonban véletlenszerű időközönként és a következőkben meghatározott okok miatt irányt válthatnak. Egyrészt, ha egy hangyánál nincs étel, de érez egy pályán lévő élelmet, akkor a szag intenzitásának nagysága alapján meghatározott valószínűséggel fordul az adott élelem irányába. Ezenkívül a hangyák szagnyomokat is hagynak maguk mögött a haladásuk során ami az élelemhez hasonlóan befolyásolja őket az útirány meghatározásánál. A hangyák szagnyoma az idő előrehaladtával folyamatosan veszít intenzitásából és egyre kevésbé befolyásolja a hangyák irányát. Amennyiben a hangyánál már van élelem, akkor célja ennek visszajuttatása a hangyabolyba, ezért ilyen esetben ez határozza meg a választott útirányt.

A pályán vannak rögzített akadályok. Ezek lehetnek különböző alakú kavicsok és tócsák. A kavicsokat a hangyáknak ki kell kerülniük, nem tudnak át menni rajtuk. A tócsák lassítják a rajtuk áthaladó hangyákat. A kavicsokat a hangyászsünek el tudják tolni. A játék indításakor véletlenszerűen lesznek lerögzítve a pályán az akadályok, mennyiségük a játék nehézségétől függően, elhelyezésük pedig annak figyelembevételével, hogy a hangyák legalább egy úton eljuthassanak a hangyabolyból minden ételraktárig. Ha egy hangya nekimenne egy akadálnak, akkor irányt vált a hangyák kanyarodási szabályait figyelembe véve.

A hangyáknak vannak természetes ellenségei is, ezeket a játékos nem tudja befolyásolni. Ők a játékost segítik, hiszen megeszik a hangyákat. Két fajtájuk van, a hangyászsün és a hangyaleső. A hangyászsün véletlenszerű időközönként megjelenik a pályán és áthalad rajta, eközben megeszik maximum 8 hangyát. A hangyaleső folyamatosan a pályán tartózkodik a tölcsérében, ami véletlenszerűen elhelyezett rögzített helyen van. Ha a hangya a tölcsérbe téved, a hangyaleső

megegy. Ha a hangyánál volt ekkor étel, akkor az ételt is megegy a hangyaleső, viszont egy új ételt helyezünk el valahova a pályán. A hangyászsun nem eszi meg a hangyánál lévő ételt.

A játékosnak rendelkezésére áll kétfajta segítő eszköz. Ezek egyike a hangyaszag-semlegesítő spray, amivel a hangyák által hátrahagyott szagnyomokat tudja eltüntetni egy adott sugarú körben. A másik eszköz a hangyairtó spray, amivel mérgező foltot tud spriccelni a pálya egy területére. A mérgező foltba érve a hangyák meghalnak, az esetlegesen náluk lévő ételt ott hagyják. A mérgező folt idővel elillan. Mindkét eszközből véges mennyiségű áll a játékos rendelkezésére. Ha elfogy, a játékos többé már nem tud beavatkozni a játékba.

A játékos célja az ételraktárak megóvása minél hosszabb ideig. Ezt egy, a játék kezdésekor elindított óra méri. A játék addig tart, amíg van a pályán élelem, amit a hangyák be tudnának vinni a hangyabolyba, tehát a játék abban a pillanatban véget ér, hogy egy hangya az utolsó élelemmel ér vissza a hangyabolyba.

2.2.3 Felhasználók

A felhasználó csak és kizárólag egy játékos módban tud a programmal játszani. A játékban nem lesz pályaszerkesztő. A játék menetét nem lehet elmenteni, mindig újból kell kezdeni.

2.2.4 Korlátozások

A BME IIT SzoftLab4. tárgyának oktatói (a megrendelők) által kiírt specifikáció megköveteli, hogy a program a HSZK gépein fusson, ahol a már előre feltelepített JRE 1.7-es környezetben kell futnia, ez alapján korlátozás, hogy csak a JDK 1.7-es verziójában már létező függvények, osztályok használhatóak a kód megírása során.

2.2.5 Feltételezések, kapcsolatok

A rendszer modellezéséhez szükséges biológiai ismeretek:

http://en.wikipedia.org/wiki/Short-beaked_echidna

<http://en.wikipedia.org/wiki/Antlion>

<http://en.wikipedia.org/wiki/Ant>

Az ételek valóságos megjelenítése és felhasználása alapjául szolgáló ismeretanyag:

http://hu.wikipedia.org/wiki/Magyar_gaszon%C3%B3mia

2.3 Követelmények

Az egyes prioritások mértéke:

- **alapvető:** a specifikált feladat teljesítéséhez, illetve a kész program futásához feltétlenül szükséges feladat. A teljesítés hiánya *végzetes* a projektre nézve.
- **fontos:** a kibővített specifikációban meghatározott, a “megrendelő” által nem kőbe vésett funkció, amelyet a korrekt működés érdekében szükséges megvalósítani. A teljesítés hiányában a projekt nem teljesen működőképes, így *kockázatos* a projektre nézve.
- **opcionális:** olyan különleges funkciók, amelyek nem szükségesek a specifikált program elvárt működéséhez. Ezeknek a követelményeknek a nem teljesítése *jelentéktelen* a projektre nézve.

Az egyes követelmények lehetséges forrásai:

- **specifikáció:** a megrendelő által specifikált paraméterei az alkalmazásnak. Az összes specifikációs követelmény alapvető, mivel enélkül a feladat biztosan nem teljesül.
- **csapat:** a csapat által támasztott egyéb elvárások a programmal szemben.
- **konzulens:** a konzulens által külön megkövetelt elvárások.

2.3.1 Funkcionális követelmények

Azo-nosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
FAT001	Játék menüje	kiértékelés	alapvető	csapat	Beállítások kezelése, Játék indítása	
FAT002	Beállítások kezelése	kiértékelés	alapvető	csapat	Beállítások kezelése	
FAT003	Játék elindítása	kiértékelés	alapvető	specifikáció	Játék indítása	
FAT004	Hangyák viselkedése	kiértékelés	alapvető	specifikáció	Étel keresése	
FAT005	Spray használat	kiértékelés	alapvető	specifikáció	Spray-k használata	
FAT006	Hangyászünök viselkedése	kiértékelés	alapvető	specifikáció	Kóricálás	
FAT007	Akadályok generálása	kiértékelés	alapvető	specifikáció	Játék indítása	
FAT008	Hangyaesők viselkedése	kiértékelés	alapvető	specifikáció		
FAT009	Játék végének ellenőrzése	kiértékelés	alapvető	specifikáció		
FFT010	Méretek felülvizsgálata	kiértékelés	fontos	csapat		
FFT011	Hangya elakadásvizsgálat	kiértékelés	fontos	csapat		
FOF012	Legyen benne pingvin	kiértékelés	opcionális	csapat		

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
EAG001	1 GB RAM	kiértékelés	alapvető	csapat	
EAG002	1 GHz CPU órajel	kiértékelés	alapvető	csapat	
EAG003	JRE 1.6 / frissebb	kiértékelés	alapvető	csapat	
EAG004	Perifériák	kiértékelés	alapvető	csapat	Input: egér Output: monitor

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
AAG001	JRE ellenőrzés	kiértékelés	alapvető	csapat	szükség esetén telepítés

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
ENK001	Kis méretű futtatható állomány (.jar)	kiértékelés	alapvető	csapat	Kicsi a pendriveom
ENK002	Tesztelési feladatok	tesztelés	fontos	specifikáció	

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Beállítások kezelése
Rövid leírás	A játékos a beállítások menüpontot nyitja meg.
Aktorok	Felhasználó
Forgatókönyv	A pálya paramétereinek változtatására alkalmas képernyő jelenik meg.

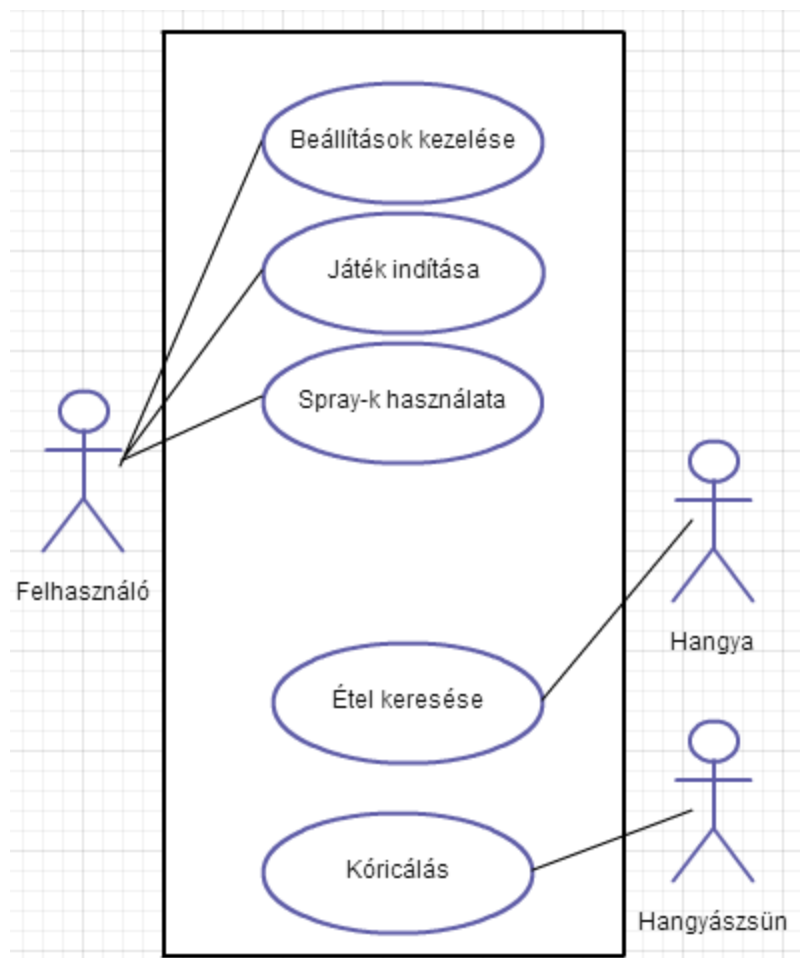
Use-case neve	Játék indítása
Rövid leírás	A játékos egy új játék indítását kezdeményezi.
Aktorok	Felhasználó
Forgatókönyv	A megfelelő függvények meghívásával létrejön és megjelenik a pálya.

Use-case neve	Spray-k használata
Rövid leírás	A játékos utasítást ad a spray-k használatára.
Aktorok	Felhasználó
Forgatókönyv	A spray-nek megfelelő hatás megtörténik a célterületen.

Use-case neve	Étel keresése
Rövid leírás	Egy hangya belép a játékba.
Aktorok	Hangya
Forgatókönyv	A hangya azonnal élelmiszer keresésébe kezd.

Use-case neve	Kóricálás
Rövid leírás	Egy hangyászsün belép a játékba.
Aktorok	Hangyászsün
Forgatókönyv	A hangyászsün véletlenszerűen elkezd a terep bejárását.

2.4.2 Use-case diagram



2.5 Szótár

Akadály: A pályán előforduló, a hangyák mozgását akadályozó elemek.

Elillanás: A pályán lévő szagnyom illetve méregfolt idővel való elhalványulása, majd eltűnése.

Élőlény: A hangyák és hangyászsün gyűjtőneve, ők a játék szereplői.

Étel: Az ételraktárakból viszik a hangyák a hangyabolyba, egyszerre egy egységet. Ha a pályáról eltűnik az összes étel, a játék véget ér.

Ételraktár: Az ételek kezdő pozíciója. A hangyáknak ide kell eljutniuk az ételekért, majd az ételeket visszajuttatni a hangyabolyba.

Folt: Idővel elillanó anyag.

Hangyaboly: A hangyák kiinduló pozíciója, innen indulnak élelmet keresni és ide térnek vissza a hangyák a megtalált ételekkel.

Hangyairtó spray: Egy kör alakú méregfoltot helyez el a pályán a kijelölt helyre, ami végez a rajta áthaladó hangyákkal. A méregfolt megadott idő elteltével elillan.

Hangyák: Élőlények, céljuk az ételraktárakból az ételek elhordása a hangyabolyba.

Hangyaleső: A pályán elhelyezkedő speciális akadály, amihez ha a hangya elegendően közel kerül, a hangya eltűnik a pályáról. A hangyánál lévő étel is eltűnik.

Hangyaszag-semlegesítő spray: Speciális spray ami a bevetése helyén azonnal eltünteti a hangyák által hagyott szagnyomot.

Hangyászsün: A játékban időnként megjelenő élőlény, ami néhány hangyát eltüntet a pályáról. Amikor elfogyaszt egy hangyát, növekszik a mérete. Ha elérte a végső méretét, mindenképp elhagyja a pályát, addig a pályán kóricál.

Kavics: Akadály, a hangyáknak meg kell kerülniük. Különböző méretűek lehetnek.

Kóricálás: A hangyászsün által végzett tevékenység. Mászkálás a pályán, közben hangya evés.

Megrendelő: A tárgyfelelős és labor konzulens.

Méregfolt: Folt. A hangyairtó spray okozza. Az áthaladó hangyákat megöli rögtön.

Pálya: a képernyőn megjelenített terület, amin a játék folyik.

Spray: A felhasználó eszközei, ezekkel tudja befolyásolni a játék menetét.

Szagnyom: Folt. A hangyák hagyják maguk után, idővel eltűnik, vonzzák a hangyákat.

Tócsa: Akadály, amit a hangyák megkerülhetnek, vagy át is kelhetnek rajta, ez esetben mozgási sebességük jelentősen visszaesik. A vízben nem tudnak szagnyomot hagyni, mert a víz elmossa. A szagokat ugyanolyan intenzitással érzik, mint a partról.

Tölcsér: Az az akadály, amely a hangyaleső(k) körül található a pályán. A tölcsérbe az arra tévedő hangyák beleesnek, és a közepén található hangyaleső elfogyasztja őket.

2.6 Projekt terv

2.6.1 Lépések és Határidők

1. Követelmény és specifikáció elkészítése - február 25.
2. Analízis modell kidolgozása 1 - március 4.
3. Analízis modell kidolgozása 2 - március 11.
4. Szkeleton tervezése - március 18.
5. Szkeleton elkészítése - március 25.
6. Prototípus koncepciója - április 2.
7. Részletes tervek - április 8.
8. Prototípus elkészítése - április 22.
9. Grafikus felület specifikációja - április 29.
10. Grafikus változat elkészítése - május 13.
11. Project átadása - május 17.

2.6.2 Felelősök

A project során mindenki részt vesz minden részfeladat elvégzésében és megoldásában, így esetlegesen egy csapattag ideiglenes munkaképtelensége esetén se áll a projekt. A különböző projektrészekért a következőképpen felelnek (részben vagy teljes egészében) a csapattagok:

Kozaróczy Zsolt: dokumentáció, munka összehangolás, kódírás, hangyászszün-szelídítő

Molnár Tamara: dokumentáció, tesztelés, grafikus felület, randompingvin-felelős

Papp Győző: kódírás, dokumentáció, diagram szerkesztés

Tóth Ákos: hangulatfelelős, git karbantartás és tanácsadó, kódírás, vezető fejlesztő

Varga Éva Árnika: dokumentáció, tesztelés, kódírás, modell tervezés

2.6.3 Használt eszközök

A dokumentációt a Google Docs segítségével szerkesztjük közösen. A dokumentumokat Google Drive-on keresztül osztjuk meg, mindenkinek teljes hozzáférése van az összes dokumentumhoz.

Az átbeszélendő teendőkhöz egy saját levelező listát használunk, amit mindannyian feliratkoztunk. Az e-maileket mindenki megkapja és közösen meg tudjuk vitatni a dolgokat akkor is, amikor nem tudunk találkozni. A sürgős ügyeket egyéb, gyorsabb kommunikációt lehetővé tevő csatornákon is megpróbáljuk megvitatni (pl. facebook chat, skype, gtalk).

A forráskódok megosztásához és szinkronizálásához Git verziókövetőt használunk, így mindenki a kódolás elején megkapja az aktuális legfrissebb verziót, illetve az esetleges hibák könnyen kijavíthatóak. A fejlesztést Eclipse fejlesztőkörnyezettel fogjuk végezni Java nyelven (jdk-1.6). A csapat az tervezés során a Sparx Enterprise Architect programot használja az UML modellek, diagrammok, stb. kezelésére.

A megjelenítéshez használt képeket több, egymástól független programmal fogjuk előállítani, például Gimp vagy Paint.net

2.7 Napló

Időpont	Időtartam	Résztevők	Eredmény
2013.02.17. 19:00:00	1,5 óra	Kozaróczy Molnár Papp Tóth Varga	Megbeszéltük a gyűléseink időpontját és az első beadandó feladatot. A csapat legközelebb 2013.02.20-án 10:00-kor gyűl össze megírni a követelmény dokumentációt.
2013.02.20. 10:30:00	2,5 óra	Kozaróczy	Dokumentálás: 2.2.2 Funkciók: vázlatos kidolgozás, 2.5 szótár, 2.7 Napló vezetés.
2013.02.20. 10:30:00	2,5 óra	Papp	Dokumentálás: 2.2.1 Általános áttekintés, 2.2.2 Funkciók: vázlatos kidolgozás, 2.5 Szótár: bővítés
2013.02.20. 10:30:00	2,5 óra	Tóth	Dokumentálás: 2.4 Lényeges use-case-ek
2013.02.20. 10:30:00	2,5 óra	Varga	Dokumentálás: 2.6.2 Felelősök, 2.2.3 Felhasználók
2013.02.20. 16:00:00	2 óra	Kozaróczy	Dokumentálás: 2.5 szótár, 2.7 Napló vezetés, 2.6.1 Lépések és Határidők

2013.02.20. 16:00:00	2 óra	Papp	Dokumentálás: 2.2.2 Funkciók: bővítés, 2.5 Szótár: bővítés
2013.02.20. 16:00:00	2 óra	Tóth	Dokumentálás: 2.4 Lényeges use-case-ek, 2.3 Követelmények, 2.2.5 Feltételezések, kapcsolatok
2013.02.20. 16:00:00	2 óra	Molnár	Dokumentálás: 2.1 Bevezetés, 2.2.4 Korlátozások
2013.02.20. 16:00:00	2 óra	Varga	Dokumentálás: 2.6.2 Felelősök, 2.2.3 Felhasználók, 2.2.2 Funkciók: Bővítés Ellenőrzés, javítás
2013.02.20. 16:00:00	2 óra	Kozaróczy Molnár Papp Tóth Varga	Döntés: Kozaróczy és Tóth véglegesíti a dokumentációt. A csapat legközelebb 2013.02.27-én gyűlik össze a következő dokumentum megszerkesztésének elkezdésére. Ekkorra Tóth prezentál a csapat számára egy objektummodell-tervet.
2013.02.24. 18:30:00	0,5 óra	Kozaróczy Tóth	A dokumentációt véglegesítettük.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Ant

Hangyák megvalósítása, mezőről-mezőre halad. Amelyik mezőt érinti ott szagnyomot hagy. Alapesetben szagnyom és étel irányába nagyobb valószínűséggel lép.

Ha ételt tartalmazó mezőre lép, az ételt felveszi és a hangyaboly irányába fog menni. Ekkor már nem törődik sem más étellel sem szagnyommal.

3.1.2 AntHill

Hangyabolyok, amelyek folyamatosan hangyákat hoznak létre. Az innen indult hangyáknak ide kell visszatérniük, ha találnak élelmet.

3.1.3 Antlion

Hangyaleső megvalósítása, ha odatéved egy hangya, akkor megöli. Nem mozog, fix helye van pályán.

3.1.4 AntScent

A hangyák által hagyott szagnyom, ami idővel veszít intenzitásából.

3.1.5 AntScentVanisherScent

A hangyaszagsemlegesítő spray által fújt szagnyom, ami eltünteti a hangyaszagokat.

3.1.6 AntScentVanisherSpray

Több egymás melletti mezőről eltünteti a hangyák által hagyott szagnyomokat.

3.1.7 Echidna

A hangyássünöket valósítja meg, mely ha nem jól lakott és hangyával találkozik, megeszi. Egy hangyászűn véges számú hangyát tud megenni, ennek függvényében nő a mérete.

3.1.8 Food

Ételek megvalósítása, ha megtalálja egy hangya, akkor fel tudja venni és elvinni a hangyabolyba. Az étel magától nem mozog, fix helye van, csak egy hangya mozgathatja.

3.1.9 FoodScent

Ételszagokat valósítja meg. Amíg az őt kibocsátó étel a helyén van, nem veszít az intenzitásából.

3.1.10 HexCell

Egy-egy mezőn található objektumok tárolására szolgál, tárolja, hogy a pályán hol helyezkedik el (szélén - nem szélén).

3.1.11 Pebble

Olyan, fix helyű akadályok a pályán, amikkel egy mezőre nem tud lépni se a hangya, se a hangyászsün.

3.1.12 Poison

Méreg, amit a PoisonSpray hagy. Csak a hangyákra hat, és egyből megöli őket. Idővel elmúlik a hatása.

3.1.13 PoisonSpray

A pályán méregfoltot hagyó spray. Egyszerre több mezőre hat.

3.1.14 Puddle

Olyan fix helyű tócsa a pályán, amin a hangya lassabban tud csak átjutni.

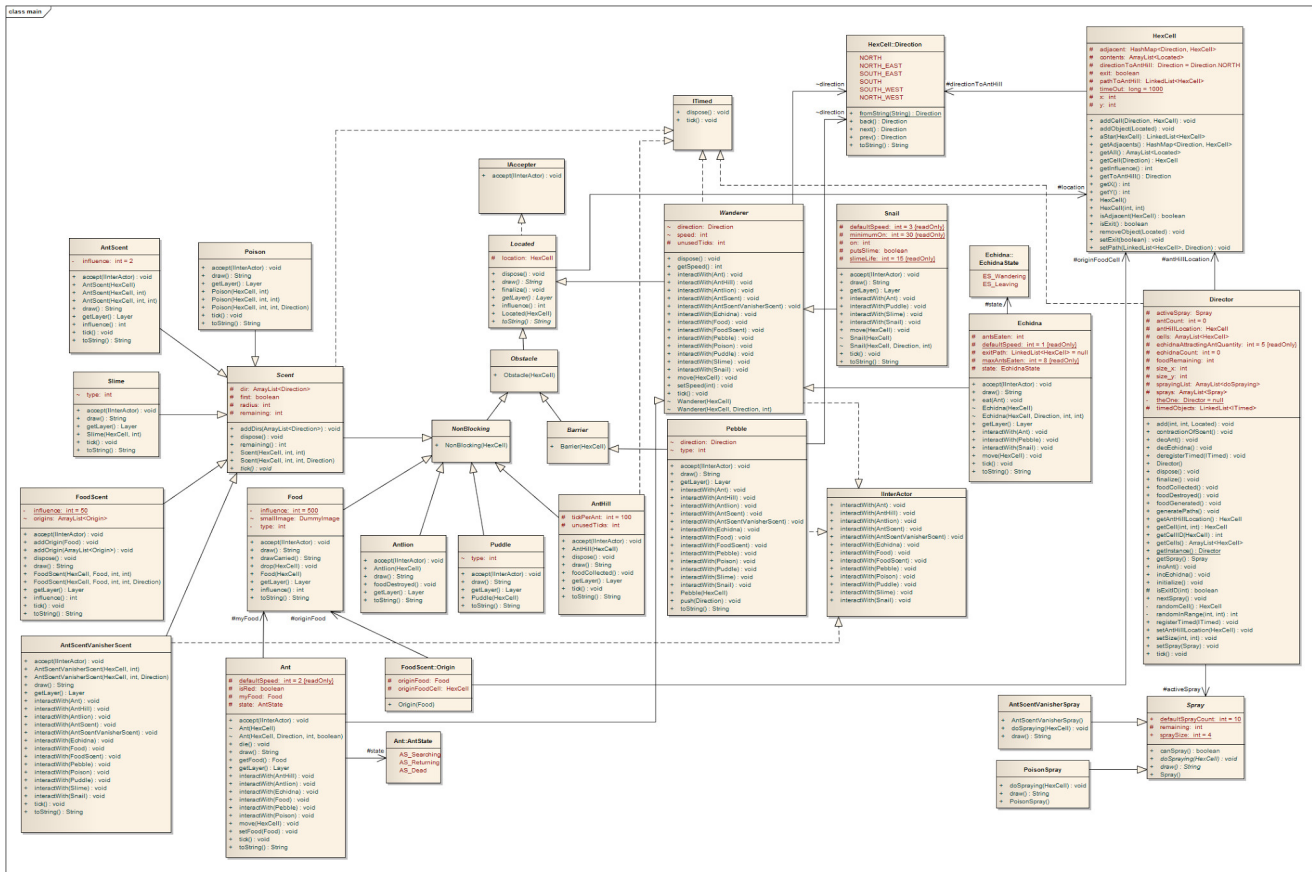
3.1.10 Snail

A csigákat. Maga után nyálkát hagy, nem tud rálépni a hangya.

3.1.11 Slime

Nyálka, amit a csiga hagy. Idővel elillan. Amíg a pályán van a hangya lassabban tud átmenni rajta.

3.2. Statikus struktúra diagramok



3.3. Osztályok leírása

3.3.1. Ant

- **Felelősség**

Az osztály a hangyák szimulálásáért felel.

- **Ösztályok**

Located -> Wanderer -> Ant

- **Interfészek**

Nincs

- **Attribútumok**

- `myFood:Food` : A hangya által vitt étel. Amikor felveszi, akkor már csak ő hivatkozik rá, a mezők nem.
- `state:AntState` : A hangya állapotára vonatkozó enum, lehet `AS_Searching` (keresés), `AS_Returning` (visszatérés) vagy `AS_Dead` (halott)
- `speed:int` : A hangya sebessége, kezdeti értéke 2.

- defaultSpeed:int : A hangya eredeti sebessége
- **Metódusok**
 - void accept(IInterActor) : Ezen keresztül jeleznek neki
 - void die(): A hangya ennek hatására AS_Dead (halott) állapotba kerül.
 - Food getFood(): A hangyánál lévő ételt (myFood) adja vissza.
 - int getSpeed(): A hangya sebességét adja vissza.
 - void interactWith(...): Ebben reagál a többiekre
 - void move(HexCell): A hangya mozgása a paraméterként kapott cellába.
 - void setFood(Food): A paraméterül kapott változót elmenti a myFood változóba, majd módosítja az állapotát AS_Returning (visszatérés) állapotba.
 - void setSpeed(int): A hangya sebességét állítja be.
 - void tick(): Az idő szimulálását végzi ez a függvény.

3.3.2. AntHill

- **Felelősség**
A hangyabolyt modellezi. Az idő múlásával folyamatosan újabb hangyákat hoz létre.
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> AntHill
- **Interfészek**
ITimed
- **Attribútumok**
 - unusedTicks : int : Ezzel számolja, hogy jöhet-e új hangya
 - tickPerAnt : Ezzel számolja, hogy mennyi időnként kell jönnie hangyának
- **Metódusok**
 - void foodCollected(): Ha étellel érkezik hangya, akkor elveszi tőle, majd továbbadja az információt.
 - void tick() : Az idő múlását modellezi

3.3.3. Antlion

- **Felelősség**
A hangyalesőket modellezi.
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> Antlion
- **Interfészek**
Nincs
- **Attribútumok**
Nincs
- **Metódusok**
 - void foodDestroyed() : Jelzi, hogy egy olyan hangyát evett meg, amelyiknél volt étel,

így az megsemmisült

3.3.4. AntScent

- **Felelősség**
Hangyaszagot reprezentáló osztály
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> Scent -> AntScent
- **Interfészek**
Nincs
- **Attribútumok**
Nincs
- **Metódusok**
 - void : tick() : Az idő múlását szimulálja. Minden meghíváskor ha tud még, akkor tovább terjed a megfelelő irányokba, egyébként pedig folyamatosan csökken az intenzitása. Nem tud terjedni.

3.3.5. AntScentVanisherScent

- **Felelősség**
Hangyaszagsemlegesítő szagot reprezentáló osztály
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> Scent -> AntScentVanisherScent
- **Interfészek**
 - IInterActor
- **Attribútumok**
Nincs
- **Metódusok**
 - void interactWith(...) : Ezen keresztül reagál mindenkivel
 - void tick() : Az idő múlását szimulálja. Minden meghíváskor ha tud még, akkor tovább terjed a megfelelő irányokba, egyébként pedig folyamatosan csökken az intenzitása.

3.3.6. AntScentVanisherScent

- **Felelősség**
Hangyaszagsemlegesítő sprayt reprezentáló osztály
- **Ősosztályok**
Spray
- **Interfészek**
Nincs
- **Attribútumok**

Nincs

- **Metódusok**

- void doSpraying(HexCell) : Az adott pozícióba létrehoz egy szagtalanító foltot.

3.3.7. Barrier

- **Felelősség**

Olyan absztrakt őszotály, amiből azok az akadályok öröklődnek, amiket a IMovable interfészt megvalósítóknak ki kell kerülniük.

- **Őszotályok**

Located -> Obstacle -> Barrier

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

- **Metódusok**

Nincs

3.3.8. Director

- **Felelősség**

Létrehozza és tárolja a mezőket (HexCell), majd szétosztja az akadályokat (Obstacle), hangyabolyt (AntHill) és ételdarabokat (Food). Ő felel a hangyászsünök (Echidna) létrehozásáért és az aktív Spray kiválasztásáért. A játék végét is ellenőrzi megfelelő függvények meghívásakor.

- **Őszotályok**

Nincs

- **Interfészek**

ITimed

- **Attribútumok**

- theOne : Director : Singleton példány
- activeSpray: Spray: Az éppen aktívan használt Spray
- cells: ArrayList<HexCell>: Az összes cella listája
- timedObjects : LinkedList<ITimed> : Az összes időzített objektum
- sprays : ArrayList<Spray> : A használható spray-k
- foodRemaining: int A még be nem vitt ételek (Food) számát tárolja.
- size_x : int : A pálya mérete
- size_y : int : A pálya mérete
- anthillLocation : HexCell : A hangyaboly helye
- antCount : int : A hangyák száma

- echidnaCount : int : A hangyászsünök száma
- echidnaAttractingAntQuantity : int : Megadja, hogy hány hangyánként rendel be a pályára egy újabb hangyászsünt

- **Metódusok**

- void add(int, int, Located) : Hozzáad egy elemet a megfelelő cellához
- void contractionOfScent() : Szagok összevonása
- void decAnt() : Csökkenti a hangyaszámlálót
- void decEchidna() : Csökkenti a hangyászsünszámlálót
- void deregisterTimed(ITimed) : Egy ITimed elem eltávolítása a listából
- void foodCollected() : Ezen keresztül kap jelzést, hogy egy étel beért a hangyabolyba. Csökkenti az ételszámlálót
- void foodDestroyed() : Ezen keresztül kap értesítést, hogy egy étel megsemmisült. Ekkor egy véletlenszerűen kiválasztott cellára rak egy új ételt. A véletlen cella csak olyan lehet, ami nem tartalmaz sem Barriert, sem Antliont, hiszen azok elérhetetlenek egy hangya számára. Továbbá AntHillt sem tartalmazhat.
- void foodGenerated() : Ezen keresztül kap jelzést, ha egy új étel jön létre
- HexCell getAntHillLocation() : Visszaadja, hol van a hangyaboly
- HexCell getCell(int, int) : Koornidáták alapján visszaad egy HexCellt. 0-tól indexel.
- int getCellID(HexCell) : Visszaadja a keresett cella ID-jét
- ArrayList<HexCell> getCells() : Visszaadja az összes cellát
- Spray getSpray() : Az aktív spray-t adja vissza.
- void incAnt() : Növeli a hangyaszámlálót. A hangya hívja meg a konstruktorában
- void incEchidna() : Növeli a hangyászsünszámlálót
- void initialize() : A játék inicializálása.
- boolean isExitID(int) : Generáláskor számozott cellák közül visszaadja, hogy az adott sorszámú szélső cella-e.
- void nextSpray() : Beállítja a aktív spraynek a következőt.
- void registerTimed(ITimed) : Egy ITimed elem hozzáadása a listához.
- void setAntHillLocation(HexCell) : Beállítja a hangyaboly helyét
- void setSize(int, int) : Beállítja a pálya méretét. Előtte letörli a teljes pályát, így kibővítésre NEM használható. Az első cella a bal felső, és tőle SOUTHEAST irányban indul el a pálya, minden sor azonos hosszúságú
- void setSpray(Spray) : Aktív spray-ként állítja be a paraméterül kapottat.
- void tick() : Az idő múlását szimulálja. Valamilyen sorrendben meghívja az összes beregisztrált ITimed objektum ITimed.tick metódusát. Ha egy tick() hívás hatására egy új ITimed kerül beregisztrálása, az ő tick metódusa NEM lesz meghívva. Ha a pályán adott mennyiségnél több hangya van, akkor berak néhány hangyászsünt a pálya széleire. Bizonyos időtartamonként az egy HexCell-en lévő hangyaszagokat összevonja, hogy ne burjánzanak el az objektumaink.

3.3.9. Echidna

- **Felelősség**

A hangyászsünök modellezésére szolgál. Ha hangyával egy mezőre (HexCell) kerül, akkor megeszi a hangyát.

- **Ősosztályok**

Located -> Wanderer -> Echidna

- **Interfészek**

Nincs

- **Attribútumok**

- antsEaten:int : A megevett hangyák számát tárolja
- maxAntsEaten:const int : A hangyák száma, amit maximum meg tud enni. Csak olvasható.
- defaultSpeed : int : Az alapértelmezett sebesség, értéke 1
- exitPath : LinkedList<HexCell> : A kijutáshoz használt útvonal
- speed: int : A hangyászsün sebessége, kezdeti értéke 4.
- state:EchidnaState : A hangyászsün állapotát tárolja. Lehet ES_Wandering (kóricál), ekkor megeszi a hangyákat vagy ES_Leaving (jóllakott), ekkor minél hamarabb megpróbál kijutni a pályáról.

- **Metódusok**

- void accept(IInterActor) : A paraméterként kapott IInterActor-on meghívja az interactWith(...) függvényt saját magát adva paraméterként.
- void eat(Ant): Megeszi a paraméterként kapott hangyát, így az meghal.
- void interactWith(...) : Ebben reagál minden mással
- void move(HexCell): A hangyászsün a paraméterül kapott cellába mozog tovább.
- void tick() : Az idő múlását szimulálja. Kóricáláskor véletlenszerű irányba mozog, távozáskor viszont megpróbál mindig a pálya szélé felé haladni

3.3.10. Food

- **Felelősség**

A ételeket reprezentálja a játékban.

- **Ősosztályok**

Located -> Obstacle -> NonBlocking -> Food

- **Interfészek**

Nincs

- **Attribútumok**

- influence : int : Hangyák befolyásolásának mértéke.

- **Metódusok**

- void drop(HexCell) : Ételt az adott mezőre ejtik ezzel.
- int influence() : Megadja, hogy mennyire befolyásolja a hangya útvonalát.

3.3.11. FoodScent

- **Felelősség**
Ételek szagát reprezentálja
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> Scent -> Food
- **Interfészek**
Nincs
- **Attribútumok**
 - origins : ArrayList<Origin> : A kibocsátó ételek
 - influence : int : Hangyák befolyásolásának mértéke.
- **Metódusok**
 - void addOrigin(Food) : Új ételt ad hozzá a forrásokhoz. Ezáltal összevonhatóak a szagok
 - int influence() : Megadja, hogy mennyire befolyásolja a hangya útvonalát.
 - void tick() : Amíg az origins-ből legalább egy a helyén van, nem csökken a szag intenzitása és terjeng az illat

3.3.12. HexCell

- **Felelősség**
A mezőket modellezi. HexCellben tárolódik minden, aminek pozíciója van (Located) és az adott cellán tartózkodik.
- **Ősosztályok**
Nincs
- **Interfészek**
Nincs
- **Attribútumok**
 - adjacent : HashMap<Direction, HexCell> : A szomszéd mezők
 - contents : ArrayList<Located> : Minden objektum, ami pozícióval rendelkezik, és az adott mezőn helyezkedik el.
 - exit:boolean : Igaz, ha a mező a pálya széle, tehát a hangyászszün (Echidna) ebben az irányban elhagyhatja a játékteret.
 - nearestExit:int : Annak a szomszédjának száma, amelyik a legközelebb van egy kijáráshoz.
 - pathToAntHill : LinkedList<HexCell> : A legközelebbi hangyaboly felé vezető útvonal

- directionToAntHill : Direction : A hangyabolyhoz vezető útvonal iránya
- x : int : A cella pozíciója
- y : int : A cella pozíciója

- **Metódusok**

- void addCell(Direction, HexCell) : Szomszédos mező hozzáadása
- void addObject(Located) : Új, pozícióval rendelkező Located objektum hozzáadása.
- HashMap<Direction, HexCell> getAdjacents() : Visszaadja az összes szomszédját
- ArrayList<Located> getAll() : Visszaadja az összes elemet a contents listából.
- HexCell getCell(Direction) : Az adott irányban lévő cellával tér vissza
- int getInfluence() : Visszaadja, hogy az adott mező mennyire befolyásolja a hangya irányát
- Direction getToAntHill() : Visszaadja, hogy milyen irányban van a hangyaboly
- boolean isExit() : Igazgal tér vissza, ha az adott cella kijárat
- void removeObject(Located) : Az adott elemet eltávolítja a contents listáról
- void setExit(boolean) : Beállítja, hogy az adott cella kijárat-e.
- void setPath(LinkedList<HexCell>, Direction) : Hangyabolyba vezető útvonal eltárolása

3.3.13. IAcceptor

- **Felelősség**

Közös interfész azok az osztályoknak, amik interact-ot tudnak kiváltani

- **Ősosztályok**

Nincs

- **Metódusok**

- void accept(IInterActor) : A paraméterként kapott IInterActor-on meghívja az interactWith(...) függvényt saját magát adva paraméterként.

3.3.14. IInterActor

- **Felelősség**

- Közös interfész azoknak, akik interakciót képesek végezni

- **Ősosztályok**

Nincs

- **Metódusok**

- void interactWith(Ant) : Hangyával való interakció
- void interactWith(AntHill) : Hangyabollyal való interakció.
- void interactWith(Antlion) : Hangyalesővel való interakció.
- void interactWith(AntScent) : Hangyaszagnyommal való interakció.
- void interactWith(AntScentVanisherScent) : Hangyaszagsemlegesítő spray-el való interakció.
- void interactWith(Echidna) : Hangyászszünnel való interakció.

- void interactWith(Food) : Étellel való interakció
- void interactWith(FoodScent) : Ételszaggal való interakció
- void interactWith(Pebble) : Kavicssal való interakció
- void interactWith(Poison) : Méreggel való interakció
- void interactWith(Puddle) : Tócsával való interakció
- void interactWith(Slime) : Nyálkával való interakció.
- void interactWith(Snail) : Csigával való interakció

3.3.15. ITimed

- **Felelősség**
Közös interfész azoknak az osztályoknak, amik reagálnak az idő múlására
- **Ősosztályok**
Nincs
- **Metódusok**
 - void dispose() : Deregisztrálja magát a Director ITimed-okat tartalmazó listájából.
 - void tick() : Az idő múlását szimulálja.

3.3.16. Located

- **Felelősség**
A közös ősosztálya mindennek, aminek van pozíciója
- **Ősosztályok**
Nincs
- **Interfészek**
Nincs
- **Attribútumok**
 - location : HexCell :
- **Metódusok**
 - void dispose() : Eltávolítja magát a cellából
 - int influence() : Megadja, hogy mennyire befolyásolja a hangya útvonalát.

3.3.17. NonBlocking

- **Felelősség**
A nem blokkoló akadályok közös ősosztálya.
- **Ősosztályok**
Located -> Obstacle -> NonBlocking
- **Interfészek**
Nincs
- **Attribútumok**
Nincs
- **Metódusok**

Nincs

3.3.18. Obstacle

- **Felelősség**
Az akadályok közös absztrakt őosztálya.
- **Őosztályok**
Located -> Obstacle
- **Interfészek**
Nincs
- **Attribútumok**
Nincs
- **Metódusok**
Nincs

3.3.19. Pebble

- **Felelősség**
Kavicsokat reprezentáló osztály
- **Őosztályok**
Located -> Obstacle -> Barrier -> Pebble
- **Interfészek**
 - IInterActor
- **Attribútumok**
 - direction : Direction : Ha meglökik a kavicsot, ez határozza meg az irányát
- **Metódusok**
 - void interactWith(...) : Ezen keresztül tud reagálni minden másra
 - void push(Direction) : Átmozog a megfelelő irányba

3.3.20. Poison

- **Felelősség**
A méregfoltokat modellező osztály. Megöli a rátévedő hangyákat.
- **Őosztályok**
Located -> Obstacle -> NonBlocking -> Scent -> Poison
- **Interfészek**
Nincs
- **Attribútumok**
Nincs
- **Metódusok**
 - void tick() : Az idő múlását szimulálja. Minden meghíváskor ha tud még, akkor tovább terjed a megfelelő irányokba, egyébként pedig folyamatosan csökken az intenzitása

3.3.21. PoisonSpray

- **Felelősség**
A méregspray-t modellezi.
- **Ősosztályok**
Spray -> PoisonSpray
- **Interfészek**
Nincs
- **Attribútumok**
Nincs
- **Metódusok**
 - void doSpraying(HexCell) : Az adott helyre méregfoltot helyez.

3.3.22. Puddle

- **Felelősség**
Tócsákat reprezentáló osztály. Mindenki lassabban tud csak átkelni rajta.
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> Puddle
- **Interfészek**
Nincs
- **Attribútumok**
Nincs
- **Metódusok**
Nincs

3.3.23. Scent

- **Felelősség**
Szagok közös ősosztálya
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> Scent
- **Interfészek**
 - ITimed
- **Attribútumok**
 - remaining : int : Az elillanásig hátralévő idő
 - radius : int : A terjedés nagysága
 - dir : ArrayList<Direction> : A terjedés iránya. Akkor használja, ha nem ő az első szagfolt
 - first : boolean : Igaz, ha ő az első szagfolt. Fontos a terjedéshez

- **Metódusok**

- addDirs(ArrayList<Direction>) : Új terjedési irányok hozzáadása
- void dispose() : Deregisztrálja magát a Director ITimed-okat tartalmazó listájából.
- int remaining() : Visszaadja az elillanásig hátralévő időt
- void tick() : Az idő múlását szimulálja. Minden meghíváskor ha tud még, akkor tovább terjed a megfelelő irányokba, egyébként pedig folyamatosan csökken az intenzitása.

3.3.24. Slime

- **Felelősség**

Csigák által hagyott nyálkákat reprezentáló osztály

- **Ősosztályok**

Located -> Obstacle -> NonBlocking -> Scent -> Slime

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

- **Metódusok**

- void tick() : Az idő múlását szimulálja. Minden meghíváskor ha tud még, akkor tovább terjed a megfelelő irányokba, egyébként pedig folyamatosan csökken az intenzitása. Nem tud terjedni.

3.3.25. Snail

- **Felelősség**

Csigákat reprezentáló osztály

- **Ősosztályok**

Located -> Wanderer -> Snail

- **Interfészek**

Nincs

- **Attribútumok**

- defaultSpeed : int : Az alapértelmezett sebesség, értéke 3
- slimeLife : int : A hagyott nyálka élettartama
- minimumOn : int : Minimim 30 tick-et tölt a pályán
- on : int : Ennyi ideje van a pályán
- putsSlime : boolean : Hamis, ha nem kell nyálkát hagynia

- **Metódusok**

- void interactWith(Ant) : Hangyával való interakció. A hangya nem lép vele egy mezőre
- void interactWith(Puddle) : Nem megy rá a tócsára

- void interactWith(Slime) : Nyálkával való interakció. Felfrissíti az elillanásig maradó idejét.
- void interactWith(Snail) : Csigával való interakció. Visszapattan a csigáról.
- void move(HexCell) : Mozgás a kijelölt cellába, majd minden ottlévőtől interakciót vár. Slime-ot hagy maga után
- void tick() : Az idő múlását szimulálja. Véletlenszerű irányba mozog.

3.3.26. Spray

- **Felelősség**

A különböző spray-eket reprezentáló absztrakt őosztály

- **Őosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- remaining: int: A maradék fújások száma
- spraySize: int: A spray hatótávolsága. Mivel minden spray ugyanolyan távolságra hat, statikus.
- defaultSprayCount : int : A kezdetben lehetséges fújások száma

- **Metódusok**

- boolean canSpray(): Igazzal tér vissza, ha még használható további fújásokra.
- void doSpray(HexCell): A paraméterként kapott mezőre ráfúj (spray-enként eltérő lehet a hatása)

3.3.27. Wanderer

- **Felelősség**

A mozogni képes lények közös őosztálya

- **Őosztályok**

- Located -> Wanderer

- **Interfészek**

- IInterActor
- ITimed

- **Attribútumok**

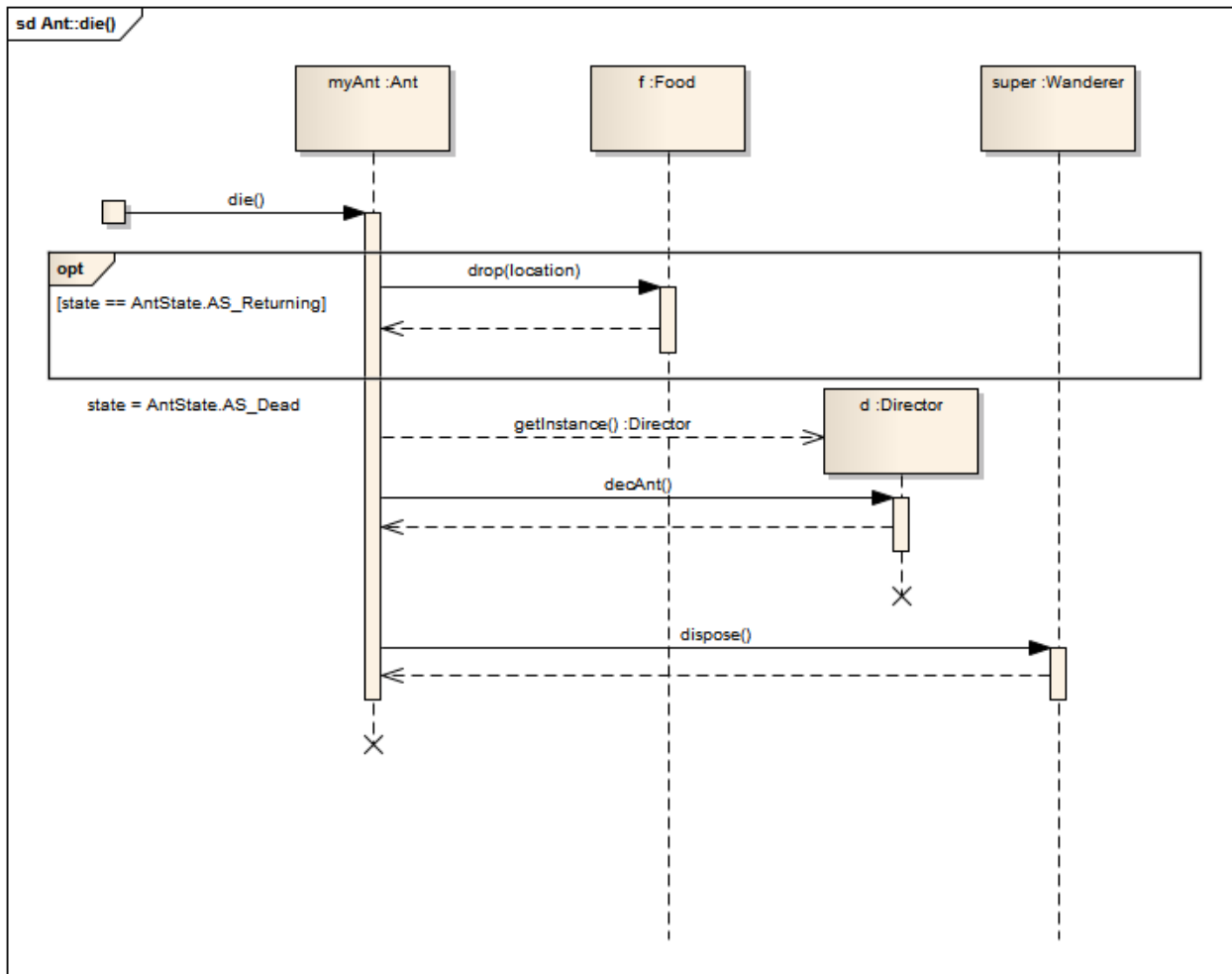
- direction : Direction : A mozgás iránya
- speed : int : A mozgás sebessége
- unusedTicks : int : Az előző mozgás óta eltelt tick-ek száma

- **Metódusok**

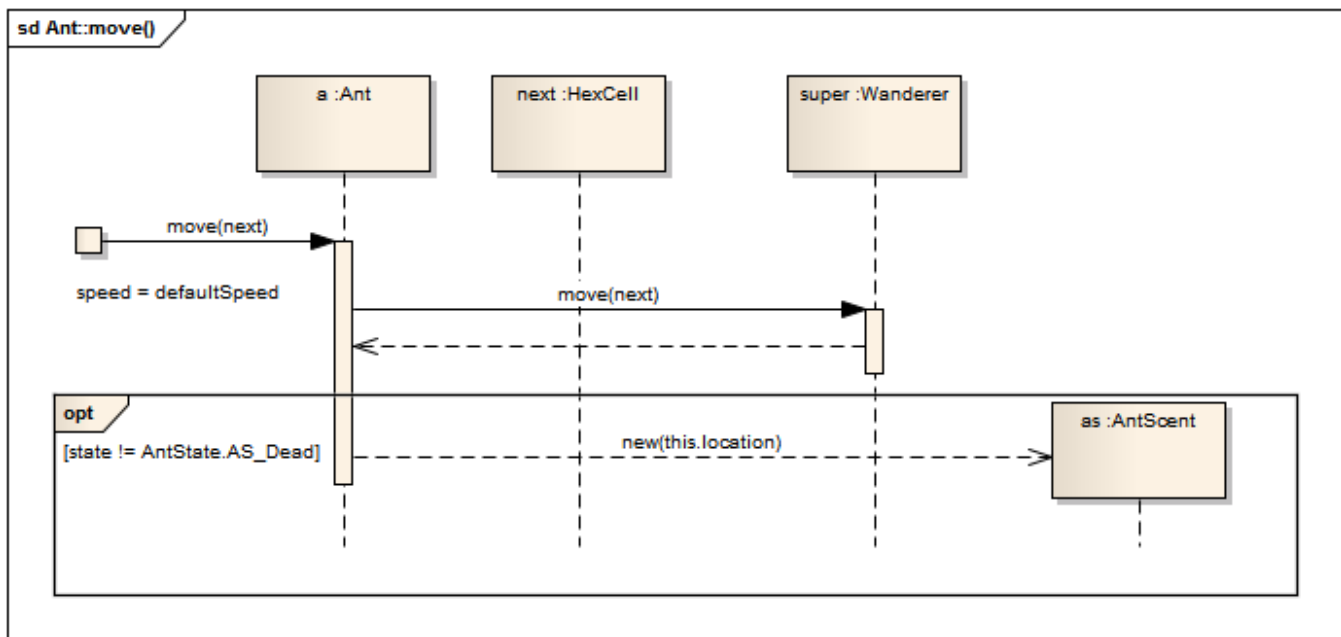
- `void dispose()` : Deregisztrálja magát a Director ITimed-okat tartalmazó listájából.
- `int getSpeed()` : A sebességet adja vissza
- `void interactWith(...)` : Ezekben reagál mindennel. Általában csak egy üres függvénytörzs
- `void interactWith(Pebble)` : Kavicssal való interakció. Visszapattan a kavicsról
- `void interactWith(Puddle)` : Tócsával való interakció. A sebességét a felére csökkenti
- `void interactWith(Slime)` : Nyálkával való interakció. A sebességét a felére csökkenti
- `void interactWith(Snail)` : Csigával való interakció. Visszapattan a csigáról.
- `void move(HexCell)` : Mozgás a kijelölt cellába, majd minden ottlévőtől interakciót vár
- `void setSpeed(int)` : A sebességet állítja be.
- `void tick()` : Az idő múlását szimulálja.

3.4. Szekvencia diagramok

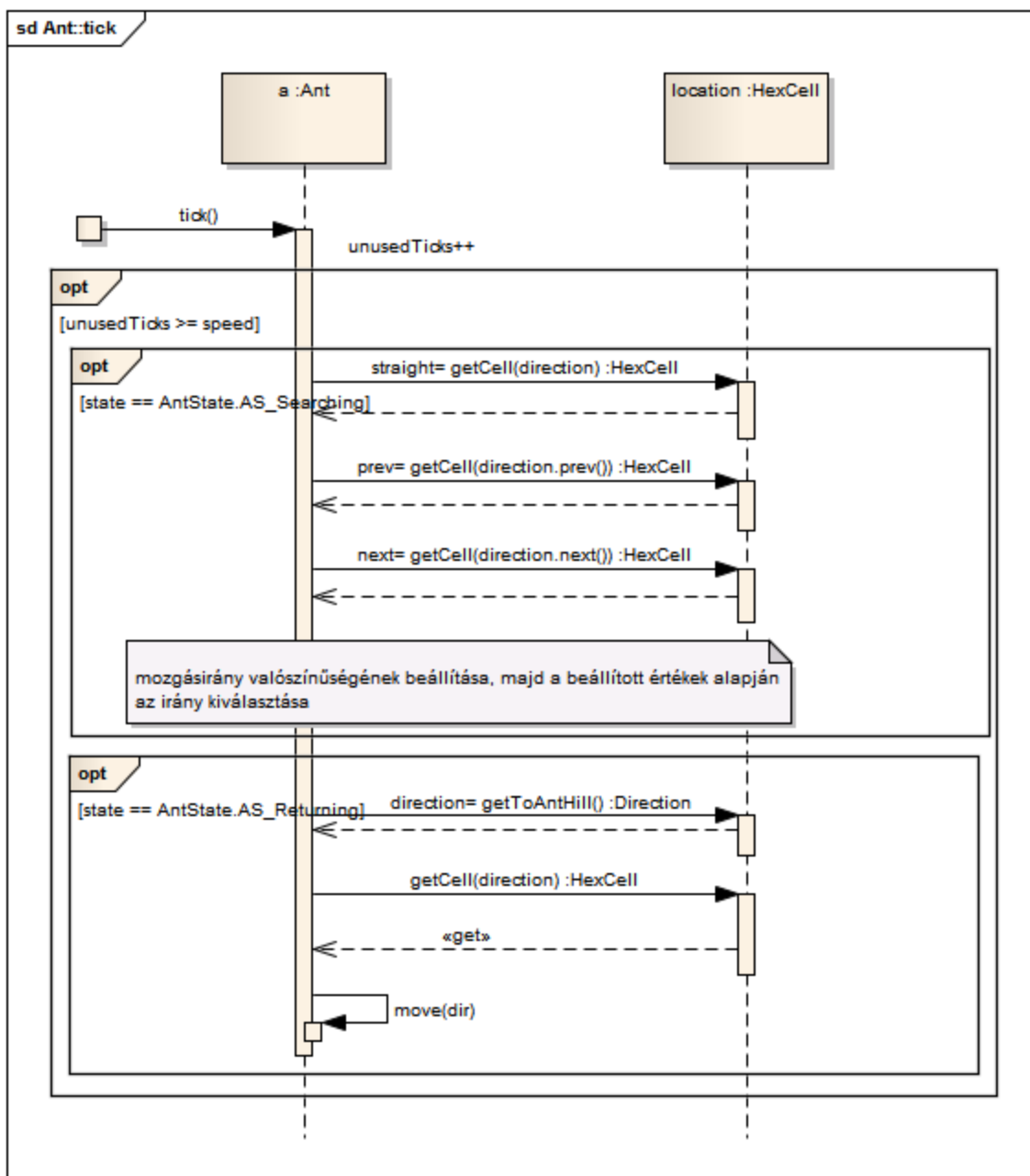
3.4.1. Ant::die



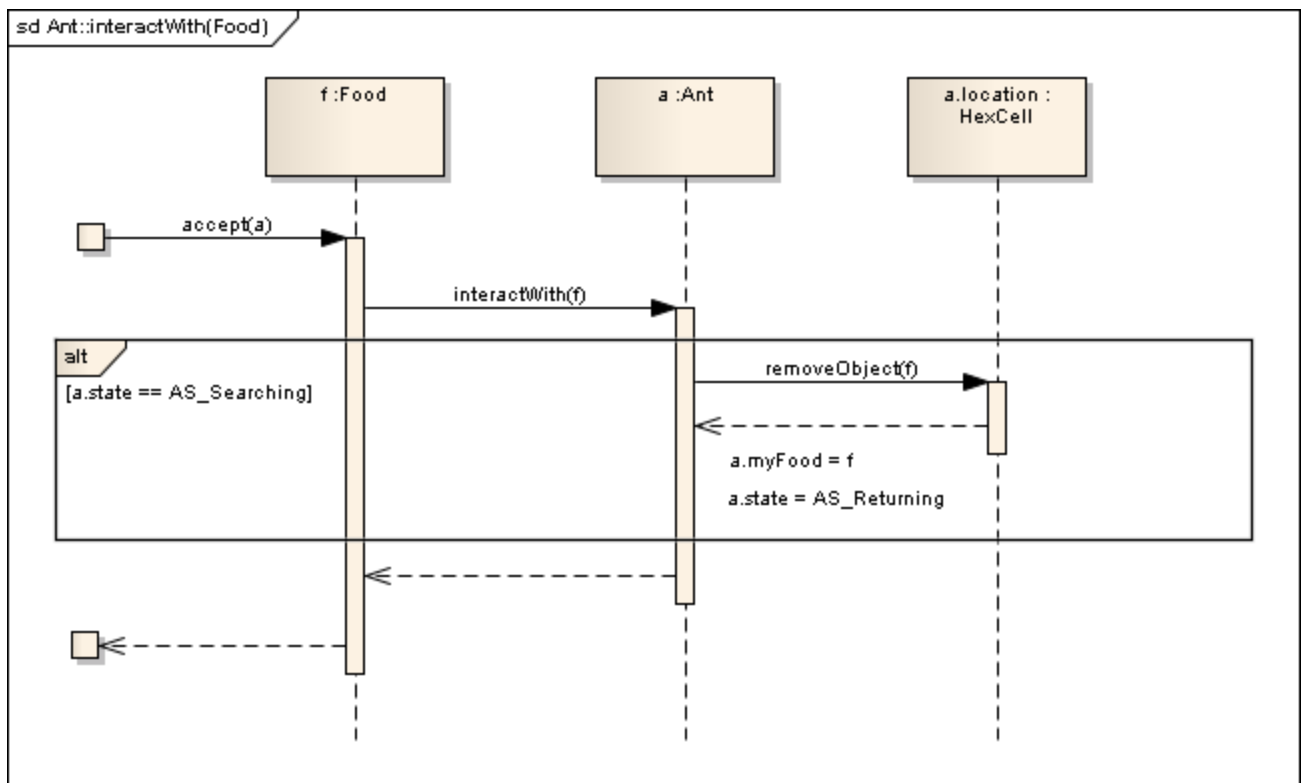
3.4.2. Ant::move



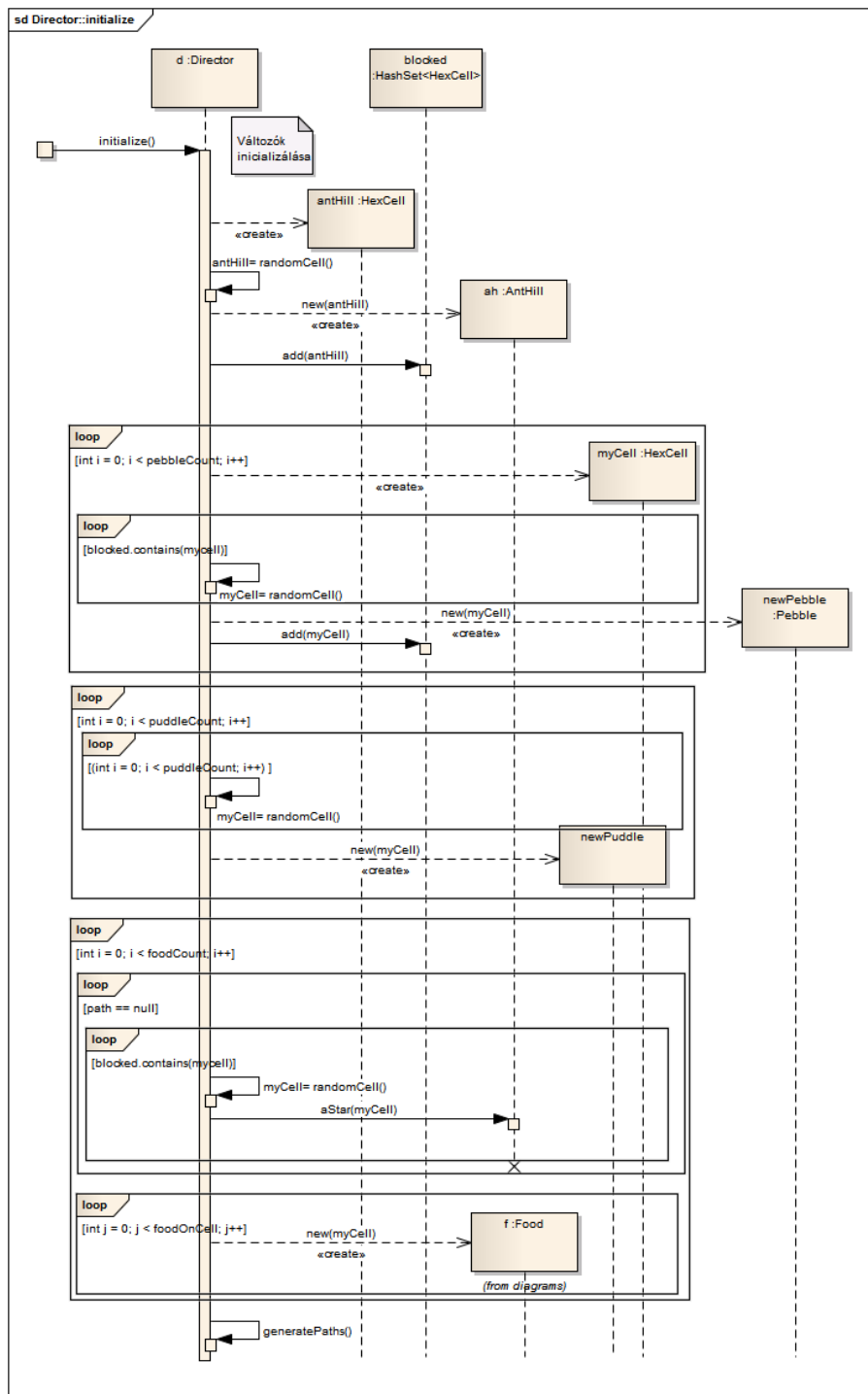
3.4.3. Ant::tick



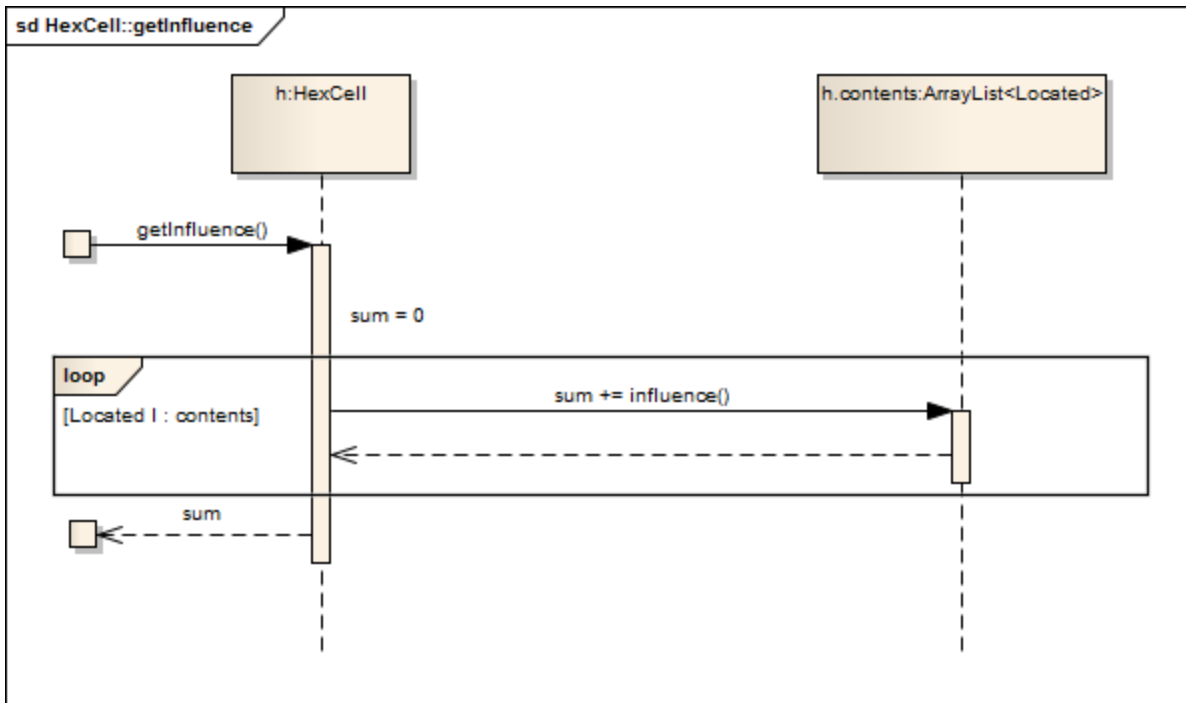
3.4.4. Ant::interactWith(Food)



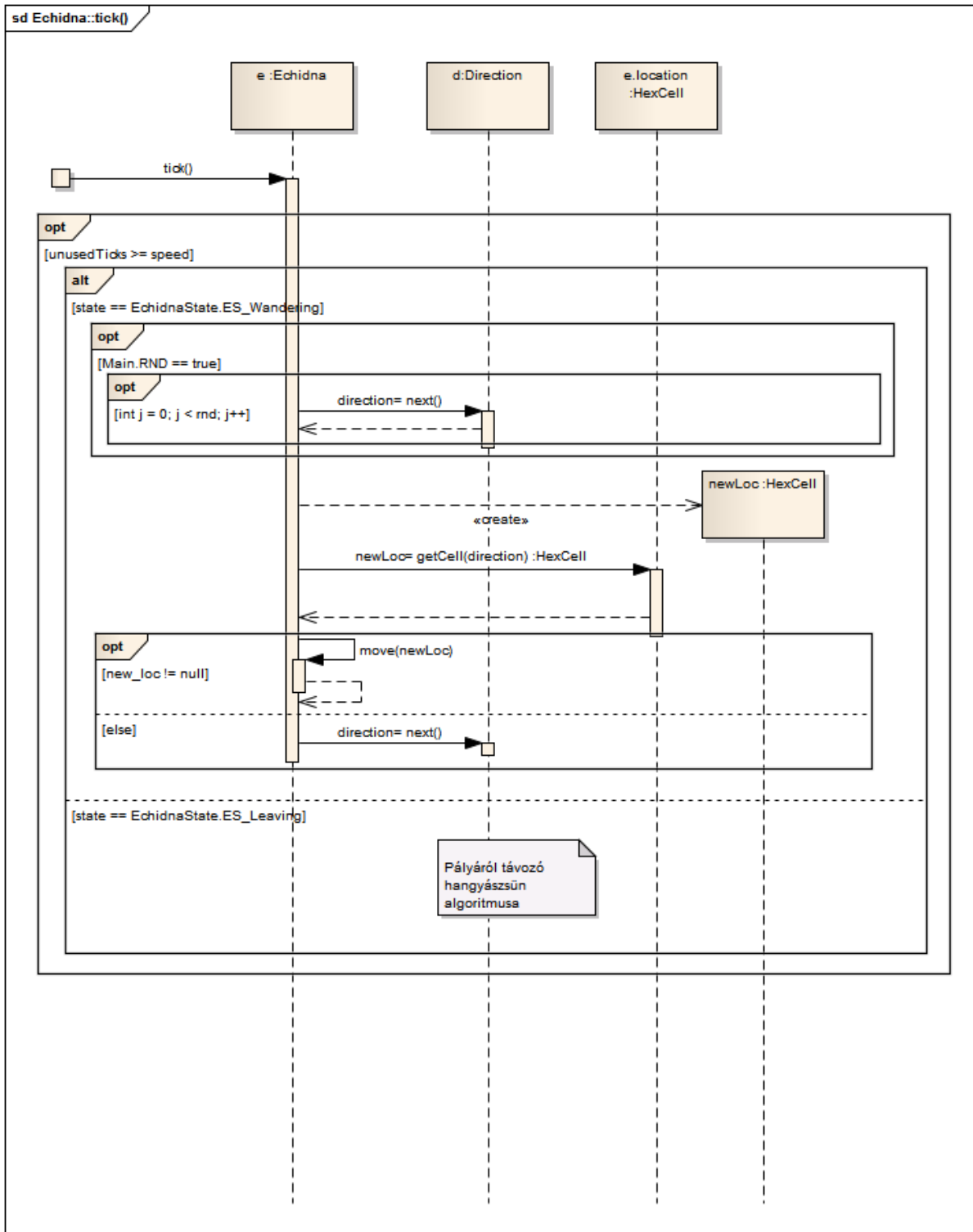
3.4.5. Director::initialize



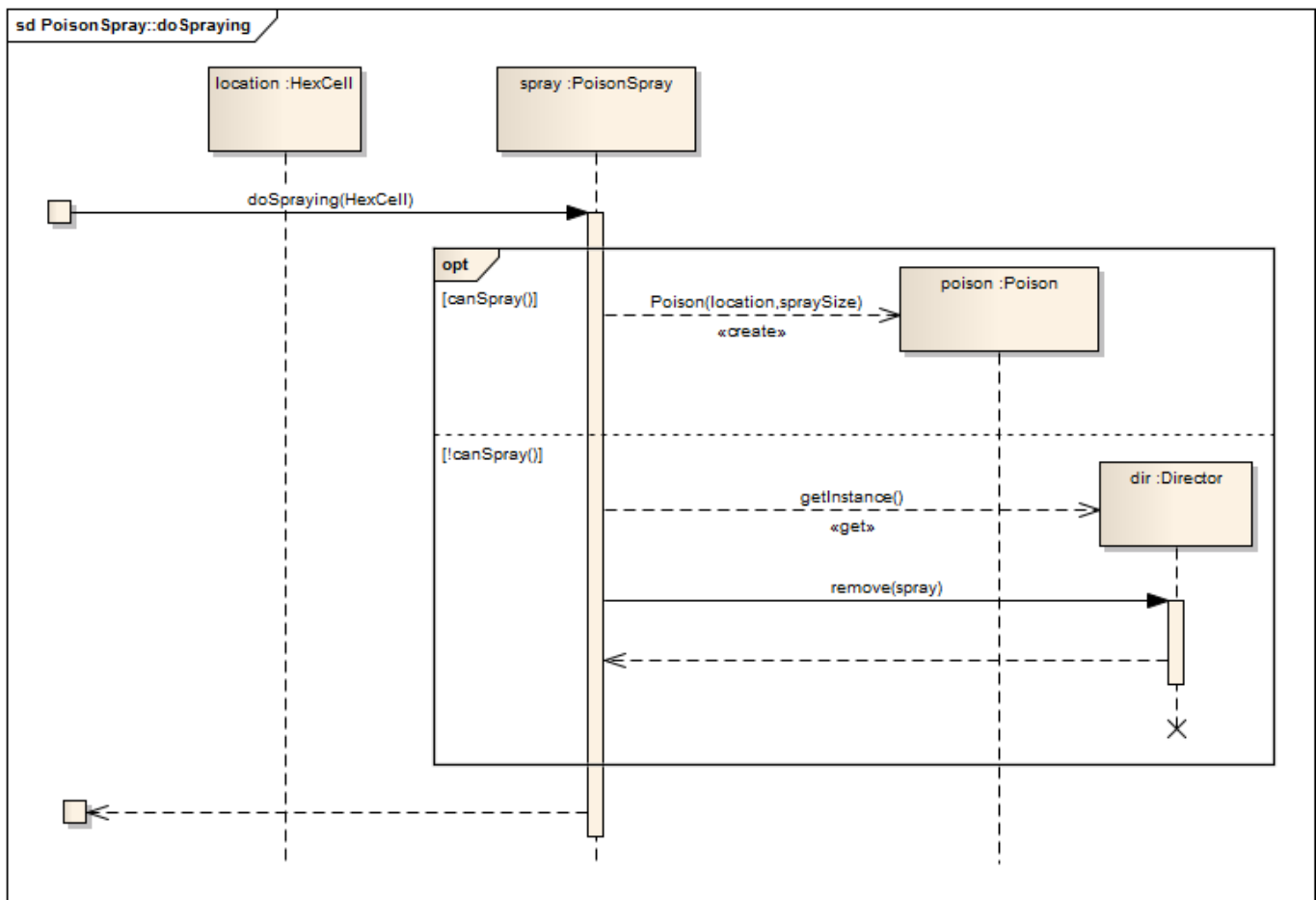
3.4.6. HexCell::getInfluence()



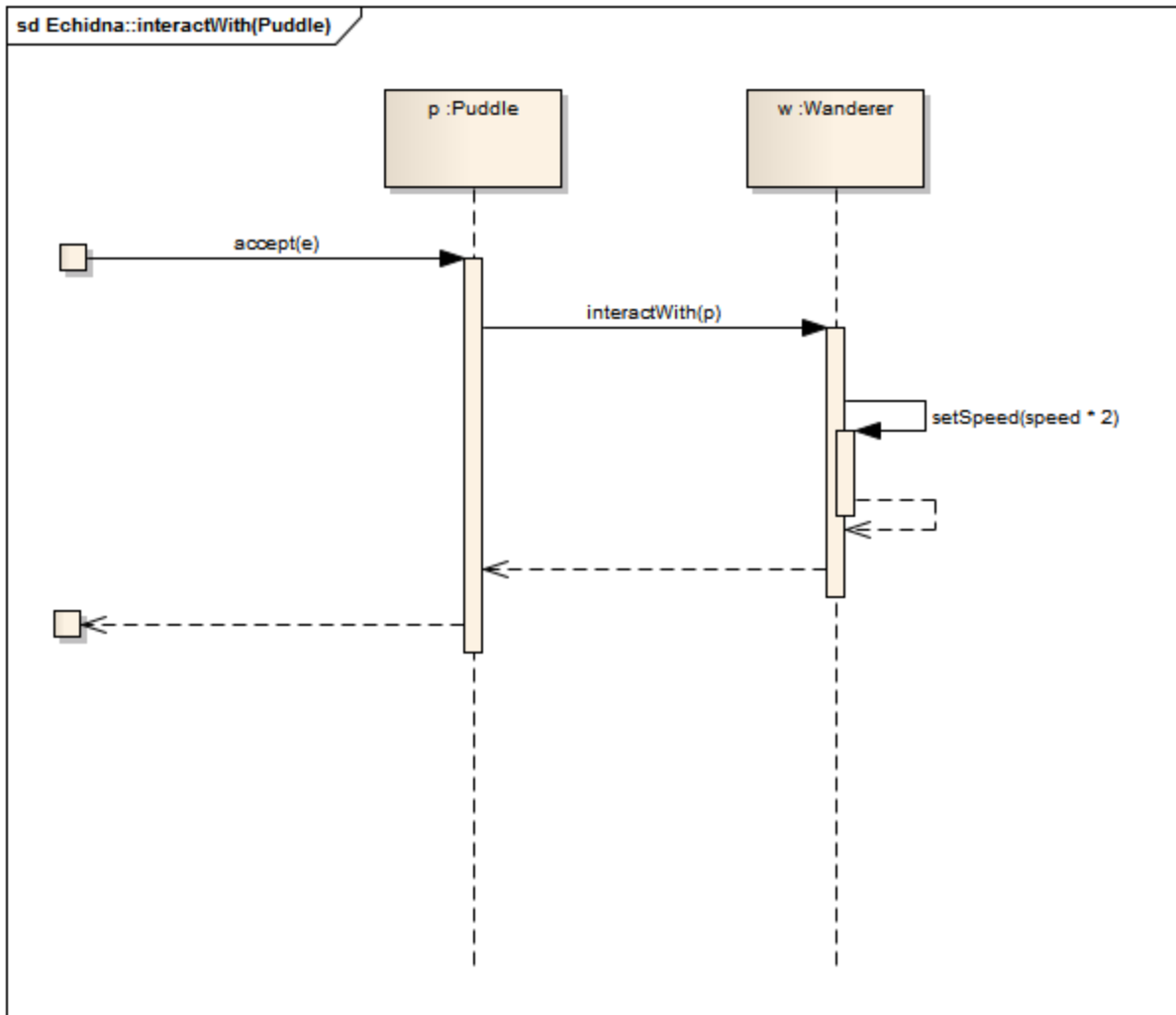
3.4.7. Echidna::tick



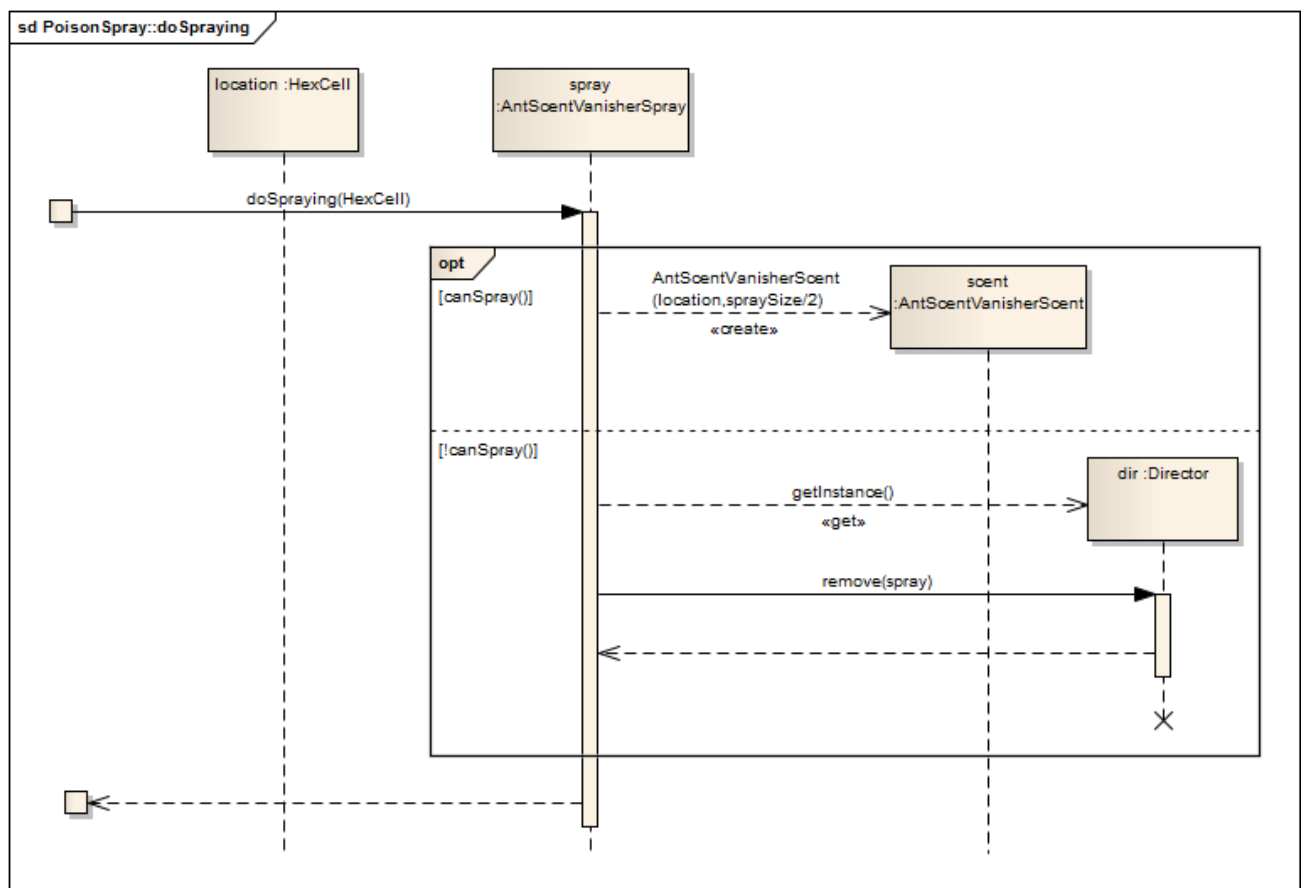
3.4.6. PoisonSpray::doSpraying



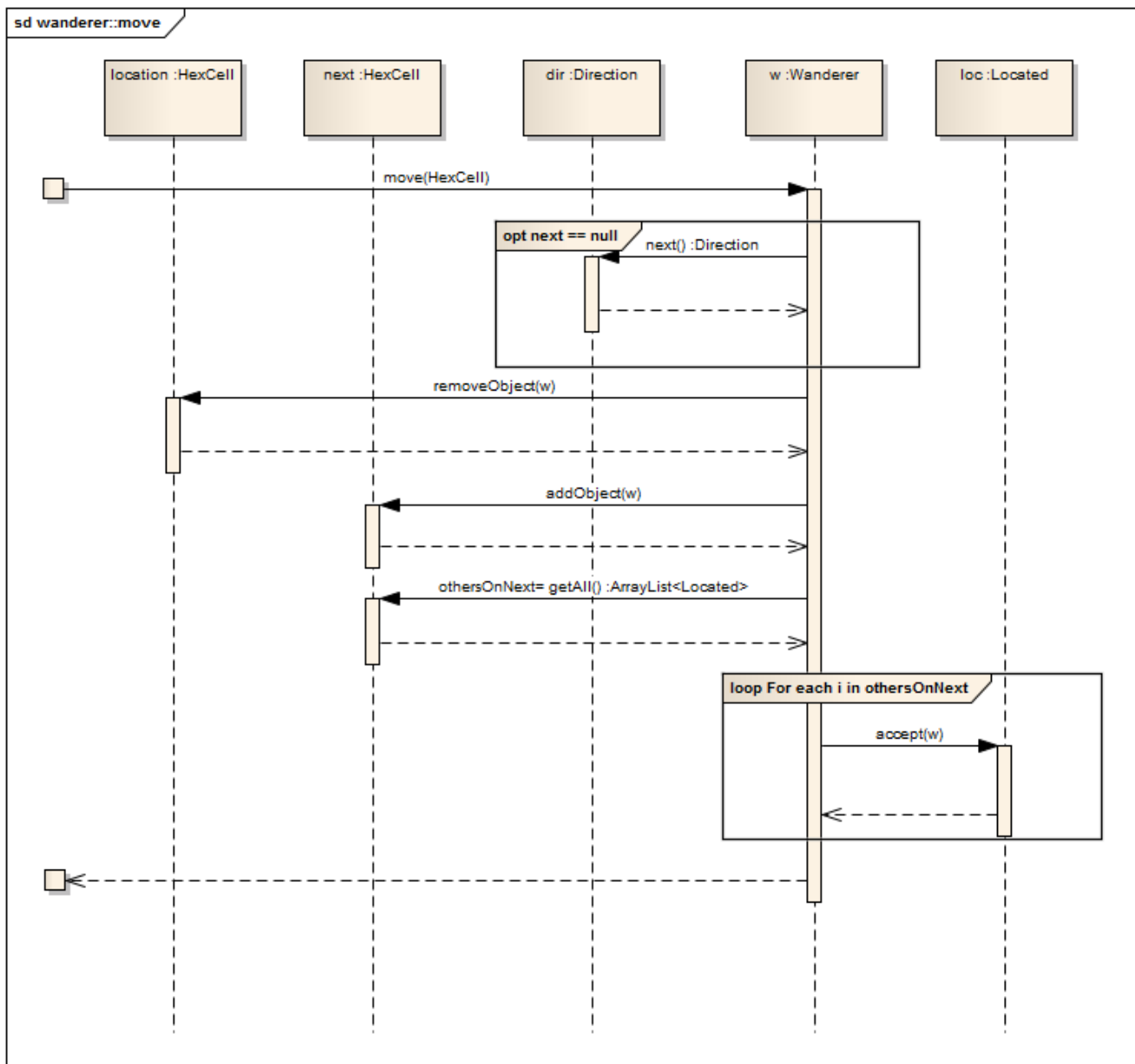
3.4.7. Puddle Slows



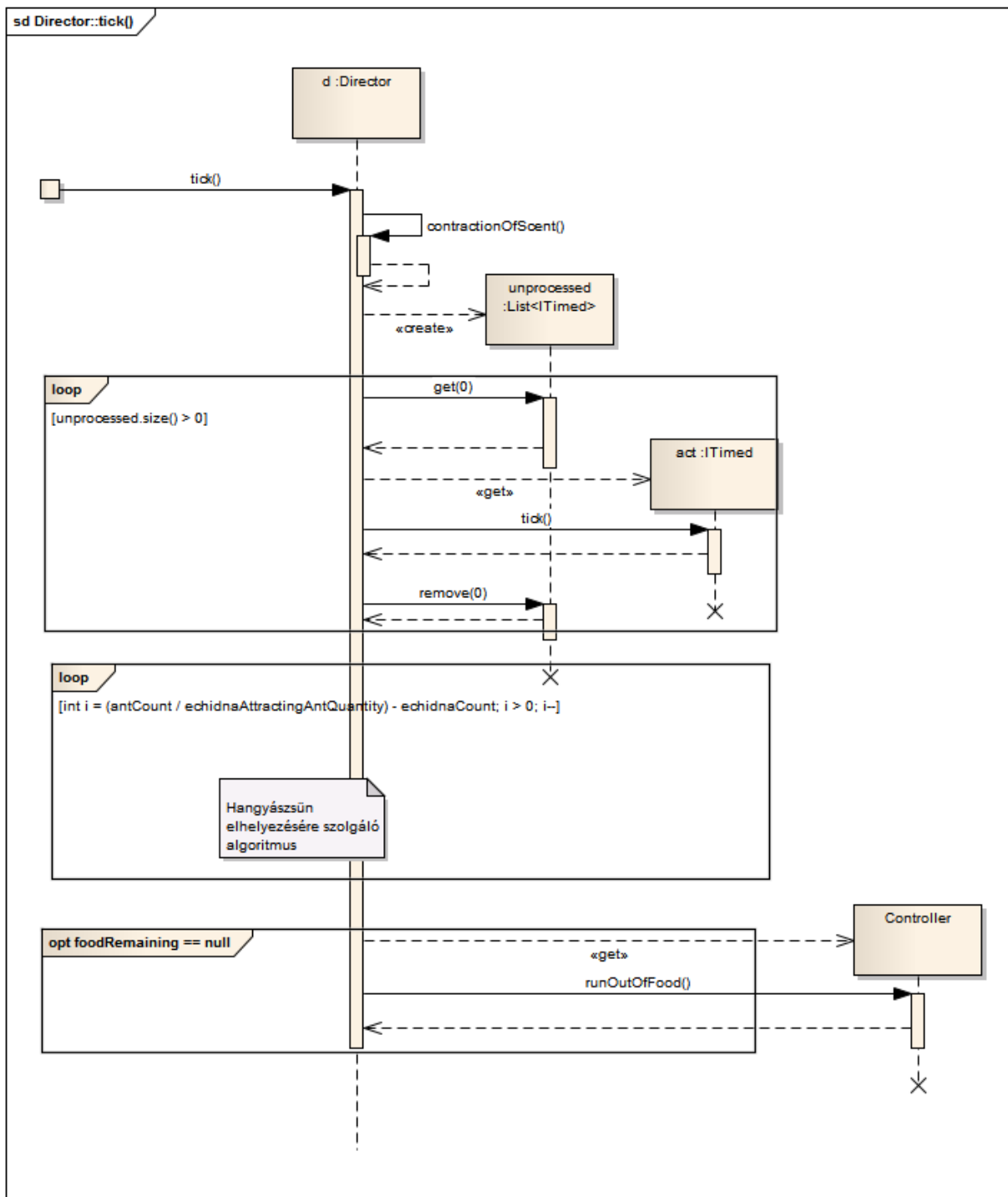
3.4.8. ScentVasinerSpray::doSpraying



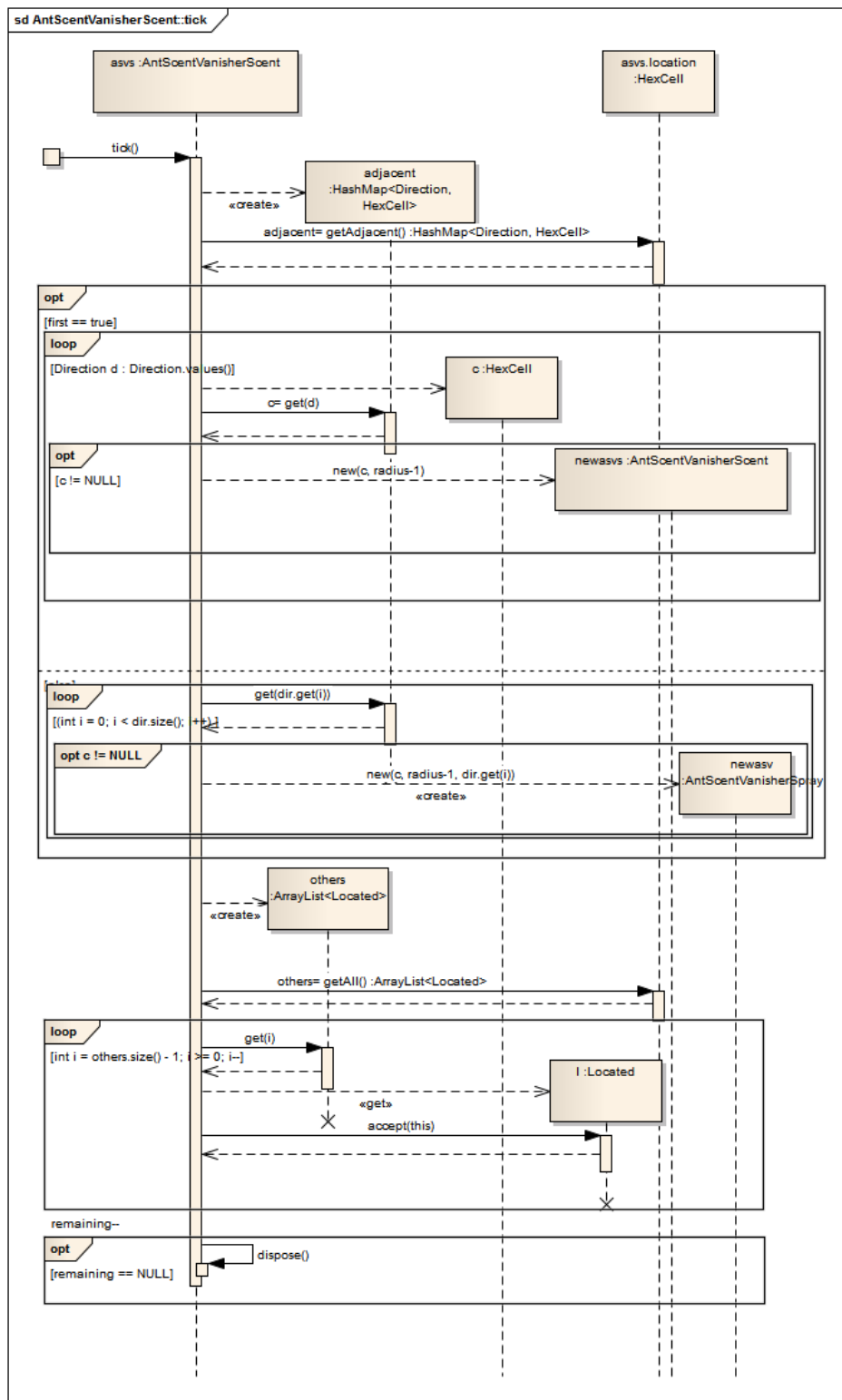
3.4.9. Wanderer:move



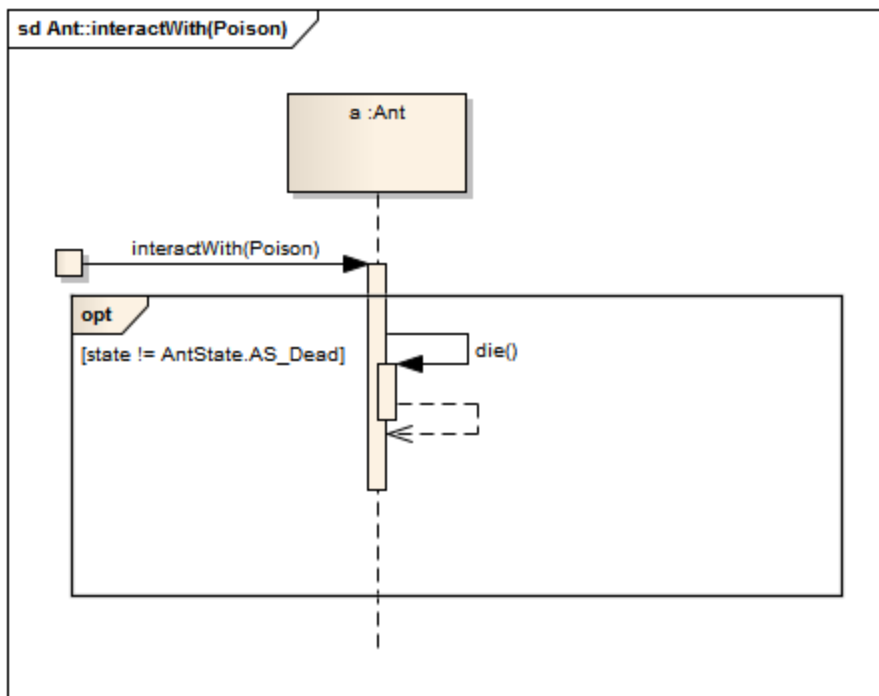
3.4.10. Director::tick()



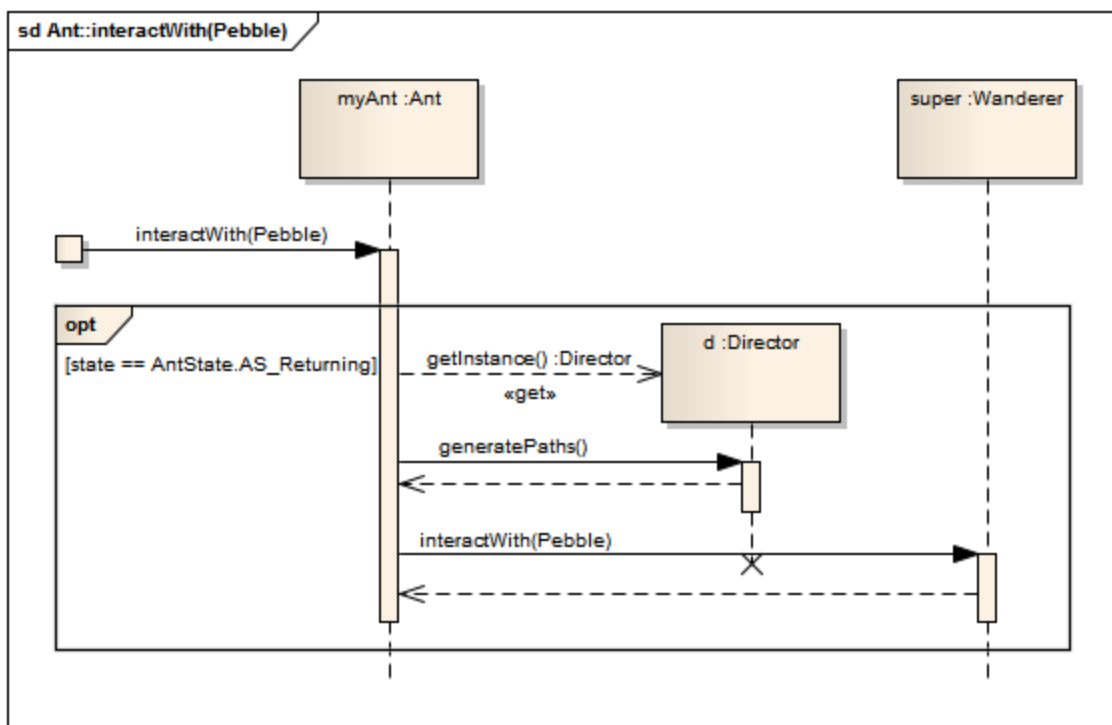
3.4.11. AntScentVanisherScent::tick()



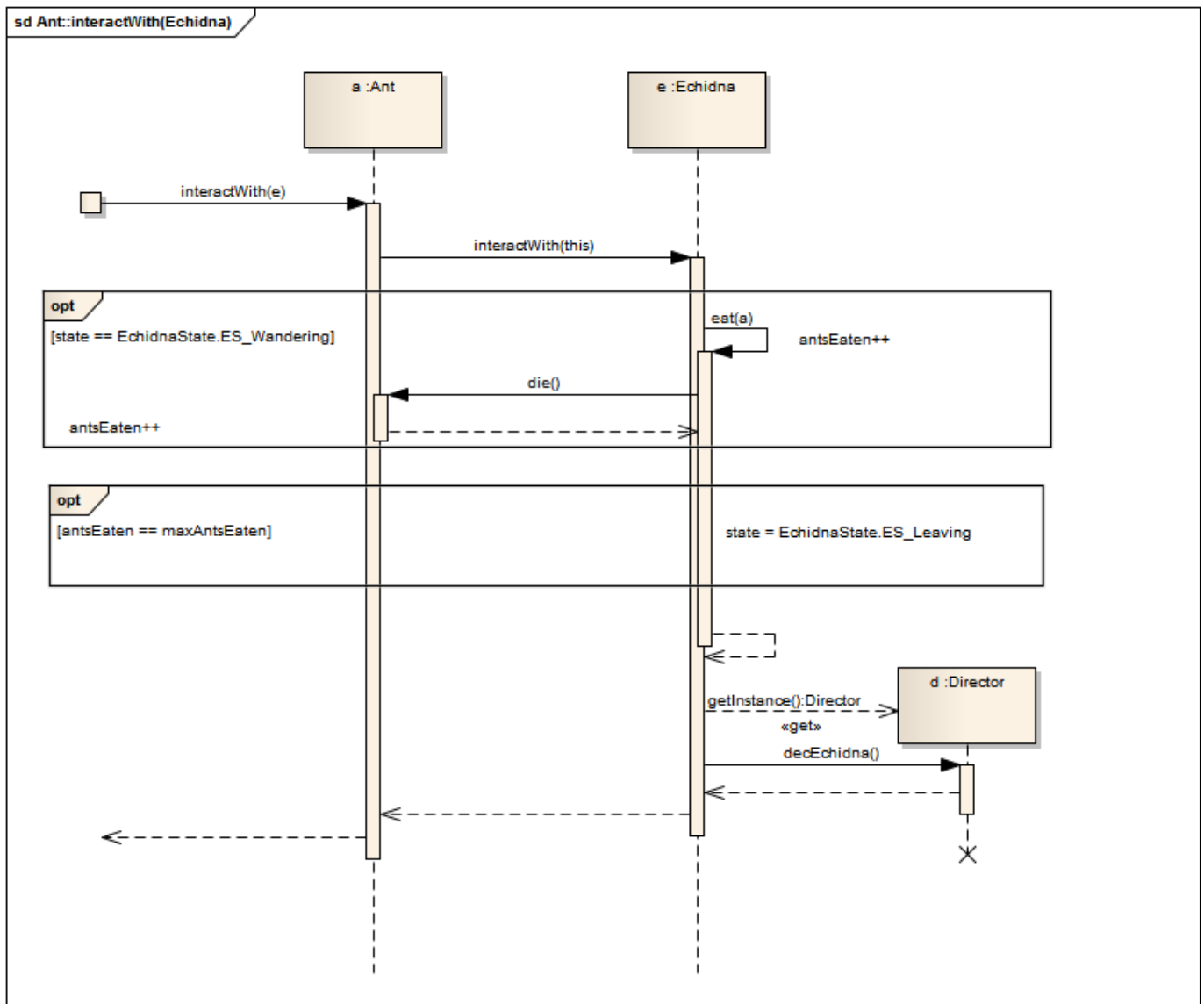
3.4.12. Ant::interactWith(Poison)



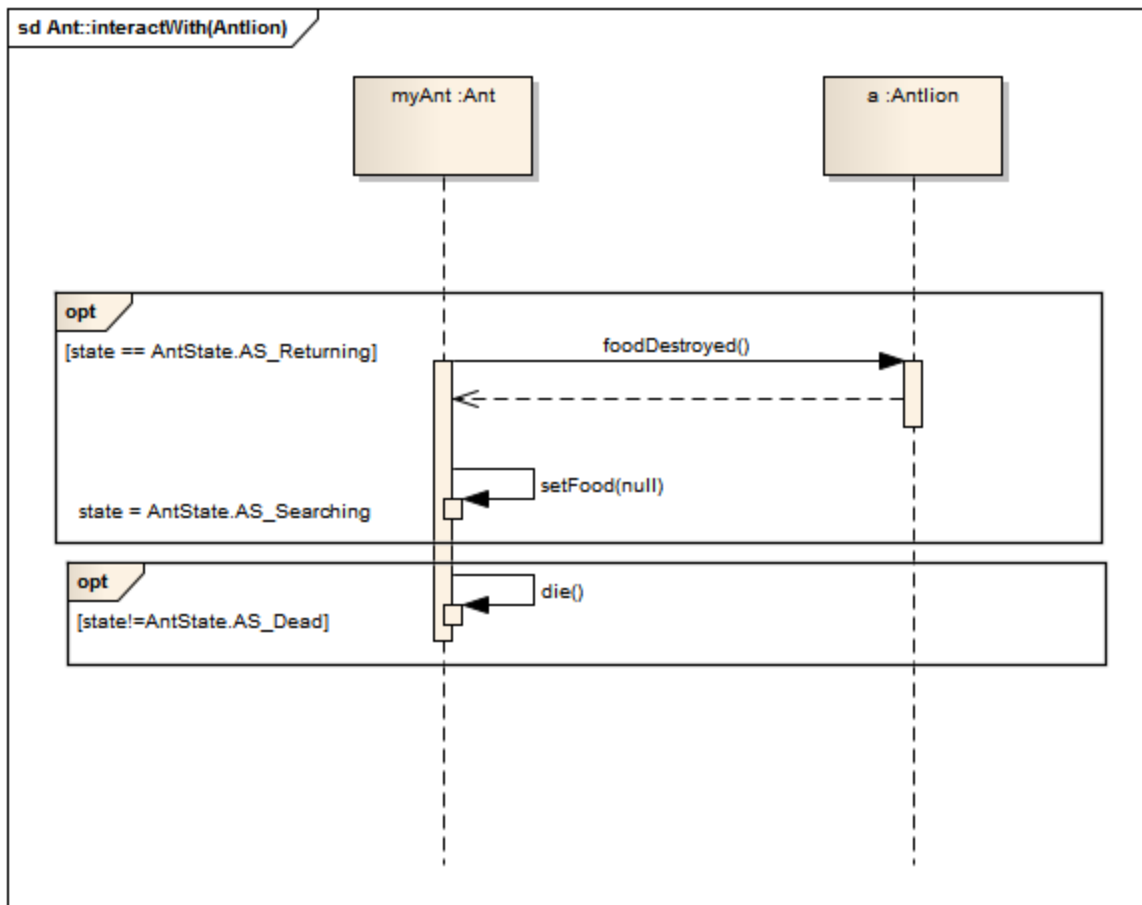
3.4.13. Ant::interactWith(Pebble)



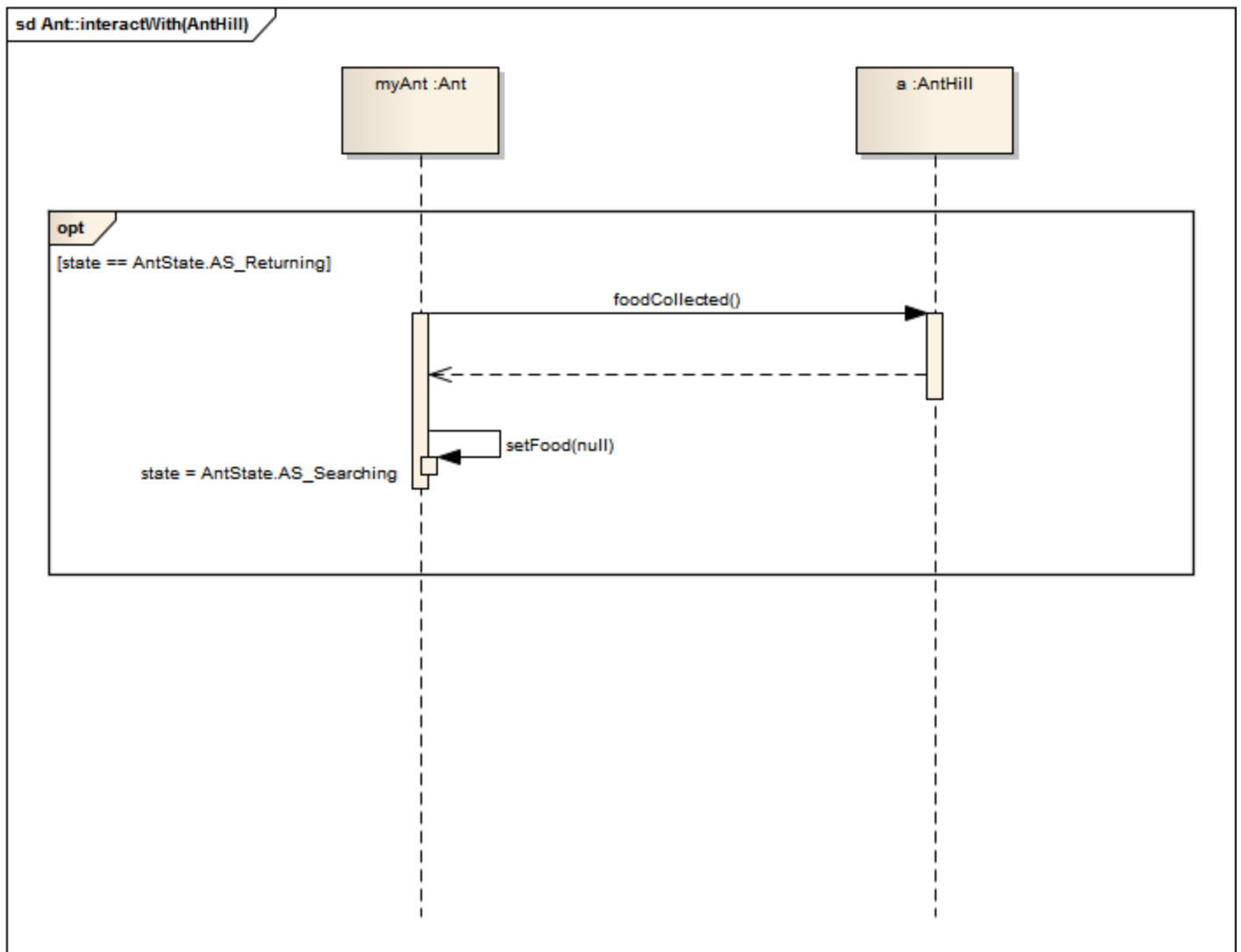
3.4.14. Ant::interactWith(Echidna) & Echidna::interactWith(Ant) & Echidna::eat()



3.4.15. Ant::interactWith(Antlion)

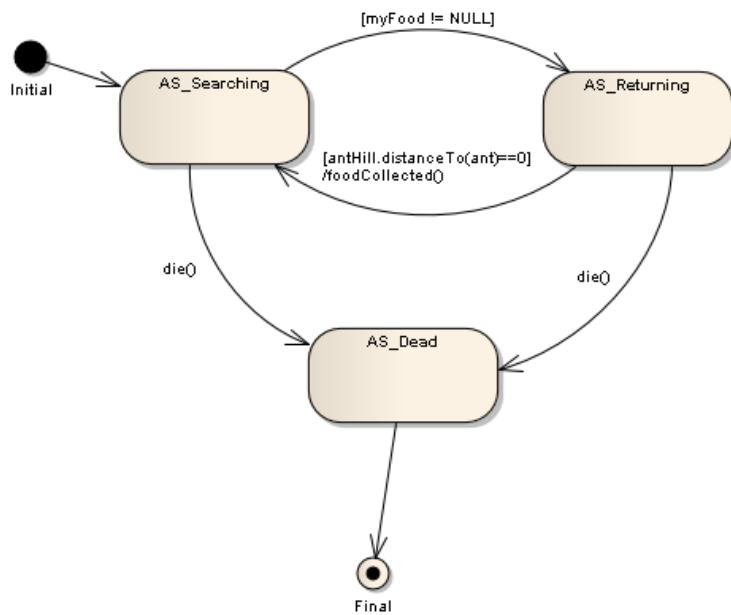


3.4.16. Ant::interactWith(AntHill)

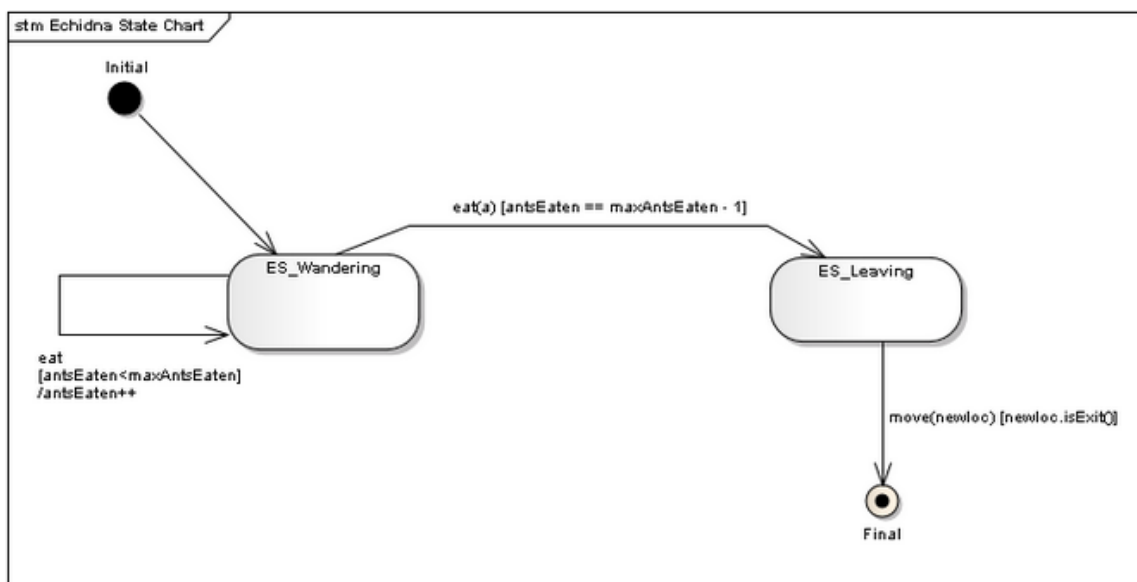


3.5. State-chartok

3.5.1 Ant életciklusa



3.5.2 Echidna életciklusa



3.6. Napló

Időpont	Időtartam	Résztevők	Eredmény
2013.02.27. 10:00:00	0,5 óra	Kozaróczy	Dokumentálás: Az első feladat hibáit Kozaróczy kijavította
2013.02.27. 11:00:00	2,5 óra	Kozaróczy Molnár Papp Tóth Varga	Megbeszéljük az az elkészítendő szoftver alapvető struktúráját. Döntés: Tóth a megbeszéltek alapján elkészíti az osztálydiagrammot.
2013.02.27. 15:30:00	2.5 óra	Molnár	Az Echidna osztály tick metódusának szekvenciadiagramja.
2013.02.27. 18:20:00	3 óra	Papp	Elkészíti az Ant osztály tick metódusának szekvenciadiagramját és az Ant életrajza állapotdiagramot.
2013.03.01. 8:45:00	0,5 óra	Kozaróczy	Kiegészíti az osztályok leírása dokumentációt.
2013.03.02. 9:00:00	1,5 óra	Papp	Elkészíti a ScentVanisherSpray.doSpraying() és a PoisonSpray.doSpraying() szekvenciadiagramokat.
2013.03.02. 15:00:00	3 óra	Varga	Elkészíti az Ant osztály die metódusának szekvenciadiagramját, tisztázza az osztálydiagrammot.
2013.03.02. 15:30:00	2 óra	Molnár	Ant osztályhoz kapcsolódó szekvenciadiagramokat elkészíti.
2013.03.02. 15:30:00	2 óra	Tóth	A Director osztály initialize metódusának szekvenciadiagramjának befejezése
2013.03.03. 14:00:00	0,5 óra	Kozaróczy	A diagram-fájlokat összefűzi egy projectbe
2013.03.03. 19:30:00	1,5 óra	Varga	3.1. Objektumkatalógus elkészítése
2013.03.03. 18:00:00	4 óra	Papp	Az Ant::move, Antlion eats, Puddle slows és Echidna::move szekvenciadiagramok megalkotása. Objektum és osztálykatalógus

			bővítése.
2013.03.03. 18:00:00	4 óra	Tóth	A cellák be- és kilépési eseményeinek integrálása az osztálydiagramba, illetve az események elsütésének és eseménykezelők regisztrációjának szekvenciadiagramjainak felépítése.
2013.03.03. 18:00:00	4 óra	Kozaróczy	Szekvenciadiagrammokat készít és a többiek munkáját összegzi, majd dokumentálja.
2013.03.03. 22:00:00	1 óra	Kozaróczy Molnár Papp Tóth Varga	A csapat összegyűlt a további teendőket megbeszélni. A jelenlegi modellel a csapat elégedett, ennek továbbfejlesztését az értékelés után, szerdán (2013.03.06-án) kezdi meg. Döntés: A csapat teljes létszámú gyűlést tart 2013.03.06.-án 10:00-tól, melyen a felmerült problémák megvitatását és orvosolását kezdik meg.
2013.03.03. 23:30:00	0,5 óra	Kozaróczy	A dokumentumot tördeli és végső javításokat tesz.

4. Analízis modell kidolgozása II

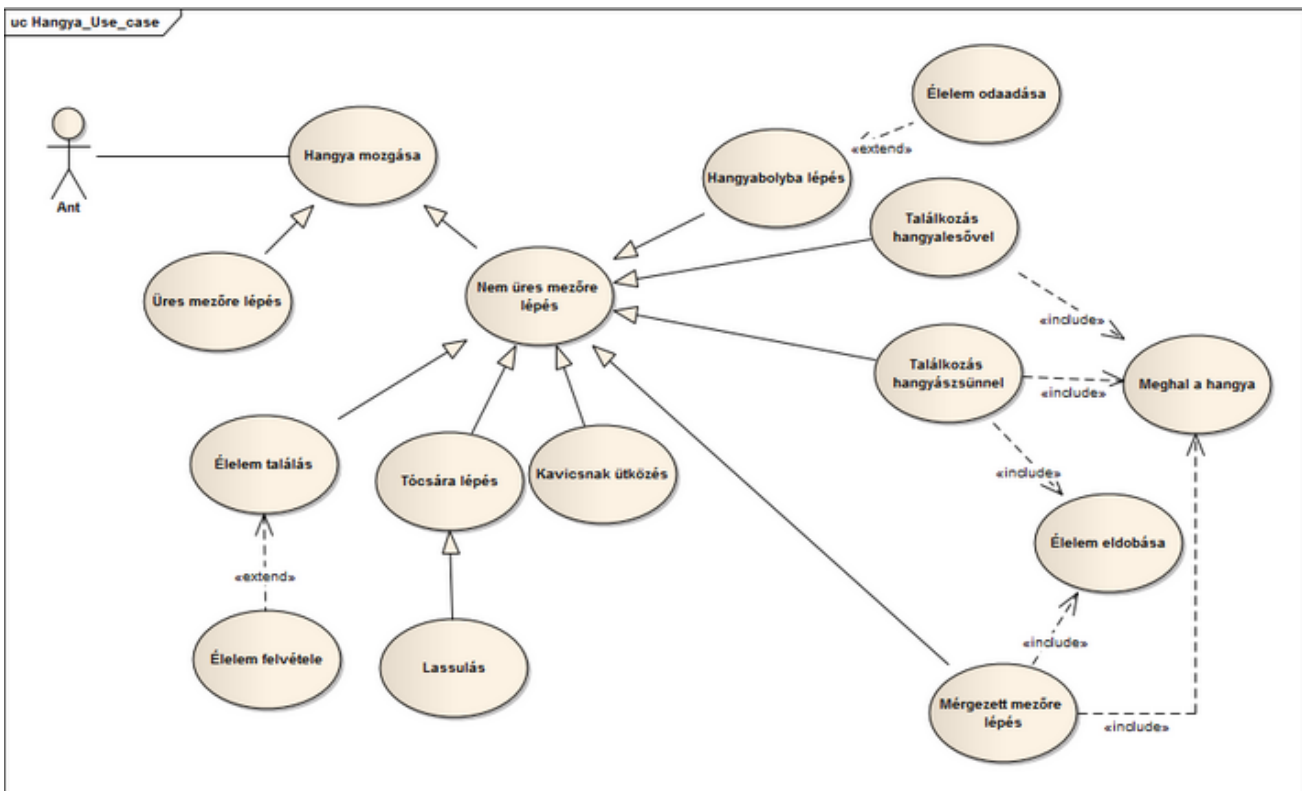
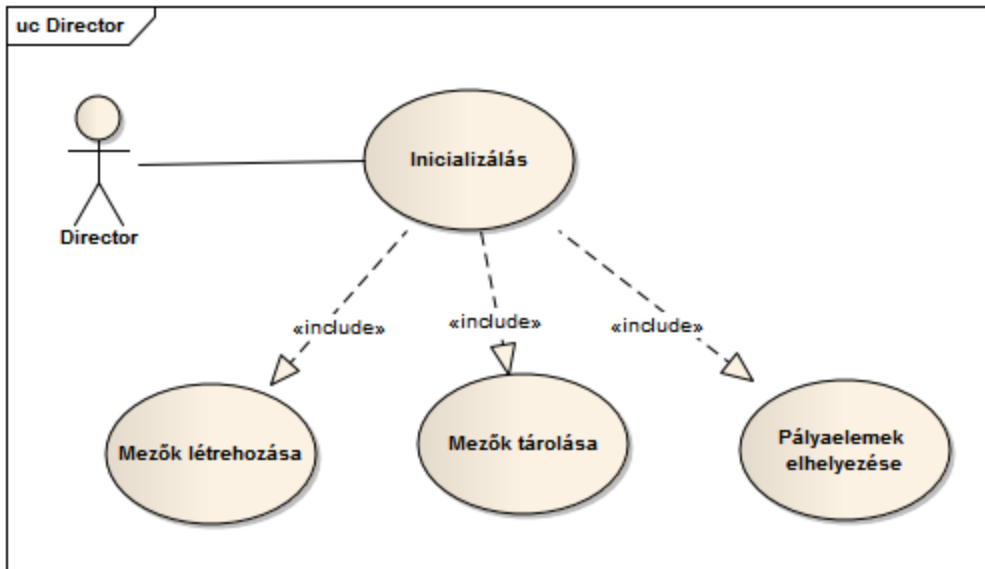
4.1. Napló

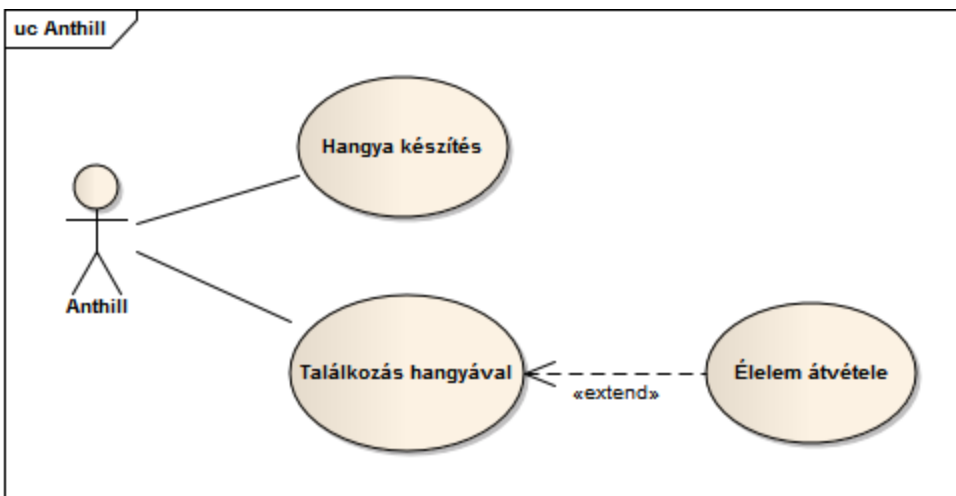
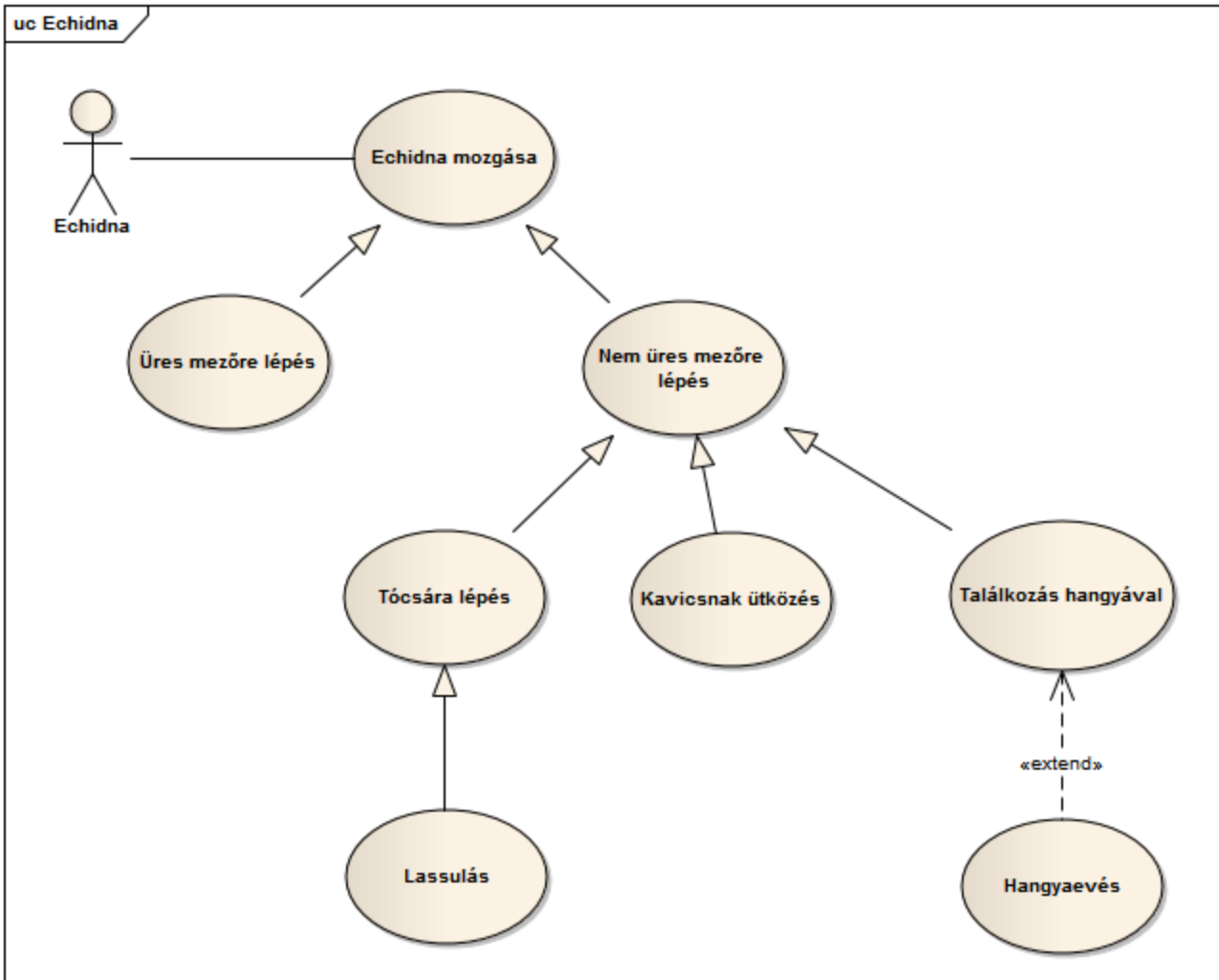
Időpont	Időtartam	Résztevők	Eredmény
2013.03.06. 12:00:00	2 óra	Kozaróczy Molnár Papp Tóth Varga	A csapat megvitatja az analízis modell 1-ben talált hibákat.
2013.03.06. 19:00:00	3 óra	Papp	Alkalmazza a Visitor patternt a program tervére. Ezen tervezési minta alkalmazása során átalakítja az osztálydiagramot. Bevezet a Wanderer őszosztályt.
2013.03.10. 16:00:00	2 óra	Varga	Echidna szekvenciadiagrammjaikat elkészíti
2013.03.10. 16:00:00	4 óra	Kozaróczy	Ant osztályhoz kapcsolódó szekvenciadiagramokat elkészíti.
2013.03.10. 17:00:00	3 óra	Tóth	A spray-ekhez kapcsolódó szekvenciadiagrammokat elkészíti.
2013.03.10. 17:00:00	2 óra	Molnár	A szekvenciadiagrammok közben felmerült javításokat felviszi az osztálydiagrammra, majd elrendezi átlátható formába.
2013.03.10. 21:00:00	2 óra	Papp	Analízis modell II kibővítése és véglegesítése A szekvenciadiagramok hibáinak kijavítása.
2013.03.10. 23:00:00	1 óra	Kozaróczy	A dokumentum végleges formázása és tördelése

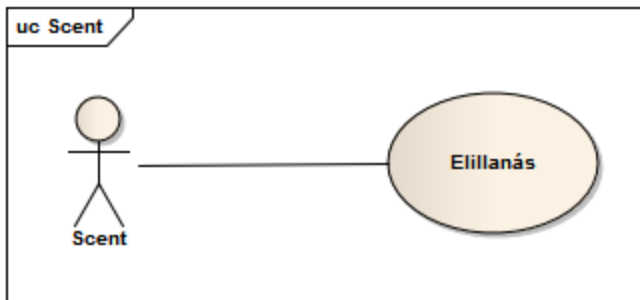
5. Szkeleton tervezése

5.1. A szkeleton modell valóságos use-case-ei

5.1.1. Use-case diagram







5.1.2. Use-case leírások

Use-case neve	Hangya mozgatása
Rövid leírás	A hangya eldönti, hogy merre lép tovább.
Aktorok	Ant
Forgatókönyv	A hangya következő lépésének kiszámítása.

Use-case neve	Nem üres mezőre lépés
Rövid leírás	A hangya olyan mezőre lép, amin akadály, vagy egy másik mozgó objektum van.
Aktorok	Ant
Forgatókönyv	A mezőn lévő objektum típusától függően reagál a hangya.

Use-case neve	Tócsára lépés
Rövid leírás	Hangya olyan mezőre lép, ahol tócsa található
Aktorok	Ant, Puddle
Forgatókönyv	A hangya sebessége felére csökken, amíg a tócsán tartózkodik

Use-case neve	Kavicsnak ütközés
Rövid leírás	A cél cellában kavics van.
Aktorok	Ant, Pebble
Forgatókönyv	Nem léphet a cellába, új úticélt kell találjon.

Use-case neve	Élelem találás
Rövid leírás	Hangya ételt tartalmazó mezőre lép
Aktorok	Ant, Food
Forgatókönyv	A hangya attól függően viselkedik, hogy visz-e már magával élelmet. (Felveszi/otthagya)

Use-case neve	Élelem felvétele
Rövid leírás	Élelemkereső hangya ételt tartalmazó mezőre lép
Aktorok	Ant, Food
Forgatókönyv	A hangya felszedi az ételt, és elindul a hangyaboly fele

Use-case neve	Lassulás
Rövid leírás	A mozgó objektum sebessége lelassul
Aktorok	Ant vagy Echidna
Forgatókönyv	A mozgó objektum sebessége felére csökken.

Use-case neve	Üres mezőre lépés
Rövid leírás	A hangya olyan mezőre lép, amin nincs se akadály, se élelem, se semmilyen egyéb pályaelem
Aktorok	Ant
Forgatókönyv	A hangya futtatja az útkereső algoritmusát, és ez alapján tovább lép

Use-case neve	Hangyabolyba lépés
Rövid leírás	A hangya olyan mezőre lép, amin hangyaboly van.
Aktorok	Ant, AntHill
Forgatókönyv	A hangya állapotától függően viselkedik - ha van nála élelem leadja - majd újra elindul.

Use-case neve	Élelem leadása
Rövid leírás	Az élelem vivő hangya hangyabolyba lépett.
Aktorok	Ant, AntHill, Food
Forgatókönyv	Leadás után tovább megy élelmet keresni.

Use-case neve	Mérgezett mezőre lépés
Rövid leírás	A hangya olyan mezőre lép, ami mérgezett
Aktorok	Ant, Poison
Forgatókönyv	A hangya meghal, amennyiben vitt élelmet, az a mezőn marad

Use-case neve	Találkozás hangyászsünnel
Rövid leírás	A hangya olyan mezőre lép, amin hangyászsün van.
Aktorok	Ant, Echidna
Forgatókönyv	A hangyászsün állapotától függően (éhes/jóllakott) a hangya meghal/továbbmegy, amennyiben meghal és volt nála élelem azt eldobja

Use-case neve	Találkozás hangyalesővel
Rövid leírás	A hangya hangyalesőt tartalmazó mezőre lép.
Aktorok	Ant, Antlion
Forgatókönyv	A hangya meghal.

Use-case neve	Pálya inicializálása
Rövid leírás	A játék kezdetekor lefutó folyamatok
Aktorok	Director
Forgatókönyv	Mezők létrehozása, és eltárolása, pályaelemek betöltése

Use-case neve	Mezők létrehozása
Rövid leírás	Az összes mező létrehozása
Aktorok	Director
Forgatókönyv	Legenerálódnak a mezők.

Use-case neve	Mezők eltárolása
Rövid leírás	A mezőket eltároljuk és összekötögetjük.
Aktorok	Director
Forgatókönyv	Létrejönnek a szomszédsági relációk és az útvonalak a kijáratokig.

Use-case neve	Pályaelemek elhelyezése
Rövid leírás	A pályaelemeket legeneráljuk és mezőkre helyezzük.
Aktorok	Director
Forgatókönyv	A pályaelemeket tartalmazó mezők megismerik tartalmukat és legenerálódnak az útvonalak a hangyabolyig.

5.2. A szkeleton kezelői felületének terve, dialógusok

A szkeleton bemeneteit csak számok képezhetik majd, amikkel egy felsorolásból lehet kiválasztani az előre megírt szekvenciákat. A kimeneten szerepelni fog mindig, hogy melyik osztály melyik függvénye hívódik meg (nem felüldefiniált, örökölt függvények esetében a szülő fog szerepelni), illetve hogy mi mikor tér vissza. A szkeleton által futtatott program egy 2 HexCell-ből álló mezőn fog játszódni, ahol a bemenet függvényében dől el, hogy mik szerepelnek a mezőkön és hogy azok mit csinálnak.

5.2.1 Szkeleton bemenetek

Sorszám	Mezők tartalma	use-case
0	-	Kilépés
1	-	Pálya inicializálása
2	hangya	Hangya mozgása üres cellába
3	kereső hangya, étel	Kereső hangya ételt talál
4	visszatérő hangya, étel	Visszatérő hangya ételt talál
5	kereső hangya, hangyászszün	Kereső hangya hangyászszünnel találkozik
6	visszatérő hangya, hangyászszün	Visszatérő hangya hangyászszünnel találkozik
7	kereső hangya, mérég	Kereső hangya mérégbe lép
8	visszatérő hangya, mérég	Visszatérő hangya mérégbe lép
9	kereső hangya, hangyaleső	Kereső hangya hangyalesővel találkozik
10	visszatérő hangya, hangyaleső	Visszatérő hangya hangyalesővel találkozik
11	hangya, tócsa	Hangya tócsán úszik át
12	hangya, kavics	Hangya kavicsra mozogna
13	kereső hangya, hangyaboly	Kereső hangya megtalálja a hangyabolyt
14	visszatérő hangya, hangyaboly	Visszatérő hangya megtalálja a hangyabolyt
15	hangyászszün	Hangyászszün üres mezőre lép

16	kóricáló hangyászsün, kereső hangya	Kóricáló hangyászsün kereső hangyával találkozik
17	kóricáló hangyászsün, visszatérő hangya	Kóricáló hangyászsün visszatérő hangyával találkozik
18		XXX
19	távozó hangyászsün, hangya	Távozó hangyászsün hangyával találkozik
20	hangyászsün, tócsa	Hangyászsün tócsán úszik át
21	hangyászsün, kavics	Hangyászsün kavicsra mozogna
22	hangyaszag-semlegesítő szag, hangyaszag	Hangyaszagsemlegesítő spray hangyaszagra hat
23	hangyaszag-semlegesítő szag	Hangyszagsemlegesítő spray fújása
24	méreg	Méreg fújása
25	hangyaszag-semlegesítő szag	Hangyaszagsemlegesítő terjedése
26	méreg	Méreg terjedése

5.2.2 Kimenetek a megfelelő bemenetekre

A szekvenciadiagramoknak megfelelő sorrendben ehhez hasonló kimenetek fognak keletkezni:

```

Food::accept (IInterActor)
Ant::interactWith (Food)
HexCell::removeObject (Food)
ret HexCell::removeObject (Food)
ret Ant::interactWith (Food)
ret Food::accept (IInterActor)

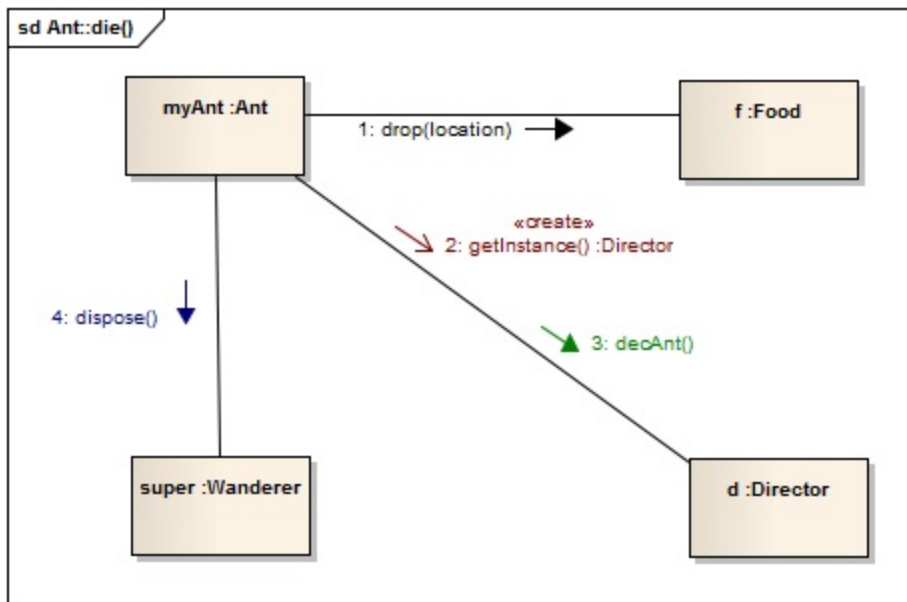
```

5.3. Szekvencia diagramok a belső működésre

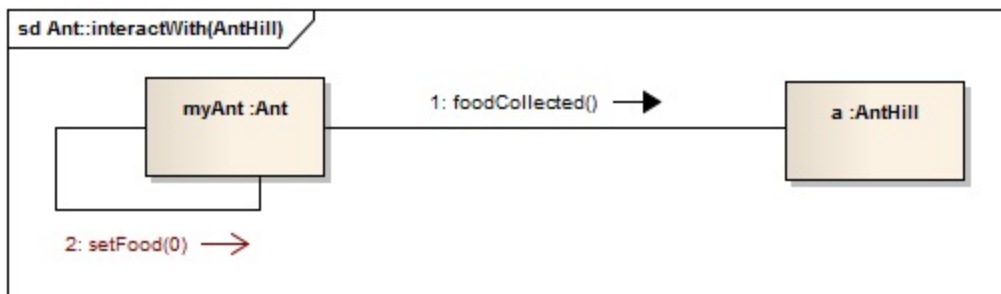
A use-case-ek az előző dokumentumban leadott viselkedéseket valósítják meg?

5.4. Kommunikációs diagramok

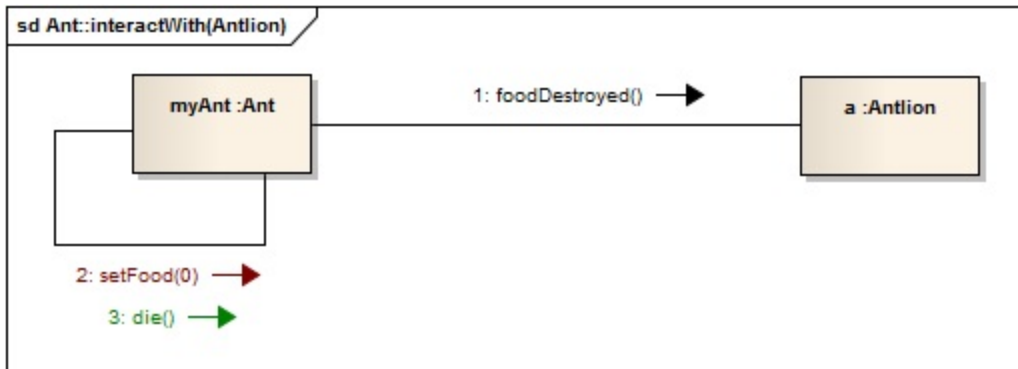
5.4.1 Ant::die()



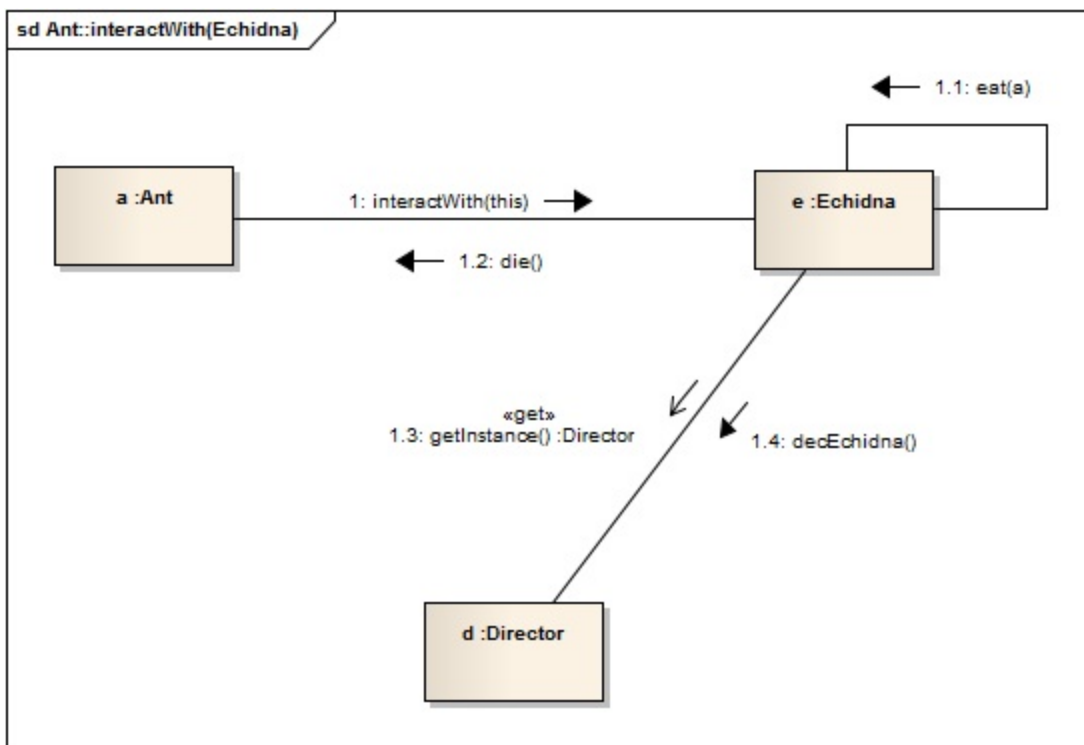
5.4.2 Ant::interactWith(AntHill)



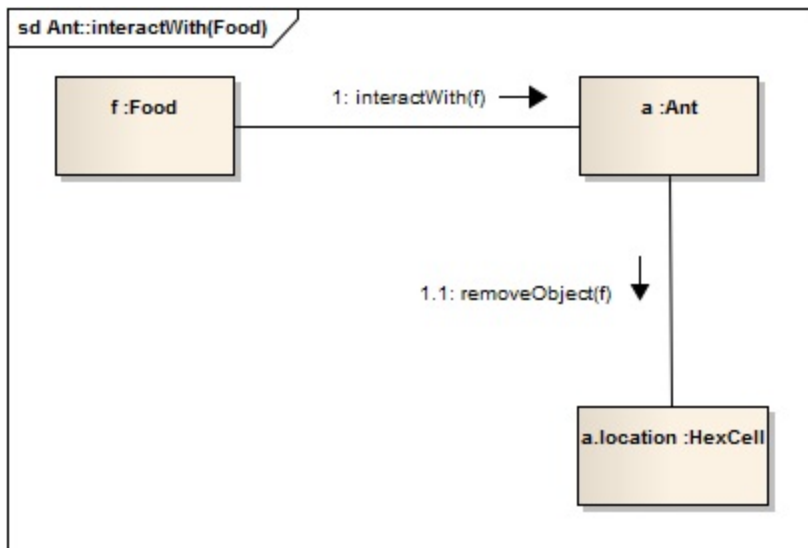
5.4.3 Ant::interactWith(Antlion)



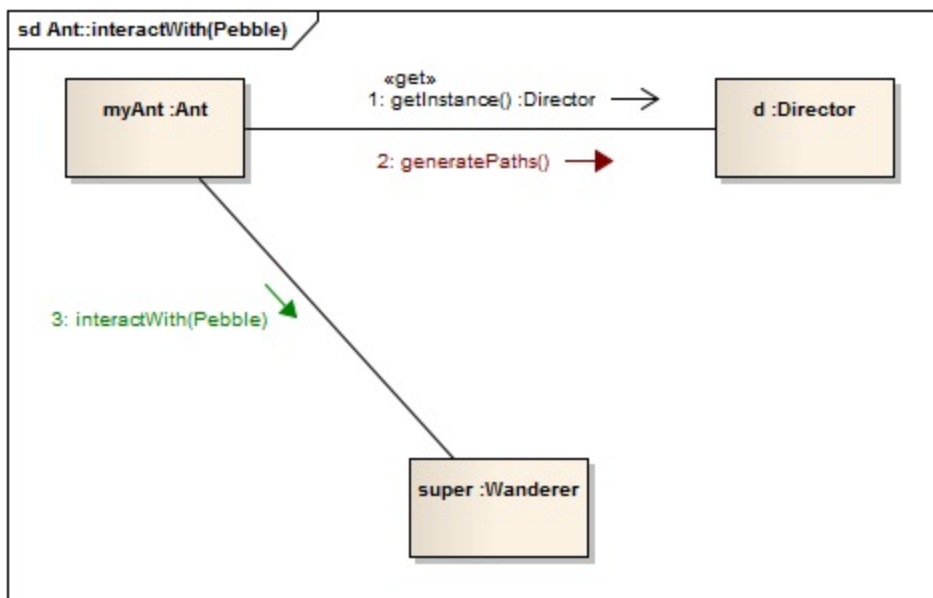
5.4.4 Ant::interactWith(Echidna)



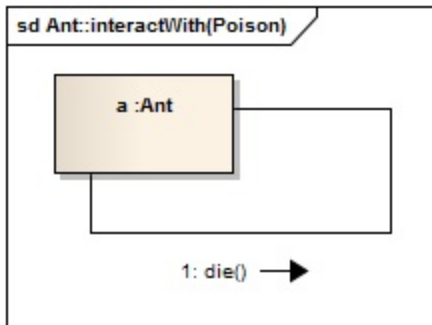
5.4.5 Ant::interactWith(Food)



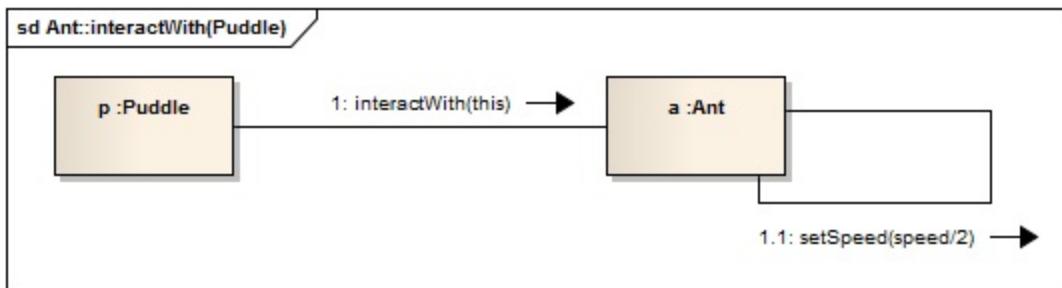
5.4.6 Ant::interactWith(Pebble)



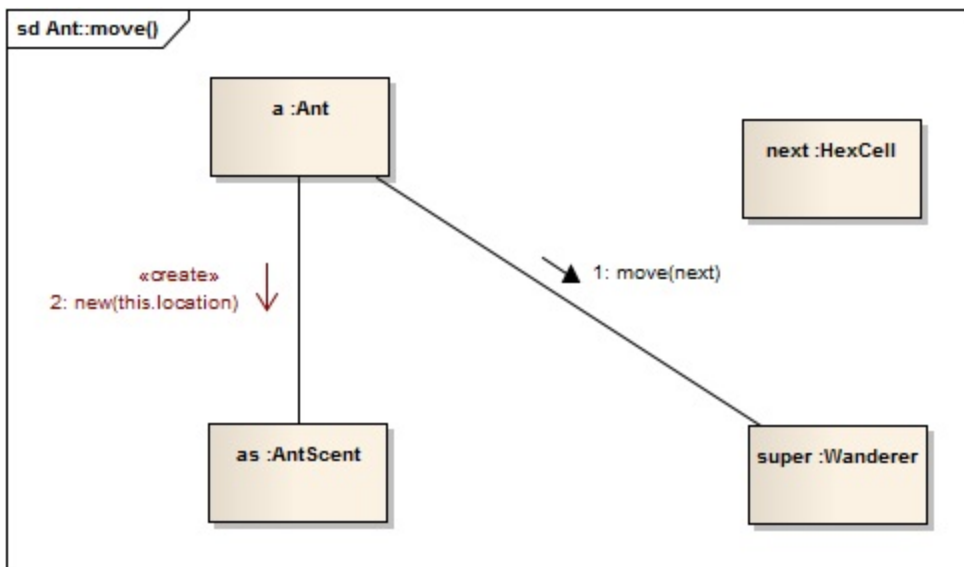
5.4.7 Ant::interactWith(Poison)



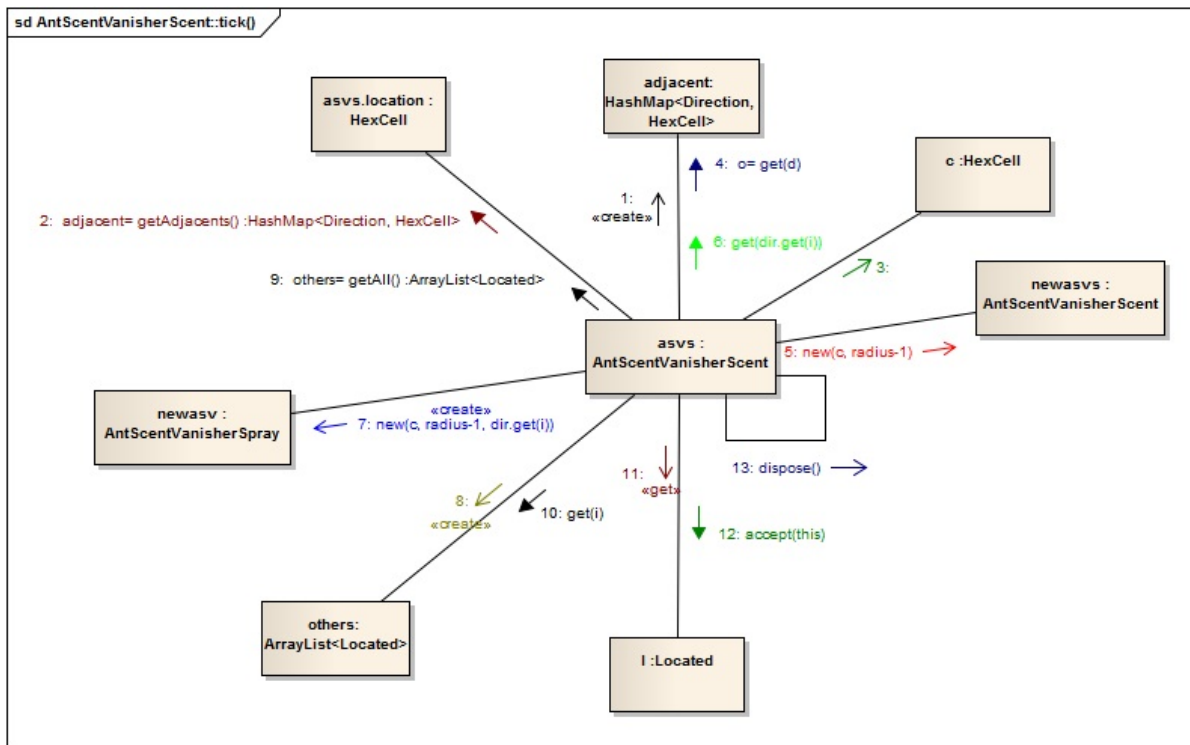
5.4.8 Ant::interactWith(Puddle)



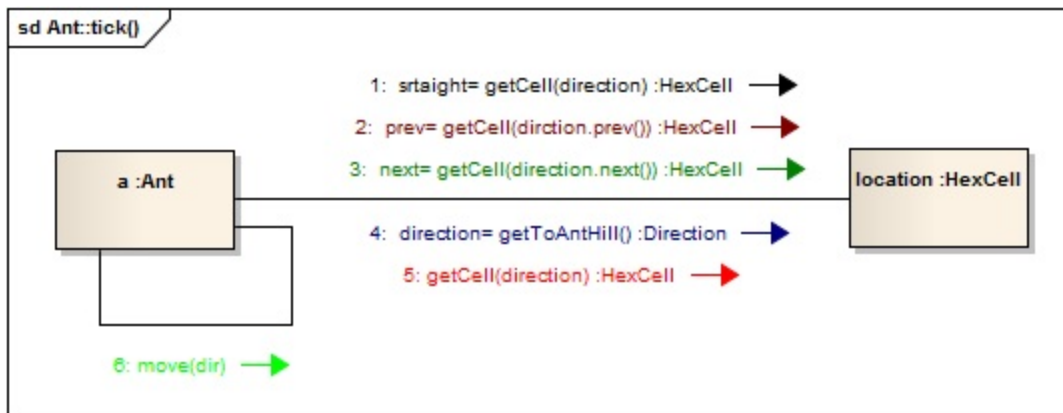
5.4.9 Ant::move()



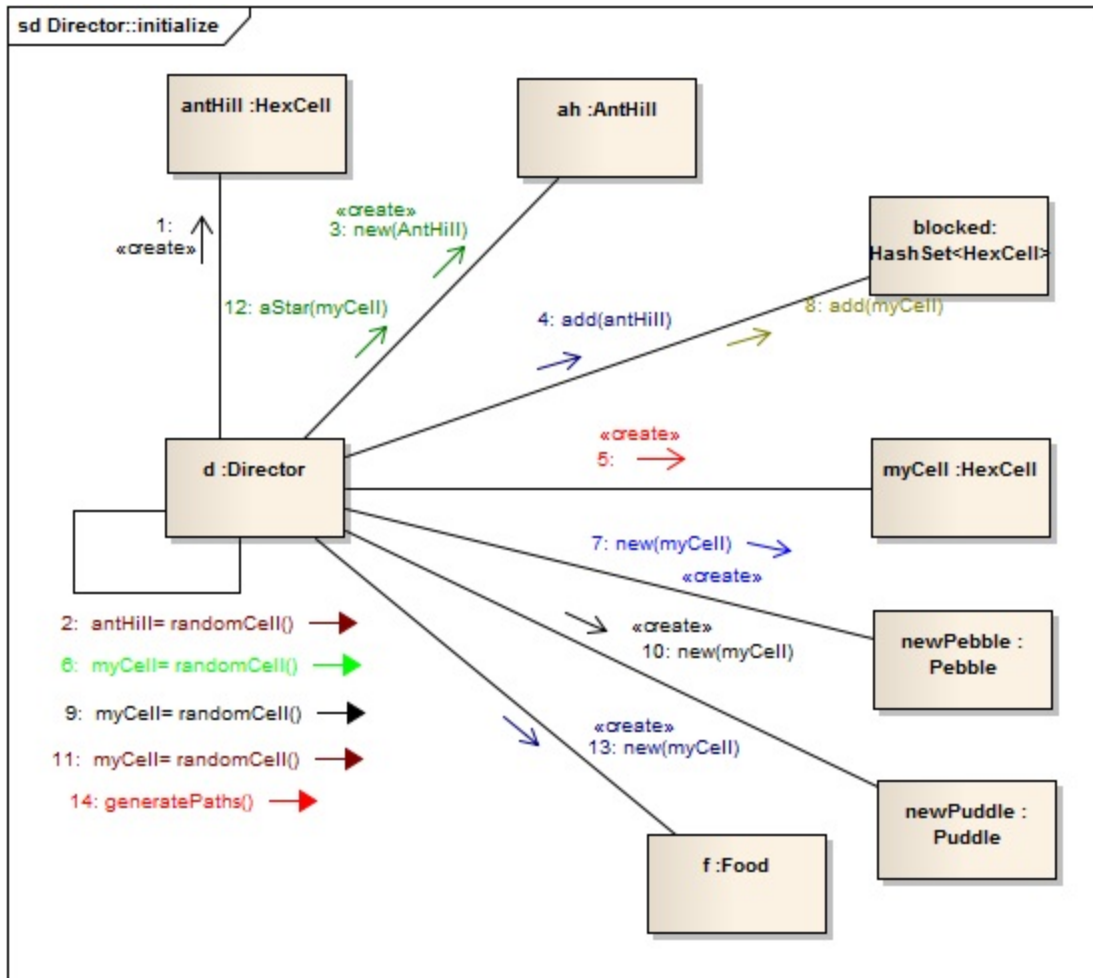
5.4.10 AntScentVanisherScent::tick()



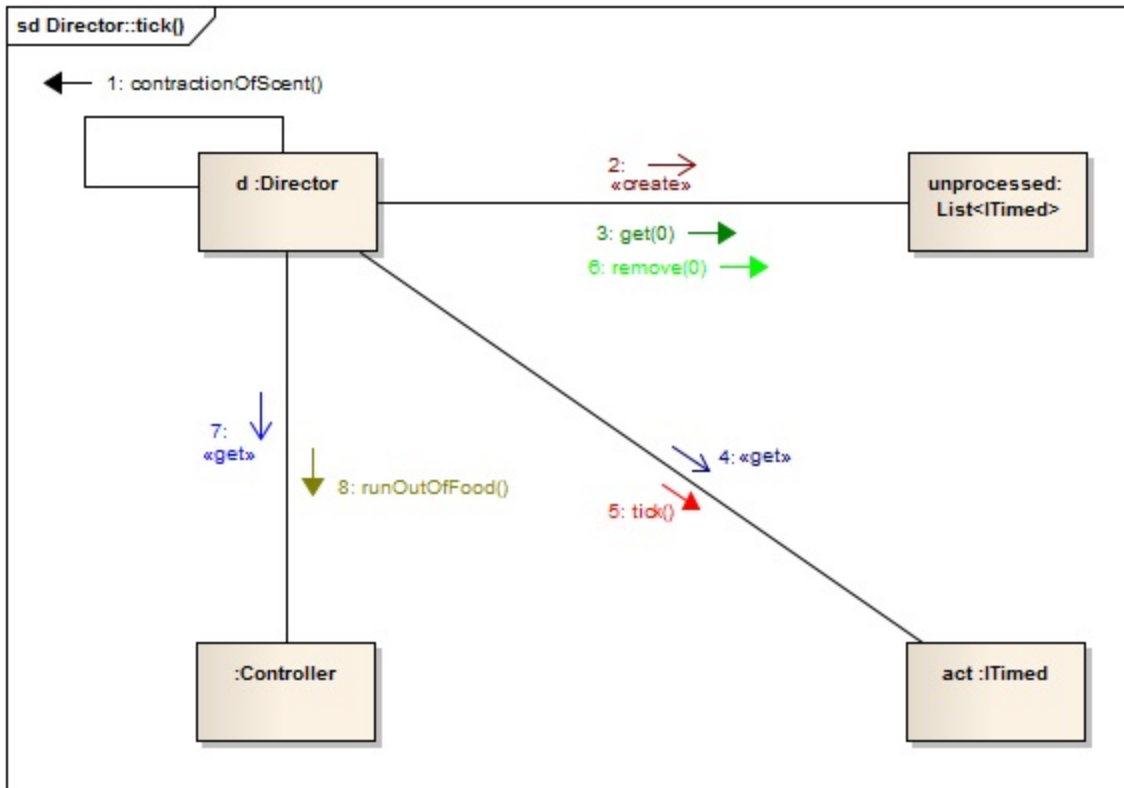
5.4.11 Ant::tick()



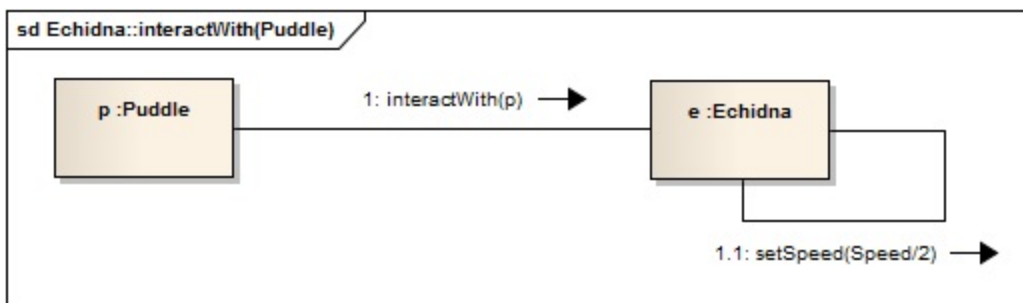
5.4.12 Director::initialize



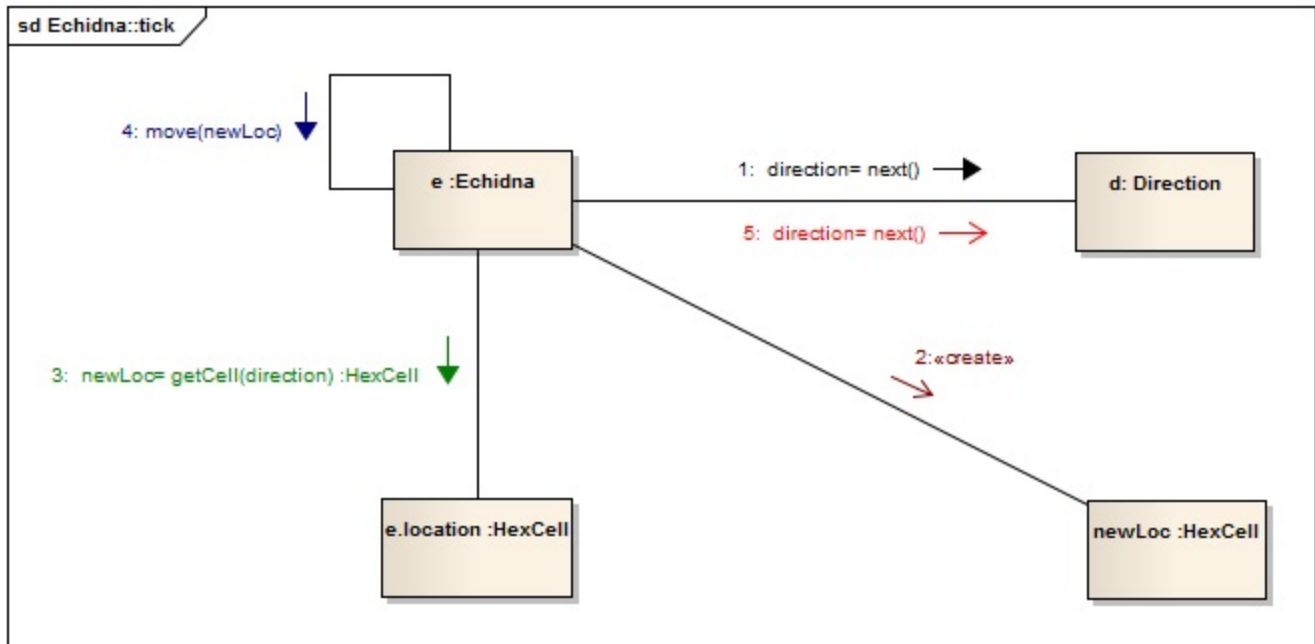
5.4.13 Director::tick()



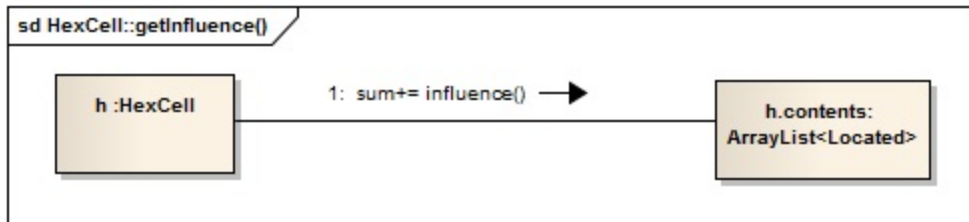
5.4.14 Echidna::interactWith(Puddle)



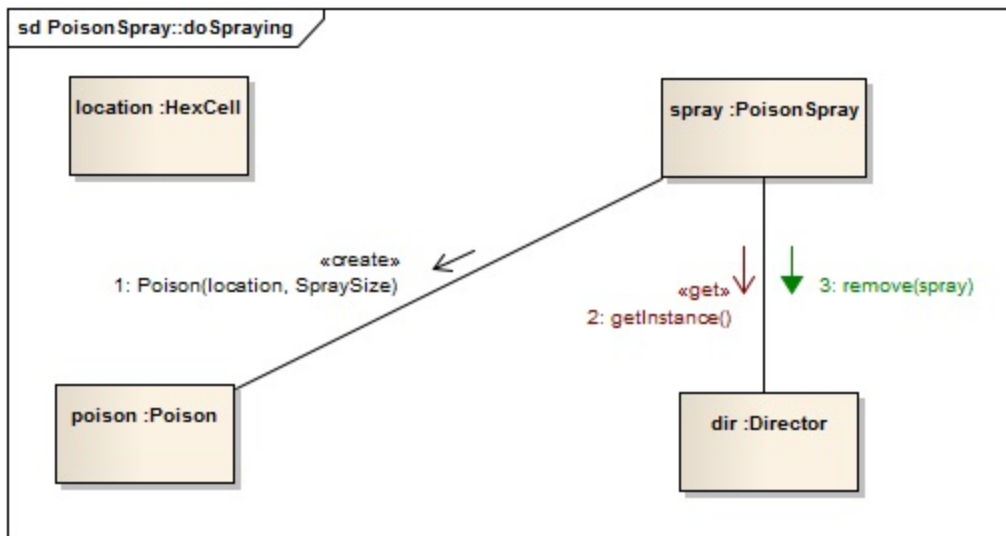
5.4.15 Echidna::tick()



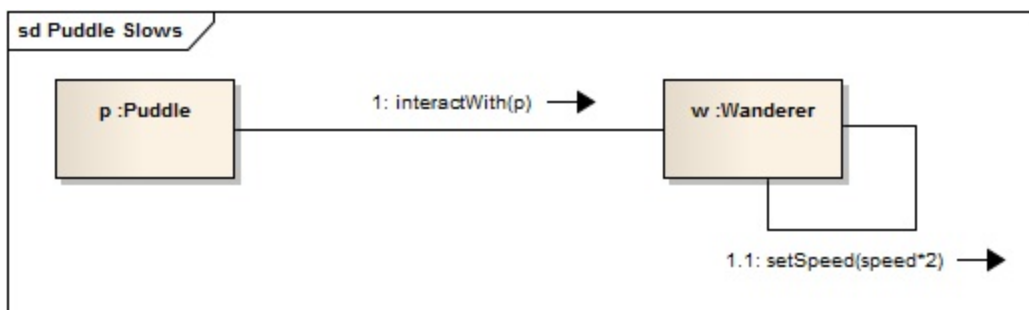
5.4.16 HexCell::getInfluence()



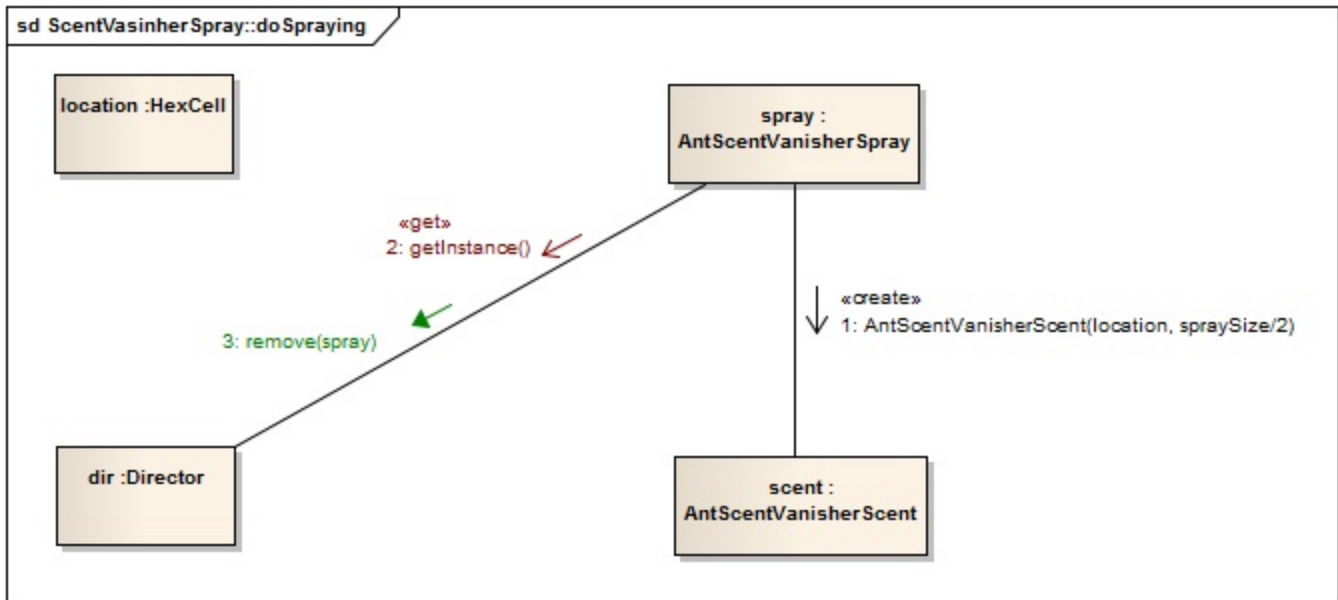
5.4.17 PoisonSpray::doSpraying()



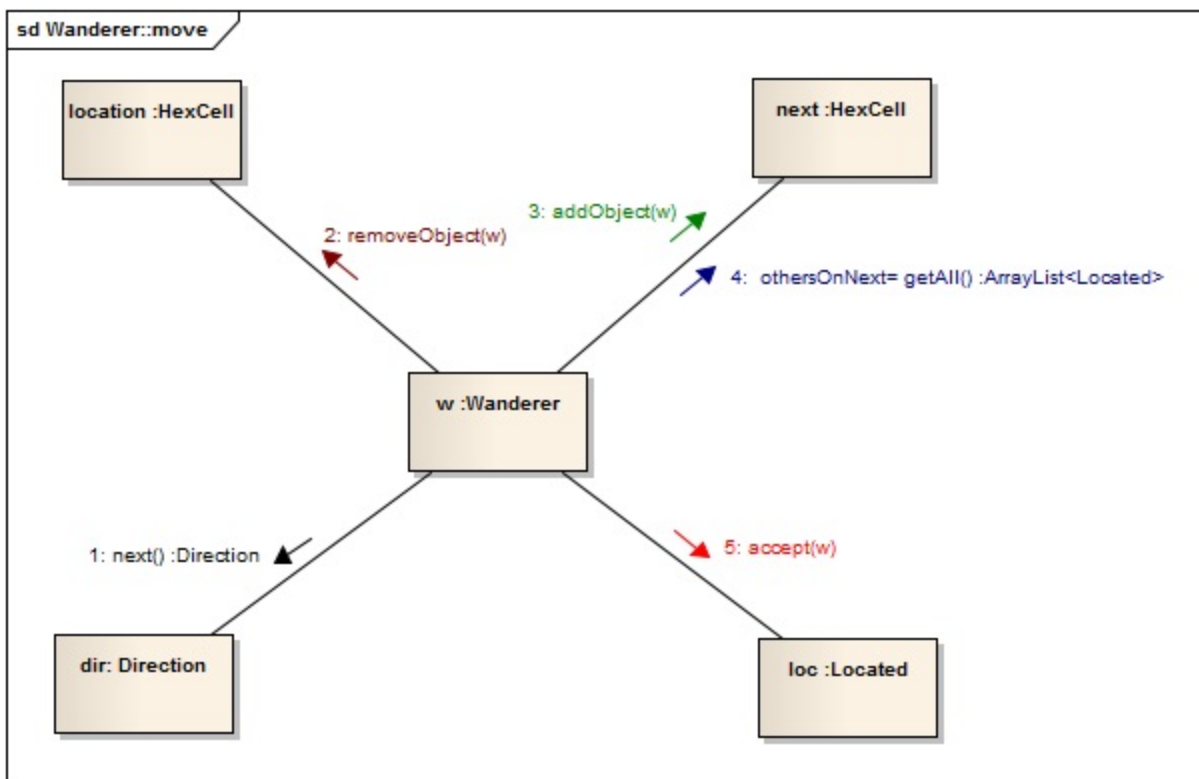
5.4.18 Puddle Slows



5.4.19 ScentVanisherSpray::doSpraying()



5.4.20 Wanderer::move()



5.5 Napló

Időpont	Időtartam	Részvevők	Eredmény
2013.03.13. 11:00:00	2 óra	Varga	Use-case-ek összeírása és leírások elkészítése.
2013.03.13. 21:00:00	1,5 óra	Varga	Use-case diagramok elkészítése, leírások frissítése/módosítása a diagramok alapján.
2013.03.17. 14:30:00	1 óra	Kozaróczy	Megírja az Ant osztály szkeletonját, lefixálja, mi fog megjelenni a kijelzőn.
2013.03.17. 18:00:00	0,5 óra	Kozaróczy	A Direction enumhoz új függvényeket talál ki és implementálja a szkeletonba.
2013.03.18. 10:00:00	2 óra	Kozaróczy	Kiegészíti a szkeleton bemeneteket, kijavítja a use-case diagrammokat, átnézi az eddigi munkát, majd megformázza a dokumentumot
2013.03.18. 10:00:00	2 óra	Tóth	Kommunikációs diagramok megszerkesztése.

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Minden fájl a megfelelő osztályt tartalmazza

Időpont	Méret(kB)	Fájlnev
2013.03.25. 13:29	4`733	Ant.java
2013.03.25. 12:36	1`499	AntHill.java
2013.03.25. 12:52	1`110	Antlion.java
2013.03.25. 12:53	1`219	AntScent.java
2013.03.25. 12:59	3`596	AntScentVanisherScent.java
2013.03.25. 13:02	635	AntScentVanisherSpray.java
2013.03.25. 13:03	382	Barrier.java
2013.03.25. 13:06	3`851	Director.java
2013.03.25. 13:08	2`454	Echidna.java
2013.03.25. 12:34	970	Food.java
2013.03.25. 13:08	1`104	FoodScent.java
2013.03.25. 13:10	6`437	HexCell.java
2013.03.25. 13:12	276	IAcceptor.java
2013.03.25. 13:13	1`209	IInterActor.java
2013.03.25. 13:13	264	ITimed.java
2013.03.25. 13:32	1`315	Located.java
2013.03.25. 13:16	3`807	Logger.java
2013.03.25. 13:32	13`794	Main.java
2013.03.25. 13:17	596	NonBlocking.java
2013.03.25. 12:35	646	Obstacle.java
2013.03.25. 13:18	617	Pebble.java
2013.03.25. 13:19	1`579	Poison.java
2013.03.25. 12:35	626	PoisonSpray.java
2013.03.25. 12:35	696	Puddle.java
2013.03.25. 12:35	943	Scent.java
2013.03.25. 12:35	207	SingletonException.java
2013.03.25. 12:35	848	Spray.java
2013.03.25. 13:20	4`454	Wanderer.java

6.1.2 Fordítás

A program helyes lefordításához a következőket kell tenni:

1. A kapott .zip állomány kicsomagolása.
2. A javac és java futtatható állományok a \$PATH környezeti változóba való felvétele a következő paranccsal: `set path=%path%;C:\Program Files\Java\jdk1.7.0_11\bin`
A verziószámot értelem szerűen ki kell cserélni, ha szükséges.
3. A parancssorban a kitömörített állomány mappájába kell navigálni.
4. Fordítás a következő paranccsal: `javac *.java`

6.1.3 Futtatás

A szkeletonnak a következőkre van szüksége a futáshoz:

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
EAG001	1 GB RAM	kiértékelés	alapvető	csapat	
EAG002	1 GHz CPU órajel	kiértékelés	alapvető	csapat	
EAG003	JRE 1.6 / frissebb	kiértékelés	alapvető	csapat	
EAG004	Perifériák	kiértékelés	alapvető	csapat	Input: billentyűzet Output: monitor

Ezekon kívül szükséges, hogy a programot tartalmazó mappánál a programnak írási joga legyen a logfájl létrehozása miatt.

6.2 Értékelés

Tag neve	Munka százalékban	Munkaóra összesítés	Alíírás
Kozaróczy Zsolt	21%	44 óra	
Molnár Tamara	18%	21 óra	
Papp Győző	21%	34 óra	
Tóth Ákos	21%	31 óra	
Varga Éva Árnika	19%	30 óra	

6.3 Napló

Időpont	Időtartam	Résztevők	Eredmény
2013.03.20. 16:30:00	0,5 óra	Kozaróczy Molnár Papp Tóth Varga	Gyűlés, kiosztottuk a feladatokat.
2013.03.20. 16:30:00	3 óra	Kozaróczy	A use-case-ekhez előre beégetett pályarészletek lekódolása
2013.03.20. 16:30:00	3,5 óra	Papp	A logolási rendszer megtervezése és a Logger osztály elkészítése. Néhány osztály metódusának javítása.
2013.03.20. 20:00:00	1 óra	Papp	Dokumentálás, a szkeleton átnézése
2013.03.21. 11:00:00	3 óra	Kozaróczy	Újabb use-case-ekhez tartozó pályarészletek lekódolása, logolást hívó függvények elhelyezése
2013.03.21. 13:00:00	1 óra	Papp	Logger osztály bővítése, pár hiba javítása az eddigi szkeletonban.
2013.03.21. 15:00:00	1,5 óra	Kozaróczy	Hibákat javít és függvénytörzseket ír

2013.03.23. 9:30:00	4,5 óra	Vara	Szekvenciadiagrammok nagyrészének újraszerkesztése a szkeleton alapján
2013.03.23. 23:20:00	2 óra	Varga	Szekvenciadiagrammok befejezése, hiányzó use-case-ek elkészítése.
2013.03.24. 16:00:00	6 óra	Molnár	Kommunikációs diagrammok kijavítása, függvények dokumentálása
2013.03.25. 10:00:00	3 óra	Varga	Szekvenciadiagrammok, kommunikációs diagrammok javítása, dokumentálás.
2013.03.25. 10:00:00	3 óra	Kozaróczy	JavaDoc kijavítása, leadandó dokumentumok és fájlok véglegesítése

7. Prototípus koncepciója

7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A program futtatásakor az alábbi kapcsolókat használhatjuk

-in

- Leírás: A bemeneti fájl neve
- Opciók: Utána stringként meg kell adni a fájl nevét. Ha a kapcsolót nem használjuk, parancssorba kell beírni a kezdeti pályaleírást

-log

- Leírás: A logolás helye
- Opciók: Utána stringként meg kell adni a fájl nevét. Ha a kapcsolót nem használjuk, parancssorba írja ki a függvényhívásokat

-rnd

- Leírás: A randomizálás bekapcsolása
- Opciók: Nincsenek további opciók, ha a kapcsolót alkalmazzuk, nem lesz determinisztikus a működés

-out

- Leírás: A kimeneti fájl neve
- Opciók: Utána stringként meg kell adni a fájl nevét. Ha a kapcsolót nem használjuk, parancssorba írja ki a teszt végeztével a pálya állapotát

7.1.2 Bemeneti nyelv

Az alábbi parancsokkal állítható elő egyedi tesztet:

width <x> : A pálya szélessége

height <y> : A pálya magassága

generate : Legenerálja magától az adott méretű pályát (nem determinisztikus)

add <x> <y> [anthill|antlion|food|pebble|puddle] : hozzáad egy elemet a kívánt helyre

add <x> <y> ant [n|ne|se|s|sw|nw] <hasFood=0> <speed=2> : Hozzáad egy hangyát az adott helyre a megfelelő paraméterekkel. Alapértelmezésben észak felé megy étel nélkül, 2 sebességgel.

add <x> <y> antscent <remaining=15> : Hozzáad egy hangyaszagot a megfelelő paraméterekkel. Alapértelmezésben 15 marad belőle.

add <x> <y> antscentvanisher <size> : Hozzáad egy hangyaszagsemlegesítő szagot a megfelelő paraméterekkel.

add <x> <y> echidna [n|ne|se|s|sw|nw] <antseaten=0> <speed=4> : Hozzáad egy hangyászszünt az adott helyre a megfelelő paraméterekkel. Alapértelmezésben észak felé megy, nulla hangyát evett és 4 a sebessége.

add <x> <y> foodscent <size=10> <remaining=15> <hasFood=0> <x=0> <y=0>: Hozzáad egy ételszagot a megfelelő helyre a kapott paraméterekkel. Ha van étel hozzákötve, annak is kell a helyzete.

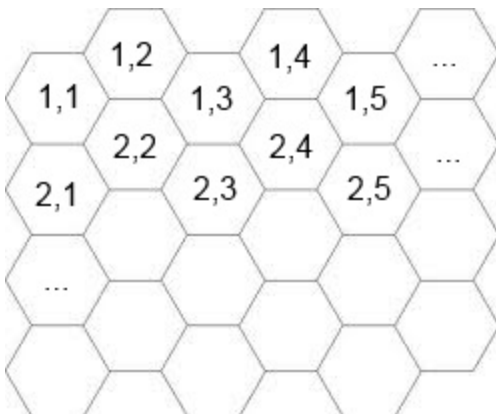
add <x> <y> poison <size=4> <remaining=15> : Hozzáad egy méregfoltot a megfelelő paraméterekkel. Alapértelmezésben 4 sugarú, és még 15 van hátra belőle.

do <x> <y> <t> [antscentvanisher|poison] : Az adott számú tick()-ben fúj egyet a spray-vel.

run <t> : Adott számú tick()-et futtat le a Director.

until <fv> <max=100>: Futtatás amíg a megadott függvény meg nem hívódik, de legfeljebb max tickig.

until <fv> <max=100> <ret=#null>: Futtatás amíg a megadott függvény ret értékkel nem tér vissza, legfeljebb max tickig (#null esetén nem vizsgáljuk a visszatérési értéket). A függvény neve osztálynév::metódus formában adandó meg. A visszatérési érték az utolsó paraméterként adható meg.



7.1.3 Kimeneti nyelv

A szabályos kimeneten a függvények hívásának sorrendje fog megjelenni, hasonló formában, mint a skeleton logolásakor. Ezen kívül (ugyanazon a nyelven, mint a bemenet) meg fog jelenni a pálya aktuális képe is.

7.2 Összes részletes use-case

Use-case neve	Pályaleírás betöltése
Rövid leírás	A játékos meghatározza a bemeneti fájl elérését.
Aktorok	Játékos
Forgatókönyv	Betöltődik a játékos által megadott fájlból a pályaleírás vagy parancssoron keresztül megadja az utasításokat.

Use-case neve	Logolás
Rövid leírás	A játékos meghatározza, hogy hova logoljon a program.
Aktorok	Játékos
Forgatókönyv	A játékos által megadott fájlba kiírodik a logolás. Ha nem ad meg semmit, parancssorba kapja meg a logot.

Use-case neve	Randomizálás
Rövid leírás	A játékos bekapcsolhatja a randomizálást.
Aktorok	Játékos
Forgatókönyv	Véletlenszerű működés indítása.

Use-case neve	Pályaleírás kimentése
Rövid leírás	A játékos meghatározza a kimeneti fájl elérését.
Aktorok	Játékos
Forgatókönyv	Pályaleírás kiírása a játékos által megadott fájlba. Ha nem ad meg fájlt, a parancssorba íródik ki a pálya állapota.

Use-case neve	Spray használat
Rövid leírás	A játékos a sprayket használja.
Aktorok	Játékos
Forgatókönyv	A játékos kiválasztja először a használandó spray-t, majd beírja a célporcella azonosítóját.

Use-case neve	Pálya szélessége
Rövid leírás	A játékos a pálya szélességét állítja be.
Aktorok	Játékos
Forgatókönyv	A játékos megadja a kívánt szélességet.

Use-case neve	Pálya magassága
Rövid leírás	A játékos a pálya magasságát állítja be.
Aktorok	Játékos
Forgatókönyv	A játékos megadja a kívánt magasságot.

Use-case neve	Pálya generálás
Rövid leírás	A játékos legeneráltatja a pályát.
Aktorok	Játékos
Forgatókönyv	A korábban megadott magasság és szélesség alapján generálódik egy játékpálya.

Use-case neve	Pályaelem elhelyezése
Rövid leírás	A játékos elhelyez egy pályaelemet a pályán.
Aktorok	Játékos
Forgatókönyv	A játékos megadja a pályán a kívánt helyet, valamint, hogy mit akar oda helyezni.

Use-case neve	Futtatás
Rövid leírás	Adott számú tick() futtatása.
Aktorok	Játékos
Forgatókönyv	A játékos megadja, hogy hány db tick() fusson le a játékban, majd az lefut.

7.3 Tesztelési terv

Teszteset neve	Étel felvétele és leszállítása
Rövid leírás	Egy hangya elindul egy tőle északkeletre levő étel felé, annak a szagát követve, azt felveszi, majd az attól északra levő hangyabolyba lerakja.
Teszt célja	Az ételek felvételének és leszállításának tesztelése

Teszteset neve	Hangyalesőbe lépés
Rövid leírás	Egy hangya elindul egy tőle északra levő étel felé, azt felveszi, majd a attól északra levő hangyalesőbe belelép. Az étel és a hangya megsemmisül.
Teszt célja	A hangyaleső szerepének tesztelése

Teszteset neve	Hangyászsun hangyaevése
Rövid leírás	Két hangya elindul észak felé és a tőle délre induló hangyászsun utolér. Útközben az egyik hangya felvesz egy ételt.
Teszt célja	Hangya megevésének tesztelése étellel és anélkül

Teszteset neve	Kavicsok mozgása
Rövid leírás	Egy hangyászsun az északra levő követ letolja a pályáról.
Teszt célja	A kavicsok mozgásának tesztelése

Teszteset neve	Méreg terjedése
Rövid leírás	Egy hangya kavicsokkal van körülhatárolva (így nem tud továbbmenni), kavicsokon kívül egy méregfolt terjed, majd megöli a hangyát
Teszt célja	Méreg terjedésének tesztelése.

Teszteset neve	Hangyaszag semlegesítése
Rövid leírás	Egy hangya 5 körön keresztül halad egyenesen, majd a nyomvonalána középpontjára hangyszag semlegesítő spray-t fújunk, amely 3 kör alatt eltünteti az első 5 körben hagyott szagot.
Teszt célja	A hangyaszag semlegesítő spray és szag tesztelése

Teszteset neve	Hangya hangyászsünre lép
Rövid leírás	Egy hangya olyan mezőre lép, amin tartózkodik már egy hangyászsün.
Teszt célja	A hangyát akkor is meg tudja enni, ha nem a hangyászsün lép rá.

Teszteset neve	Kavicsok mozgatása hangyára
Rövid leírás	Egy hangyászsün eltol egy kavicsot egy olyan mezőre, amin egy hangy tartózkodik.
Teszt célja	Hangya kavics által való összenyomásának tesztelése.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

7.4.1 logView

Ha a tesztelő csapat úgy kívánja, akkor a függvényhívások logolásával keletkező fájlból egy segédprogram treeview elrendezésben megjeleníti a program menetét. Az egymásba ágyazott függvényhívások összecukhatóak lesznek, így a kívánt részletességgel tekinthető meg a futtatott program.

7.4.2 Cél-ellenőrző

Az előre definiált tesztekhez cél-ellenőrzés lesz megvalósítva, ami a kimeneti pályaképen egy-egy tulajdonságot tesztel. (Például ha a cél az, hogy az étel visszajusson a hangyabolyba, megvizsgálja, hogy tényleg eltűnt-e az étel a pályáról)

7.5 Napló

Időpont	Időtartam	Résztevők	Eredmény
2013.03.26. 11:00:00	1 óra	Papp	A feladatváltozások implementálása a szkeletonban.
2013.03.27. 12:00:00	1 óra	Kozaróczy	logView segédprogram létrehozása
2013.03.27. 16:00:00	1 óra	Kozaróczy	logView véglegesítése
2013.04.01. 12:30:00	0,5 óra	Kozaróczy	Prototípus ki és bemeneti nyelvének megtervezése
2013.04.01. 16:00:00	1 óra	Papp	Kimeneti nyelv pontosítása
2013.04.02. 10:00:00	1 óra	Kozaróczy	Tesztesetek írása
2013.04.02. 10:00:00	1 óra	Tóth	Tesztesetek írása
2013.04.02. 10:00:00	1 óra	Varga	Use-case-ek írása
2013.04.02. 11:30:00	0,5 óra	Kozaróczy	Dokumentum végső formázása és tördelése

8. Részletes tervek

8.1 Osztályok és metódusok tervei

8.1.1 Ant

- **Felelősség**

Hangyákat reprezentáló osztály

- **Ősosztályok**

Located -> Wanderer -> Ant

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **# myFood:Food** : A hangya által vitt étel. Amikor felveszi, akkor már csak ő hivatkozik rá, a mezők nem.

- **# state:AntState** : A hangya állapotára vonatkozó enum, lehet AS_Searching (keresés), AS_Returning (visszatérés) vagy AS_Dead (halott)

- **# defaultSpeed:int** : A hangya sebessége, kezdeti értéke 2.

- **Metódusok**

- **+ Ant(HexCell location)** : Hangya konstruktora

- **+ Ant(HexCell location, Direction dir, int speed, boolean hasFood)** : Minden paraméterre kiterjedő konstruktor a pályabetöltéshez

- **+ void accept(IInterActor interactor)** : Meghívja a IInterActor interactWith(Ant) metódusát, paraméterként adva saját magát.

- **+ void die()** : A hangya ennek hatására AS_Dead (halott) állapotba kerül, és ha volt nála étel, akkor eldobja

- **+ Food getFood()** : A hangyánál lévő ételt adja vissza

- **+ void interactWith(AntHill a)** : Hangyabollyal való interakció. Ha van nála étel, jelez.

- **+ void interactWith(Antlion a)** : Hangyalesővel való interakció. Meghal. Ha volt nála étel, máshova kerül.

- **+ void interactWith(Echidna e)** : Hangyászszünnel való interakció. Visszahívja az Echidna.interactWith(Ant) függvényt

- **+ void interactWith(Food f)** : Étellel való interakció. Ha tudja, felveszi.

- **+ void interactWith(Pebble p)** : Kavicssal való interakció. Visszapattan a kavicsról. Ha vitt ételt, újragondolja az útvonalát.

- **+ void interactWith(Poison p)** : Méreggel való interakció. Meghal tőle.

- **+ void move(HexCell next)** : A hangya mozgása a paraméterként kapott cellába.

A Wanderer move(HexCell) függvényét hívja tovább, majd szagnyomot hagy.

- **+ void setFood(Food newVal)** : A paraméterül kapott változót elmenti a myFood változóba.
- **+ void tick()** : Az idő múlását szimulálja. A következő irányt a szomszédos cellák befolyása alapján állapítja meg kereső állapotban. Visszatérő állapotban a legközelebbi hangyaboly felé mozog.

8.1.2 AntHill

- **Felelősség**

A hangyabolyt modellezi. Az idő múlásával folyamatosan újabb hangyákat hoz létre.

- **Ősosztályok**

Located -> Obstacle -> NonBlocking -> AntHill

- **Interfészek**

ITimed

- **Attribútumok**

- **# unusedTicks int** : Ezzel számolja, hogy jöhet-e új hangya
- **# tickPerAnt : int** : Ezzel számolja, hogy mennyi időnként kell jönnie hangyának

- **Metódusok**

- **+ AntHill(HexCell location)** : Konstruktorként, beeregisztrálja magát a Director-nál.
- **+ void accept(IInterActor interactor)** : Meghívja az IInterActor interactWith(AntHill) metódusát, paraméterként adva saját magát.
- **+ void dispose()** : Deregisztrálja magát a Director-ból.
- **+ void foodCollected()** : A hangya ezen keresztül jelzi, hogy behozott egy ételt.
- **+ void tick()** : Az idő múlását szimulálja.

8.1.3 Antlion

- **Felelősség**

A hangyalesőket modellezi.

- **Ősosztályok**

Located -> Obstacle -> NonBlocking -> Antlion

- **Interfészek**

Nincsenek.

- **Attribútumok**

Nincsenek.

- **Metódusok**

- **+ Antlion(HexCell location)** : Konstruktorként
- **+ void accept(IInterActor interactor)**: Meghívja az IInterActor interactWith(Antlion) metódusát, paraméterként adva saját magát.
- **+ void foodDestroyed()** : Jelzi, hogy egy olyan hangyát evett meg, amelyiknél volt

étel, így az megsemmisült.

8.1.4 AntScent

- **Felelősség**
A hangyák által hagyott szagnyomokat valósítja meg.
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> Scent -> AntScent
- **Interfészek**
Nincsenek.
- **Attribútumok**
 - **# influence:int** : Hangyák befolyásolásának mértéke.
- **Metódusok**
 - **+ AntScent(HexCell location)** : Konstruktor
 - **+ AntScent(HexCell location, int remaining)** : Részletes konstruktor
 - **+ void accept(IInterActor interactor)** : Meghívja az IInterActor interactWith(AntScent) metódusát, paraméterként adva saját magát.
 - **+ int influence()** : A hangyák befolyásolása. Függs a befolyásoló képességtől és a szag intenzitásától
 - **+ void tick()** : Az idő múlását szimulálja. Folyamatosan csökken az intenzitása. Nem tud terjedni.

8.1.5 AntScentVanisherScent

- **Felelősség**
Hangyaszag-semlegesítő spray által hagyott szagnyom.
- **Ősosztályok**
Located -> Obstacle -> NonBlocking -> Scent -> AntScentVanisherScent
- **Interfészek**
IInterActor
- **Attribútumok**
Nincsenek.
- **Metódusok**
 - **+ AntScentVanisherScent(HexCell location, int size)** : Konstruktor
 - **+ AntScentVanisherScent(HexCell location, int size, Direction dir)** : Konstruktor
 - **+ void accept(IInterActor interactor)** : Meghívja az IInterActor interactWith(AntScent) metódusát, paraméterként adva saját magát.
 - **+ void interactWith (Ant a)** : Hangyával való interakció
 - **+ void interactWith (AntHill a)** : Hangyabollyal való interakció
 - **+ void interactWith (Antilon a)** : Hangyalesővel való interakció

- **+ void interactWith (AntScent a)** : A hangya szagnyomásnak semlegesítése.
- **+ void interactWith (AntScentVanisherScent a)** : Hangyaszagsemlegesítő szaggal való interakció
- **+ void interactWith (Echidna e)** : Hangyászszünnel való interakció
- **+ void interactWith (Food f)** : Étellel való interakció
- **+ void interactWith (FoodScent s)** : Ételszaggal való interakció
- **+ void interactWith (Pebble p)** : Kavicssal való interakció
- **+ void interactWith (Poison p)** : Méreggel való interakció
- **+ void interactWith (Puddle p)** : Tócsával való interakció
- **+ void interactWith (Slime s)** : Nyálkával való interakció
- **+ void interactWith (Snail s)** : Csigával való interakció
- **+ void tick()** : Időnként csökkenti az erősségét, ha ez nulla lesz, meghívja a dispose() függvényt.

8.1.6 AntScentVanisherSpray

- **Felelősség**
Hangyaszag-semlegesítő sprayt reprezentáló osztály.
- **Ősosztályok**
Spray -> AntScentVanisherSpray
- **Interfészek**
Nincsenek.
- **Attribútumok**
Nincsenek.
- **Metódusok**
 - **+ AntScentVanisherSpray()** : Konstruktor
 - **+ void doSpraying(HexCell location)** : Az adott pozícióba létrehoz egy méregfoltot.

8.1.7 Barrier

- **Felelősség**
A blokkoló akadályok absztrakt ősztyála.
- **Ősosztályok**
Located -> Obstacle -> Barrier
- **Interfészek**
Nincsenek.
- **Attribútumok**
Nincsenek.
- **Metódusok**
Nincsenek.

8.1.8 Director

- **Felelősség**

A játék inicializálásáért és futtatásáért felelő osztály.

Létrehozza és tárolja a mezőket (HexCell), majd szétosztja az akadályokat (Obstacle), hangyabolyt (AntHill) és ételdarabokat (Food). Ő felel a hangyászsünök (Echidna) létrehozásáért és az aktív Spray kiválasztásáért. A játék végét is ellenőrzi megfelelő függvények meghívásakor.

- **Ősosztályok**

Nincs.

- **Interfészek**

ITimed.

- **Attribútumok**

- **# activeSpray:Spray** : A kiválasztott spray
- **# antCount : int** : A hangyák száma
- **# antHillLocation : HexCell** : A hangyaboly helye
- **# cells:ArrayList<HexCell>** : A pálya elemei
- **# echidnaAttractingAntQuantity : int** : Megadja, hogy hány hangyánként rendel be a pályára egy újabb hangyászsünt
- **# echidnaCount : int** : A hangyászsünök száma
- **# foodRemaining:int** : A maradék ételek száma
- **# size_x:int** : A pálya mérete
- **# size_y:int** : A pálya mérete
- **# sprayingList : ArrayList<doSpraying>** : A betöltéskor meghatározott késleltetett fűjásokhoz
- **# sparys:ArrayList<Spray>** : A lehetséges spray-k
- **# timedObjects:ArrayList<ITimed>** : A időzített objektumok listája
- **- theOne:Director** : Egyetlen statikus példány az osztályhoz.

- **Metódusok**

- **+ Director()** : Konstruktor
- **+ void add(int x, int y, Located l)** : Hozzáad egy elemet a megfelelő cellához.
- **+ Director getInstance()** : Az aktuális példányt adja vissza. Ha nem létezik, akkor létrehoz egy példányt.
- **+ void contractionOfScent()** : Szagok összevonása
- **+ void decAnt()** : Csökkenti a hangyaszámlálót.
- **+ void decEchidna()** : Csökkenti a hangyászsünszámlálót
- **+ void deregisterTimed(ITimed t)** : Egy ITimed elem eltávolítása a listából.
- **+ void dispose()** : Az egyetlen példány megsemmisítése.

- **+ void finalize()** : A példány törlése
- **+ void foodCollected()** : Ezen keresztül kap jelzést, hogy egy étel beért a hangyabolyba vagy a pályán megsemmisült.
- **+ void foodDestroyed()** : Ezen keresztül kap értesítést, hogy egy étel megsemmisült. Ekkor egy véletlenszerűen kiválasztott cellára rak egy új ételt.
- **+ void foodGenerated()** : Ezen keresztül kap jelzést, ha egy új étel jön létre
- **+ void generatePaths()** : Legenerálja az összes cellához a hangyabolyba vezető utat szélességi bejárással. Gyors.
- **+ void getAntHillLocation()** : Visszaadja, hol van a hangyaboly
- **+ void HexCell getCell(int, int)** : Koornidáták alapján visszaad egy HexCellt. 0-tól indexel.
- **+ void getCells()** : Visszaadja az összes cellát
- **+ Spray getSpray()** : Az aktív spray-t adja vissza.
- **+ void incAnt()** : Növeli a hangyaszámlálót. A hangya hívja meg a konstruktorában
- **+ void incEchidna()** : Növeli a hangyászszümszámlálót
- **+ void initialize()** : A játék inicializálása.
- **+ boolean isExitID(int id)** : Generáláskor számozott cellák közül visszaadja, hogy az adott sorszámú szélső cella-e.
- **+ void nextSpray()** : Beállítja a aktív spraynek a következőt.
- **- HexCell randomCell()** : Segédfüggvények inicializálásához.
- **- int randomInRange()** : Visszaad egy randomszámot a két érték között
- **+ void registerTimed(ITimed t)** : Egy ITimed elem hozzáadása a listához.
- **+ void setAntHillLocation(HexCell c)** : Beállítja a hangyaboly helyét
- **+ void setSize(int x, int y)** : Beállítja a pálya méretét
- **+ void setSpray(Spray newVal)** : Aktív spray-ként állítja a paraméterül kapottat.
- **+ void tick()** : Az idő múlását szimulálja.

8.1.9 Echidna

- **Felelősség**

A hangyászszünök modellezésére szolgál. Ha hangyával egy mezőre (HexCell) kerül, akkor megeszi a hangyát.

- **Össztályok**

Located -> Wanderer -> Echidna

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **# antsEaten:int** : Megevett hangyák száma.
- **#maxAntsEaten:int** : Maximum ennyi hangyát tud megenni.

- **# defaultSpeed:int** : Alapsebesség.
- **# state:EchidnaState** : A hangyászsün helyzete.
- **# exitPath : LinkedList<HexCell>** : A kijutáshoz használt útvonal
- **Metódusok**
 - **+ Echidna(HexCell location)** : Konstruktor
 - **+ Echidna(HexCell location, Direction dir, int speed, int antsEaten)** : Részletes konstruktor
 - **+ void accept(InterActor interactor)** : Meghívja az InterActor interactWith(Echidna) metódusát, paraméterként adva saját magát.
 - **+ void eat(Ant ant)** : Megeszei a paraméterül kapott hangyát.
 - **+ void interactWith(Ant a)** : Hangyával való interakció.
 - **+ void interactWith(Pebble p)** : A hangyászsün kavicssal való interakciója; a hangyászsün arrébb tolhatja a kavicsot.
 - **+ void interactWith(Snail s)** : A csigára nem lép rá a hangyászsün.
 - **+ void move()** : Mozgás a kijelölt cellába, majd minden ottlévőtől interakciót vár
 - **+ void tick()** : Az idő múlását szimulálja. Kóricáláskor véletlenszerű irányba mozog, távozáskor viszont megpróbál mindig a pálya széle felé haladni

8.1.10 Food

- **Felelősség**
A ételeket reprezentálja a játékban.
- **Össztályok**
Located -> Obstacle -> NonBlocking -> Food
- **Interfészek**
Nincsenek.
- **Attribútumok**
 - **+ influence : int** : Hangyák befolyásolásának mértéke.
- **Metódusok**
 - **+ Food(HexCell location)** : Konstruktor, maga mellé rak egy ételszagot
 - **+ void accept(InterActor)** : A paraméterként kapott InterActor-on meghívja az interactWith(...) függvényt saját magát adva paraméterként.
 - **+ void drop(HexCell target)** :Ételt az adott mezőre tesz.
 - **+ int influence()** : Megadja, hogy mennyire befolyásolja a hangya útvonalát.

8.1.11 FoodScent

- **Felelősség**
Az ételek szagát reprezentálja.
- **Össztályok**
Located -> Obstacle -> NonBlocking -> Scent -> FoodScent

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **# origins : ArrayList<Origin>** : A kibocsátó ételek
- **# influence : int** : Hangyák befolyásolásának mértéke.

- **Metódusok**

- **+ FoodScent(HexCell location, int size, Food origin, Direction dir)** : Konstruktor
- **+ FoodScent(HexCell l, Food o, int s, int r, Direction d)** : Konstruktor terjedéshez
- **+ void accept(IInterActor interactor)** : Meghívj az IInterActor interactWith(FoodScent) metódusát, paraméterként adva saját magát.
- **+ void addOrigin(ArrayList<Origin> origins)** : Új ételeket listaként is hozzá lehet adni
- **+ void addOrigin(Food origin)** : Új ételt ad hozzá a forrásokhoz. Ezáltal összevonhatóak a szagok
- **+ void dispose()** : Deregisztrálja magát a Director ITimed-okat tartalmazó listájából. Kiüríti az ételek listáját is
- **+ int influence()** : A hangyák befolyásolása. Függ a befolyásoló képességtől és a szag intenzitásától
- **+ void tick()** : Amíg az originFood a helyén van, nem csökken a szag intenzitása, és terjeng az illat.

8.1.12 HexCell

- **Felelősség**

A mezőket modellezi. HexCellben tárolódik minden, aminek pozíciója van (Located) és az adott cellán tartózkodik.

- **Ősosztályok**

Nincsenek.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **# adjacent:HashMap<Direction, HexCell>** : Szomszédos mezők
- **# contents ArrayList<Located>** : A rajta lévő objektumok
- **# directionToAntHill : Direction** : A hangyabolyhoz vezető útvonal iránya
- **# exit:boolean** : Kijárat-e
- **# pathToAntHill : LinkedList<HexCell>** : A legközelebbi hangyaboly felé vezető útvonal

- **# timeout : long** : AStar függvény timeout-ja.
- **# x : int** : A cella pozíciója
- **# y : int** : A cella pozíciója
- **Metódusok**
 - **+ HexCell()** : Konstruktor
 - **+ HexCell(int x, int y)** : Konstruktor
 - **+ void addCell(Direction direction, HexCell hexCell)** : Szomszédos mező hozzáadása.
 - **+ void addObject(Located newltem)** : Új pozícióval rendelkező objektum hozzáadása.
 - **+ LinkedList<HexCell> aStar(HexCell target)** : Súlyozott A* útkeresés saját cellából. Timeout: Ha túl sok ideig fut, akkor nullal tér vissza.
 - **+ HashMap<Direction, HexCell> getAdjacent()** : Visszaadja az összes szomszédját.
 - **+ ArrayList<Located> getAll()** : Visszaadja az össze selemet a contents listából.
 - **+ HexCell getCell(Direction dir)** : Az adott irányban lévő cellával tér vissza.
 - **+ int getInfluence()** : Visszaadja, hogy adott mező mennyire befolyásolja a hangya irányát.
 - **+ Direction getToAntHill()** : Visszaadja, hogy milyen irányban van a hangyaboly
 - **+ int getX()** : X koordináta
 - **+ int getY()** : Y koordináta
 - **+ boolean isAdjacent(HexCell other)** : Ellenőrzés szempontjából lekérhető, hogy tényleg szomszédos-e a két cella.
 - **+ boolean isExit()** : Igazzal tér vissza, ha az adott calla kijárat
 - **+ void removeObject(Located removed)** : Az adott elemet eltávolítja a contents listáról.
 - **+ void setExit(boolean newVal)** : beállítja, hogy az adott cella kijárat-e.
 - **+ void setPath(LinkedList<HexCell> path, Direction direction)** : Hangyabolyba vezető útvonal eltárolása

8.1.13 IAcceptor

- **Felelősség**

Közös interfész azon osztályoknak, amik interact-ot tudnak kiváltani.

- **Ősosztályok**

Nincsenek.

- **Metódusok**

- **+void accept()** : A paraméterként kapott IInterActor-on keresztül meghívja az interactWith() függvényt saját magát adva paraméterként.

8.1.14 IInterActor

- **Felelősség**
Közös interfész azoknak, akik interakciót képesek végezni.
- **Ősosztályok**
Nincsenek.
- **Metódusok**
 - **+ void interactWith(Ant a)** : Hangyával való interakció.
 - **+ void interactWith(AntHill a)** : Hangyabollyal való interakció.
 - **+ void interactWith(Antilon a)** : Hangyalesővel való interakció.
 - **+ void interactWith(AntScent a)** : Hangya-szagnyommal való interakció.
 - **+ void interactWith(AntScentVanisherScent a)** : Hangyaszag-semlegesítő spray-el való interakció.
 - **+ void interactWith(Echidna e)** : Hangyászszünnel való interakció.
 - **+ void interactWith(Food f)** : Étellel való interakció.
 - **+ void interactWith(FoodScent s)** : Ételszaggal való interakció.
 - **+ void interactWith(Pebble p)** : Akadállyal való interakció.
 - **+ void interactWith(Poison p)** : Méreggel való interakció.
 - **+ void interactWith(Puddle p)** : Tócsával való interakció.
 - **+ void interactWith(Slime s)** : Nyálkával való interakció
 - **+ void interactWith(Snail s)** : Csigával való interakció

8.1.15 ITime

- **Felelősség**
Közös interfész azoknak az osztályoknak, amik reagálnak az idő múlására.
- **Ősosztályok**
Nincsenek.
- **Metódusok**
 - **+ void dispose()** : Deregisztrálja magát a Director ITime-okat tartalmazó listájából.
 - **+ void tick()** : Az idő múlását szimulálja.

8.1.16 LoaderSaver

- **Felelősség**
Betöltéshez és mentéshez használt osztály
- **Ősosztályok**
Nincsenek.
- **Interfészek**
Nincsenek.
- **Attribútumok**

Nincsenek

- **Metódusok**

- **+ void exportMap(Printstream out)** : Kíírja a pályát a prototípus koncepciója 7.1.2-es pontjának megfelelően. A függvény meghívásához a Directort inicializálni kell.
- **+ RunCondition importMap(InputStream in)** : A pályakép beolvasása. Egy hatalmas állapotgéppel történik.

8.1.17 Located

- **Felelősség**

A pozícióval rendelkező osztályok közös absztrakt őse.

- **Ősosztályok**

Nincsenek.

- **Interfészek**

- IAcceptor

- **Attribútumok**

- **# location:HexCell** : Az objektum pozíciója

- **Metódusok**

- **+ Located(HexCell c)** : Az alapértelmezett konstruktort felüldefiniálja, így csak akkor hozható létre, ha megadjuk a helyét. Hozzáadja magát az adott cellához, ha az nem null
- **+ void dispose()** : Eltávolítja magát a cellából.
- **+ void finalize()** : Finalize, amit a GC használ.
- **+ int influence()** : Megadja, hogy mennyire befolyásolja a hangya útvonalát. Alapértelmezett értéke 0.

8.1.18 Logger

- **Felelősség**

A választott kimenetre naplóz (alapértelmezésben a skeleton.log fájlba), mindene statikus.

- **Ősosztályok**

Nincsenek.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **# out:PrintStream** : A kimenet
- **- calls:HashMap<String, HashSet<String>>** : Függvényhívások és visszatérési értékek logja, az until tesztbemenet megvalósításához.
- **- indentLevel:int** : A behúzás mértéke
- **- ident:String** : A behúzás egy egysége

- **- isLogging: boolean** : Ki-be kapcsolhatóság
- **Metódusok**
 - **+ void begin()** : Elindítja a logolást.
 - **+ boolean called(String, String)** : Igazsal tér vissza, ha meghívódott a keresett függvény a várt visszatérési értékkel
 - **+ void dispose()** : Logfájl erőforrás felszabadítása. Szabályosan bezárja a logfájl.
 - **+ void end()** : Leállítja a logolást.
 - **+ void entering(Object that, Object[] params)** : Függvényhívást logoló metódus.
 - **+ void exiting(Object that, Object returnValue)** : Függvényvisszatérést logoló metódus. Kiírja, hogy melyik osztály melyik metódusából hívták, illetve megadható, hogy a hívó metódus mivel fog visszatérni.
 - **+ void exiting(Object that)** : Egyparaméteres exiting, visszatérési értéket nem logol.
 - **+ void init()** : Default init().
 - **+ void init(PrintStream destination)** : Inicializálja a Loggert. A Logger metódusai közül ezt kell először elindítani.
 - **+ void meanwhile(String str)** : Logolja a paramterként kapott szöveget a megfelelő ehízással.
 - **+ reset()** : Alaphelyzetbe állítja a Logger függvényhívás-cachét.

8.1.19 Main

- **Felelősség**

A program belépési pontja. Kezeli a parancssori argumentumokat és a logolást. Egy ciklusban folyamatosan meghívja a Director tick() metódusát.
- **Ősosztályok**
- **Interfészek**
- **Attribútumok**
 - **+ RND: boolean**: Flag, alapértelmezésben false. Ha igaz, akkor a játék működése nemdeterminisztikus. Egyébként az.
- **Metódusok**
 - **+ void main(String[] args)**: A program belépési pontja.

8.1.20 NonBlocking

- **Felelősség**

A nem blokkoló akadályok közös ősosztálya.
- **Ősosztályok**

Located -> Obstacle -> NonBlocking
- **Interfészek**

Nincsenek.

- **Attribútumok**

Nincsenek.

- **Metódusok**

Nincsenek.

8.1.21 Obstacle

- **Felelősség**

Az akadályok közös absztrakt ősztyála.

- **Ősztyályok**

Located -> Obstacle

- **Interfészek**

Nincsenek.

- **Attribútumok**

Nincsenek.

- **Metódusok**

Nincsenek.

8.1.22 Pebble

- **Felelősség**

Kavicsokat reprezentáló osztály.

- **Ősztyályok**

Located -> Obstacle -> Barrier -> Pebble

- **Interfészek**

lnterActor

- **Attribútumok**

- **direction:Direction** : A legutolsó lökésének iránya

- **Metódusok**

- **+ Pebble(HexCell location)** : Konstruktör
- **+ void accept(lnterActor interactor)** : Meghívja az lnterActor interactWith(Pebble) metódusát, paraméterként önmagát adva.
- **+ void interactWith(Ant a)** : Hangyával való interakció. Megöli a hangyát.
- **+ void interactWith(AntHill a)** : Hangyabollyal való interakció. Visszalöki a követ.
- **+ void interactWith(Antilon a)** : Hangyalesővel való interakció. Elpusztul a hangyaleső.
- **+ void interactWith(AntScent a)** : Hangyszaggal való interakció.
- **+ void interactWith(AntScentVanisherScent a)** : Hangyszag-semlegesítővel való interakció.

- **+ void interactWith(Echidna e)** : Hangyászsünnel való interakció. Ahova lép, onnan visszalöki a hangyászsünt.
- **+ void interactWith(Food f)** : Étellel való interakció. Elpusztul az étel.
- **+ void interactWith(FoodScent s)** : Ételszaggal való interakció.
- **+ void interactWith(Pebble p)** : Kavicssal való interakció. Ha már van ott kavics, ahova lép, onnan visszalép.
- **+ void interactWith(Poison p)** : Méreggel való interakció.
- **+ void interactWith(Puddle p)** : Tócsával való interakció. Megöli a hangyát.
- **+ void interactWith(Slime s)** : Nyálkával való interakció
- **+ void interactWith(Snail s)** : Csigával való interakció
- **+ void push(Direction dir)** : Átmozog a megfelelő irányba.

8.1.23 Poison

- **Felelősség**

A méregfoltokat modellező osztály.

- **Ősosztályok**

Located -> Obstacle -> NonBlocking -> Scent -> Poison

- **Interfészek**

Nincsenek.

- **Attribútumok**

Nincsenek

- **Metódusok**

- **+ Poison(HexCell location, int size)** : Konstruktor
- **+ Poison(HexCell location, int size, int remaining, Direction dir)** : Részletes konstruktor
- **+ void accept(IInterActor interactor)**: Meghívja az IInterActor interactWith(Poison) metódusát, paraméterként önmagát adva.
- **+ void tick()** : Csökkenti önmaga erősségét, ha eléri a nullát, akkor meghívja a dispose()-t és megsemmisíti önmagát.

8.1.24 PoisonSpray

- **Felelősség**

A méregspray-t modellezi.

- **Ősosztályok**

Spray -> PoisonSpray

- **Interfészek**

Nincsenek.

- **Attribútumok**

Nincsenek.

- **Metódusok**

- **+ void doSpraying(HexCell location)** :Az adott helyre méregfoltot helyez.

8.1.25 Puddle

- **Felelősség**

A tócsákat valósítja meg. Ezen a hangyák át tudnak kelni, de lassabban és nem hagynak nyomot.

- **Ősosztályok**

Located -> Obstacle -> NonBlocking -> Puddle

- **Interfészek**

Nincsenek.

- **Attribútumok**

Nincsenek.

- **Metódusok**

- **+ void accept(IInterActor interactor)** : Meghívja az IInterActor interactWith(Puddle) metódusát, paraméterként önmagát adva.

8.1.26 Scent

- **Felelősség**

Az összes szagnyom közös absztrakt őszosztálya.

- **Ősosztályok**

Located -> Obstacle -> NonBlocking -> Scent

- **Interfészek**

ITimed

- **Attribútumok**

- **# remaining:int** : Az elillanásig hátralévő idő.
- **# radius:int** : A terjedés nagysága.
- **# dir : ArrayList<Direction>** : A terjedés iránya
- **# first : boolean** : A legelső szagnyom

- **Metódusok**

- **+ Scent(HexCell location, int size, int remaining)** : Konstruktor
- **+ Scent(HexCell location, int size, int remaining, Direction dir)** : Konstruktor
- **+ void dispose()**: Eltávolítja önmagát a játékból.
- **+ void tick()** : Az időt szimulálja.

8.1.27 SingletonException

- **Felelősség**

Singleton osztályok konstruktorai által dobott kivétel. Megakadályozza, hogy egynél több példány legyen egy singleton osztályból.

- **Ősosztályok**

Exception

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **- serialVersionUID:long**

- **Metódusok**

Nincsenek.

8.1.28 Slime

- **Felelősség**

Csigák által hagyott nyálkákat reprezentáló osztály

- **Ősosztályok**

Located -> Obstacle -> NonBlocking -> Scent -> Slime

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

- **Metódusok**

- **+ Slime(HexCell, int)** : Konstruktor
- **+ void accept(IInterActor)** : A paraméterként kapott IInterActor-on meghívja az interactWith(...) függvényt saját magát adva paraméterként.
- **+ void tick()** : Az idő múlását szimulálja. Minden meghíváskor ha tud még, akkor tovább terjed a megfelelő irányokba, egyébként pedig folyamatosan csökken az intenzitása. Nem tud terjedni.

8.1.29 Snail

- **Felelősség**

Csigákat reprezentáló osztály

- **Ősosztályok**

Located -> Wanderer -> Snail

- **Interfészek**

Nincs

- **Attribútumok**

- **+ defaultSpeed : int** : Az alapértelmezett sebesség, értéke 3
- **+ slimeLife : int** : A hagyott nyálka élettartama
- **+ minimumOn : int** : Minimim 30 tick-et tölt a pályán
- **+ on : int** : Ennyi ideje van a pályán
- **+ putsSlime : boolean** : Hamis, ha nem kell nyálkát hagynia

- **Metódusok**

- **+ void move(HexCell)** : Mozgás a kijelölt cellába, majd minden ottlévőtől interakciót vár. Slime-ot hagy maga után
- **+ void tick()** : Az idő múlását szimulálja. Véletlenszerű irányba mozog

8.1.30 Spray

- **Felelősség**

A különböző spray-eket reprezentáló absztrakt őosztály.

- **Őosztályok**

Nincsenek.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **# remaining:int** : A maradék fújások száma
- **+ spraySize:int** : A spray mérete
- **+ defaultSprayCount:int** : Az alapértelmezett maximum fújásszám

- **Metódusok**

- **+ boolean canSpray()** : Igazzal tér vissza, ha még használható további fújásra a spray.
- **+ void doSpraying(HexCell location)** : A paraméterként kapott mezőre ráfúj (spray-enként eltérő lehet a hatása)

8.1.31 Wanderer

- **Felelősség**

A mozgó lények közös absztrakt őosztálya.

- **Őosztályok**

Located -> Wanderer

- **Interfészek**

- ITimed
- IInterActor

- **Attribútumok**

- **direction:Direction** : A kóricáló iránya
- **speed:int** : A kóricáló sebessége
- **Metódusok**
 - **+ void dispose()** : A kóricáló eltüntetése
 - **+ int distanceTo(HexCell target)** : Az adott mezőtől mért távolságot adja vissza.
 - **+ int getSpeed()** : A sebességet adja vissza.
 - **+ void interactWith(Ant a)** : Hangyával való interakció.
 - **+ void interactWith(AntHill a)** : Hangyabollyal való interakció.
 - **+ void interactWith(Antilon a)** : Hangyalesővel való interakció.
 - **+ void interactWith(AntScent a)** : Hangya-szagnyommal való interakció.
 - **+ void interactWith(AntScentVanisherScent a)** : Hangyaszag-semlegesítő spray-el való interakció.
 - **+ void interactWith(Echidna e)** : Hangyászszünnel való interakció.
 - **+ void interactWith(Food f)** : Étellel való interakció.
 - **+ void interactWith(FoodScent s)** : Ételszaggal való interakció.
 - **+ void interactWith(Pebble p)** : Akadállyal való interakció.
 - **+ void interactWith(Poison p)** : Méreggel való interakció.
 - **+ void interactWith(Puddle p)** : Tócsával való interakció.
 - **+ void interactWith(Scent s)** : Szagokkal való interakció.
 - **+ void move(HexCell next)** : Mozgás a kijelölt cellába.
 - **+ void setSpeed(int newVal)** : A sebességet állítja be.
 - **+ void tick()** : Az időt szimulálja.

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

A teszteset kimeneteinél a hangya-, és ételszagok helye nem definiáltak, hiszen a tesztek kimeneteit az elvárt kimenetekkel a ResultChecker programmal hasonlítjuk össze ami nem veszi figyelembe a hangyaszagokat és ételszagokat a 8.3.2. pontban leírtak miatt.

8.2.1 Étel felvétele és leszállítása

- **Leírás**

Egy hangya elindul egy tőle északkeletre levő étel felé, annak a szagát követve, azt felveszi, majd az attól északra levő hangyabolyba lerakja.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
width 2
height 4
add 2 4 ant n
add 1 2 food
add 2 1 anthill
run 2
```

- **Elvárt kimenet**

```
width 2
height 4
add 2 1 ant [n|e] 0 2
add 2 1 anthill
add 1 2 foodscent 10 15
[add 1 1 foodscent 9 14]
```

8.2.2 Hangyalesőbe lépés

- **Leírás**

Egy hangya elindul egy tőle északra levő étel felé, azt felveszi, majd az attól északra levő hangyalesőbe belelép. Az étel és a hangya megsemmisül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
width 1
height 3
add 1 3 ant
add 1 2 food
add 1 1 antlion
run 1
```

- **Elvárt kimenet**

```
width 1
height 3
add 1 1 antlion
add 1 2 foodscent
```


8.2.3 Hangyászsün hangyaevése

- **Leírás**

Két hangya elindul észak felé és a tőle délre induló hangyászsün utoléri. Útközben az egyik hangya felvesz egy ételt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
width 1
height 5
add 1 5 echidna
add 1 3 ant n 0 0
add 1 4 ant n 1 0
run 1
```

- **Elvárt kimenet**

```
width 1
height 5
add 1 1 echidna n 2 4
add 1 4 food
```

8.2.4 Hangyászsün hangyaevése II

- **Leírás**

Egy hangya olyan mezőre lép, amin tartózkodik már egy hangyászsün, ami már csak egy hangyát tud megenni. Ezután odalép egy másik hangya is

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
width 1
height 2
add 1 2 ant n 0 1
add 1 2 ant n 0 1
add 1 1 echidna n 7 0
run 1
```

- **Elvárt kimenet**

```
width 1
height 2
add 1 1 ant n 0 2
add 1 1 echidna n 8 4
```

8.2.5 Kavicsok mozgása

- **Leírás**

Egy hangyászsün az északra levő követ letolja a pályáról.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
width 1
height 2
add 1 1 pebble
add 1 2 echidna n 0 1
run 1
```

- **Elvárt kimenet**

```
width 1
height 2
add 1 1 echidna n 0 4
```

8.2.5 Kavicsok mozgása II

- **Leírás**

Egy hangyászszün eltol egy kavicsot egy olyan mezőre, amin egy hangya tartózkodik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
width 1
height 3
add 1 3 echidna n 0 1
add 1 2 pebble
add 1 1 ant n 0 0
run 1
```

- **Elvárt kimenet**

```
width 1
height 3
add 1 1 pebble
add 1 2 echidna n 0 4
```

8.2.6 Méreg terjedése

- **Leírás**

Egy hangya kavicsokkal van körülhatárolva (így nem tud továbbmenni), a kavicsokon kívül egy méregfolt terjed, majd megöli a hangyát

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
width 3
height 3
add 1 2 pebble
add 1 3 pebble
add 2 1 pebble
add 2 2 ant
add 2 3 pebble
add 3 1 poison 4 15
```

```
add 3 2 pebble
add 3 3 pebble
run 2
```

- **Elvárt kimenet**

```
width 3
height 3
add 1 2 pebble
add 1 3 pebble
add 2 1 pebble
add 2 3 pebble
add 1 1 poison 0
add 2 1 poison 0
add 3 1 poison 0
add 1 2 poison 0
add 2 2 poison 0
add 3 2 poison 0
add 3 3 poison 0
add 3 2 pebble
add 3 3 pebble
```

8.2.7 Hangyszag semlegesítése

- **Leírás**

Egy hangya 5 körön keresztül halad egyenesen, majd a nyomvonalának középpontjára hangyszag semlegesítő spray-t fújunk, amely 3 kör alatt eltünteti az első 5 körben hagyott szagot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
width 1
height 13
add 1 13 ant
do 1 4 3 antscentvanisher
run 6
```

- **Elvárt kimenet**

```
width 1
height 13
add ant n 0 2
```

8.3 A tesztelést támogató programok tervei

8.3.1 logView

A program célja a függvényhívások megfelelő részletességgel való végigkövetése.

A logolásból keletkezett fájl tartalmát jeleníti meg treeview elrendezésben. A program Java Swing-et használ, azon belül a JTree elrendezést. Logoláskor az egymásba ágyazott függvényeket jelezzük, így a program könnyen tudja törölni a kapott szöveges fájlt.

8.3.2 ResultChecker

A program ellenőrzi, hogy a hangyás játék kimenete megfelel-e az elvárt kimenetnek.

Két fájlt kap bementeként:

- A hangyás játék elvárt kimenetét tartalmazó fájlt.
- A hangyás játék ténylegesen kapott kimenetét.

A ResultChecker lényegében a játéktér celláit hasonlítja össze, megnézi, hogy ugyanazokat a Located-eket tartalmazzák-e. Ha a kapott kimenet által leírt minden cella tartalma megegyezik az elvárttal, akkor a program jelzi, hogy a teszt sikeres volt. Egyébként nem volt az.

A ResultChecker az ellenőrzéskor nem veszi figyelembe az étel és hangyaszagokat, mert azok elhelyezkedése a determinisztikus működés során is nagyon nehezen jósolható meg csak a bementet ismeretében. (Pl.: nehezen eldönthető, hogy előbb veszi fel az ételt, így az nem hagy új szagot, vagy előbb hagy új szagot és utána veszi fel egy hangya)

8.4 Napló

Időpont	Időtartam	Résztevők	Eredmény
2013.04.04. 14:00:00	3 óra	Varga	Hiánypótlások nagy részének elkészítése (módosított osztálydiagramm, új szekvenciadiagrammok, teszteset, kimeneti nyelv)
2013.04.05. 13:00:00	6 óra	Molnár	Osztályok leírása
2013.04.06. 13:00:00	2 óra	Kozaróczy	Tesztesetek bemenete és kimenete
2013.04.06. 15:00:00	0,5 óra	Kozaróczy	Tesztelést támogató programok leírása
2013.04.06. 17:00:00	2,5 óra	Papp	Dokumentáció kiegészítése, poison terjedés javítása
2013.04.08.	0,5 óra	Kozaróczy	Dokumentum végső formázása és tördelése

10:00:00			
----------	--	--	--

10. Prototípus

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Ahol nincs külön kiemelve a fájl funkciója, ott az adott nevű osztályt valósítjuk meg benne.

Időpont	Méret	Fájl	Funkció
2013.04.22. 10:09	464	compile.bat	Bat fájl a fordításhoz
2013.04.22. 11:17	0	expectedTest.txt	Az elvárt kimenetet ide kell megadni
2013.04.22. 10:00	144	logView.bat	A logView segédprogram futtatása
2013.04.22. 09:43	44	manifest.mf	A proto jar fájljához
2013.04.22. 09:54	47	manifest_logView.mf	A logView jar fájljához
2013.04.22. 09:56	62	manifest_resultChecker.mf	A resultChecker jar fájljához
2013.04.22. 10:02	193	resultChecker.bat	A resultChecker futtatása
2013.04.22. 11:16	0	resultTest.txt	A kimenet ebbe kerül
2013.04.22. 09:43	222	run.bat	A proto futtatása
2013.04.22. 11:16	0	skeleton.log	A függvényhívások logolása
2013.04.22. 11:16	0	Test.txt	A tesztesetek bemenete

Az **src/main** mappában található a prototípus kódja

Időpont	Méret	Fájl
2013.04.22. 05:10	7'352	Ant.java
2013.04.22. 05:10	1'817	AntHill.java
2013.04.18. 13:08	984	Antlion.java
2013.04.22. 10:13	1'602	AntScent.java
2013.04.18. 16:09	4'109	AntScentVanisherScent.java
2013.04.18. 13:20	591	AntScentVanisherSpray.java
2013.04.18. 13:21	202	Barrier.java
2013.04.22. 10:47	14'548	Director.java
2013.04.22. 05:10	5'042	Echidna.java
2013.04.22. 10:29	1'425	Food.java
2013.04.22. 10:12	4'926	FoodScent.java
2013.04.22. 10:40	9'386	HexCell.java

2013.04.18.	13:07	403	IAccepter.java
2013.04.22.	05:10	1'490	IInterActor.java
2013.04.18.	12:27	369	ITimed.java
2013.04.22.	10:25	6'210	LoaderSaver.java
2013.04.22.	09:43	1'274	Located.java
2013.04.22.	09:43	6'608	Logger.java
2013.04.22.	09:43	2'025	Main.java
2013.04.18.	14:04	215	NonBlocking.java
2013.04.18.	14:05	190	Obstacle.java
2013.04.18.	15:38	3'742	Pebble.java
2013.04.18.	15:56	2'267	Poison.java
2013.04.18.	14:12	518	PoisonSpray.java
2013.04.22.	05:10	528	Puddle.java
2013.04.22.	10:14	1'889	Scent.java
2013.04.22.	05:10	214	SingletonException.java
2013.04.18.	14:17	932	Spray.java
2013.04.18.	14:24	4'323	Wanderer.java

Az **src/logView** mappában található a logView segédprogram kódja

Időpont	Méret	Fájl
2013.03.27. 17:45	215	Main.java
2013.04.22. 09:43	5'822	Window.java

Az **src/resultChecker** mappában található a resultChecker segédprogram kódja

Időpont	Méret	Fájl
2013.04.17. 17:28	2'688	ResultChecker.java
2013.04.18. 17:46	911	ResultDifferenceCellRenderer.java
2013.04.18. 17:46	1'198	ResultListModel.java
2013.04.22. 05:10	1'200	ResultVisualizer.java
2013.04.22. 05:10	2'834	ResultVisualizerDataWindow.java
2013.04.22. 05:10	12'597	ResultVisualizerPanel.java
2013.04.18. 17:46	5'952	ResultWindow.java

10.1.2 Fordítás

A fordításhoz és futtatáshoz csomagoljuk ki a kapott zip fájlt, majd Windows alatt parancssorból a `compile.bat` fájljal fordítsuk le.

10.1.3 Futtatás

A `run.bat` segítségével futtassuk a fő programot, illetve a `logView.bat` és

resultChecker.bat fájlokkal futtasuk a segédprogramokat.

10.2 Tesztek jegyzőkönyvei

10.2.1 Beolvasás

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.17.
Teszt eredménye	Nem várt kivételdobás a méreg terjedésekor
Lehetséges hibaok	Nem inicializált változók
Változtatások	Scent-ek konstruktorából kimaradt az érték átadása

10.2.2 Étel felvétele és leszállítása

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18.
Teszt eredménye	Nem várt kivételdobás.
Lehetséges hibaok	Nem kezeltük le azt, hogyha a hangya minden irányban pálya szélével találkozik
Változtatások	Probléma feloldása (ilyen esetben csak fordul egyet)

10.2.2 Étel felvétele és leszállítása

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18.
Teszt eredménye	A vártnál több hangya van a pályán
Lehetséges hibaok	A hangyaboly minden óraütemre kiállít még egy hangyát.
Változtatások	A tesztek elvárt kimeneteit újragondoljuk a hangyaboly működésével.

10.2.3 Étél felvétele és leszállítása

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18

10.2.4 Hangyászsün hangyaevése

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18.
Teszt eredménye	A hangyászsün máshol tartózkodik
Lehetséges hibaok	Az utolsó lépést nem tudta megtenni.
Változtatások	Rosszul volt beállítva a szomszédosság

10.2.5 Hangyászsün hangyaevése

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18

10.2.6 Hangyászsün hangyaevése II

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18

10.2.7 Kavicsok mozgása

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18

10.2.8 Kavicsok mozgása II

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18.
Teszt eredménye	Végtelen ciklusba került
Lehetséges hibaok	A kavics önmagával lép interakcióba, és oda-vissza lökdösi magát
Változtatások	Magával nem fog interakcióba lépni

10.2.9 Kavicsok mozgása II

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18

10.2.10 Méreg terjedése

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18.
Teszt eredménye	Méreg máshol helyezkedik el
Lehetséges hibaok	Rosszul volt a szomszédosság implementálva.
Változtatások	Szomszédosság beállításának újradeifniálása

10.2.11 Méreg terjedése

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18.
Teszt eredménye	Méreg nem ölte meg a hangyát
Lehetséges hibaok	A méreg nem kér interakciót a hangyától, csak a hangya tick() metódusában lesz eredmény
Változtatások	Egyel több tick() szükségeltetik

10.2.12 Méreg terjedése

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18

10.2.13 Hangyaszag semlegesítése

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.18

10.2.14 Hangyalesőbe lépés

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.22.
Teszt eredménye	Megjelent egy új étel a pályán
Lehetséges hibaok	Hibás konstruktorhívás
Változtatások	Hiba kijavítása

10.2.15 Hangyalesőbe lépés

Tesztelő neve	Kozaróczy
Teszt időpontja	2013.04.22.
Megjegyzés	A korábban definiált elvárt kimenet alapján nem kellene ételnek lennie, ezért módosult, mert mostmár nem elpusztul az étel hangyalesőné, hanem új helyre sorsolódik.

10.3 Értékelés

Tag neve	Munka százalékban	Aláírás
Kozaróczy Zsolt	21%	
Molnár Tamara	18%	
Papp Győző	21%	
Tóth Ákos	21%	
Varga Éva Árnika	19%	

10.4 Napló

Időpont	Időtartam	Résztevők	Eredmény
2013.04.08. 16:30:00	1,5 óra	Molnár	Kódolás: Wanderer és leszármazottai, Scent és leszármazottai, Director setSize függvénye
2013.04.12. 16:30:00	3 óra	Papp	Director.tick() megalkotása és javadoc dokumentáció néhány függvényekhez
2013.04.15. 10:00:00	6 óra	Tóth	resultChecker teszteset-ellenőrző csomag
2013.04.16. 15:00:00	3 óra	Papp	HexCell.BFS(), HexCell.directionTo(), HexCell.distanceTo() metódusai
2013.04.17. 12:00:00	1 óra	Molnár	Beolvasások kezelése
2013.04.17. 13:00:00	1 óra	Kozaróczy	Felmerülő hibák javítása
2013.04.17. 16:30:00	3 óra	Kozaróczy	logView paraméterhetősége és Logger javítása
2013.04.17. 16:00:00	3 óra	Papp	toAnthill, toNearestExit implementálása, kommentelés, hibajavítás részletesen a 3f2cabdb... commit megjegyzésben
2013.04.18. 12:30:00	2 óra	Varga	A kód ellátása javadoc kommentekkel
2013.04.18. 14:30:00	2 óra	Kozaróczy	Tesztesetek vizsgálata és hozzájuk kapcsolódó hibák javítása
2013.04.22. 1:00:00	8 óra	Tóth	Kódolás
2013.04.22. 10:00:00	1 óra	Kozaróczy	Teszttek felülvizsgálata, dokumentum tördelése

11. Grafikus felület specifikációja

11.1 A grafikus interfész

Három fő megjelenítésünk. Lesz egy a játék előtti főmenü, itt be lehet állítani a pálya méretét, nehézségét és a toplistát lekérdezni. A következő képernyőn a játék folyik, lehet spray-kkel fújni, követni a hangyák mozgását.. A harmadik képernyő a másodiktól annyiban tér el, hogy itt már vesztett a játékos, és nem irányíthat semmit, csak lehetősége van új játékot kezdeni, vagy tovább figyelni az eseményeket. Itt már nem kell kijelezni a maradék spray-ket és ételek számát.

A látható mintaképek csak vázlatok, a tényleges megjelenítésben ettől eltérhetnek.

11.1.1 Start Screen



A kezdőképernyőn lehetőség lesz elindítani a játékot, néhány beállítást elvégezni, illetve megtekinteni a toplistát. Megtekintéskor a középső gombok két oldalán jelennek meg a beállítható paraméterek, illetve a toplista.

11.1.2 Field Screen



A játék alatt a képernyőn folyamatosan látszani fog a pálya hálójá, ezzel könnyítve a spray-kkel való fújást. A játéktéren megjelenő elemek fontosságuk szerint sorrendben fognak megjelenni, legalul lesznek a szagok és mérgek (a csoporton belül viszont nem meghatározott), e felé kerülnek a tócsák, kavicsok és hangyalesők és ételek. Legfelül (mindig láthatóan) mozognak majd a hangyák és hangyászsünek. A képernyő jobb felső sarkában fog látszani, hogy mennyi van még hátra a játékból (ételek, mérgek és szagtalanítók száma).

11.1.3 End Screen




A játék elvesztése utáni képernyő háttérében tovább fognak mozogni a pályán, csupán a játékosnak nem lesz lehetősége beleszólni a játék menetébe. Lehetősége lesz viszont új játékot kezdeni, megtekinteni az eddigi toplistát, vagy elhagyni a játékot. A toplistába a kezdőképernyő beállításainál megadott névvel kerül be automatikusan, illetve ha ott nem adott meg semmit, akkor elégséges eredmény esetén a játék itt megkérdezi a nevét.

11.1.4 Felhasznált képek

A miniatűr ábrák a leadásig még változhatnak, vagy bővíülhetnek.

Hangya


Mindig arra forog, amerre éppen halad, illetve több színben is feltűnhet, ezeket véletlenszerűen sorsoljuk. Ha felvesz egy ételt, akkor annak a miniatűr mását fogja vinni a hátán. 

Étel

Az ételekből is többféle jelenik meg a játékban, ezeket az étel létrehozásakor sorsoljuk ki. Amikor

egy hangya felveszi, akkor annak a hátán megjelenik kicsiben, majd ha a hangya meghal, akkor visszakerülve a mezőre megtartja eredeti formáját 

Hangyászsün

A hangyászsün is tud majd forogni, hasonlóan a hangyákhoz. Ezen kívül képes lesz a méretét változtatni, ahogy eszi a hangyákat és növekszik. Jelenleg három mérete létezik. 

További alakzatok

A játékban a többi megjelenítést is próbáltuk változatosra készíteni, így szinte mindennek van több formája, ezekről mindig a modellbeli létrehozáskor döntünk.



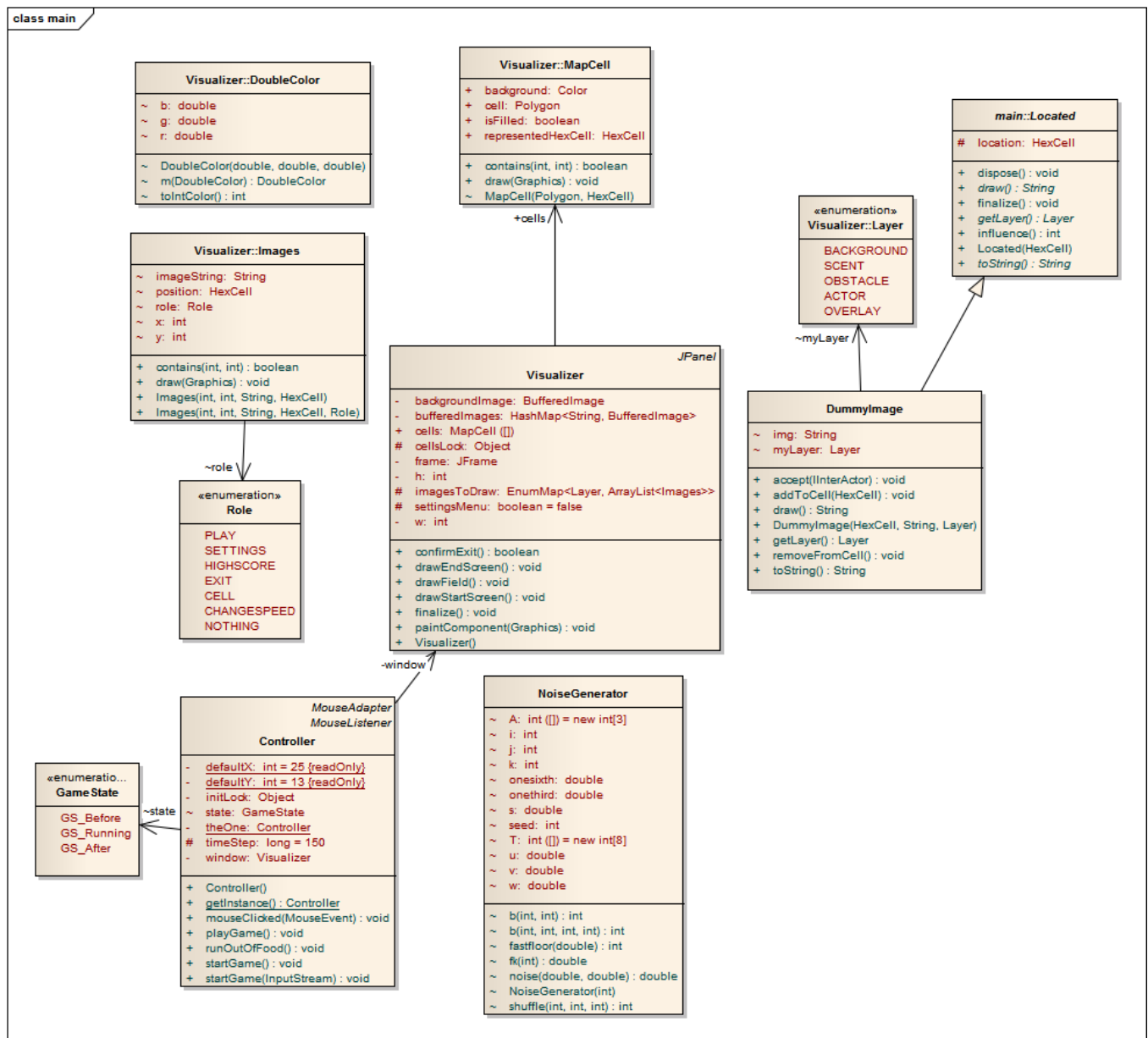
Balról-jobbra: kavics, kavics, tócsa, tócsa, hangyaszag, ételszag, mérreg

11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

Visualizer fog felelni a kirajzolásért. Controller egyszerű mouseListener, aki a Directorral kommunikál, illetve a statikus függvényein keresztül lehet jelezni, hogy melyik mezők változtak (HexCell-eket tart számon). Amikor jelzést kap, hogy egy teljes tick() lefutott, akkor a Visualizer a kirajzolandó mintákat statikus függvényeken keresztül kéri le az osztályoktól (draw() - fájl névvel tér vissza), így bármi újat belerakhatunk. A gombok megnyomására a Controller figyel és jelez a Director-nak, hogy mit kell tenni, így a Controller tartja számon az idő múlását is.

11.2.2 A felület osztály-struktúrája



11.3 A grafikus objektumok felsorolása

11.3.1 Controller

- **Felelősség**
Ez kezeli a játékos interakcióit, ütemezi a modell futását és rajzoltatja ki a megfelelő képernyőket.
- **Ősosztályok**
-
- **Interfészek**
ActionListener
- **Attribútumok**
 - **- theOne: Controller**: Singleton modell alapján csak egy fog belőle létezni.
 - **- window: Visualizer**: Az ablakot megjelenítő osztály egy példánya
- **Metódusok**
 - **+ Controller getInstance()**: Singleton modell szerint ezzel érjük el.
 - **+ void actionPerformed(ActionEvent arg0)**: ActionListener lévén ezen keresztül észeli, hogy történt valami.

11.3.2 DummyImage

- **Felelősség**
Az megjelenítendő objektumok képeit reprezentáló osztály. Minden Locateddel (kivéve DummyImage-el) azonos cellában szerepelni fog, egy a Located képét tartalmazó DummyImage.
- **Ősosztályok**
Located
- **Interfészek**
-
- **Attribútumok**
 - **# img:String**: A reprezentált képfájl.
 - **# myLayer:Layer**: A hozzá tartozó megjelenítési réteg.
- **Metódusok**
 - **+ void accept(IInterActor interactor)**: Formalitás, hogy a HexCell.contents mezőben tárolhassuk, üres függvény.
 - **+ void removeFromCell()**: Eltávolítja magát az aktuálisan elfoglalt HexCell-ből.
 - **+ void addToCell(HexCell location)**: Hozzáadja magát a location HexCell-hez.
 - **+ String draw()**: Get metódus az img tagváltozóra.
 - **+ Layer getLayer()**: Get metódus a myLayer tagváltozóra.

11.3.3 Located (*módosított osztály*)

- (új) Metódusok
 - + **String draw()**: a pályán megjelenő objektumok kirajzolásáért felelős függvény (minden leszármazott megvalósítja)
 - + **Layer getLayer()**: Visszaadja, hogy melyik rétegben kell lennie. (minden leszármazott megvalósítja)

11.3.4 Visualizer

- **Felelősség**
A megjelenítésért felelős osztály.
- **Ősosztályok**
JPanel
- **Interfészek**
-
- **Attribútumok**
 - **# frame:JFrame**: A megjelenítéshez szükséges JFrame
 - **# w:int**: A pálya szélessége. Elmenti, hogy ne kelljen mindig lekérdezni
 - **# h:int**: A pálya magassága. Elmenti, hogy ne kelljen mindig lekérdezni
 - **# cells:MapCell[]**: Ebben tárolja a hexa-hálót
 - **# imagesToDraw:ArrayList<Images>**: Ebben tárol mindet, amit ki kell rajzolni
 - **# bufferedImages:HashMap<String, BufferedImage>**: A képeket hashMap-ben tárolja, hogy ne kelljen többször ugyanazt beolvasnia.
 - **# backgroundImage:BufferedImage**: A háttérképet duplabufferelve tartalmazó változó.
 - **# Layer: enum**: (*BACKGROUND, SCENT, OBSTACLE, ACTOR, OVERLAY*) : A megfelelő rétegzéshez layer-ekbe csoportosítjuk a kirajzolandó dolgokat.
- **Metódusok**
 - + **void drawField()**: A játéktár kirajzolásának előkészítése
 - + **void drawStartScreen()**: A kezdőképernyő kirajzolása.
 - + **void drawEndScreen()**: A játék vége képernyő kirajzolása.
 - + **void paintComponent(Graphics page)**: A függvény felüldefiniálása, hogy az általunk kért képeket kirajzolja az imagesToDraw listából

11.3.5 Visualizer.Images

- **Felelősség**
Segédosztály, amiben tárolható, hogy mit és hova kell rajzolni.
- **Ősosztályok**
-
- **Interfészek**
-
- **Attribútumok**
 - **# x:int:** Pozíció
 - **# y:int:** Pozíció
 - **# imageString:String:** Kép elérési útvonala
 - **# position:HexCell:** Modellbeli pozíciója a képnek (frissítéshez kell)
- **Metódusok**
 - **+ void draw(Graphics page):** A kirajzoláshoz használt függvény

11.3.6 Visualizer.MapCell

- **Felelősség**
A hexa-háló kirajzolásához használt segédosztály.
- **Ősosztályok**
-
- **Interfészek**
-
- **Attribútumok**
 - **+ cell: Polygon:** A hexa-háló egy szemének pontjai
 - **+ background: Color:** A hexa-háló egy szemének színe
 - **+ isFilled: boolean:** A hexa-háló egy szemének kitöltöttsége
- **Metódusok**
 - **+ void draw(Graphics page):** A kirajzoláshoz használt függvény

11. 3. 7. Visualizer.NoiseGenerator

- **Felelősség**
A háttér generálását segítő osztály, lényegében egy álvéletlen zaj generátor.
- **Ősosztályok**
-
- **Interfészek**
-
- **Attribútumok**

- **#i:int**: A zaj osztály randomgenerátorának segédváltozója (x irányú kitérés egészrésze).
- **#j:int**: A zaj osztály randomgenerátorának segédváltozója (y irányú kitérés egészrésze).
- **#k:int**: A zaj osztály randomgenerátorának segédváltozója (z irányú kitérés egészrésze).
- **#seed:int**: A zaj osztály randomgenerátorának seed-je. Két kódsor, amely ugyanazt a seedet használja, majd ugyanolyan sorrendben generálja a zajt ugyanazt a képet fogja előállítani.
- **#u:double**: A zaj osztály randomgenerátorának segédváltozója (x irányú kitérés törtrésze).
- **#v:double**: A zaj osztály randomgenerátorának segédváltozója (y irányú kitérés törtrésze).
- **#w:double**: A zaj osztály randomgenerátorának segédváltozója (z irányú kitérés törtrésze).
- **#s:double**: A zaj osztály randomgenerátorának segédváltozója (átlagos távolság).
- **#onethird:double**: Segédváltozó, amelynek értéke 1/3.
- **#onesixth:double**: Segédváltozó, amelynek értéke 1/6.
- **#T: int[]**: Pszeudorandom számok előállításához előre előállított bitminták.
- **#A: int[]**: Zajeltolás (tengelyenként szekvenciálisan).
- **Metódusok**
 - **+ int fastfloor(double n)**: Gyors alsó egészrész.
 - **+ double fk(int a)**: A véletlenszámokat előállító függvény.
 - **+ int shuffle(int i, int j, int k)**: A T bitmintákból egy 8 tagú összeget előállító függvény.
 - **+ int b(int i, int j, int k, int B)**: A tengelyirányú eltolások alapján eldönti, hogy mely T bitmintát használjuk fel.
 - **+ int b(int N, int B)**: N szám B bittel jobbra toltjának paritását adja vissza.
 - **+ double noise(double x, double y)**: A Perlin-zajt előállító függvény a textúrasík x,y koordinátájában. Determinisztikus, de csak abban az esetben, ha pontosan definiált a beadott koordináták sorrendje.

11. 3. 8 Visualizer.DoubleColor

- **Felelősség**
A háttér generálása során használt RGB színeket reprezentáló osztály.
- **Ősosztályok**
-
- **Interfészek**

-
- **Attribútumok**
 - **# r:double:** A vörös komponens.
 - **#g:double:** A zöld komponens.
 - **# b:double:** A kék komponens.
- **Metódusok**
 - **+ DoubleColor m(DoubleColor o):** Megszorozza a szín komponenseit o szín megfelelő komponenseivel.

11.5 Napló

Időpont	Időtartam	Résztevők	Eredmény
2013.04.25. 17:00:00	1 óra	Varga	Megrajzolta a piktorgammokat, amiket használni fogunk.
2013.04.25. 16:00:00	2 óra	Papp	Kódoptimalizálás, jelentős sebességnövelés
2013.04.27. 17:00:00	3 óra	Kozaróczy	A grafikus felület működési elvének alapjai, graikus interfész főbb elemeinek összeszedése, és a megvalósítás első lépései
2013.04.28. 5:30:00	2 óra	Kozaróczy	A szagok mennyiségének optimalizálása
2013.04.28. 11:30:00	2 óra	Kozaróczy	Optimalizálás, debuggolás
2013.04.28. 18:30:00	1 óra	Papp	"Visualize-ban néha exception dobás" hiba debuggolása és megoldása
2013.04.28. 16:00:00	3 óra	Varga	Hiányzó grafikus elemek elkészítése, új grafikus osztályok felsorolása.
2013.04.29. 9:00:00	2,5 óra	Varga	Módosítások alapján osztályleírások frissítése, kiegészítése, szekvenciadiagrammok elkészítése, a felület osztály-struktúrájának elkészítése.
2013.04.29. 9:30:00	1 óra	Kozaróczy	11.1-es pont elkészítése
2013.04.29. 10:00:00	1 óra	Papp	Dokumentálás
2013.04.28. 16:00:00	6 óra	Tóth	Képek antialiasolása, zajgenerátor lefejlesztése.
2013.04.29. 9:30:00	2 óra	Tóth	Zajgenerátor fejlesztése, dokumentálása
2013.04.29. 11:30:00	0,5 óra	Kozaróczy	Dokumentum tördelése és formázása

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

A gyökérkönyvtárban a fordításhoz és futtatáshoz használt fájlok vannak.

Időpont	Méret	Fájl	Tartalom
2013.05.13. 10:49	202	compile.bat	Fordítási parancsok
2013.04.28. 18:33	46	manifest.mf	Jar fájlhoz
2013.05.13. 10:45	114	run.bat	Futtató parancs

A \img mappában a képek vannak

Időpont	Méret	Fájl
2013.04.28. 18:33	333	ant.png
2013.04.29. 03:46	1`880	anthill.png
2013.04.29. 03:46	1`963	antlion.png
2013.04.29. 03:46	1`030	antscent.png
2013.04.29. 03:46	1`463	antscentvanisherscent.png
2013.04.29. 13:53	2`787	antscentvanisherspray.png
2013.04.29. 03:46	1`177	ant_n.png
2013.04.28. 18:33	1`124	ant_ne.png
2013.04.28. 18:33	1`171	ant_nw.png
2013.04.29. 03:46	1`189	ant_red.png
2013.04.29. 03:46	1`189	ant_red_n.png
2013.04.28. 18:33	1`067	ant_red_ne.png
2013.04.28. 18:33	1`087	ant_red_nw.png
2013.04.29. 03:46	1`296	ant_red_s.png
2013.04.28. 18:33	1`068	ant_red_se.png
2013.04.28. 18:33	1`051	ant_red_sw.png
2013.04.29. 03:46	1`335	ant_s.png
2013.04.28. 18:33	1`133	ant_se.png
2013.04.28. 18:33	1`106	ant_sw.png
2013.05.08. 10:59	821	echidna_0_n.png
2013.05.08. 10:58	1`319	echidna_0_ne.png
2013.05.08. 10:58	1`330	echidna_0_nw.png

2013.05.08. 10:59	824	echidna_0_s.png
2013.05.08. 10:57	1`317	echidna_0_se.png
2013.05.08. 10:57	1`326	echidna_0_sw.png
2013.05.08. 11:01	1`061	echidna_1_n.png
2013.05.08. 11:01	1`348	echidna_1_ne.png
2013.05.08. 11:01	1`351	echidna_1_nw.png
2013.05.08. 11:02	1`051	echidna_1_s.png
2013.05.08. 11:00	1`344	echidna_1_se.png
2013.05.08. 11:01	1`349	echidna_1_sw.png
2013.05.08. 11:03	1`198	echidna_2_n.png
2013.05.08. 11:03	1`366	echidna_2_ne.png
2013.05.08. 11:02	1`380	echidna_2_nw.png
2013.05.08. 11:03	1`197	echidna_2_s.png
2013.05.08. 11:02	1`378	echidna_2_se.png
2013.05.08. 11:02	1`386	echidna_2_sw.png
2013.05.12. 15:58	1`099	empty.png
2013.04.28. 18:33	1`458	exit.png
2013.04.29. 03:46	1`873	food00.png
2013.04.29. 03:46	620	food00_p.png
2013.04.29. 03:46	1`833	food01.png
2013.04.29. 03:46	600	food01_p.png
2013.04.29. 03:46	1`891	food02.png
2013.04.29. 03:46	595	food02_p.png
2013.04.29. 13:21	1`135	foodscent.png
2013.05.08. 11:16	2`974	gameover.png
2013.04.28. 18:33	2`392	highscore.png
2013.04.28. 18:33	2`311	newgame.png
2013.04.27. 19:39	1`206	null.png
2013.04.29. 03:46	2`177	pebble0.png
2013.04.29. 03:46	1`035	pebble1.png
2013.04.29. 03:46	1`992	pebble2.png
2013.05.11. 20:54	1`202	penguin.png
2013.04.28. 18:33	1`697	play.png
2013.04.27. 19:39	1`362	poison.png
2013.04.29. 13:21	1`906	poisonspray.png
2013.04.29. 03:46	2`172	puddle0.png
2013.04.29. 03:46	2`154	puddle1.png
2013.04.29. 03:46	1`892	puddle2.png
2013.04.28. 18:33	1`982	settings.png

2013.05.11. 20:54	1'628	shadow.png
2013.04.29. 13:21	1'917	slime0.png
2013.04.29. 13:21	1'827	slime1.png
2013.04.29. 13:21	1'632	slime2.png
2013.04.29. 13:22	2'117	snail.png
2013.04.28. 18:33	75'889	startscreen.png