

A két torony

74 – Don Serege

Konzulens:

Dr. László Zoltán

Csapattagok

Barabás Péter	RDRCAE	medve9213@gmail.com
Révész Péter	E41E7A	rpeter10110@gmail.com
Boczán Tamás	A5X61F	boczan.tamas@gmail.com
Apáti-Nagy Attila	SSLY1O	tntiti@hotmail.com
Csörgő Szabolcs	XZIK7J	csorgosz@gmail.com

2014. május 16.

Tartalomjegyzék

2.	Követelmény, projekt, funkcionalitás	8
2.1	Bevezetés	8
2.1.1	Cél	8
2.1.2	Szakterület.....	8
2.1.3	Definíciók, rövidítések.....	8
2.1.4	Hivatkozások	8
2.1.5	Összefoglalás	9
2.2	Áttekintés	9
2.2.1	Általános áttekintés.....	9
2.2.2	Funkciók	9
2.2.3	Felhasználók.....	11
2.2.4	Korlátozások	11
2.2.5	Feltételezések, kapcsolatok.....	11
2.3	Követelmények	12
2.3.1	Funkcionális követelmények.....	12
2.3.2	Erőforrásokkal kapcsolatos követelmények.....	13
2.3.3	Átadással kapcsolatos követelmények	13
2.3.4	Egyéb nem funkcionális követelmények.....	14
2.4	Lényeges use-case-ek.....	14
2.4.1	Use-case leírások	14
2.4.2	Use-case diagram.....	16
2.5	Szótár.....	16
2.6	Projekt terv	17
3.	Analízis modell kidolgozása	20
3.1	Objektum katalógus	20
3.1.1	InputHandler:	20
3.1.2	GameLogic:	20
3.1.3	Renderer:.....	20
3.1.4	UtCella:	20
3.1.5	TerepCella:	20
3.1.6	VegCella:	20
3.1.7	Hobbit:	20
3.1.8	Torp:	20
3.1.9	Tunde:.....	21
3.1.10	Ember:	21
3.1.11	Torony:.....	21
3.1.12	Akadaly:	21
3.2	Statikus struktúra diagramok	22
3.3	Osztályok leírása	22
3.4	Szekvencia diagramok.....	25
3.5	State-chartok.....	31
4.	Analízis modell kidolgozása 2	32
4.1	Objektum katalógus	32
4.1.1	Barricade	32
4.1.2	Construct	32
4.1.3	ConstructManager	32

4.1.4	Dwarf	32
4.1.5	Elf.....	32
4.1.6	EndTile.....	32
4.1.7	Enemy.....	32
4.1.8	EnemyGenerator	32
4.1.9	Geometry	32
4.1.10	Hobbit	33
4.1.11	Human	33
4.1.12	FieldTile	33
4.1.13	Mana.....	33
4.1.14	MagicGem	33
4.1.15	PathGenerator	33
4.1.16	PathTile.....	33
4.1.17	Tile	33
4.1.18	Tower.....	33
4.1.19	Updater	34
4.2	Statikus struktúra diagramok	35
4.3	Osztályok leírása	36
4.3.1	Barricade	36
4.3.2	Construct	36
4.3.3	ConstructManager	37
4.3.4	Dwarf	37
4.3.5	Elf.....	38
4.3.6	EndTile.....	38
4.3.7	Enemy.....	39
4.3.8	EnemyGenerator	39
4.3.9	FieldTile	40
4.3.10	Geometry	40
4.3.11	Hobbit	41
4.3.12	Human	41
4.3.13	MagicGem	42
4.3.14	Mana.....	42
4.3.15	PathGenerator	43
4.3.16	PathTile.....	43
4.3.17	Tile	44
4.3.18	Tower.....	44
4.3.19	Updater	45
4.4	Szekvencia diagramok.....	46
4.5	State-chartok.....	53
5.	Analízis modell kidolgozása 3	54
5.1	Objektum katalógus	54
5.1.1	Barricade	54
5.1.2	Construct	54
5.1.3	ConstructManager	54
5.1.4	Dwarf	54
5.1.5	Elf.....	54
5.1.6	EndTile.....	54
5.1.7	Enemy.....	54

5.1.8	EnemyGenerator	54
5.1.9	Geometry	54
5.1.10	Hobbit	55
5.1.11	Human	55
5.1.12	FieldTile	55
5.1.13	Mana	55
5.1.14	MagicGem	55
5.1.15	PathGenerator	55
5.1.16	PathTile	55
5.1.17	Tile	55
5.1.18	Tower	55
5.1.19	Updater	56
5.2	Statikus struktúra diagramok	57
5.3	Osztályok leírása	58
5.3.1	Barricade	58
5.3.2	Construct	58
5.3.3	ConstructManager	59
5.3.4	Dwarf	59
5.3.5	Elf	60
5.3.6	EndTile	60
5.3.7	Enemy	61
5.3.8	EnemyGenerator	61
5.3.9	FieldTile	62
5.3.10	Geometry	62
5.3.11	Hobbit	63
5.3.12	Human	63
5.3.13	MagicGem	64
5.3.14	Mana	64
5.3.15	PathGenerator	65
5.3.16	PathTile	65
5.3.17	Tile	66
5.3.18	Tower	66
5.3.19	Updater	67
5.4	Szekvencia diagramok	68
5.5	State-chartok	73
6.	Szkeleton tervezése	74
6.1	A szkeleton modell valóságos use-case-ei	74
6.1.1	Use-case diagram	74
6.1.2	Use-case leírások	74
6.2	A szkeleton kezelői felületének terve, dialógusok	76
6.3	Szekvencia diagramok a belső működésre	78
6.4	Kommunikációs diagramok	89
7.	Szkeleton beadás	98
7.1	Fordítási és futtatási útmutató	98
7.1.1	Fájllista	98
7.1.2	Fordítás	100
7.1.3	Futtatás	101
7.2	Értékelés	102

8.	Prototípus koncepciója	103
8.0	Változtatások a módosított feladatra	103
8.0.1	A változások leírása	103
8.0.2	Megváltozott osztálydiagram	104
8.0.3	Megváltozott szekvencia diagramok	105
8.1	Prototípus interface-definíciója	109
8.1.1	Az interfész általános leírása	109
8.1.2	Bemeneti nyelv	109
8.1.3	Kimeneti nyelv	112
8.2	Összes részletes use-case	115
8.3	Tesztelési terv	117
8.4	Tesztelést támogató segéd- és fordítóprogramok specifikálása	119
9.	Részletes tervek	120
9.0	Módosított ki- és bemeneti nyelv	120
9.0.1	Bemeneti nyelv	120
9.0.2	Kimeneti nyelv	122
9.1	Osztályok és metódusok tervei	126
9.1.1	Barricade	126
9.1.2	Construct	126
9.1.3	ConstructManager	127
9.1.4	Dwarf	127
9.1.5	Elf	128
9.1.6	EndTile	128
9.1.7	Enemy	129
9.1.8	EnemyGenerator	129
9.1.9	FieldTile	130
9.1.10	Geometry	131
9.1.11	Hobbit	131
9.1.12	Human	132
9.1.13	MagicGem	132
9.1.14	Mana	133
9.1.15	PathGenerator	133
9.1.16	PathTile	134
9.1.17	Tile	134
9.1.18	Tower	135
9.1.19	Updater	135
9.2	A tesztek részletes tervei, leírásuk a teszt nyelvén	137
9.2.1	Pálya betöltése	137
9.2.2	Torony építése	137
9.2.3	Akadály építése	138
9.2.4	Torony fejlesztése	139
9.2.5	Torony lő	139
9.2.6	Akadály lassít	140
9.2.7	Kód	141
9.2.8	Ellenség halála	143
9.2.9	Ellenség osztódik	143
9.3	A tesztelést támogató programok tervei	144
10.	Prototípus beadása	145

10.1	Fordítási és futtatási útmutató	145
10.1.1	Fájllista.....	145
10.1.2	Fordítás.....	147
10.1.3	Futtatás.....	150
10.2	Tesztek jegyzőkönyvei	150
10.2.1	Teszteset 1	150
10.2.2	Teszteset 2	150
10.2.3	Teszteset 3	150
10.2.4	Teszteset 4	150
10.2.5	Teszteset 5	150
10.2.6	Teszteset 6	152
10.2.7	Teszteset 7	152
10.2.8	Teszteset 8	152
10.2.9	Teszteset 9	152
10.3	Értékelés	152
10.4	A tesztek részletes tervei módosítás	153
10.4.1	Teszteset 1 - Pálya betöltése	153
10.4.2	Teszteset 2 - Torony építése	154
10.4.3	Teszteset 3 - Akadály építése	155
10.4.4	Teszteset 4 - Torony fejlesztése	155
10.4.5	Teszteset 5 - Torony lő	156
10.4.6	Teszteset 6 - Akadály lassít.....	157
10.4.7	Teszteset 7 - Kód.....	158
10.4.8	Teszteset 8 - Ellenség halála.....	159
10.4.9	Teszteset 9 - Ellenség osztódik	160
11.	Grafikus felület specifikációja	161
11.1	A grafikus interfész	161
11.2	A grafikus rendszer architektúrája	163
11.2.1	A felület működési elve	163
11.2.2	A felület osztály-struktúrája.....	164
11.3	A grafikus objektumok felsorolása.....	165
11.3.1	BarricadeView.....	165
11.3.2	Controller	165
11.3.3	Drawable.....	166
11.3.4	DwarfView.....	166
11.3.5	ElfView	166
11.3.6	EndTileView	167
11.3.7	FieldTileView	167
11.3.8	HobbitView.....	168
11.3.9	HumanView	168
11.3.10	PathTileView	168
11.3.11	TowerView	169
11.3.12	View	169
11.4	Kapcsolat az alkalmazói rendszerrel	171
12.	Grafikus változat beadása	174
12.1	Fordítási és futtatási útmutató	174
12.1.1	Fájllista.....	174
12.1.2	Fordítás.....	178

12.1.3	Futtatás	180
12.2	Értékelés	180
13.	Összefoglalás	181
13.1	A projektre fordított összes munkaidő	181
13.1.1	Egyéni munkaidő percben	181
13.1.2	Egyéni munkaidő órában	181
13.1.3	Munkaidő diagramok	182
13.1.4	Végső összesítés:.....	184
13.2	A projekt során beadott forrásfájlok és azok sorainak a száma	185
13.2.1	Szkeleton	185
13.2.2	Prototípus	186
13.2.3	Grafikus változat	187
13.3	Projekt összegzés	189
13.3.1	Mit tanultak a projektből konkrétan és általában?.....	189
13.3.2	Mi volt a legnehezebb?	190
13.3.3	Mi volt a legkönnyebb?	190
13.3.4	Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?	190
13.3.5	Milyen változtatási javaslatuk van?.....	190
13.3.6	Milyen feladatot ajánlanának a projektre?	190
14.	Napló	191
2.hét	191
3. hét	192
4. hét	193
5. hét	194
6. hét	196
7. hét	197
8. hét	198
10. hét	199
11. hét	200
13. hét	201

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

Jelen dokumentum a 2013/14 –es tanév második félévében meghirdetett Szoftver laboratórium 4 tárgy által kitűzött követelményeknek megfelelő szoftver tervezési és fejlesztési fázisainak bemutatását, továbbá a program strukturális felépítését tartalmazza.

2.1.2 Szakterület

A dokumentumban tárgyalt szoftver kitűzött célja a szórakoztatás.

2.1.3 Definíciók, rövidítések

UML - Unified Modelling Language, szoftvermodellezési nyelv

MVC - Modell - View - Controller tervezési minta.

JDK - Java Development Kit

JRE - Java Runtime Environment

2.1.4 Hivatkozások

A Szoftver laboratórium 4 tantárgy honlapja.

<https://www.iit.bme.hu/~szoftlab4/>

Az dokumentumban tárgyalt szoftver leírása.

<https://www.iit.bme.hu/~szoftlab4/feladat.shtml>

Az UML honlapja :

<http://www.uml.org>

2.1.5 Összefoglalás

A dokumentum „Analízis modell” részében a programot felépítő osztályok felépítését, működését, felelősségeit és a viselkedését taglalja. Az objektumok és az azok közti kapcsolatok leírása szövegesen, illetve az UML felhasználásával történik.

A következő nagy egysége a dokumentumnak a program úgy nevezett szkeleton változatának részletezése. Itt lesz meghatározva, hogy a végső program milyen bemeneteket fog elfogadni és ezen bemeneteknek milyen hatásai vannak a programra nézve. A leírás tartalmazza, hogy a bemenet hatására a programon belül milyen folyamatok, milyen objektumokon és milyen sorrendben futnak le.

A dokumentum továbbá tartalmazza a program prototípusát, amelyben szerepelnek az interfészek általános leírásai. Ezután következik a program részletes terve, melyben található minden objektum részletes ismertetése. A program tesztelésének menete is itt található. Itt kifejtésre kerül, hogy a tesztelés során mit tesztelünk, hogy ez miként történik meg, és az egész tesztelés hogyan valósul meg.

A dokumentum végén a szoftver grafikus megjelenésének és működési elvének ismertetése a cél.

2.2 Áttekintés

2.2.1 Általános áttekintés

A program a többretegű MVC szerkezeti mintát követ. A modell rész a virtuális világot írja le, a view a grafikus megjelenítéséért felelős rész, a controller pedig a felhasználói interakciókat dolgozza fel (egér kezelés). A controller rész egy-egy interface-en keresztül tudja manipulálni a modellben leírt virtuális világot és a view állapotát. Továbbá létezik egy olyan interface ami által a view el tudja érni a modellt de azt csak használni tudja változtatni nem.

2.2.2 Funkciók

Az Irányítástechnikai és Informatikai tanszéktől kapott feladatunk a Két torony nevű játék elkészítése. A feladat alapját a klasszikus 'Tower Defense' játéklógika adja. Lényege, hogy egy kijelölt bázispontot különböző ellenfelek támadnak. A játékos úgy védekezhetsz, hogy tornyokat helyez el a térképre, amelyek sebzést mérnek az ellenfelekre, így elpusztítva azokat. A játékos célja, hogy megakadályozza a bázispont ellenségek általi elérését.

Konkrét feladatunkban az ellenfeleket a Gyűrűk Urából ismeretes lények alkotják, melyek a következők: ember, tünde, törp és hobbit. Ezen ellenfelek feladata, hogy elérjék a bázispontot, ezáltal eljuttatva a gyűrűt a Végzet hegyéhez. (A gyűrű csak szimbolikus, nem jelenik meg a játékban). Amiben azonban a feladat eltér a klasszikus értelemben vett 'Tower Defense' játékoktól az az, hogy egyetlen egy ellenséges lénynek sem szabad elérni a

bázispontot, mert akkor veszítünk. A lények kinézetben és tulajdonságban térnek el egymástól. A tulajdonság az egyes ellenségek haladási sebessége. Ez a Gyűrűk Ura valóságának próbál minél hűebben megfelelni. Így a leggyorsabban a tünde tud a pályán közlekedni. Őt követi a hobbit, majd az ember, és legvégül a törp. Azonban mozgásuk nem lehet véletlenszerű a térképen, ugyanis van pár előre kijelölt útvonal. Ezeken véletlenszerűen elosztva mennek végig a lények. Fontos azonban, hogy miután egy ellenséges lény rálépett egy bizonyos útra, arról semmilyen körülmények között nem térhet le, attól fogva az adott útvonalon kell végigmennie.

Lényeges, hogy a játék úgynevezett hullámokra van osztva. Ez azt jelenti, hogy jön egy csapat ellenség, majd egy bizonyos idő eltelte után jön egy másik csapat, amely azonban számosságában nagyobb, mint az előző. Ez a minta ismétlődik, de nem a végtelenségig, mert egy idő után az ellenség elfogy. Ilyenkor hátradőlhetünk, mivel beérett a munkánk gyümölcse, és sikerült megvédeni a Végzet hegyét.

A játék egy felhasználót szolgál ki, akinek feladata a már említett Végzet hegyének védelme. A játék közben a Gyűrűk Urából ugyancsak jól ismert varázsló, Szarumán bőrébe bújhatunk virtuálisan.

Ő varázserejét felhasználva próbálja megakadályozni az ellenséges lények eljutását a Végzet hegyéhez. Ehhez alapvetően két játékbeli elem áll szolgálatára: a tornyok és az akadályok.

Tornyok az alapvető védekező egységek. Nélkülük semmi esély nem lenne arra, hogy megvédjük a Végzet hegyét. Szarumán (játékos) bármennyi tornyot elhelyezhet a térképen, csupán két dologra kell figyelnie: A térképen pontosan annyi torony lehet, amennyi terepcella van. Az útra és a Végzet hegyére nem helyezhető el torony, csupán a terepcellákra. A végtelen számú torony elhelyezésében továbbá a varázserő gátolja. Ez egy mérték, amiből gazdálkodva tornyokat építhetünk, vagy akadályokat helyezhetünk el. Varázserőt úgy szerezhetünk, ha az ellenséges lényeket megöljük. Azonban varázserőnkkel nem csak új tornyokat építhetünk, hanem a már jól elhelyezett tornyaink tulajdonságát is befolyásolhatjuk. A torornak alapvetően három tulajdonsága van: hatósugara, tüzelési gyorsasága, és sebzése.

Hatósugár meghatározza, hogy milyen messze tud ellőni a torony. Ennek határa a torony köré rajzolt képzeletbeli kör íve. Ha ennek a képzeletbeli körnek a belsejébe érkezik az ellenséges lény, akkor már elég közel van a toronyhoz, így a torony képes sebzést mérni az ellenségre. A torony egyszerre egy ellenséges lényt tud támadni. Ha több ellenséges lény is hatósugáron belülre kerül, nyilvánvalóan azt az ellenséges lényt támadja, aki a legközelebb van hozzá. Ha vele végzett, és még tartózkodnak a hatósugárban, akkor szintén a hozzá legközelebb lévő kezd el támadni, és továbbra is ezt a mintát követi.

A tüzelési gyorsaság meghatározza, hogy milyen időközönként mér sebzést az ellenséges lényre. Mértékegysége az általunk választott lövés / másodperc, amely megadja, hogy egy másodperc alatt mennyi lövést tud leadni a torony. A lövés egy meghatározott mértékű sebzést bevívó támadás. A lövedéket egy a toronyból indított 'lézersugár' reprezentál, amely az ellenséges lényt (célpont) és a tornyot összekötő egyenes. A harmadik tulajdonság a sebzés mértéke. A sebzés mértékének megértéséhez először definiálnunk kell az ellenséges lények életerejét. Az életerő lényegében az adott ellenséges lény ellenálló-képessége. Ez a számérték adja meg, hogy az adott ellenség mennyi sebzést bír ki anélkül, hogy elpusztulna. A hatósugáron belülre kerülő ellenségre a torony sebzést tud kimérni. A lövés által bevitt sebzés állandóan kivonódik az előbb definiált életerőből. Így, amennyiben az ellenségnek az életereje 0-ra csökken, úgy az ellenség elpusztul, ezáltal megszűnik létezni a játékban.

A játékosnak még egy eszköze van arra, hogy nehezítse a Végzet hegyéhez való eljutást.

Ezek az ún. akadályok abban különböznek a tornyoktól, hogy sebzést nem képesek bevinni, és csak az ellenséges csapatok által járható utakra helyezhetők el. A konkrét játékban levő akadályozza Szarumán által varázssolt gonosz mocsár jelképezi. Amennyiben egy ellenség rálép a játékos által elhelyezett akadályra, akkor amíg az akadály által befolyásolt útszakaszon van, mozgása lelassul. A lassulás mértéke előre meghatározott, mindegyik lény sebességének egy bizonyos százalékát veszíti el. A mocsár megidézéséhez Szarumánnak a toronyépítésnél már említett varázserőre van szüksége, amit ugyanolyan módon tud megszerezni, ahogy azt már a tornyoknál megtárgyaltuk.

A varázserőnek van egy harmadik felhasználási területe is. Azon kívül, hogy tornyokat, mocsarakat helyezhetünk el segítségével a pályán, lehetőségünk van a tornyainkat erősíteni is. Az erősítést úgynevezett varázskövek segítségével tehetjük meg. A varázskövek megidézése úgy történik, hogy a megfelelő tornyot felruházzuk valamilyen különleges tulajdonsággal. Fontos, hogy ha egy tornyot megerősítettünk egy varázskövel, akkor továbbival már nem tehetjük meg, így mindig körültekintően kell eljárunk. Az adott varázskövel való felruházást mutatja, a játékos számára szemléletesebb reprezentálás érdekében, hogy a toronynak megváltozik a színe. Az erősítés lehet az egyes élőlények (tündék, törpök, emberek, hobbitok) szembeni nagyobb sebzés. Ilyenkor az adott lényre kiadott lövésnek nagyobb lesz a sebzése. A varázskövel azonban nem csak a sebzés mértékét lehet megváltoztatni. A varázskő képes a további két tulajdonságát is megváltoztatni a toronynak. Így képes hatósugarat növelni, ezáltal elérni, hogy a torony messzebbre is tudjon támadást bevinni. Egy másik varázskő képes a tüzelési sebességet, azaz a másodpercenkénti lövések számát megnövelni.

Egy akadály varázskövel felruházva nagyobb mértékben lassítja az ellenséges lényt.

2.2.3 Felhasználók

A játék egy felhasználót szolgál ki egyszerre. A felhasználótól elvártak az alapvető számítógép-kezelési készségek. Előnyös, hogy ha van már korábbi számítógépes játékokkal szerzett tapasztalata.

2.2.4 Korlátozások

Fontos igény a szoftverre nézve, hogy alapvető számítástechnikai ismeretek nélkül is használható legyen.

Követelmény, hogy ne igényeljen nagy teljesítményű számítógépet, ugyanakkor a számítógépnek képesnek kell lennie a Java keretrendszer, és az egyszerű grafikus programok futtatására.

2.2.5 Feltételezések, kapcsolatok

A Szoftver laboratórium 4 tantárgy honlapja:

<https://www.iit.bme.hu/~szoftlab4/>

Az dokumentumban tárgyalt szoftver leírása:

<https://www.iit.bme.hu/~szoftlab4/feladat.shtml>

Az UML honlapja:

<http://www.uml.org>

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
1	Új torony építhető	Bemutató	alapvető	feladatkiírás	Torony - építés	
2	Új akadály építhető	Bemutató	alapvető	feladatkiírás	Akadály - építés	
3	Torony hatótáv fejleszthető	Bemutató	alapvető	feladatkiírás	Hatótáv - fejlesztés	
4	Torony tűzgyorsaság fejleszthető	Bemutató	alapvető	feladatkiírás	Tűzgyorsaság - fejlesztés	
5	Torony sebzés fejleszthető	Bemutató	alapvető	feladatkiírás	Sebzés - fejlesztés	
6	Akadály fejleszthető	Bemutató	alapvető	feladatkiírás	Lassítás - fejlesztés	

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
1	Eclipse		Fontos	Csapat	Fejlesztőkörnyezet
2	JDK 1.6		Alapvető	Csapat	Futtatókörnyezet
3	JRE 1.6 rendszerkövetelményeinek eleget tevő számítógép		Alapvető	Csapat	
4	Internetelérés		Fontos	Csapat	Dokumentáció és forráskód megosztása, kommunikáció
5	Microsoft Word, Google Docs		Fontos	Csapat	Dokumentáció egyes elemeinek elkészítése.
6	JUnit		Fontos	Csapat	Program tesztelése
7	GitHub		Fontos	Csapat	Forráskód megosztása, verziókezelés

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
1	Operációs rendszer		Alapvető	Csapat	
2	JRE 1.6		Alapvető	Csapat	Futtatókörnyezet
3	Internetelérés vagy USB port		Fontos	Csapat	Program feltöltése

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
1	Tesztelhetőség		fontos	Csapat	A funkciók adott bemenetre mindig adott kimenettel válaszoljanak.
2	Modularitás		fontos	Csapat	Segíti a tesztelést, illetve későbbi követelmény-változások követését.
3	Hordozhatóság		fontos	Csapat	Platformfüggetlenség, olyan mértékben, amit a Java lehetővé tesz.

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Építés
Rövid leírás	A játékos új épületet hoz létre.
Aktorok	Játékos
Forgatókönyv	Új épület jön létre a megadott cellán. Az akadály- és toronyépítés őse. Varázserő levonódik.

Use-case neve	Akadály-építés
Rövid leírás	A játékos kiválaszt egy útcellát, és épít rá.
Aktorok	Játékos
Forgatókönyv	A megadott útcellán létrejön egy akadály.

Use-case neve	Torony-építés
Rövid leírás	A játékos kiválaszt egy terepcellát, és épít rá.

Aktorok	Játékos
Forgatókönyv	A megadott cellán létrejön egy torony.

Use-case neve	Fejlesztés
Rövid leírás	A játékos kiválaszt egy épületet, és fejleszti.
Aktorok	Játékos
Forgatókönyv	A varázskő típusától függően az épület egy tulajdonsága fejlődik. A 4-féle fejlesztés őse. Varázserő levonódik.

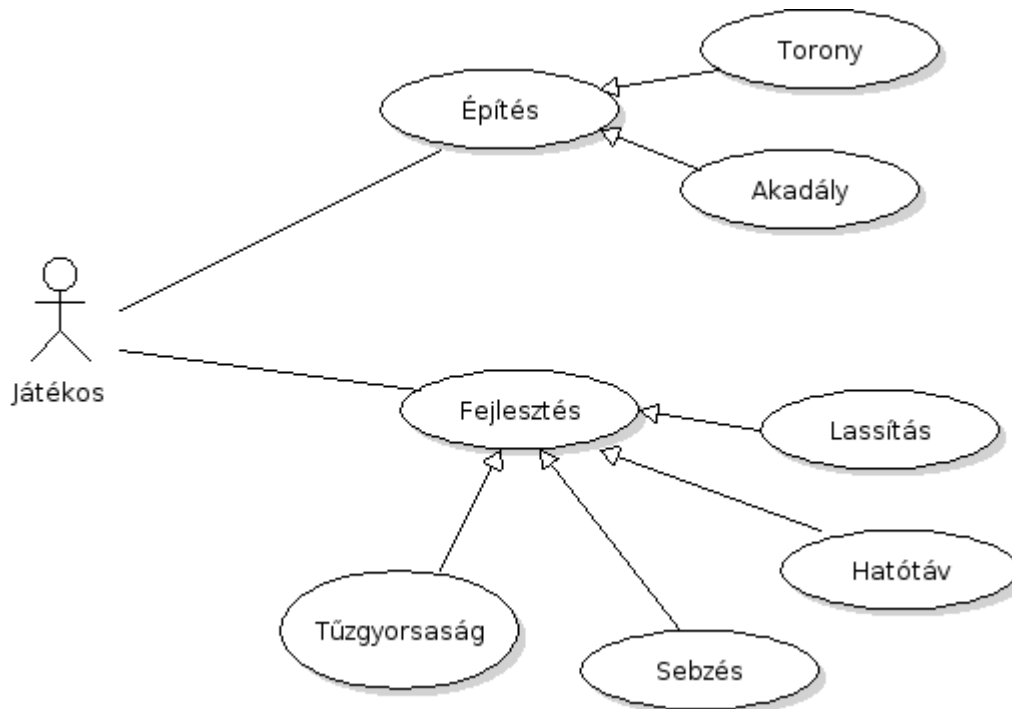
Use-case neve	Lassítás-fejlesztés
Rövid leírás	A játékos kiválaszt egy akadályt, és fejleszti.
Aktorok	Játékos
Forgatókönyv	Az akadály lassítási képessége nő.

Use-case neve	Hatótáv-fejlesztés
Rövid leírás	A játékos kiválaszt egy tornyot, és fejleszti a hatótávját.
Aktorok	Játékos
Forgatókönyv	A torony hatótávja nő.

Use-case neve	Tűzgyorsaság-fejlesztés
Rövid leírás	A játékos kiválaszt egy tornyot, és fejleszti a tűzgyorsaságát.
Aktorok	Játékos
Forgatókönyv	A torony tűzgyorsasága nő.

Use-case neve	Sebzés-fejlesztés
Rövid leírás	A játékos kiválaszt egy tornyot, és fejleszti a sebzését egy adott ellenség ellen.
Aktorok	Játékos

2.4.2 Use-case diagram



2.5 Szótár

ellenség: a játék célja ezeknek a kiiktatása. Ha mindegyiket sikerül kiiktatni, a játék győzelemmel ér véget. Lehet különböző mennyiségű életerejük, mozgási sebességük, és kinézetük. A játék során egyre több ellenséget kell elpusztítani. Egy útvonal mentén mennek végig a pályán, miközben tornyok lőhetnek rájuk. Ha egy ellenség eléri az útvonal végét, a játék vereséggel ér véget.

ellenségfajta: azonos életerővel, sebességgel és kinézettel ellátott ellenségek csoportja.

hullám: ellenségcsapatok időközönkénti megjelenése a pályán. A hullámok megadott időközönként érkeznek, és ez idővel egyre gyakoribbá válik.

életerő: az ellenségek egyik tulajdonsága, mely minden lövésnél csökken. Mikor elfogyott, az ellenség elpusztul.

lövés: a torony tud lőni. Egy lövés hatására a célba vett ellenség életereje csökken a torony sebzése alapján.

sebzés: az az életerő-mennyiség amit egy lövés levesz az ellenségektől.

hatótáv: az a távolság, amin belül a torony lőni tud.

tűzelési sebesség: a torony által leadott lövések gyakorisága.

varázserő: az egyetlen nyersanyag, ami a játékos rendelkezésére áll. Ellenségek elpusztításával juthat a játékos több varázserőhöz.

torony: a játékos építhet tornyot egy kijelölt terep cellára varázsereje felhasználásával. Útra és a báziscellára nem lehet építeni. A tornyok tudnak ellenségekre lőni, és van saját sebzésük, hatótávuk és tűzelési sebességük. Varázserőért varázskövekkel lehet őket fejleszteni, miután celláját kijelöltük.

akadály: útcellára építhető, a rajta áthaladó ellenségek sebességét csökkentik. A lassítás

mértéke egy varázskővel növelhető.

épület: akadály vagy torony.

játekos: a személy, aki a játékkal játszik.

fejlesztés: varázskövek toronyhoz vagy akadályhoz való rendelése.

varázskő: a tornyok és akadályok erősítésére szolgálnak. Minden ellenségtípusra külön varázskő jut mely növeli a torony egy ellenségtípuson okozta sebzését. Ezen felül lehet olyan varázskő mely a torony hatótávját növeli, illetve olyan, ami a tüzelési sebességét növeli. Az akadályra rakott varázskő a lassítás mértékét növeli. A varázskövek különböző mennyiségű varázserőbe kerülhetnek. Egyszerre egy épületen csak egy varázskő lehet.

cella: egy terület, amit kijelölhetünk.

terepcella: egy olyan cella, melyre tornyot lehet építeni.

útcella: egy olyan cella, melyen mozoghatnak az ellenségek, és akadályt lehet rá építeni.

útvonal: útcellák egy olyan sorozata, melyen az ellenségek végigmennek. Ha egy ellenség a végére ér, a játékot elvesztjük.

bázispont: az a cella, amire ha elér az ellenség, a játék vereséggel végződik.

pálya: az összes játékban létező cella, és a rajta elhelyezkedő tornyok és ellenségek gyűjteménye.

2.6 Projekt terv

A projekt három megvalósítási lépésre bontható, ezen lépések és határidejük a következők:

- szkeleton – március 26.
- prototípus – április 23.
- grafikus – május 14.

A projekt megvalósítási lépései:

- **Szkeleton:** A szkeleton változat célja annak bizonyítása, hogy az objektum és dinamikus modellek a definiált feladat egy modelljét alkotják. A szkeleton egy program, amelyben már valamennyi, a végső rendszerben is szereplő business objektum szerepel. Az objektumoknak csak az interfésze definiált. Valamennyi metódus az indulás pillanatában az ernyőre szöveges változatban kiírja a saját nevét, majd meghívja azon metódusokat, amelyeket a szolgáltatás végrehajtása érdekében meg kell hívnia. Amennyiben a metódusból valamely feltétel fennállása esetén hívunk meg más metódusokat, akkor a feltételre vonatkozó kérdést interaktívan az ernyőn fel kell tenni és a kapott válasz alapján kell a továbbiakban eljárni. A szkeletonnak alkalmasnak kell lenni arra, hogy a különböző forgatókönyvek és szekvencia diagramok ellenőrizhetők legyenek.
- **Prototípus:** A prototípus program célja annak demonstrálása, hogy a program elkészült, helyesen működik, valamennyi feladatát teljesíti. A prototípus változat egy elkészült program kivéve a kifejlett grafikus

interfészt. A változat tervezési szempontból elkészült, az ütemezés, az aktív objektumok kezelése megoldott. A business objektumok - a megjelenítésre vonatkozó részeket kivéve - valamennyi metódusa a végleges algoritmusokat tartalmazza. A megjelenítés és működtetés egy alfanumerikus ernyőn követhető, ugyanakkor a megjelenítés fájlban is logolható, ezzel megteremtve a rendszer tesztelésének lehetőségét.

- **Teljes/grafikus változat:** A prototípustól csak a kezelőfelület minőségében tér el.

A projekt végrehajtásának egyéb határidői:

- febr. 24. - Követelmény, projekt, funkcionalitás - beadás
- márc. 3. - Analízis modell kidolgozása 1. - beadás
- márc. 10. - Analízis modell kidolgozása 2. - beadás
- márc. 17. - Szkeleton tervezése - beadás
- márc. 24. - Szkeleton - beadás
- márc. 31. - Prototípus koncepciója - beadás
- ápr. 7. - Részletes tervek - beadás
- ápr. 22. - Prototípus - beadás
- ápr. 28. - Grafikus felület specifikációja - beadás
- máj. 12. - Grafikus változat - beadás
- máj. 16. - Összefoglalás - beadás

Csapattagok és csapat felépítése

Az alapvető koncepciónk az, hogy minden csapattag megközelítőleg egyenlő részt vállaljon minden feladatból, így biztosítva a munkamegosztást, valamint a minél szélesebb körű ismeretek elsajátítását. Ennek megfelelően igyekszünk minden feladattípust egyenlő részekre szétosztani a tagok között. A feladatmegosztást szükség esetén módosítjuk a tagok egyéni képességeinek, idejének és egyéb tényezőknek megfelelően.

A megfelelő szervezettség és a kérdések eldöntésének megkönnyítése érdekében választunk egy csapatvezetőt, aki koordinálja a folyamatokat.

Csapattagok közti kommunikáció:

Heti minimum két alkalommal meeting, lehetőség szerint minden csapattag jelenlétében. Ilyenkor tisztázzuk a hátralévő feladatokat, majd ezek az egységes megoldási menet megbeszélése után szétosztásra kerülnek a tagok között. Emellett ezeken az alkalmakon korrigáljuk az esetleges eltéréseket az egyénileg megoldott feladatrészek között, majd véglegesítjük a dokumentáció eddigi részeit.

A meetingek között a kommunikáció körlevelekben zajlik, valamint a Google Drive lehetőséget ad, hogy a közösen szerkesztett dokumentációba megjegyzéseket szúrjunk be. Utóbbi nagyban segíti a feladatok közti eltérések feltárását és javítását.

A csapat:

- Barabás Péter (csapatvezető)
- Boczán Tamás
- Apáti-Nagy Attila
- Csörgő Szabolcs
- Révész Péter

Szükséges erőforrások:

lásd: 2.3.2. Erőforrásokkal kapcsolatos követelmények rész

Csoportmunkát támogató eszközök, választott technológiák:

Microsoft Word, Google Drive – dokumentáció megosztása és szerkesztése

GIT – forráskód megosztása

Eclipse – fejlesztőkörnyezet

Visual Paradigm, Violet UML Editor – UML diagramok készítése és forráskódból való generálása

JUnit – tesztelés

3. Analízis modell kidolgozása

3.1 *Objektum katalógus*

3.1.1 **InputHandler:**

Ez az osztály kezeli a felhasználótól érkező bemeneteket. (ide még lki kéne találni, hogy hogyan lesz ezek alapján módosítva az épületek)

3.1.2 **GameLogic:**

Ez a játék lelke. Ez az osztály kezeli, generálja és vezérli az összes ellenséget és épületet, valamint ütemezi ezek tevékenységét. Szintén ez felel a játék és a hullámok kezdetéért és végéért.

3.1.3 **Renderer:**

A megjelenítésért felelős osztály. Ez jelenít meg minden látható elemet a képernyőn.

3.1.4 **UtCella:**

A pálya olyan cellája, ami útnak minősül, azaz az ellenségek haladhatnak rajta és akadályok helyezhetők rá. Ezekre a cellákra nem építhető torony.

3.1.5 **TerepCella:**

A pálya olyan cellája, amin nem haladhat ellenség, ezekre építhetjük a tornyokat.

3.1.6 **VegCella:**

Ezen a cellán van a Végzet hegye. Léphet rá ellenfél és amennyiben ez megtörténik, a játék vereséggel véget ér.

3.1.7 **Hobbit:**

A négy ellenségfajta egyike. Életerővel és egyedi sebesség paraméterrel rendelkezik. Minden példánya egy útvonalat ismer, ezen halad végig a játék során. A példányok tárolják az aktuális életerejüket és pozíciójukat a pályán, valamint hogy hat-e rá és milyen mértékben lassítás.

3.1.8 **Torp:**

A négy ellenségfajta egyike. Életerővel és egyedi sebesség paraméterrel rendelkezik. Minden példánya egy útvonalat ismer, ezen halad végig a játék során. A példányok tárolják az aktuális életerejüket és pozíciójukat a pályán, valamint hogy hat-e rá és milyen mértékben lassítás.

3.1.9 Tunde:

A négy ellenségfajta egyike. Életerővel és egyedi sebesség paraméterrel rendelkezik. Minden példánya egy útvonalat ismer, ezen halad végig a játék során. A példányok tárolják az aktuális életerejüket és pozíciójukat a pályán, valamint hogy hat-e rá és milyen mértékben lassítás.

3.1.10 Ember:

A négy ellenségfajta egyike. Életerővel és egyedi sebesség paraméterrel rendelkezik. Minden példánya egy útvonalat ismer, ezen halad végig a játék során. A példányok tárolják az aktuális életerejüket és pozíciójukat a pályán, valamint hogy hat-e rá és milyen mértékben lassítás.

3.1.11 Torony:

A játékos helyezheti el őket terep cellákon, elhelyezés után varázskővel fejleszthetők. A példányai ismerik a pályán lévő helyzetüket, hatótávolságukat, tüzelési gyorsaságukat és hogy milyen varázskő van bennük. Rá tud lőni az ellenségekre és megsebezni őket, amennyiben azok a hatótávolságán belül vannak.

3.1.12 Akadaly:

A játékos helyezheti el őket az út cellákon, elhelyezés után varázskővel fejleszthetők. A példányai ismerik a pályán lévő helyzetüket, hogy milyen varázskő van bennük, valamint az általuk okozott lassítás mértékét. Az rajta áthaladó ellenségek mozgását lassítja addig, amíg az általa lefedett cellában tartózkodnak

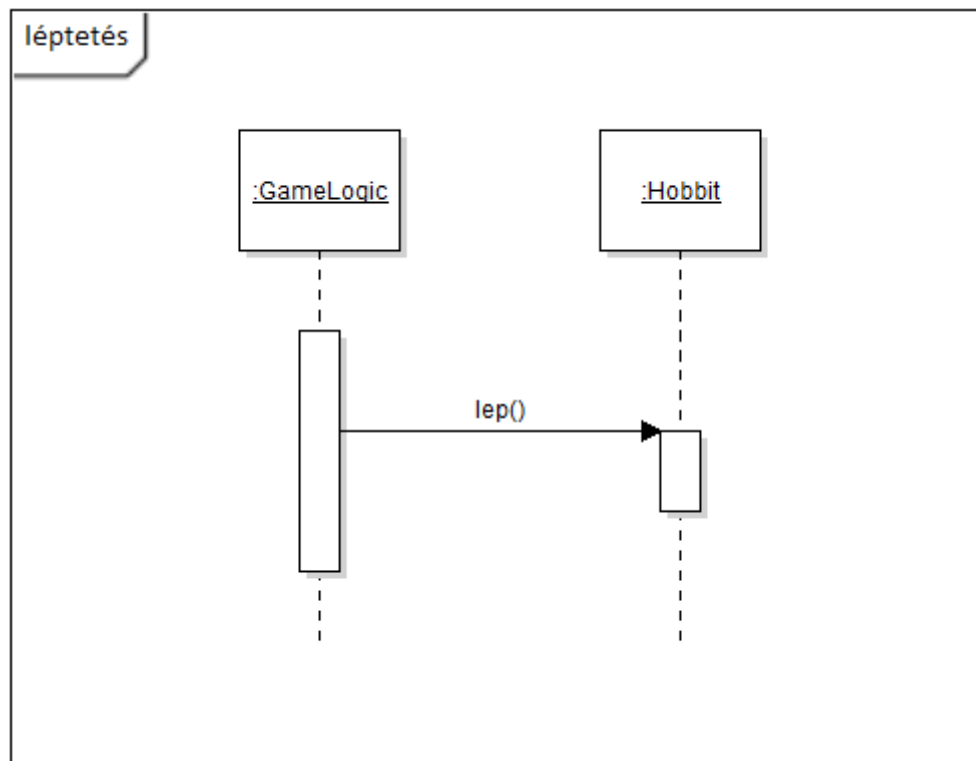
- lassitas: float
- **Metódusok:**
 - **getHatotav(): float** - visszaadja az akadály hatótávolságát, ami akadály esetében fix 1.
 - **interakco(ArrayList<Ellenseg>):void** - a paraméterként kapott listában lévő ellenségeket lelassítja a lassitas attribútumnak megfelelő mértékben.
 - **fejleszt(int):void** - a paraméter által azonosított varázskövet helyezi az akadályba és ennek megfelelően módosítja a lassitas attribútumot.
- **Cella**
 - **Felelősség:** A pálya celláit reprezentálja, absztrakt osztály.
 - **Attribútumok:**
 - vanEpulet: boolean
 - cellaTipusID: int
 - **Metódusok:**
 - **getEpulet(): boolean** - visszaadja, hogy található-e a cellán épület, vagy nem
- **Ellenseg**
 - **Felelősség:** Az ellenségeket reprezentáló absztrakt ősosztály. Az ellenségek szükséges adatait tárolja.
 - **Attribútumok:**
 - sebesseg: float
 - sebessegModosito: float
 - elet: float
 - ellensegTipusID: int
 - pozicioX: int
 - pozicioY: int
 - utvonal: ArrayList<UtvonalSzegmens>
 - megtettTav: float
 - **Metódusok:**
 - **serul(float): void** - a paraméterként kapott értéket levonja az ellenség életéből
 - **setSebessegModosito(float):void** - módosítja az ellenség sebességModosito értékét, aminek hatására változik a sebessége is
 - **getTipusID(): int** - egy számértéket ad vissza, ami alapján azonosítható, hogy milyen fajta ellenségről van szó
 - **getPozicioX(): int** - az ellenség X tengely menti koordinátáját adja vissza
 - **getPozicioY(): int** - az ellenség Y tengely menti koordinátáját adja vissza
 - **lep(): void** - a sebesseg és sebessegModosito értékeknek megfelelően elmozdítja az ellenséget a benne tárolt útvonalnak megfelelően, valamint hozzáadja az elmozdulást a megtettTav változóhoz.
- **Ember**
 - **Felelősség:** Az ember típusú ellenségek osztálya
 - **Ősosztályok:** Ellenseg
- **Epulet**

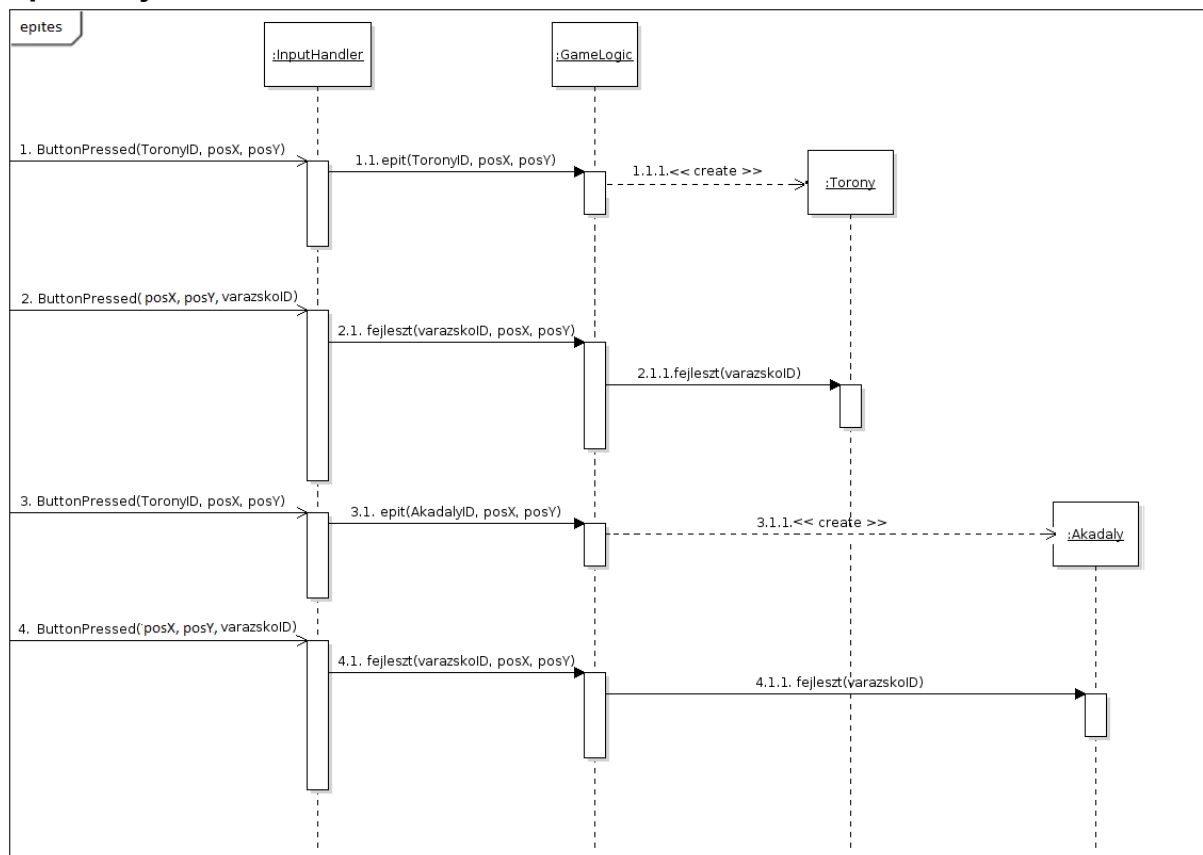
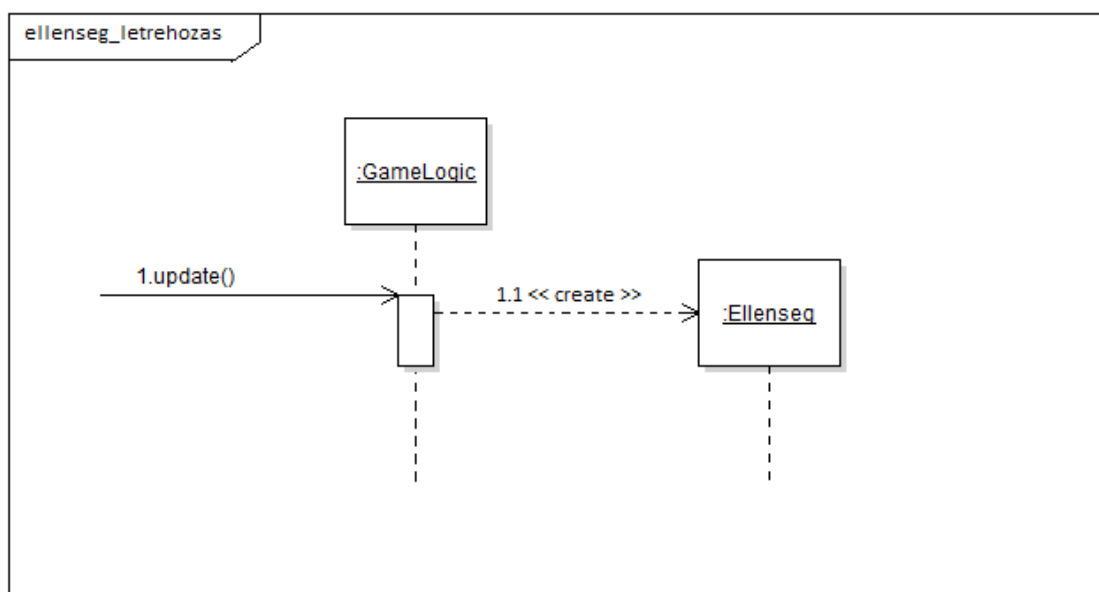
- **Felelősség:** Az épületeket reprezentáló absztrakt őszosztály. Az épületek alapvető adatait tárolja.
- **Attribútumok:**
 - varazskold: int
 - pozicioX: int
 - pozicioY: int
- **Metódusok:**
 - **getPozicioX(): int** - az épület X tengely menti koordinátáját adja vissza
 - **getPozicioY(): int** - az épület Y tengely menti koordinátáját adja vissza
- **GameLogic**
 - **Felelősség:** Ez az osztály vezérli a játékot. Itt tároljuk az épületek és ellenségek listáját, valamint a pályát és minden egyéb, a játékmenethez szükséges adatokat. Az időszimuláció ebben az osztályban történik. Szintén ebben az osztályban található a játék fő ciklusa, ami a frissítéseket végzi minden játékelemen, mint például időszelentenként léptetés, lövés, új ellenségek létrehozása és ezek a hatásainak végrehajtása. A kapott felhasználói beavatkozásoknak változtatásait is itt eszközöljük.
 - **Attribútumok**
 - cellak: ArrayList<Cella>
 - ellensegek: ArrayList<Ellenseg>
 - epuletek: ArrayList<Epulet>
 - utvonal: ArrayList<UtvonalSzegmens>
 - kezdoPontok: int[]
 - regi_ido: long
 - ido: long
 - hullam: int
 - varazsero: int
 - **Metódusok:**
 - **epit(int, int, int): void** - a paraméterként kapott int alapján azonosítja, hogy milyen épületet kell elhelyeznie a szintén paraméterként kapott koordinátára.
 - **fejleszt(int, int, int): void** - a paraméterként kapott int alapján azonosítja, hogy milyen épületet kell fejleszteni a szintén paraméterként kapott koordinátán.
 - **init(): void** - a program indulásakor fut le a függvény. Ennek hatására töltődik be a pálya, generálódnak le ez alapján az ellenségek útvonalai, valamint beállítja az alapértékeket.
- **Hobbit**
 - **Felelősség:** A hobbit típusú ellenségek osztálya
 - **Őszosztályok:** Ellenseg
- **InputHandler**
 - **Felelősség:** A felhasználótól érkező bemeneteket kezeli és továbbítja a GameLogic felé.
 - **Attribútumok:**
 - gameLogic: GameLogic
- **Renderer**
 - **Felelősség:** a játék grafikus megjelenítéséért felel.
 - **Attribútumok:**

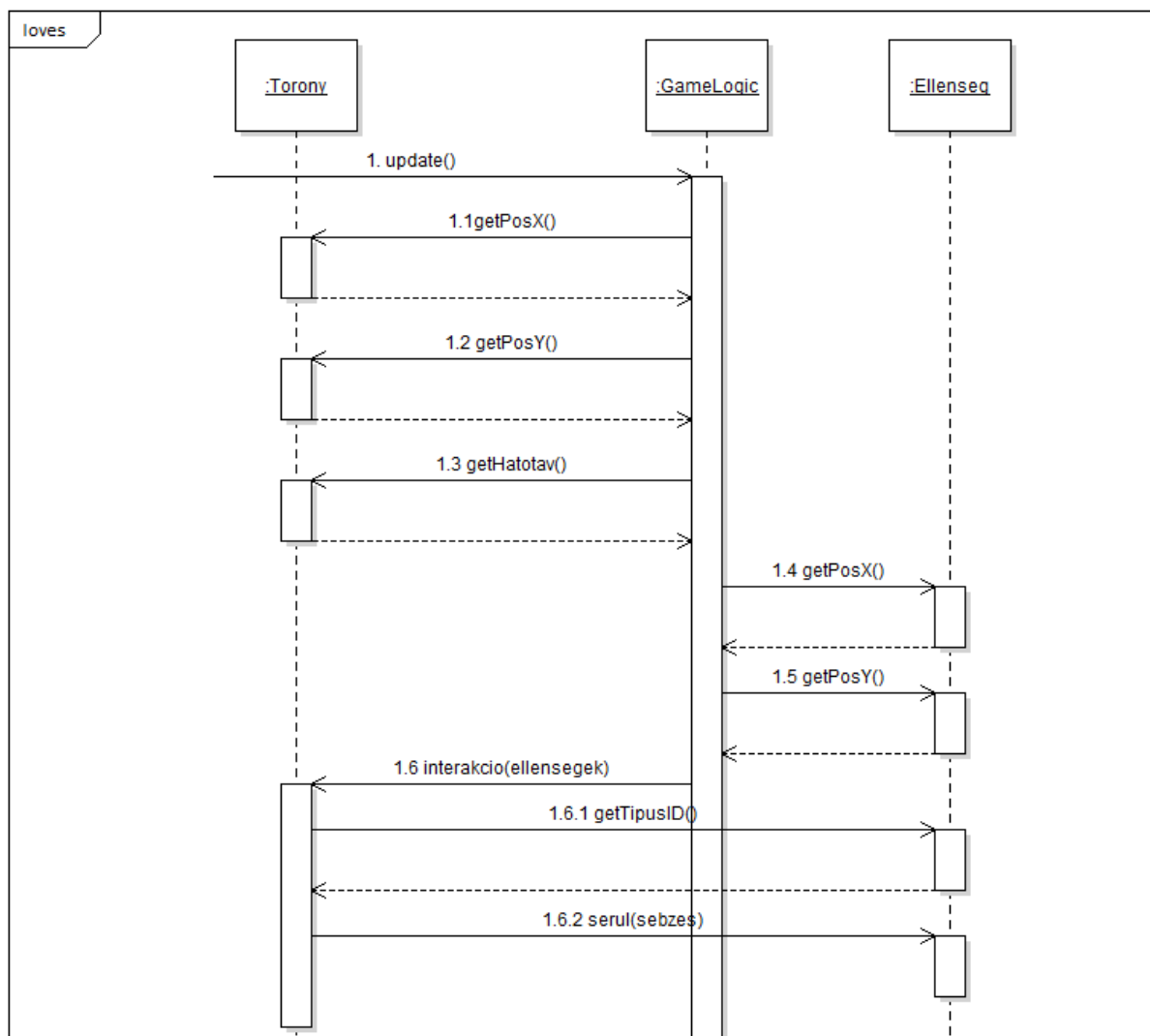
- gameLogic: GameLogic
- **TerepCella**
 - **Felelősség:** A pálya terep típusú celláit reprezentálja.
 - **Ősosztályok:** Cella
- **Torony**
 - **Felelősség:** a TerepCella cellákra elhelyezhető tornyok osztálya.
 - **Ősosztályok:** Epulet
 - **Attribútumok:**
 - sebzes: float
 - tuzelésiSebesseg: float
 - hatotav: float
 - **Metódusok:**
 - **getHatotav(): float** - visszaadja a torony hatótávolságát, amit a hatotav attribútum határoz meg.
 - **interakco(ArrayList<Ellenseg>): void** - a paraméterként kapott listában lévő ellenségeket sebzi meg az esetlegesen behelyezett varázskövet is figyelembe véve.
 - **fejleszt(int):void** - a paraméter által azonosított varázskövet helyezi a toronyba és ennek megfelelően módosítja a hatotav vagy sebzes attribútumot.
- **Torp**
 - **Felelősség:** A törp ttípusú ellenségek osztálya
 - **Ősosztályok:** Ellenseg
- **Tunde**
 - **Felelősség:** A tünde ttípusú ellenségek osztálya
 - **Ősosztályok:** Ellenseg
- **TerepCella**
 - **Felelősség:** A pálya terep típusú celláit reprezentálja.
 - **Ősosztályok:** Cella
- **VegCella**
 - **Felelősség:** A Végzet hegylt reprezentálja. Ha ellenség lép rá, a játékot elvesztjük.
 - **Ősosztályok:** Cella
 - **Attribútumok:**
 - pozicioX: int
 - pozicioY: int
 - **Metódusok:**
 - **beertek(ArrayList<Ellenseg>): boolean** - igazgal tér vissza, ha a paraméterként kapott listában szereplő ellenségek közül bármelyik is ezen a cellán tartózkodik

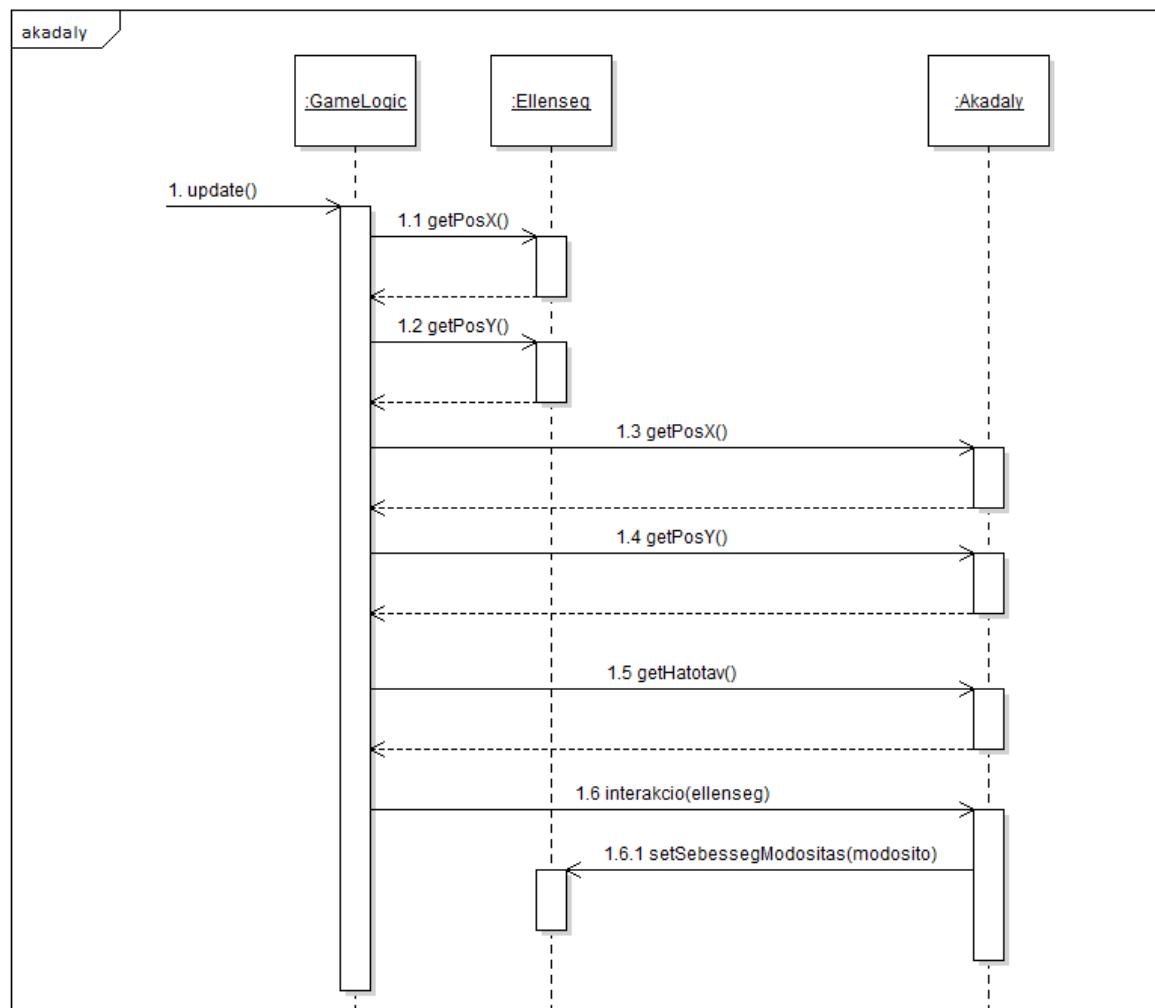
3.4 Szekvencia diagramok

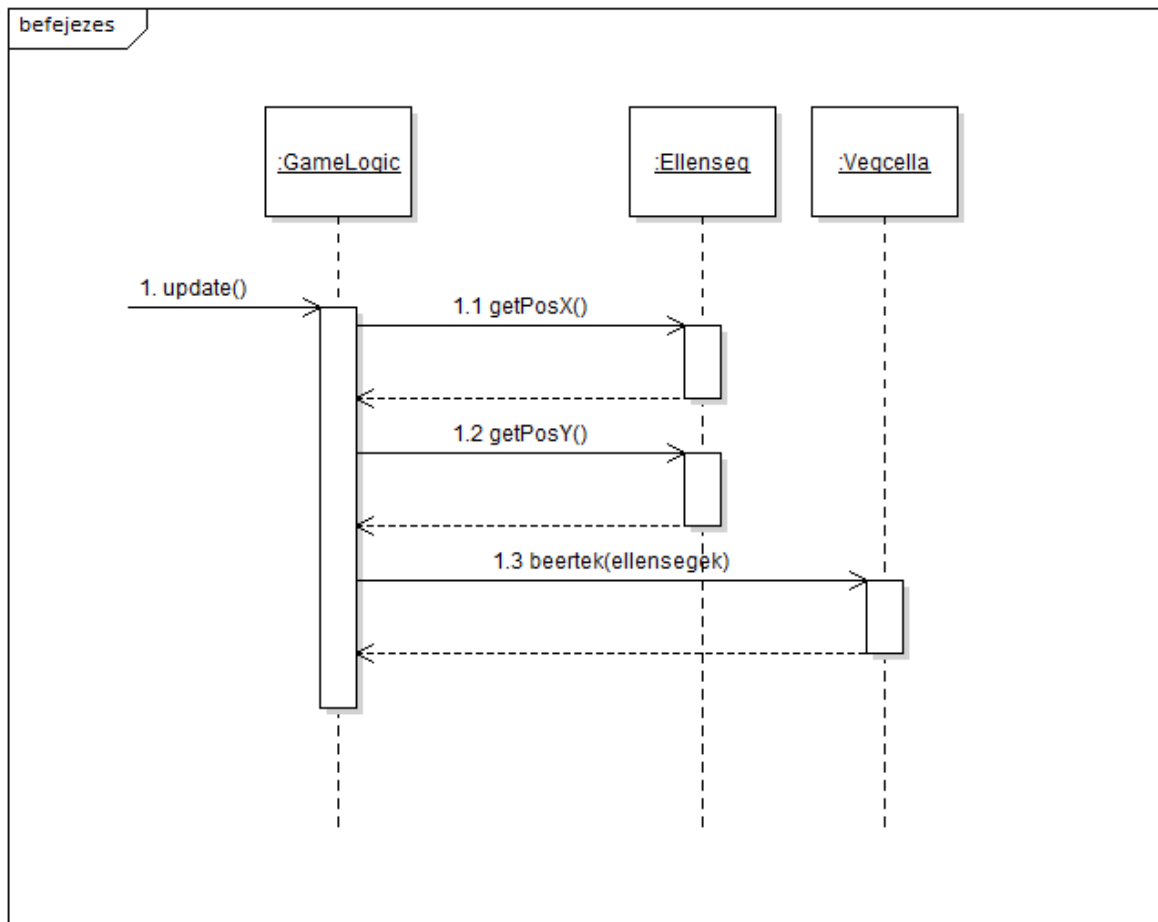
Lépés:



Építés-fejlesztés:**Ellenség létrehozás :**

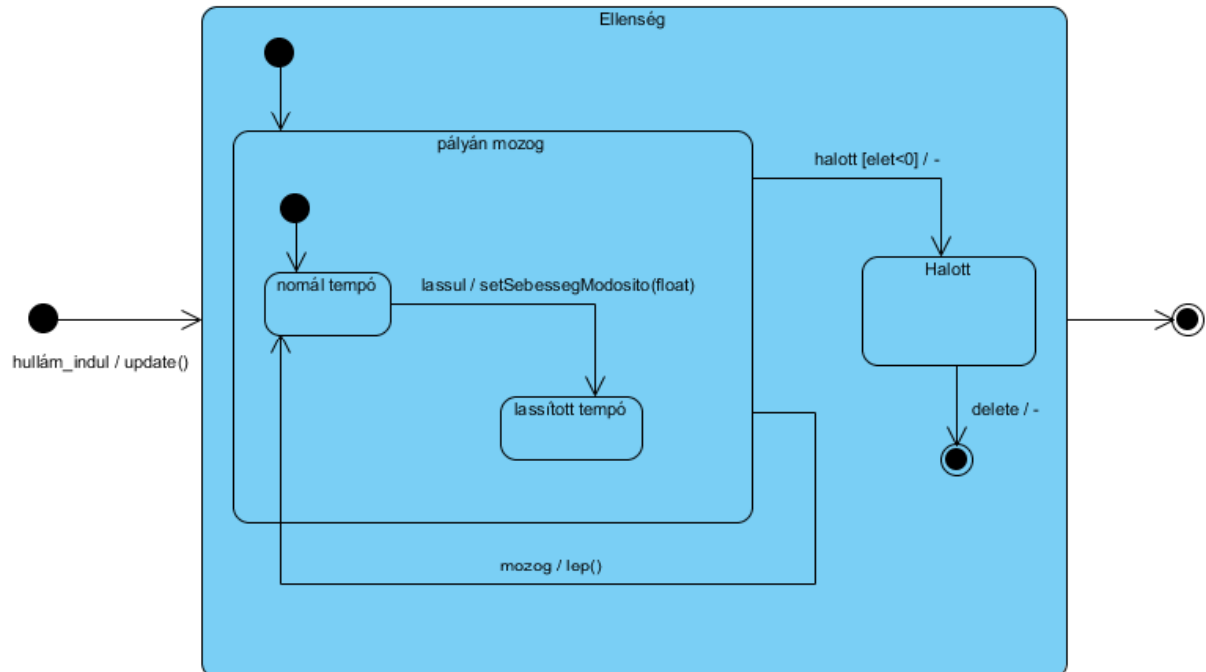
Lövés :

Akadályra lépés :

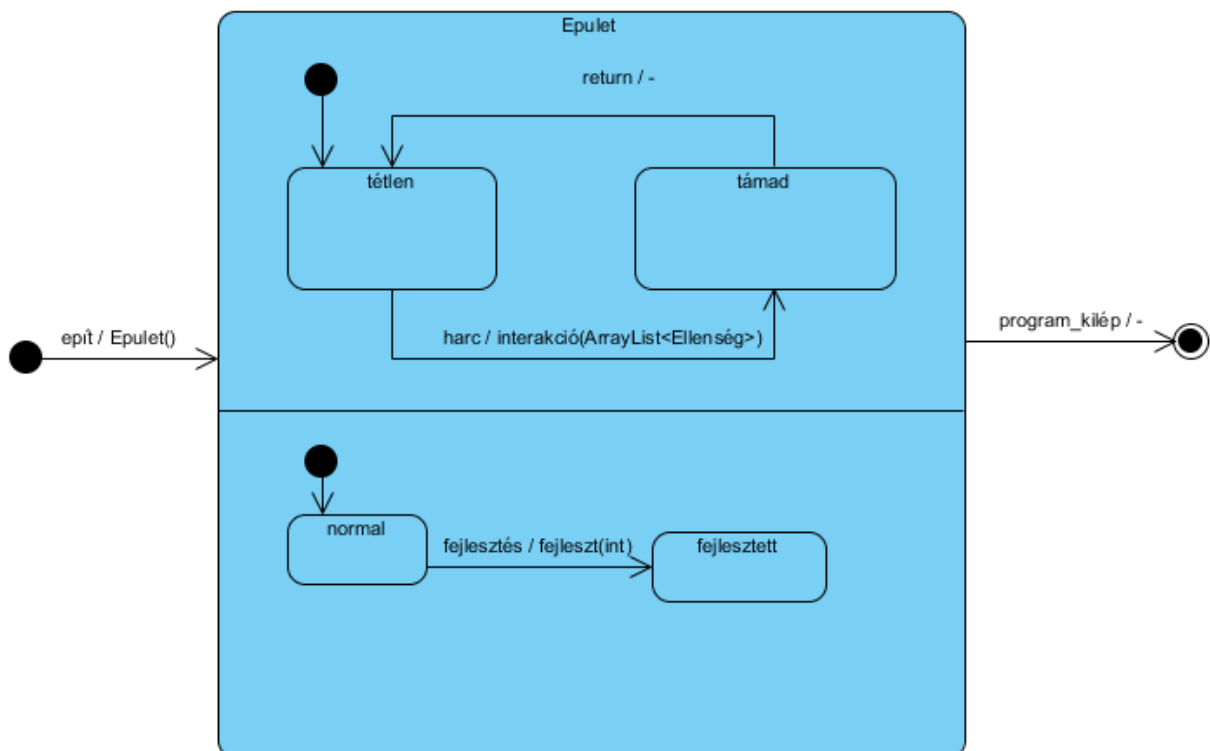
Játék vége:

3.5 State-chartok

Ellenség:



Épület:



4. Analízis modell kidolgozása 2

4.1 Objektum katalógus

4.1.1 Barricade

Olyan Construct, amit csak PathTile-re lehet építeni, ahol lelassítja az ott áthaladó Enemy-eket, azaz később léphetnek tovább. Benne található egy varázskő (MagicGem), ami a lassítás mértékét növeli. Ezt a követ fejleszteni is lehet. Ismeri a Tile-t ami őt tárolja.

4.1.2 Construct

Ezzel az absztrakt osztállyal lehetséges a Tower-ök és Barricade-ek egy tömbben tárolása.

4.1.3 ConstructManager

Ez az osztály felelős a Construct-ok létrehozásáért és fejlesztéséért. Meg tudja nézni, hogy az építendő Construct-hoz vagy MagicGem-hez van-e elég Mana, ha van létre tudja hozni, és le tudja vonni a Mana-t.

4.1.4 Dwarf

Az Enemy leszármazottja, tárolja a saját életerejét, a halálakor kapható varázserő értékét és a sebességét. Tud lépni, és le tudja vonni az életerejét. Egyedi attribútumai vannak a többi ellenség típushoz képest.

4.1.5 Elf

Az Enemy leszármazottja, tárolja a saját életerejét, a halálakor kapható varázserő értékét és a sebességét. Tud lépni, és le tudja vonni az életerejét. Egyedi attribútumai vannak a többi ellenség típushoz képest.

4.1.6 EndTile

Olyan Tile, amin ha ellenség tartózkodik, a játék vereséggel ér véget.

4.1.7 Enemy

Ezzel az absztrakt osztállyal lehetséges a különböző ellenségtípusok egy tömbben tárolása.

4.1.8 EnemyGenerator

Ez az osztály felelős az ellenségek létrehozásáért. Ismeri a PathGeneratort, ami segítségével a létrehozott Enemy-eket le tudja rakni az útvonalakra. Tudja, hogy mikor kell ellenséget létrehozni, illetve meg tudja mondani, hogy létrehozta-e az utolsó ellenséget.

4.1.9 Geometry

Felelős a csempék tárolásáért és elrendezéséért. Meg tudja adni, hogy egy pont adott környezetében melyik cellák vannak. Vissza tudja adni az összes csempét.

4.1.10 Hobbít

Az Enemy leszármazottja, tárolja a saját életerejét, a halálakor kapható varázserő értékét és a sebességét. Tud lépni, és le tudja vonni az életerejét. Egyedi attribútumai vannak a többi ellenség típushoz képest.

4.1.11 Human

Az Enemy leszármazottja, tárolja a saját életerejét, a halálakor kapható varázserő értékét és a sebességét. Tud lépni, és le tudja vonni az életerejét. Egyedi attribútumai vannak a többi ellenség típushoz képest.

4.1.12 FieldTile

Egy csempe típus, amire egy tornyot lehet építeni. Tárolja a rajta lévő tornyot.

4.1.13 Mana

A játék nyersanyaga, tárolja az elérhető varázserő mennyiségét. Meg tudja mondani, hogy adott mennyiség rendelkezésre áll-e, illetve le tudja csökkenteni és meg tudja növelni adott értékekkel.

4.1.14 MagicGem

Varázskő, melyet egy toronyba lehet tenni varázserőért cserébe. Növelheti a torony hatótávát, tüzelési gyorsaságát és a sebzését a különböző ellenségtípusok ellen. Meg tudja mondani, milyen típusú.

4.1.15 PathGenerator

Legenerálja és tárolja az útvonalakat amiken ellenségek járhatnak végig. Le tudja rakni az ellenségeket az útvonalra.

4.1.16 PathTile

Egy csempe típus, amin ellenségek léphetnek és akadályt lehet rá építeni. Tárolja a rajta lévő akadályt, és a rajta lévő ellenségeket. Ismeri a következő útcsempét amire róla lépni lehet.

4.1.17 Tile

Ezzel az absztrakt osztállyal lehetséges az EndTile-ok, FieldTile-ok és PathTile-ok egy tömbben tárolása. Minden Tile ismeri az őt tároló geometriát és meg tudja mondani milyen típusú (pl. EndTile).

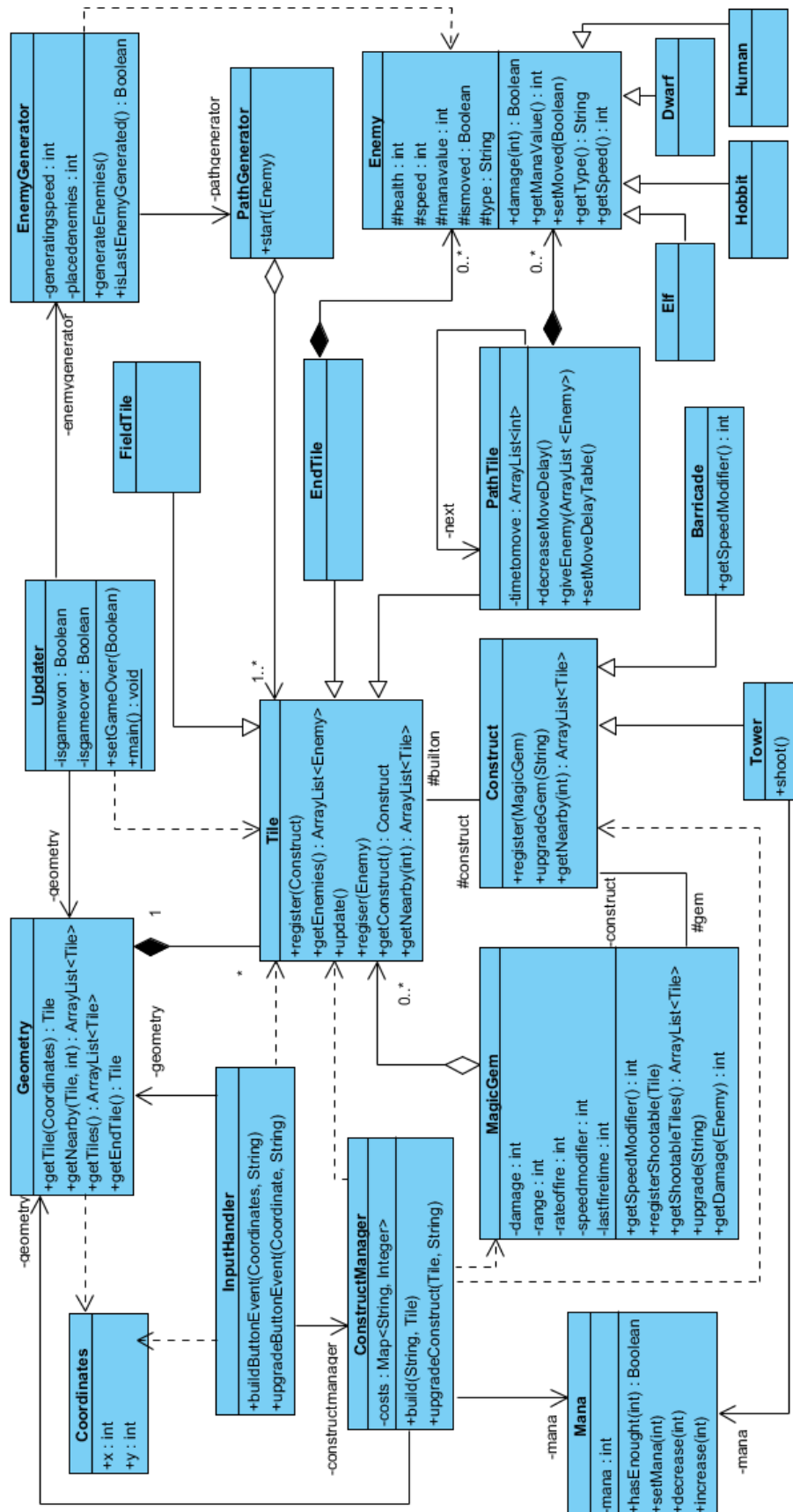
4.1.18 Tower

Egy Construct, amit FieldTile-ra lehet építeni, ha ott nincs már torony. Van saját hatótávolsága, lövési gyorsasága, és sebzése. Bele tehető egy varázskő (MagicGem). Képes löni. Ismeri a Tile-t ami őt tárolja.

4.1.19 Updater

Felelős az időszimulációért. Tárolja az összes épületet és ellenséget, melyeken időszakosan lövést és léptetést hív. Lövés után megnézi, hogy meghalt-e az ellenség, ha igen kitörli, illetve ha az az utolsó ellenség volt, véget vet a játéknak.

4.2 Statikus struktúra diagramok



4.3 Osztályok leírása

4.3.1 Barricade

- **Felelősség**
 - adott értékkel lassítja a csempéjén áthaladó ellenségeket
 - varázskövet lehet beletenni

- **Ősosztályok**

Object ->Construct

- **Interfészek**

nincs

- **Attribútumok**

- **int speedModifier:** a lassítás mértéke

- **Metódusok**

- **int getSpeedModifier():** visszatér a lassítás mértékével

4.3.2 Construct

- **Felelősség**

- ezzel az absztrakt osztállyal lehetséges a Tower-ök és Barricade-ek egy tömbben tárolása.

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **Tile builtOn:** az a csempe, amin a Construct tartózkodik
- **MagicGem gem:** a constructban található varázskő
- **String type:** a construct típusa, például torony vagy akadály

- **Metódusok**

- **String getType():** az épület típusával tér vissza
- **void setMagicGem(MagicGem magicgem):** beállítja a gem attribútumot a kapott értékre

4.3.3 ConstructManager

- **Felelősség**

- létrehozza a különböző épületeket
- létrehoz varázsköveket, melyeket bele tesz épületekbe
- megkérdezi, hogy van-e elég mana az építéshez/fejlesztéshez, és levonja a varázserőt az építés/fejlesztés után

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

String[] types: egyes épület és varázskő típusokat tárolja

int[] costs: egyes épület és varázskő típusokhoz tartozó árakat tárolja

Mana mana: referencia a varázserőre

- **Metódusok**

- **void build(String type, Tile location):** type alapján létrehozza egy épület típus példányát. Megkérdezi a Mana-t, hogy van-e elég az építéshez, ha igen levonja, és létrehozza a Constructot. Ezek után beregisztrálja a kapott Tile-ra.
- **void upgrade(String type, Construct location):** type alapján tesz az adott tile-on lévő constructba varázskövet. Megkérdezi a Mana-t, hogy van-e elég, ha igen levonja, és utána hozza létre a varázskövet. Beállítja a type-nak megfelelő attribútumát a constructnak.

4.3.4 Dwarf

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

String type: az osztály típusát jelzi tehát az hogy, dwarf

- **Metódusok**

csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül

4.3.5 Elf

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

String type: az osztály típusát jelzi tehát az, hogy elf

- **Metódusok**

csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül

4.3.6 EndTile

- **Felelősség**

- ha rálép valaki veszítünk
- nem mehetnek róla tovább ellenségek

- **Ősosztályok**

Object->Tile

- **Interfészek**

nincs

- **Attribútumok**

Enemy[] enemies: rajta tartózkodó ellenségeket tárolja

- **Metódusok**

csak öröklött metódusok, melyeket nem definiál felül

4.3.7 Enemy

- **Felelősség**

- ezzel az absztrakt osztállyal lehetséges a különböző ellenségtípusok egy tömbben tárolása.
- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tudja hogy lépet-e már

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **int health:** az ellenség életerejét jelenti
- **int speed:** a lépes gyakoriságát mutatja
- **int manaValue:** az a mana értéket jelenti melyet a megölésért kap a játékos
- **int moveDelay:** megmondja mikor léphet az ellenség legközelebb
- **String type:** az ellenség típusát adja meg mint például hobbit vagy dwarf
- **Tile currentTile:** a csempe melyen az ellenség jelenleg tartózkodik

- **Metódusok**

- **void damage(int value):** a value értékével csökkenti az életerejét
- **void decreaseMoveDelay():** csökkenti a moveDelay attribútum értékét egyel
- **int getHealth():** az ellenség életerejével tér vissza
- **int getManaValue():** visszaadja, hogy mennyi manát ér a megölése
- **Tile getTile():** visszaadja a csempét melyen az ellenség jelenleg tartózkodik
- **String getType():** az ellenség típusával tér vissza
- **boolean move():** lépteti az ellenséget a currentTile-ről a currentTile nextTile-jára. Megnézi, hogy a végzet hegyére léptünk-e, ha igen akkor true-val tér vissza. Lépés után meghívja a setMoveDelay-t az esetleges akadálytól kapott lassítás mértékével
- **void setMoveDelay(int modifier):** beállítja a moveDelay-t a sebesség és a kapott modifier összegére

4.3.8 EnemyGenerator

- **Felelősség**

- létrehozza az ellenségeket
- gondoskodik az idővel egyre nagyobb ellenséghordák biztosításáról
- tudja, hogy létre hozta-e az utolsó ellenséget

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **int generatingSpeed:** a jelenlegi gyakorisága egy ellenség lerakásának
- **int placedEnemies:** a lerakott ellenségek száma. Ha elér egy bizonyos értéket kinullázzuk, és dekrementáljuk a generatingSpeed-et, így biztosítva az egyre nagyobb ellenségsűrűséget

- **Metódusok**

- **void generateEnemies():** ez a metódus hoz létre egy véletlenszerű fajta ellenséget, majd átadja a PathGenerator-nak, ami lerakja az egyik útvonalra
- **boolean isLastEnemyGenerated():** ha elértük a végleges generatingSpeed-et és az utolsó ellenséget is leraktuk, igazgal tér vissza. Egyébként hamis.

4.3.9 FieldTile

- **Felelősség**

- el lehet rá helyezni tornyokat

- **Ősosztályok**

Object->Tile

- **Interfészek**

nincs

- **Attribútumok**

- **Construct construct:** a Tile-ra épített épületre mutató referencia.

- **Metódusok**

- **addConstruct(Construct construct):** a kapott épületet letárolja

4.3.10 Geometry

- **Felelősség**

- a csempék elrendezése
- meg tudja mondani, hogy adott csempétől adott sugarú körön belül melyik csempék vannak

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **Tile[] tiles:** az összes csempét tároló attribútum

- **Metódusok**

- **Tile[] getNearby(Tile tile, int range):** egy csempe adott sugarú körén belüli csempéit adja vissza
- **Tile[] getTiles():** a tiles attribútumát adja vissza

4.3.11 Hobbít

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

String type: az osztály típusát jelzi tehát az, hogy hobbít

- **Metódusok**

- *csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül*

4.3.12 Human

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

String type: az osztály típusát jelzi tehát az, hogy human

- **Metódusok**

- *csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül*

4.3.13 MagicGem

- **Felelősség**

- toronyba lehet elhelyezni, egyszerre csak egy darab lehet. Elhelyezéskor növelheti a tornyának valamely attribútumát
- tudja, milyen ellenség ellen mennyi plusz sebzést ad

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **String type:** jelenlegi varázserő mennyiség

- **Metódusok**

- **int getDamageBonus(String type):** a paraméter és a saját type attribútuma alapján visszaadja, hogy mennyi plusz sebzést ad az adott ellenségtípusra

4.3.14 Mana

- **Felelősség**

- meg tudja mondani, hogy van-e elég varázserő
- változtatni tudja a varázserő mennyiségét

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **int mana:** jelenlegi varázserő mennyiség

- **Metódusok**
 - **boolean hasEnough(int amount):** megadja, hogy van-e paraméterként kapott mennyiségű varázserő
 - **void decrease(int value):** kivonja a jelenlegi varázserőből a kapott értéket
 - **void increase(int value):** hozzáadja a jelenlegi varázserőhöz a kapott értéket

4.3.15 PathGenerator

- **Felelősség**
 - létre tudja hozni az útvonalakat, amin később ellenségek mehetnek végig
 - rá tud helyezni egy ellenséget egy útvonalra
- **Ősosztályok**
Object
- **Interfészek**
nincs
- **Attribútumok**
 - **Tile[] pathStarts:** útvonalak kezdőcsempéinek tömbje. Ezekre teheti le az ellenségeket
- **Metódusok**
 - **void start(Enemy enemy):** leteszi a kapott ellenséget valamelyik kezdőcsempére.

4.3.16 PathTile

- **Felelősség**
 - tartózkodhat rajta ellenség
 - el lehet rá helyezni akadályokat
- **Ősosztályok**
Object->Tile
- **Interfészek**
nincs
- **Attribútumok**
 - **PathTile nextTile:** az útvonalon következő csempe referenciája, ahova az ellenségek lépnek
 - **Enemy[] enemies:** a csempén tartózkodó ellenségeket tárolja
- **Metódusok**
 - **Enemy getEnemy():** visszaad egy, az ezen a csempén lévő ellenséget

- **Tile getNextTile():** visszaadja a következő útcsempét
- **void removeEnemy():** visszaad egy, az ezen a csempén lévő ellenséget

4.3.17 Tile

- **Felelősség**
 - ezzel az absztrakt osztállyal lehetséges az EndTile-ok, FieldTile-ok és PathTile-ok egy tömbben tárolása

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **Construct construct:** a csempén lévő épület referenciája
- **Geometry geometry:** a Tile-t tároló Geometry-re mutató referencia.

- **Metódusok**

- **void addConstruct(Construct construct):** a kapott épületet letárolja, ha lehet rajta
- **void addEnemy(Enemy enemy):** a kapott ellenséget felveszi az ellenségek közé, ha lehet rajta
- **Construct getConstruct():** visszaadja a csempén lévő épületet
- **Geometry getGeometry():** visszaadja a csempét tároló Geometry-t
- **String getType():** visszaadja a csempe típusát

4.3.18 Tower

- **Felelősség**

- tud a hatáskörén belüli csempéken tartózkodó ellenségekre lőni
- lehet fejleszteni különböző varázskövekkel

- **Ősosztályok**

Object->Construct

- **Interfészek**

nincs

- **Attribútumok**

- **int damage:** a torony sebzése
- **int fireRate:** tüzelési sebesség, azaz mennyit lő egy torony egyszerre
- **int range:** a torony hatótávolsága

- **Metódusok**

- **Enemy shoot():** Lekérdezi a hatótávon belüli csempéket, majd kér valamelyikről egy ellenséget. Ha van varázskő a toronyban, megkérdezi milyen plusz sebzést biztosít a varázskő, és a saját értékéhez hozzáadva belesebzti azt az ellenségbe, annyiszor, amennyi a fireRate attribútumának az értéke. Visszatér azzal az ellenséggel, akit meglőtt.
- **void setDamage(int damage):** beállítja a damage attribútumot
- **void setFireRate(int fireRate):** beállítja a fireRate attribútumot
- **void setRange(int range):** beállítja a range attribútumot

4.3.19 Updater

- **Felelősség**

- időszimuláció
- periodikus tevékenységek, például mozgás, lövés, ellenség generálás indítása

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

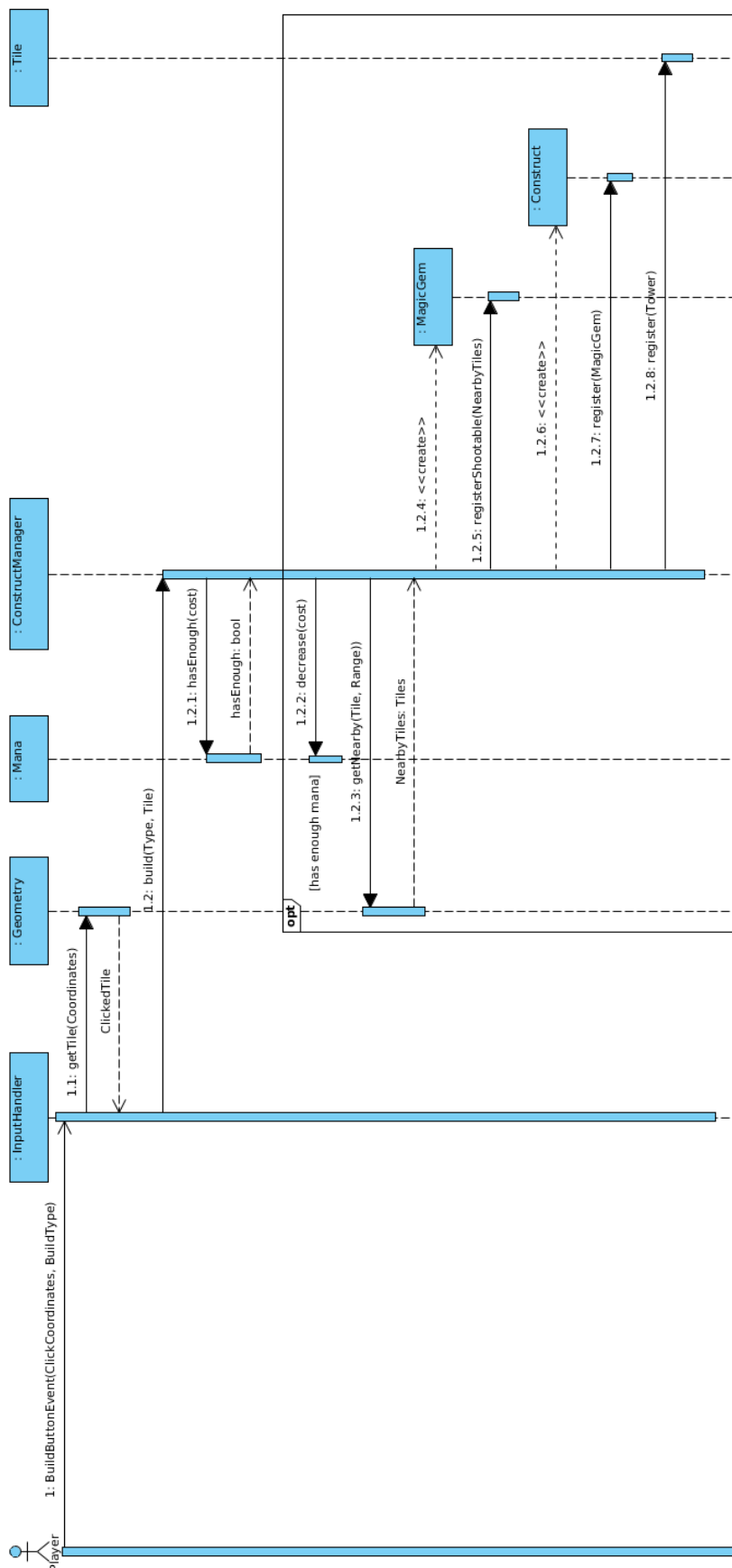
- **EnemyGenerator enemyGenerator:** az ellenségeket generáló osztályt tartalmazza
- **Geometry geometry:** egy referencia a geometry osztályra
- **Enemy[] enemies:** a játékban létező összes ellenséget tárolja
- **Construct[] constructs:** a játékban létező összes épületet tárolja

- **Metódusok**

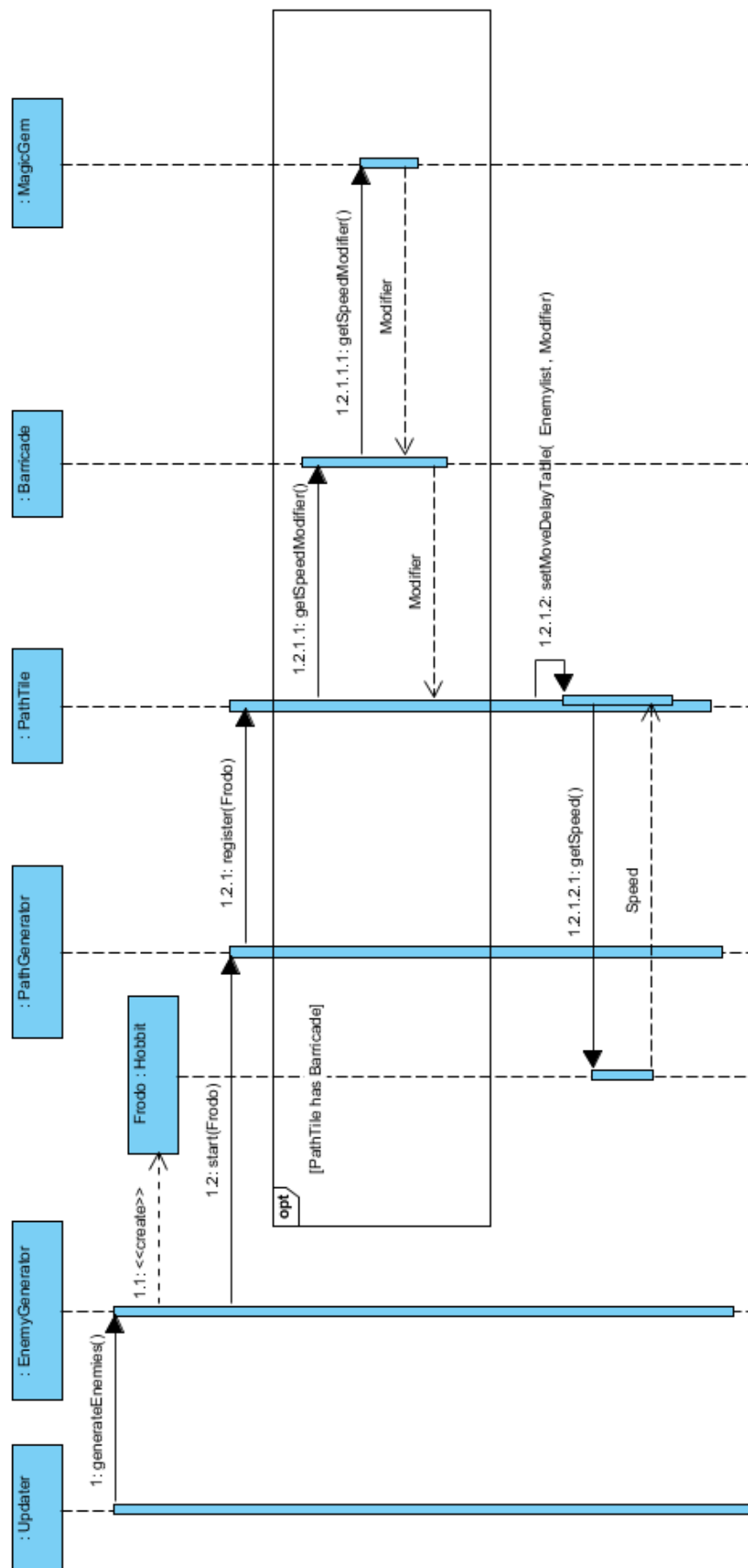
- **addEnemy (Enemy enemy):** hozzáad egy ellenséget az enemies attribútumhoz
- **void gameOver(boolean outcome):** véget vet a játéknak győzelemmel, ha igaz a paraméter, vereséggel ha hamis
- **static void main():** main metódus. Ebben jönnek létre az első objektumok, majd innen lépünk be a játék főciklusába

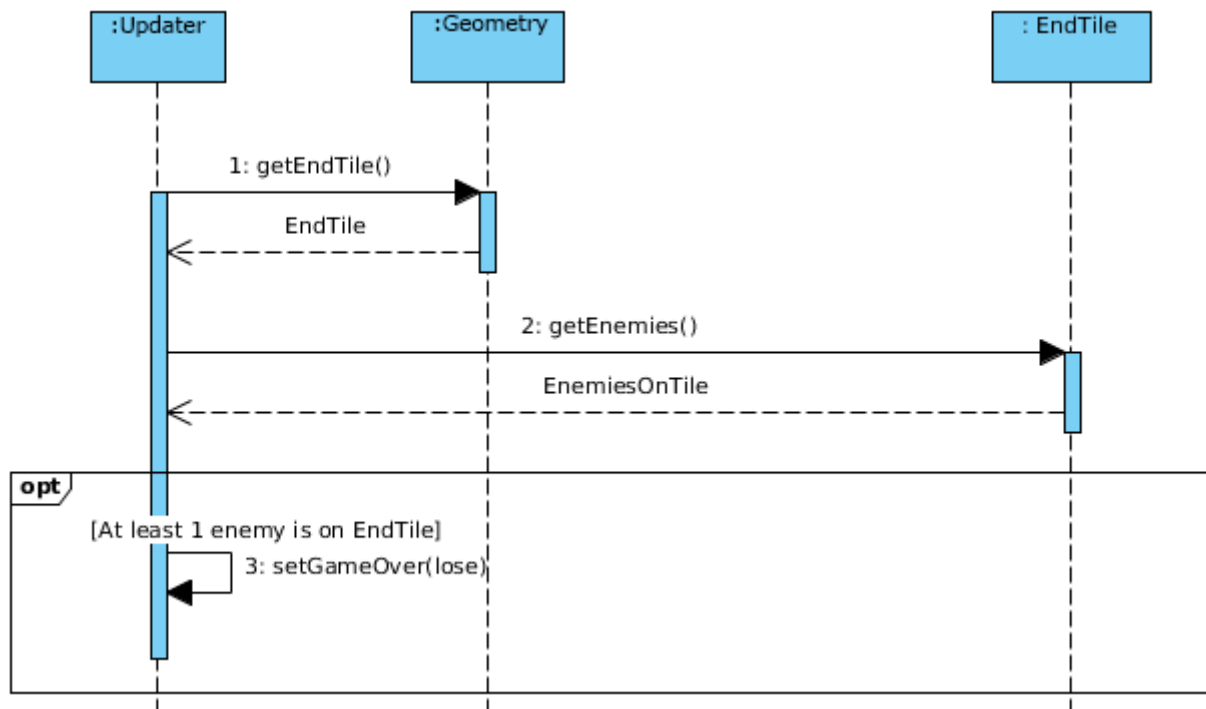
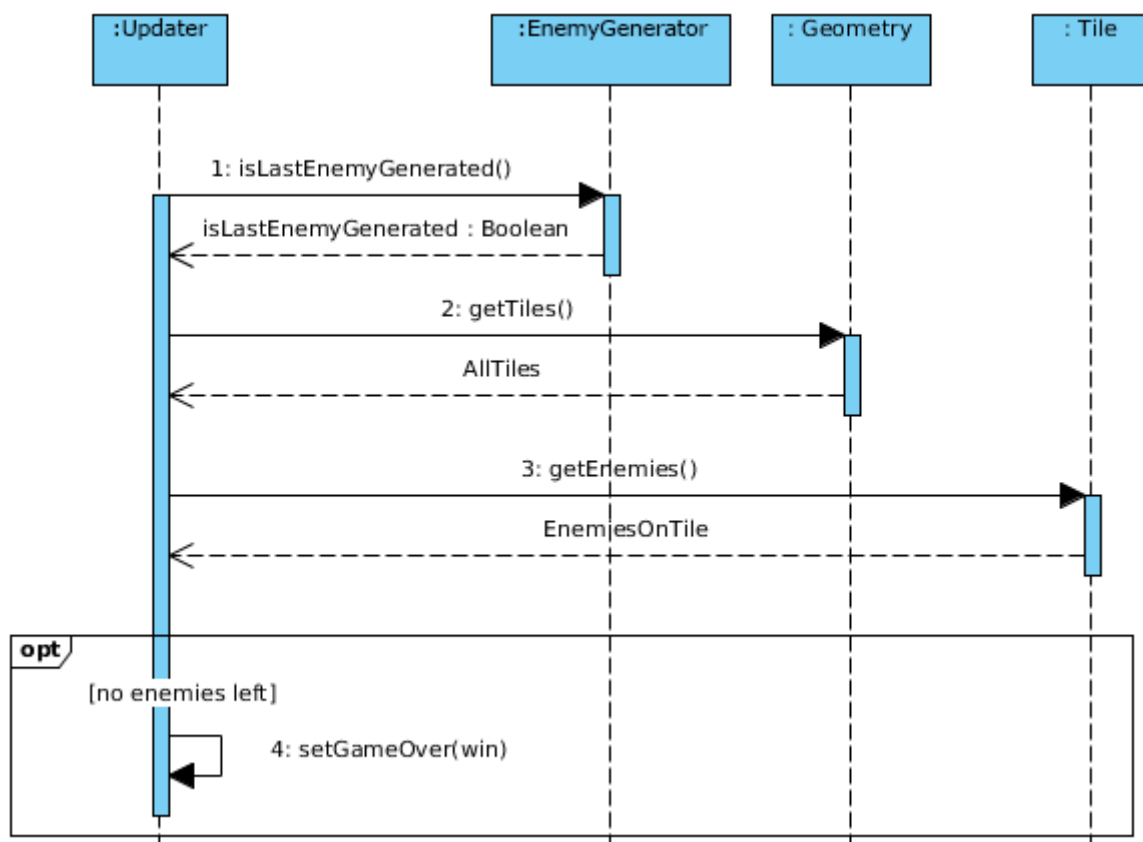
4.4 Szekvencia diagramok

Építés

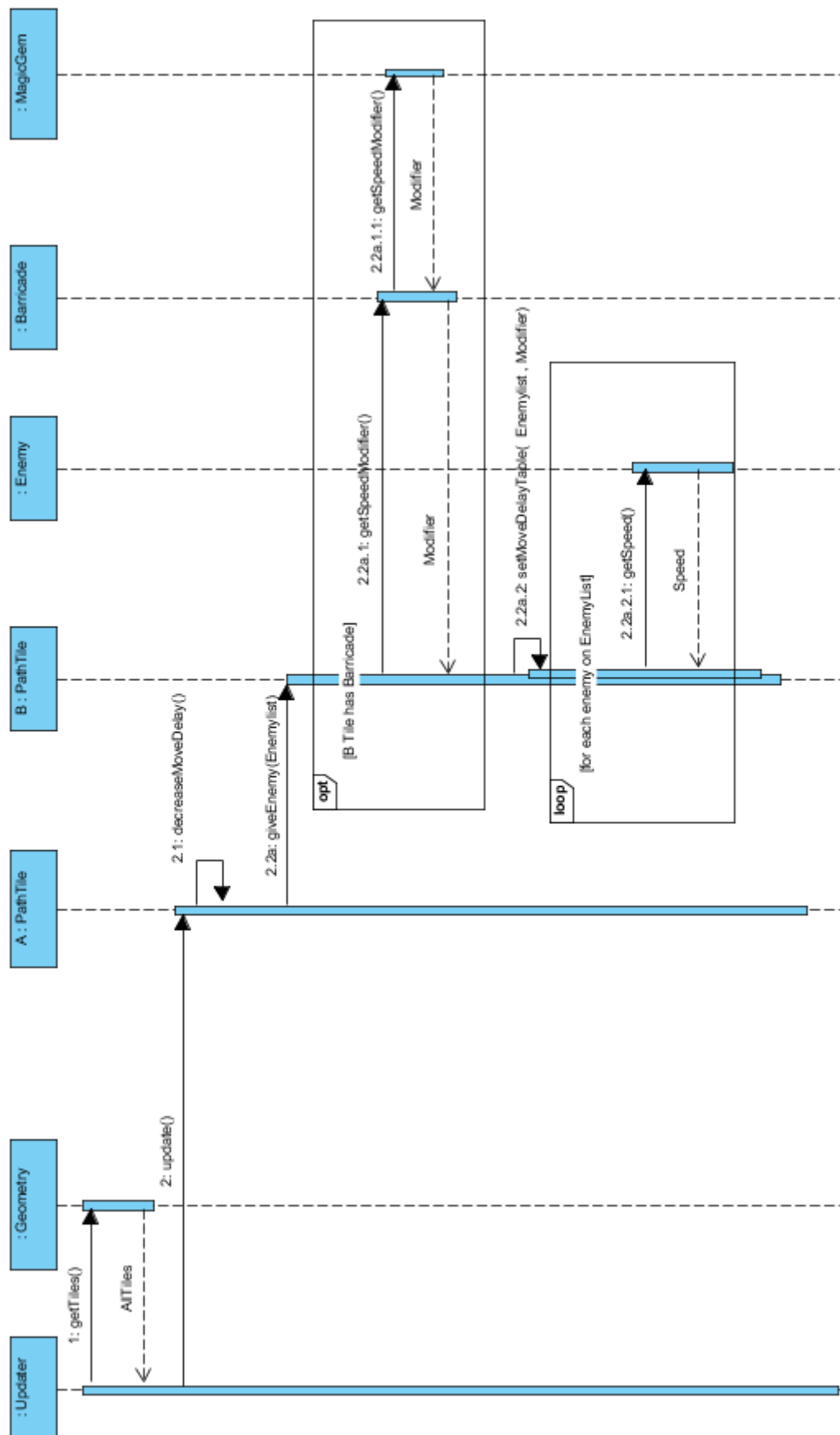


Enemy Generálás

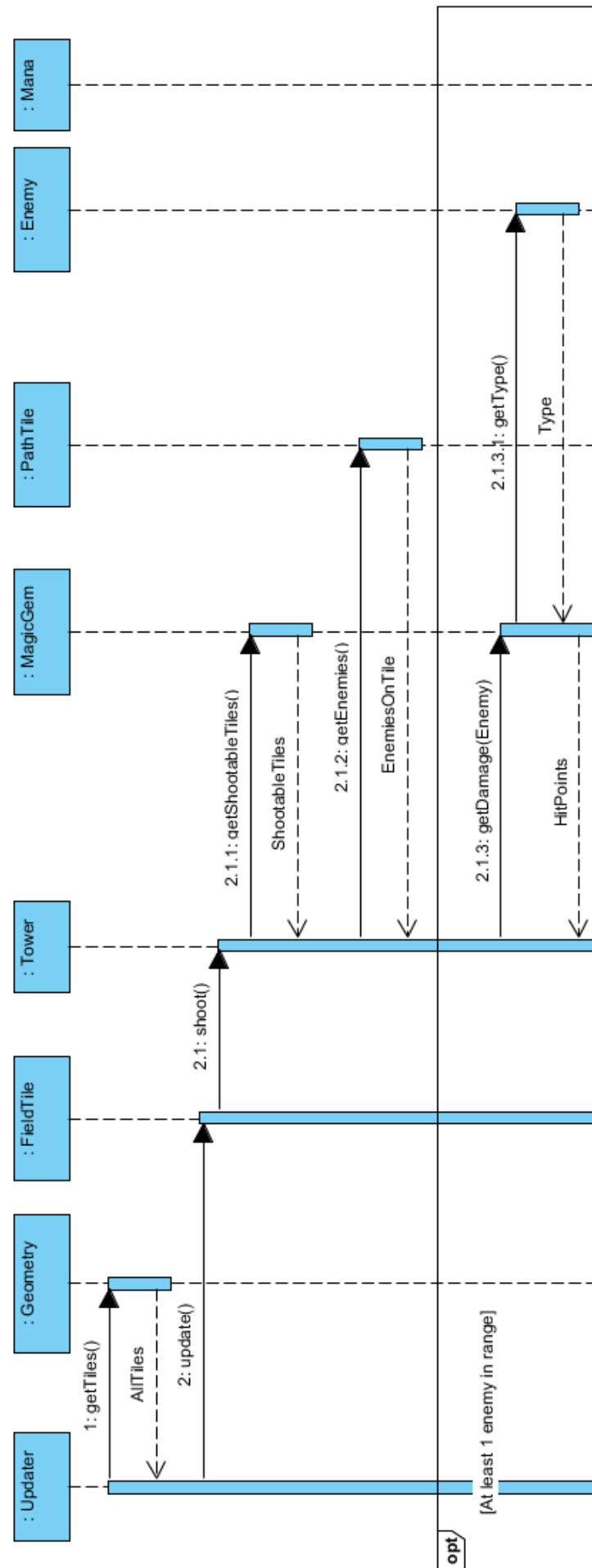


Vesztés**Nyerés**

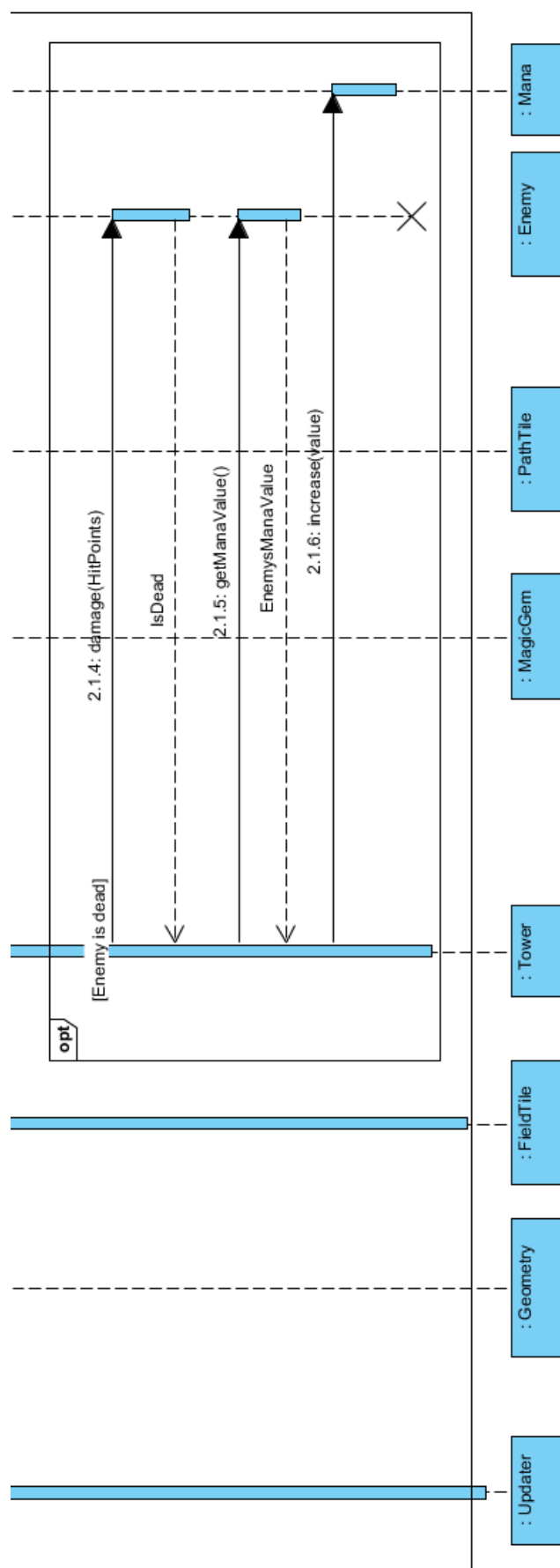
Mozgás



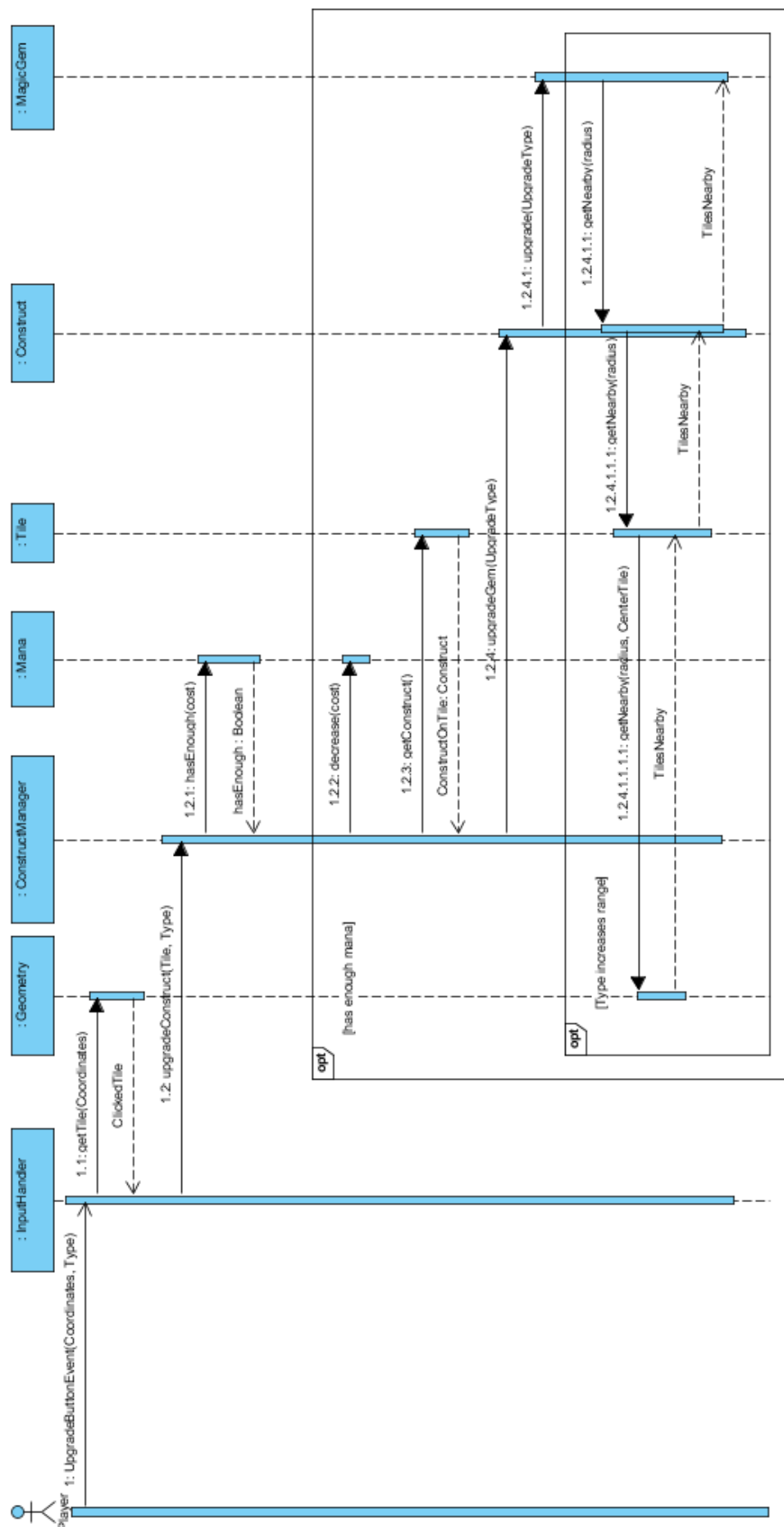
Lövés 1



Lövés 2



Fejlesztés (hatótáv)



4.5 State-chartok

Nincs állapotfüggő viselkedés a programban.

5. Analízis modell kidolgozása 3

5.1 Objektum katalógus

5.1.1 Barricade

Olyan Construct, amit csak PathTile-re lehet építeni, ahol lelassítja az ott áthaladó Enemy-eket, azaz később léphetnek tovább. Benne található egy varázskő (MagicGem), ami a lassítás mértékét növeli. Ezt a követ fejleszteni is lehet. Ismeri a Tile-t ami őt tárolja.

5.1.2 Construct

Ezzel az absztrakt osztállyal lehetséges a Tower-ök és Barricade-ek egy tömbben tárolása.

5.1.3 ConstructManager

Ez az osztály felelős a Construct-ok létrehozásáért és fejlesztéséért. Meg tudja nézni, hogy az építendő Construct-hoz vagy MagicGem-hez van-e elég Mana, ha van létre tudja hozni, és le tudja vonni a Mana-t.

5.1.4 Dwarf

Az Enemy leszármazottja, tárolja a saját életerejét, a halálakor kapható varázserő értékét és a sebességét. Tud lépni, és le tudja vonni az életerejét. Egyedi attribútumai vannak a többi ellenség típushoz képest.

5.1.5 Elf

Az Enemy leszármazottja, tárolja a saját életerejét, a halálakor kapható varázserő értékét és a sebességét. Tud lépni, és le tudja vonni az életerejét. Egyedi attribútumai vannak a többi ellenség típushoz képest.

5.1.6 EndTile

Olyan Tile, amin ha ellenség tartózkodik, a játék vereséggel ér véget.

5.1.7 Enemy

Ezzel az absztrakt osztállyal lehetséges a különböző ellenségtípusok egy tömbben tárolása.

5.1.8 EnemyGenerator

Ez az osztály felelős az ellenségek létrehozásáért. Ismeri a PathGeneratort, ami segítségével a létrehozott Enemy-eket le tudja rakni az útvonalakra. Tudja, hogy mikor kell ellenséget létrehozni, illetve meg tudja mondani, hogy létrehozta-e az utolsó ellenséget.

5.1.9 Geometry

Felelős a csempék tárolásáért és elrendezéséért. Meg tudja adni, hogy egy pont adott környezetében melyik cellák vannak. Vissza tudja adni az összes csempét.

5.1.10 Hobbít

Az Enemy leszármazottja, tárolja a saját életerejét, a halálakor kapható varázserő értékét és a sebességét. Tud lépni, és le tudja vonni az életerejét. Egyedi attribútumai vannak a többi ellenség típushoz képest.

5.1.11 Human

Az Enemy leszármazottja, tárolja a saját életerejét, a halálakor kapható varázserő értékét és a sebességét. Tud lépni, és le tudja vonni az életerejét. Egyedi attribútumai vannak a többi ellenség típushoz képest.

5.1.12 FieldTile

Egy csempe típus, amire egy tornyot lehet építeni. Tárolja a rajta lévő tornyot.

5.1.13 Mana

A játék nyersanyaga, tárolja az elérhető varázserő mennyiségét. Meg tudja mondani, hogy adott mennyiség rendelkezésre áll-e, illetve le tudja csökkenteni és meg tudja növelni adott értékekkel.

5.1.14 MagicGem

Varázskő, melyet egy toronyba lehet tenni varázserőért cserébe. Növelheti a torony hatótávát, tüzelési gyorsaságát és a sebzését a különböző ellenségtípusok ellen. Meg tudja mondani, milyen típusú.

5.1.15 PathGenerator

Legenerálja és tárolja az útvonalakat amiken ellenségek járhatnak végig. Le tudja rakni az ellenségeket az útvonalra.

5.1.16 PathTile

Egy csempe típus, amin ellenségek léphetnek és akadályt lehet rá építeni. Tárolja a rajta lévő akadályt, és a rajta lévő ellenségeket. Ismeri a következő útcsempét amire róla lépni lehet.

5.1.17 Tile

Ezzel az absztrakt osztállyal lehetséges az EndTile-ok, FieldTile-ok és PathTile-ok egy tömbben tárolása. Minden Tile ismeri az őt tároló geometriát és meg tudja mondani milyen típusú (pl. EndTile).

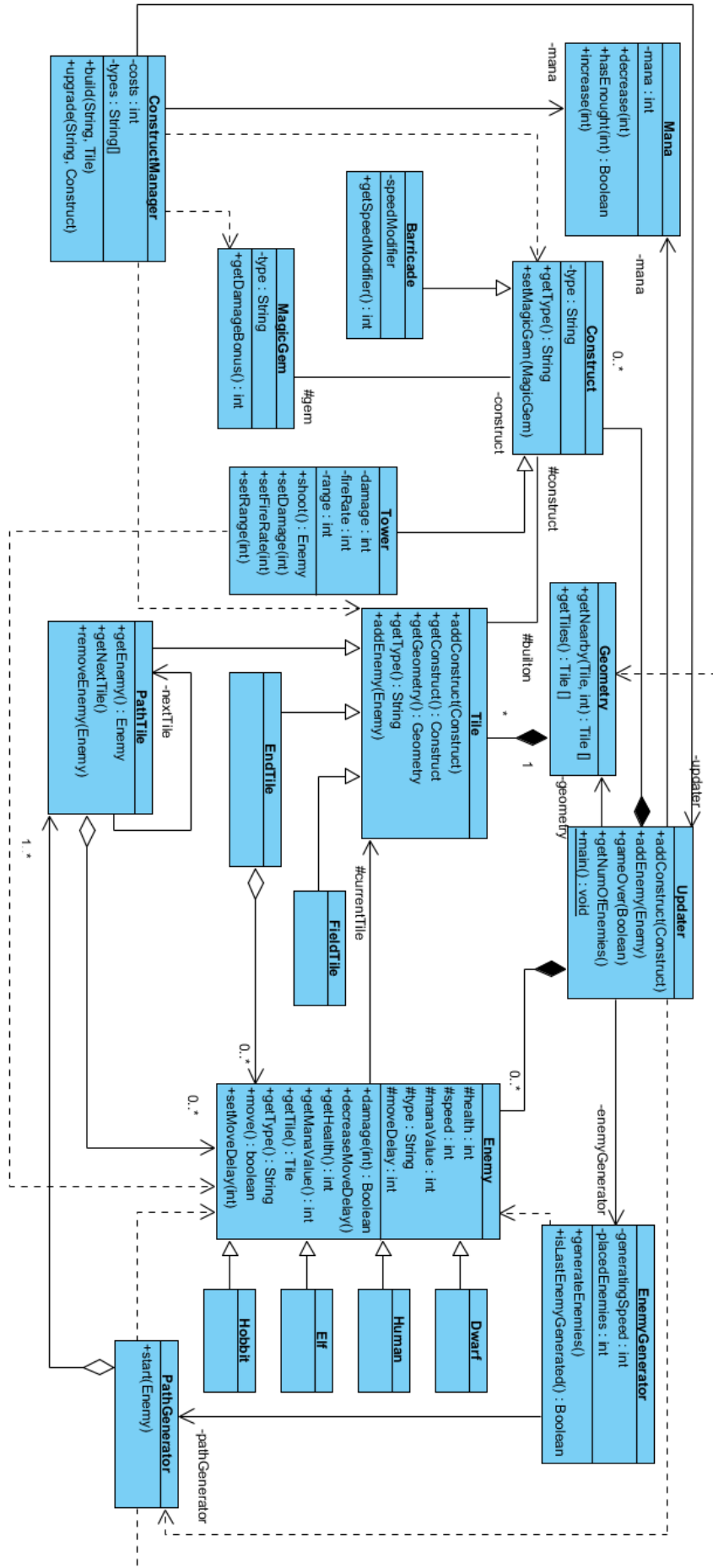
5.1.18 Tower

Egy Construct, amit FieldTile-ra lehet építeni, ha ott nincs már torony. Van saját hatótávolsága, lövési gyorsasága, és sebzése. Bele tehető egy varázskő (MagicGem). Képes löni. Ismeri a Tile-t ami őt tárolja.

5.1.19 Updater

Felelős az időszimulációért. Tárolja az összes épületet és ellenséget, melyeken időszakosan lövést és léptetést hív. Lövés után megnézi, hogy meghalt-e az ellenség, ha igen kitörli, illetve ha az az utolsó ellenség volt, véget vet a játéknak.

2014-05-14



5.3 Osztályok leírása

5.3.1 Barricade

- **Felelősség**
 - adott értékkel lassítja a csempéjén áthaladó ellenségeket
 - varázskövet lehet beletenni

- **Ősosztályok**

Object ->Construct

- **Interfészek**

nincs

- **Attribútumok**

- **int speedModifier:** a lassítás mértéke

- **Metódusok**

- **int getSpeedModifier():** visszatér a lassítás mértékével

5.3.2 Construct

- **Felelősség**

- ezzel az absztrakt osztállyal lehetséges a Tower-ök és Barricade-ek egy tömbben tárolása.

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **Tile builtOn:** az a csempe, amin a Construct tartózkodik
- **MagicGem gem:** a constructban található varázskő
- **String type:** a construct típusa, például torony vagy akadály

- **Metódusok**

- **String getType():** az épület típusával tér vissza
- **void setMagicGem(MagicGem magicgem):** beállítja a gem attribútumot a kapott értékre

5.3.3 ConstructManager

- **Felelősség**

- létrehozza a különböző épületeket
- létrehoz varázsköveket, melyeket bele tesz épületekbe
- megkérdezi, hogy van-e elég mana az építéshez/fejlesztéshez, és levonja a varázserőt az építés/fejlesztés után

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

int[] costs: egyes épület és varázskő típusokhoz tartozó árakat tárolja

Mana mana: referencia a varázserőre

String[] types: egyes épület és varázskő típusokat tárolja

Updater[] updater: referencia az időkezelőre

- **Metódusok**

- **void build(String type, Tile location):** type alapján létrehozza egy épület típus példányát. Megkérdezi a Mana-t, hogy van-e elég az építéshez, ha igen levonja, és létrehozza a Constructot. Ezek után beregisztrálja a kapott Tile-ra és hozzáadja az updater épületeihez
- **void upgrade(String type, Construct location):** type alapján tesz az adott tile-on lévő constructba varázskövet. Megkérdezi a Mana-t, hogy van-e elég, ha igen levonja, és utána hozza létre a varázskövet. Beállítja a type-nak megfelelő attribútumát a constructnak.

5.3.4 Dwarf

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

String type: az osztály típusát jelzi tehát az hogy, dwarf

- **Metódusok**

csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül

5.3.5 Elf

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

String type: az osztály típusát jelzi tehát az, hogy elf

- **Metódusok**

csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül

5.3.6 EndTile

- **Felelősség**

- ha rálép valaki veszítünk
- nem mehetnek róla tovább ellenségek

- **Ősosztályok**

Object->Tile

- **Interfészek**

nincs

- **Attribútumok**

Enemy[] enemies: rajta tartózkodó ellenségeket tárolja

- **Metódusok**

csak öröklött metódusa van

5.3.7 Enemy

- **Felelősség**

- ezzel az absztrakt osztállyal lehetséges a különböző ellenségtípusok egy tömbben tárolása.
- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tudja hogy lépet-e már

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **int health:** az ellenség életerejét jelenti
- **int speed:** a lépes gyakoriságát mutatja
- **int manaValue:** az a mana értéket jelenti melyet a megölésért kap a játékos
- **int moveDelay:** megmondja mikor léphet az ellenség legközelebb
- **String type:** az ellenség típusát adja meg mint például hobbit vagy dwarf
- **Tile currentTile:** a csempe melyen az ellenség jelenleg tartózkodik

- **Metódusok**

- **void damage(int value):** a value értékével csökkenti az életerejét
- **void decreaseMoveDelay():** csökkenti a moveDelay attribútum értékét egyel
- **int getHealth():** az ellenség életerejével tér vissza
- **int getManaValue():** visszaadja, hogy mennyi manát ér a megölése
- **Tile getTile():** visszaadja a csempét melyen az ellenség jelenleg tartózkodik
- **String getType():** az ellenség típusával tér vissza
- **boolean move():** lépteti az ellenséget a currentTile-ről a currentTile nextTile-jára. Megnézi, hogy a végzet hegyére léptünk-e, ha igen akkor true-val tér vissza. Lépés után meghívja a setMoveDelay-t az esetleges akadálytól kapott lassítás mértékével
- **void setMoveDelay(int modifier):** beállítja a moveDelay-t a sebesség és a kapott modifier összegére

5.3.8 EnemyGenerator

- **Felelősség**

- létrehozza az ellenségeket
- gondoskodik az idővel egyre nagyobb ellenséghordák biztosításáról
- tudja, hogy létre hozta-e az utolsó ellenséget

- **Össosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **int generatingSpeed:** a jelenlegi gyakorisága egy ellenség lerakásának
- **int placedEnemies:** a lerakott ellenségek száma. Ha elér egy bizonyos értéket kinullázzuk, és dekrementáljuk a generatingSpeed-et, így biztosítva az egyre nagyobb ellenségsűrűséget

- **Metódusok**

- **Enemy generateEnemies():** ez a metódus hoz létre egy véletlenszerű fajta ellenséget, majd átadja a PathGenerator-nak, ami lerakja az egyik útvonalra. Visszatér a leggenerált ellenséggel
- **boolean isLastEnemyGenerated():** ha elértük a végleges generatingSpeed-et és az utolsó ellenséget is leraktuk, igazzal tér vissza. Egyébként hamis.

5.3.9 FieldTile

- **Felelősség**

- el lehet rá helyezni tornyokat

- **Össosztályok**

Object->Tile

- **Interfészek**

nincs

- **Attribútumok**

- **Construct construct:** a Tile-ra épített épületre mutató referencia.

- **Metódusok**

csak örökölt metódusa van

5.3.10 Geometry

- **Felelősség**

- a csempék elrendezése
- meg tudja mondani, hogy adott csempétől adott sugarú körön belül melyik csempék vannak

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **Tile[] tiles:** az összes csempét tároló attribútum

- **Metódusok**

- **Tile[] getNearby(Tile tile, int range):** egy csempe adott sugarú körén belüli csempéit adja vissza
- **Tile[] getTiles():** a tiles attribútumát adja vissza

5.3.11 Hobbit

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

String type: az osztály típusát jelzi tehát az, hogy hobbit

- **Metódusok**

- *csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül*

5.3.12 Human

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

String type: az osztály típusát jelzi tehát az, hogy human

- **Metódusok**

- *csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül*

5.3.13 MagicGem

- **Felelősség**

- toronyba lehet elhelyezni, egyszerre csak egy darab lehet. Elhelyezéskor növelheti a tornyának valamely attribútumát
- tudja, milyen ellenség ellen mennyi plusz sebzést ad

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **String type:** jelenlegi varázserő mennyiség

- **Metódusok**

- **int getDamageBonus(String type):** a paraméter és a saját type attribútuma alapján visszaadja, hogy mennyi plusz sebzést ad az adott ellenségtípusra

5.3.14 Mana

- **Felelősség**

- meg tudja mondani, hogy van-e elég varázserő
- változtatni tudja a varázserő mennyiségét

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **int mana:** jelenlegi varázserő mennyiség

- **Metódusok**
 - **boolean hasEnough(int amount):** megadja, hogy van-e paraméterként kapott mennyiségű varázserő
 - **void decrease(int value):** kivonja a jelenlegi varázserőből a kapott értéket
 - **void increase(int value):** hozzáadja a jelenlegi varázserőhöz a kapott értéket

5.3.15 PathGenerator

- **Felelősség**
 - létre tudja hozni az útvonalakat, amin később ellenségek mehetnek végig
 - rá tud helyezni egy ellenséget egy útvonalra
- **Ősosztályok**
Object
- **Interfészek**
nincs
- **Attribútumok**
 - **Tile[] pathStarts:** útvonalak kezdőcsempéinek tömbje. Ezekre teheti le az ellenségeket
- **Metódusok**
 - **void start(Enemy enemy):** leteszi a kapott ellenséget valamelyik kezdőcsempére.

5.3.16 PathTile

- **Felelősség**
 - tartózkodhat rajta ellenség
 - el lehet rá helyezni akadályokat
- **Ősosztályok**
Object->Tile
- **Interfészek**
nincs
- **Attribútumok**
 - **PathTile nextTile:** az útvonalon következő csempe referenciája, ahova az ellenségek lépnek
 - **Enemy[] enemies:** a csempén tartózkodó ellenségeket tárolja
- **Metódusok**
 - **Enemy getEnemy():** visszaad egy, az ezen a csempén lévő ellenséget

- **Tile getNextTile():** visszaadja a következő útcsempét
- **void removeEnemy():** visszaad egy, az ezen a csempén lévő ellenséget

5.3.17 Tile

- **Felelősség**
 - ezzel az absztrakt osztállyal lehetséges az EndTile-ok, FieldTile-ok és PathTile-ok egy tömbben tárolása

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **Construct construct:** a csempén lévő épület referenciája
- **Geometry geometry:** a Tile-t tároló Geometry-re mutató referencia.

- **Metódusok**

- **void addConstruct(Construct construct):** a kapott épületet letárolja, ha lehet rajta
- **void addEnemy(Enemy enemy):** a kapott ellenséget felveszi az ellenségek közé, ha lehet rajta
- **Construct getConstruct():** visszaadja a csempén lévő épületet
- **Geometry getGeometry():** visszaadja a csempét tároló Geometry-t
- **String getType():** visszaadja a csempe típusát

5.3.18 Tower

- **Felelősség**

- tud a hatáskörén belüli csempéken tartózkodó ellenségekre lőni
- lehet fejleszteni különböző varázskövekkel

- **Ősosztályok**

Object->Construct

- **Interfészek**

nincs

- **Attribútumok**

- **int damage:** a torony sebzése
- **int fireRate:** tüzelési sebesség, azaz mennyit lő egy torony egyszerre
- **int range:** a torony hatótávolsága

- **Metódusok**

- **Enemy shoot():** Lekérdezi a hatótávon belüli csempéket, majd kér valamelyikről egy ellenséget. Ha van varázskő a toronyban, megkérdezi milyen plusz sebzést biztosít a varázskő, és a saját értékéhez hozzáadva belesebzti azt az ellenségbe, annyiszor, amennyi a fireRate attribútumának az értéke. Visszatér azzal az ellenséggel, akit meglőtt.
- **void setDamage(int damage):** beállítja a damage attribútumot
- **void setFireRate(int fireRate):** beállítja a fireRate attribútumot
- **void setRange(int range):** beállítja a range attribútumot

5.3.19 Updater

- **Felelősség**

- időszimuláció
- periodikus tevékenységek, például mozgás, lövés, ellenség generálás indítása

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

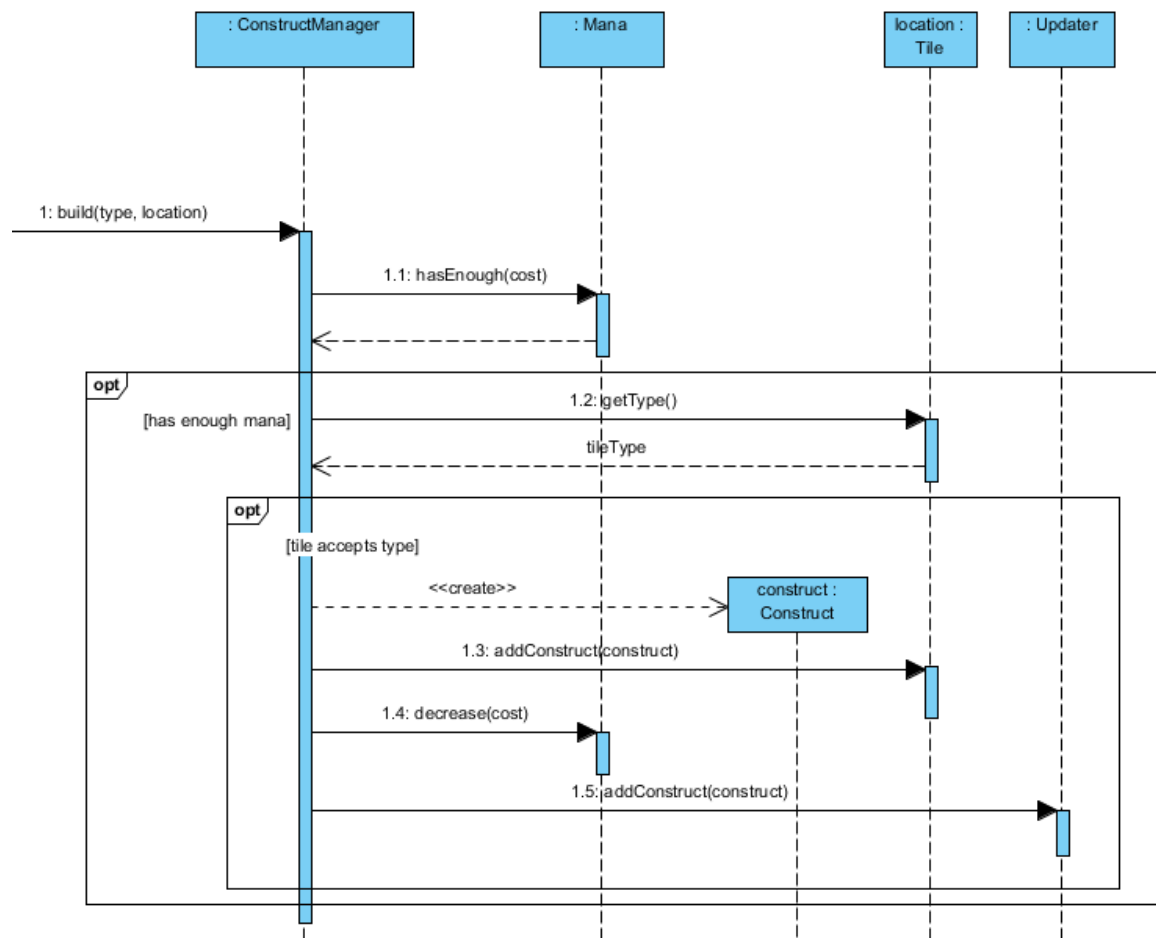
- **EnemyGenerator enemyGenerator:** az ellenségeket generáló osztályt tartalmazza
- **Geometry geometry:** egy referencia a geometry osztályra
- **Enemy[] enemies:** a játékban létező összes ellenséget tárolja
- **Construct[] constructs:** a játékban létező összes épületet tárolja

- **Metódusok**

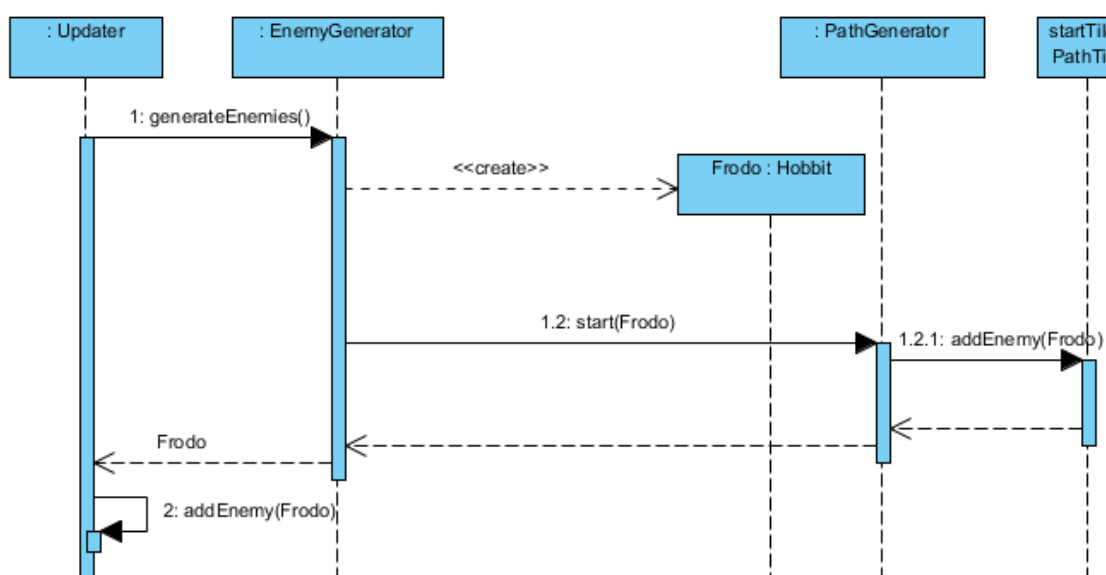
- **addEnemy (Enemy enemy):** hozzáad egy ellenséget az enemies attribútumhoz
- **addConstruct (Construct construct):** hozzáad egy épületet a constructs attribútumhoz
- **void gameOver(boolean outcome):** véget vet a játéknak győzelemmel, ha igaz a paraméter, vereséggel ha hamis
- **int getNumOfEnemies():** megszámolja az ellenségeket, és visszaadja az ellenségek számát
- **static void main():** main metódus. Ebben jönnek létre az első objektumok, majd innen lépünk be a játék főciklusába

5.4 Szekvencia diagramok

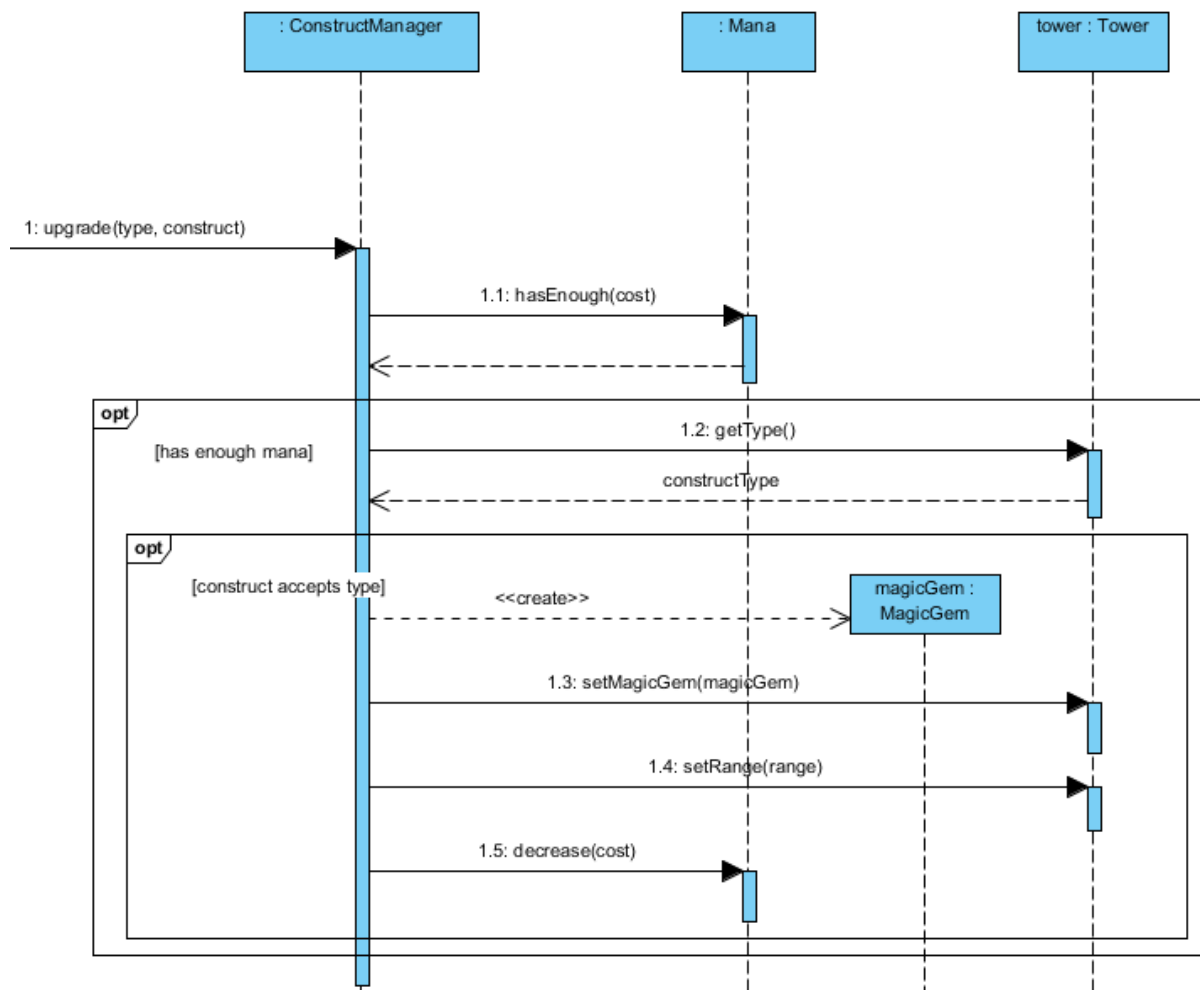
Építés



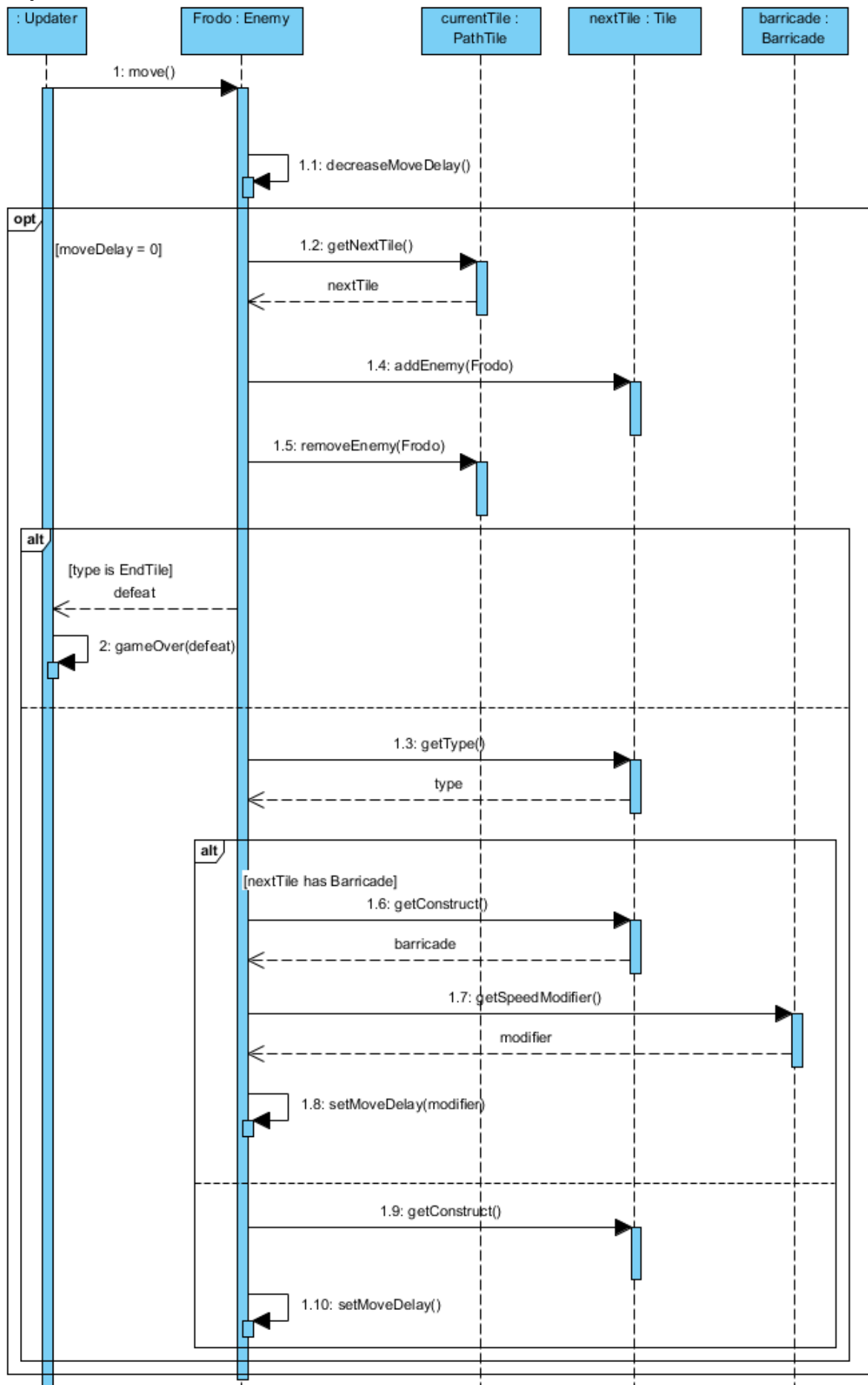
Enemy Generálás



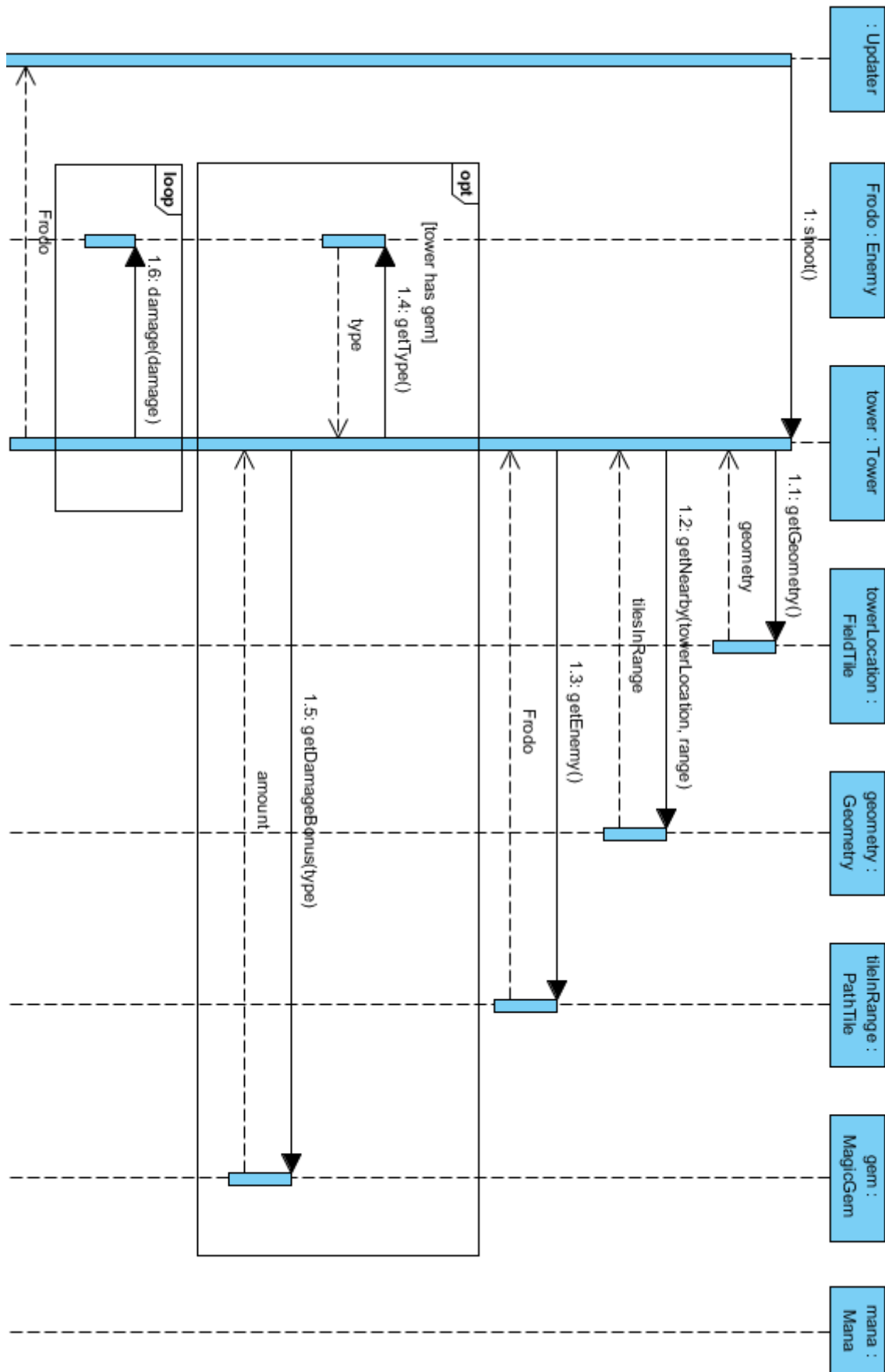
Fejlesztés (hatótáv)



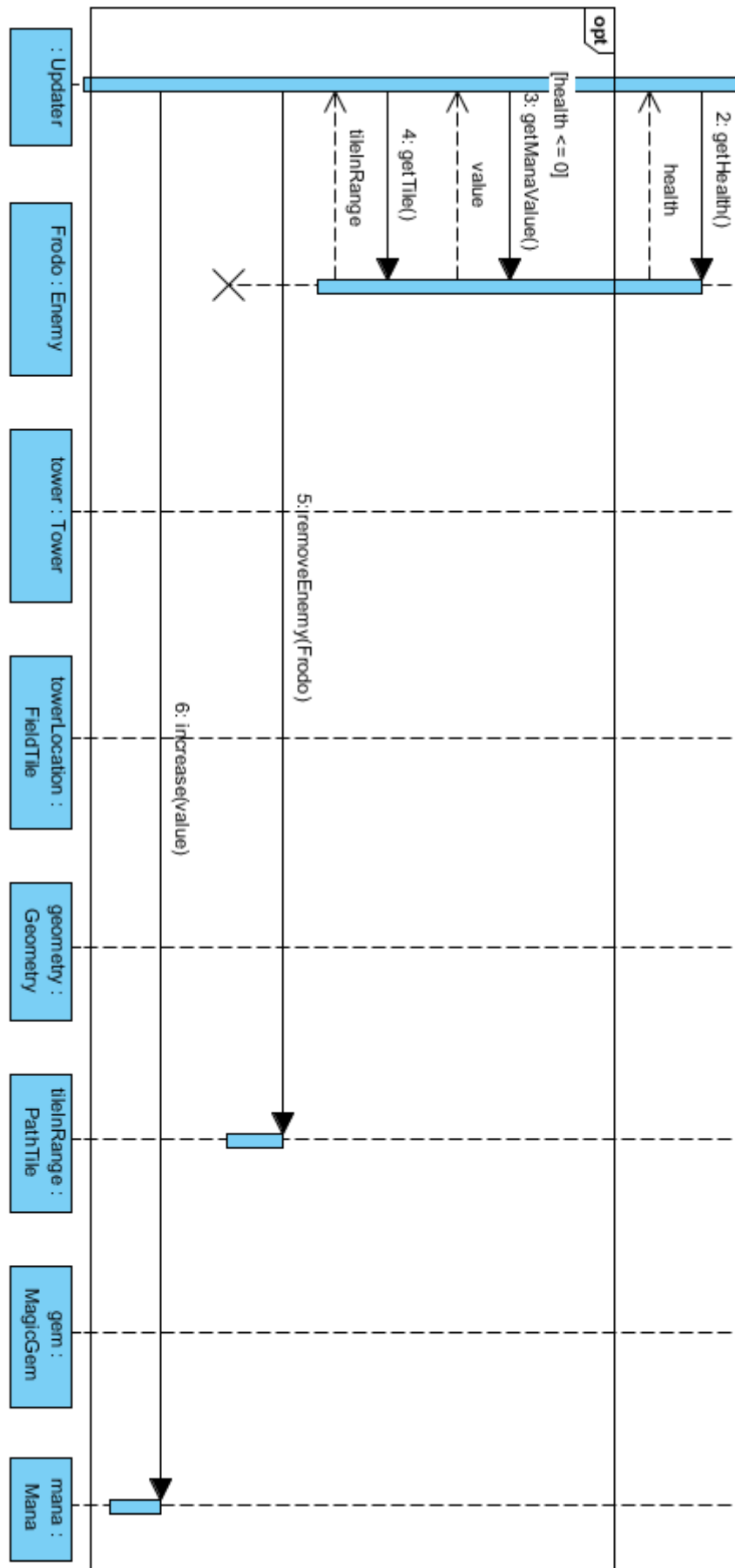
Lépés



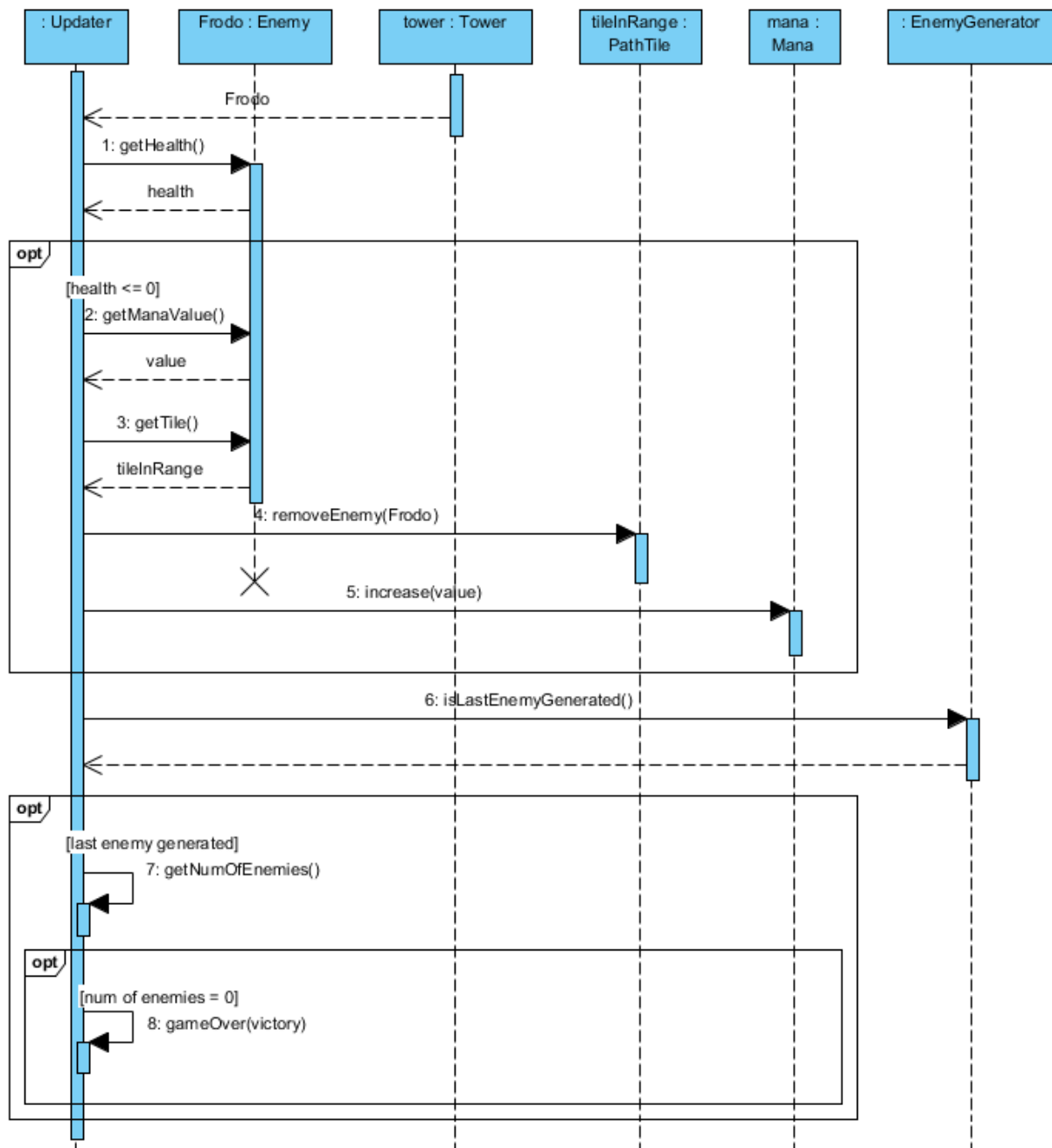
Lövés 1



Lövés 2



Lövés után győzelem



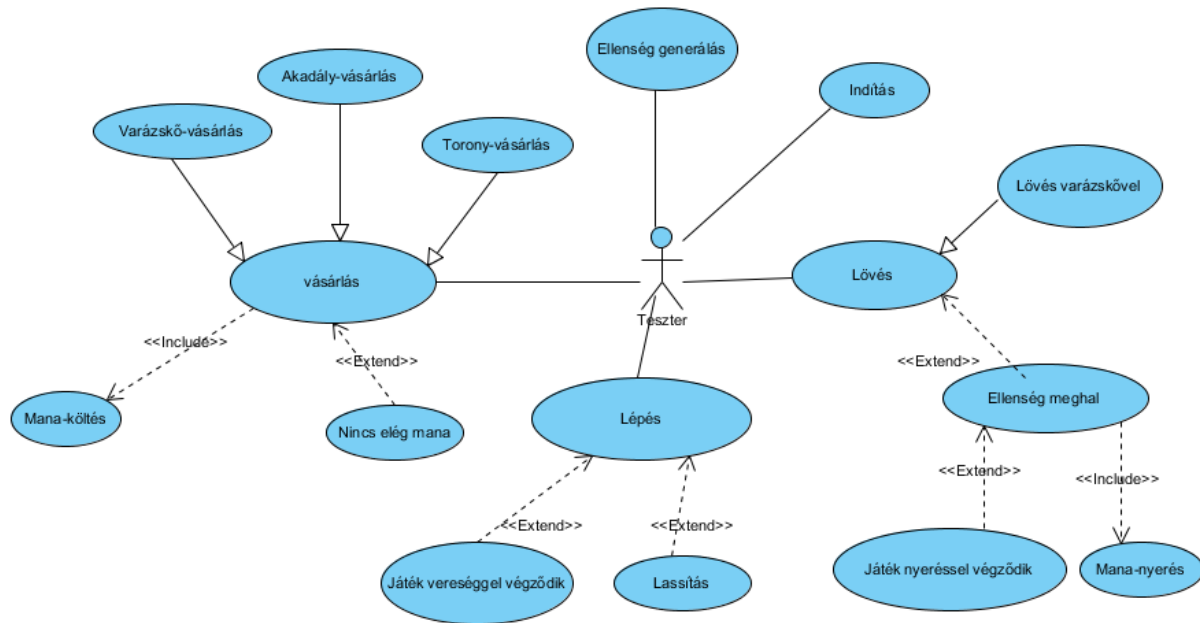
5.5 State-chartok

Nincs állapotfüggő viselkedés a programban.

6. Szkeleton tervezése

6.1 A szkeleton modell valóságos use-case-ei

6.1.1 Use-case diagram



6.1.2 Use-case leírások

Use-case neve	Vásárlás
Rövid leírás	Új épületeket vagy fejlesztéseket kapunk varázserőért.
Aktorok	Teszter
Forgatókönyv	Varázserőért cserébe veszünk valamit egy adott csempére. Tartalmazza a Mana-költés use-case-t.

Use-case neve	Mana-költés
Rövid leírás	Varázserőt von le a játékosról.
Aktorok	Teszter
Forgatókönyv	A „Vásárlás” tartalmazza. A játékos varázssereje csökken az éppen vásárolt dolog értékével.

Use-case neve	Varázskő-vásárlás
Rövid leírás	Varázskő vásárlása épület fejlesztéséhez.
Aktorok	Teszter
Forgatókönyv	A Vásárlás örökölt use-case-e. Egy olyan vásárlás, amikor varázskövet veszünk fejlesztés céljából.

Use-case neve	Akadály-vásárlás
Rövid leírás	Új akadály vásárlása.
Aktorok	Teszter
Forgatókönyv	A Vásárlás örökölt use-case-e. Egy olyan vásárlás, amikor akadályt veszünk egy adott csempére.

Use-case neve	Torony-vásárlás
Rövid leírás	Új torony vásárlása.
Aktorok	Teszter
Forgatókönyv	A Vásárlás örökölt use-case-e. Egy olyan vásárlás, amikor tornyot veszünk egy adott csempére.

Use-case neve	Nincs elég mana
Rövid leírás	Ez történik, ha nincs elég varázserő a kiválasztott vásárláshoz
Aktorok	Teszter
Forgatókönyv	A vásárlás kiterjesztése, ha nincs elég varázserő a vásárlás ellenértékének kifizetéséhez, a vásárlás nem jön létre.

Use-case neve	Ellenség generálás
Rövid leírás	Létrejön egy ellenség.
Aktorok	Teszter
Forgatókönyv	Létrejön egy ellenség, mely aztán a pályára kerül.

Use-case neve	Indítás
Rövid leírás	A program inicializálása.
Aktorok	Teszter
Forgatókönyv	A játék kezdése, létrejön a térkép, a csempék, a játék futásához szükséges példányok.

Use-case neve	Lövés
Rövid leírás	Egy torony ellenséget sebez.
Aktorok	Teszter
Forgatókönyv	A torony hatótávában az egyik ellenség életerőpontjait csökkenti.

Use-case neve	Lövés varázskövel
Rövid leírás	Egy varázskövel ellátott torony ellenséget sebez.
Aktorok	Teszter
Forgatókönyv	A torony hatótávában az egyik ellenség életerőpontjait csökkenti, a benne lévő varázskő plusz sebzését figyelembe véve.

Use-case neve	Ellenség meghal
Rövid leírás	Ellenség életerőpontjai elfogynak a lövés után.
Aktorok	Teszter
Forgatókönyv	A lövés kiterjesztése. Amennyiben a lövés hatására az ellenség életerőpontjai elfogytak. Minden esetben, mikor az ellenség meghal, Mana-nyerés történik.

Use-case neve	Mana-nyerés
Rövid leírás	Varázserőt kap a játékos.
Aktorok	Teszter
Forgatókönyv	Az „Ellenség meghal” use-case tartalmazza. A játékos varázsereje jutalmul az ellenség mana-értékével növekszik.

Use-case neve	Játék nyéréssel végződik
Rövid leírás	Az „Ellenség meghal” kiterjesztése, ha a meghalt ellenség volt az utolsó, akkor a játékos nyer.
Aktorok	Teszter
Forgatókönyv	Ha az épp elpusztított ellenség volt az utolsó, akkor a játékos megnyerte a játékot.

Use-case neve	Lépés
Rövid leírás	Az ellenségek a következő csempére lépnek.
Aktorok	Teszter
Forgatókönyv	Az összes ellenség az úton számára következő csempére lép, figyelembe véve az akadályok lassítását.

Use-case neve	Játék vereséggel végződik
Rövid leírás	A Lépés kiterjesztése. A lépés hatására egy ellenség a végzet hegyére ér.
Aktorok	Teszter
Forgatókönyv	A lépés után van olyan ellenség, ami a Végzet Hegye csempén áll.

Use-case neve	Lassítás
Rövid leírás	A lépés kiterjesztése.
Aktorok	Teszter
Forgatókönyv	A lépés olyan csempére történik, amin akadály van.

6.2 A szkeleton kezelői felületének terve, dialógusok

A szkeleton az adott programnak egy váza, amelyben a metódusok és objektumok megtalálhatók, de a metódusok nincsenek implementálva.

A szkeleton által lehetőségünk nyílik a programunk tesztelésére futás közben. Ezt parancssori argumentumban megadva, valósíthatjuk meg, tehát az egyes teszteseteket

számmal azonosítjuk, majd ezt kapja a program parancssori argumentumnak. A metódusok kiírják a metódus nevét és paramétereit, és az osztály nevét, amin hívjuk. Így ellenőrizhetőek a szekvencia diagramok.

Karakteres képernyőkezelést használunk, és az egyes használati esetek a következő konvenciót követik:

- Minden metódushíváskor kiírjuk a metódus nevét, paramétereit, tartalmazó osztályának nevét
- metódushívás : '-->' , visszatérés : '<--'
- amikor egy metódus másik metódust hív, azt behúzással (tabulátorral) jelöljük

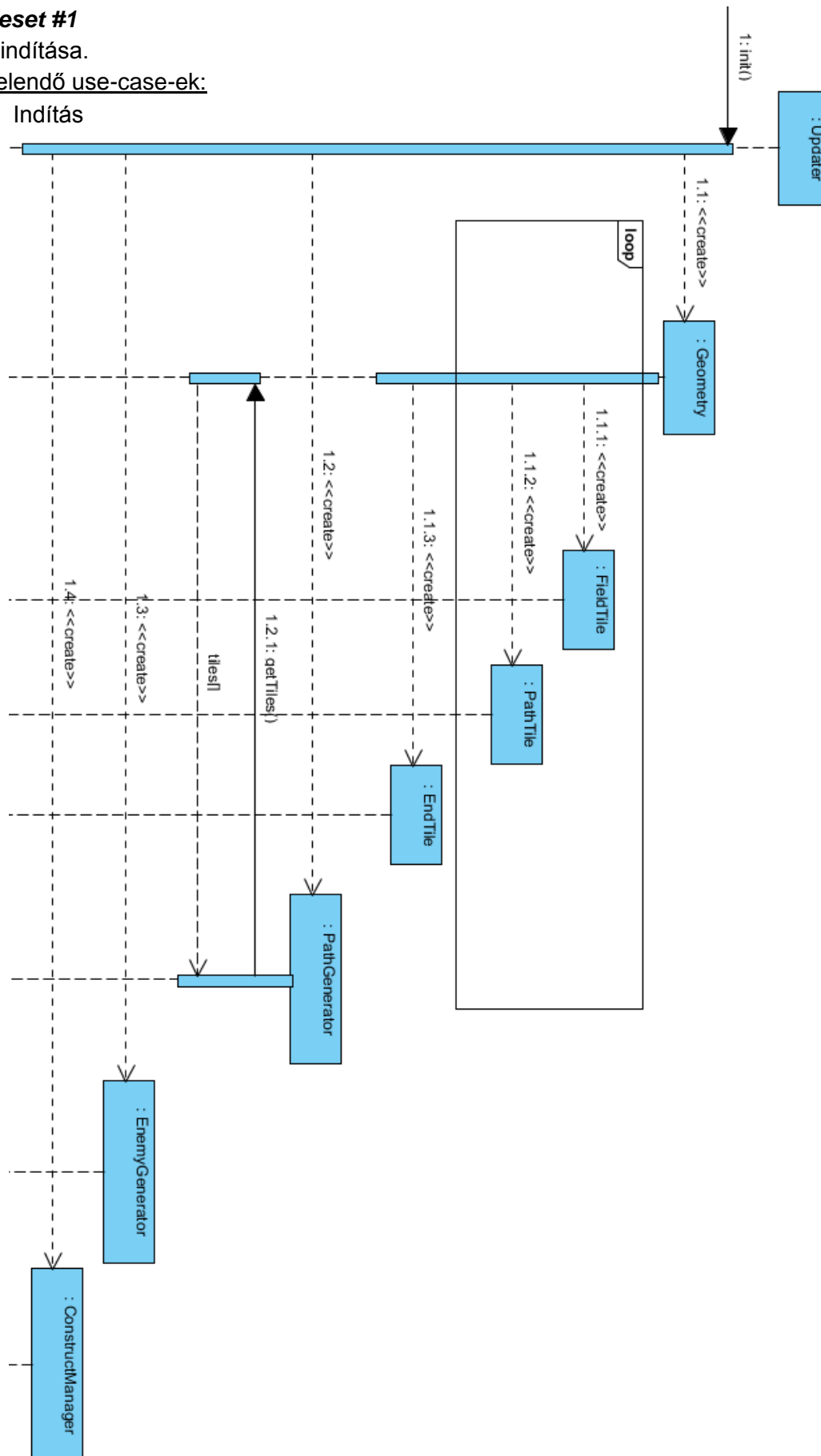
6.3 Szekvencia diagramok a belső működésre

Teszt eset #1

Játék indítása.

Tesztelendő use-case-ek:

- Indítás

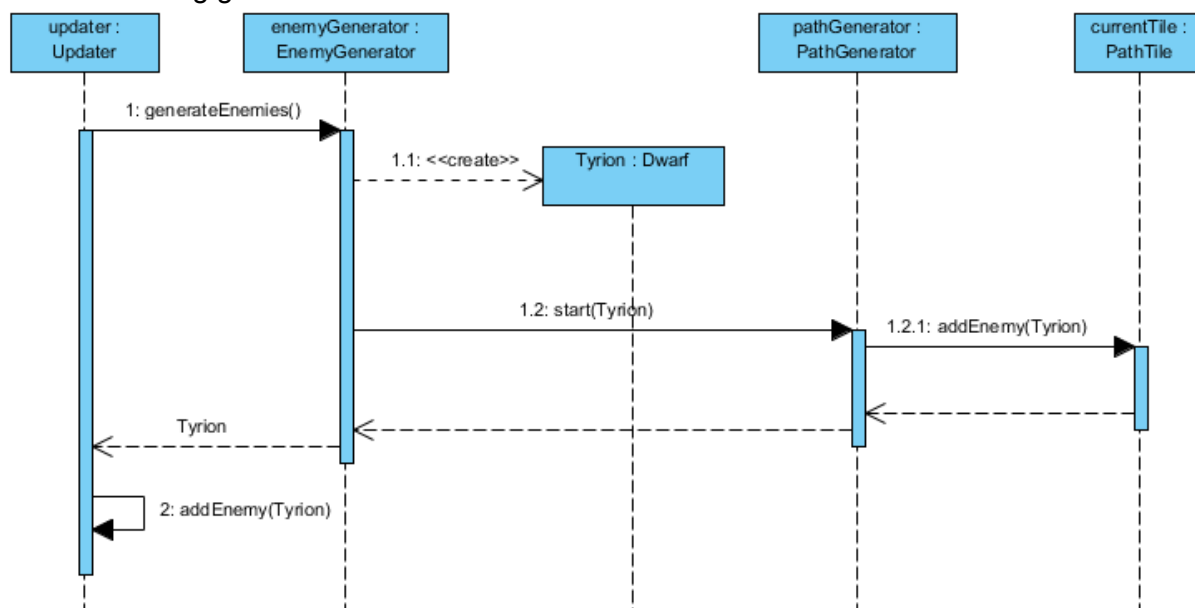


Teszt eset #2

Létrehozunk egy ellenséget.

Tesztelendő use-case-ek:

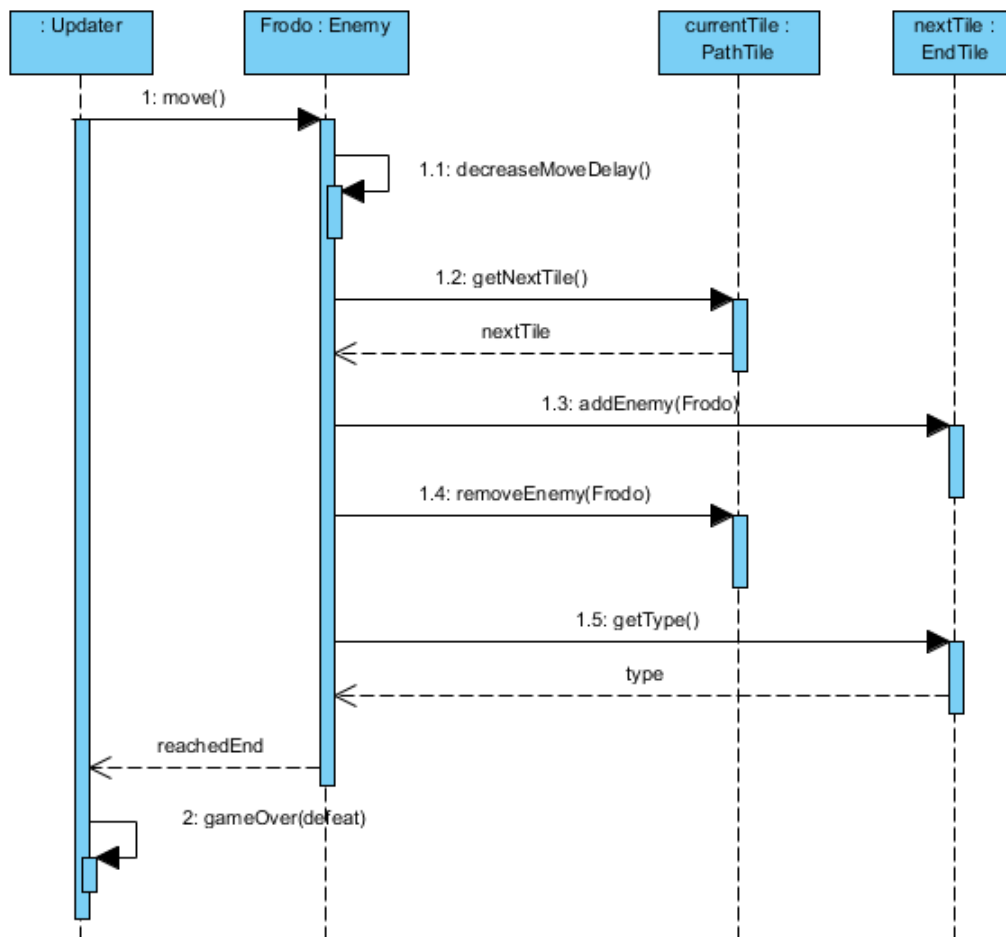
- Ellenség generálás

**Teszt eset #3**

Egy ellenség lép, majd mivel a Végzet hegyére lépett, veszítünk.

Tesztelendő use-case-ek:

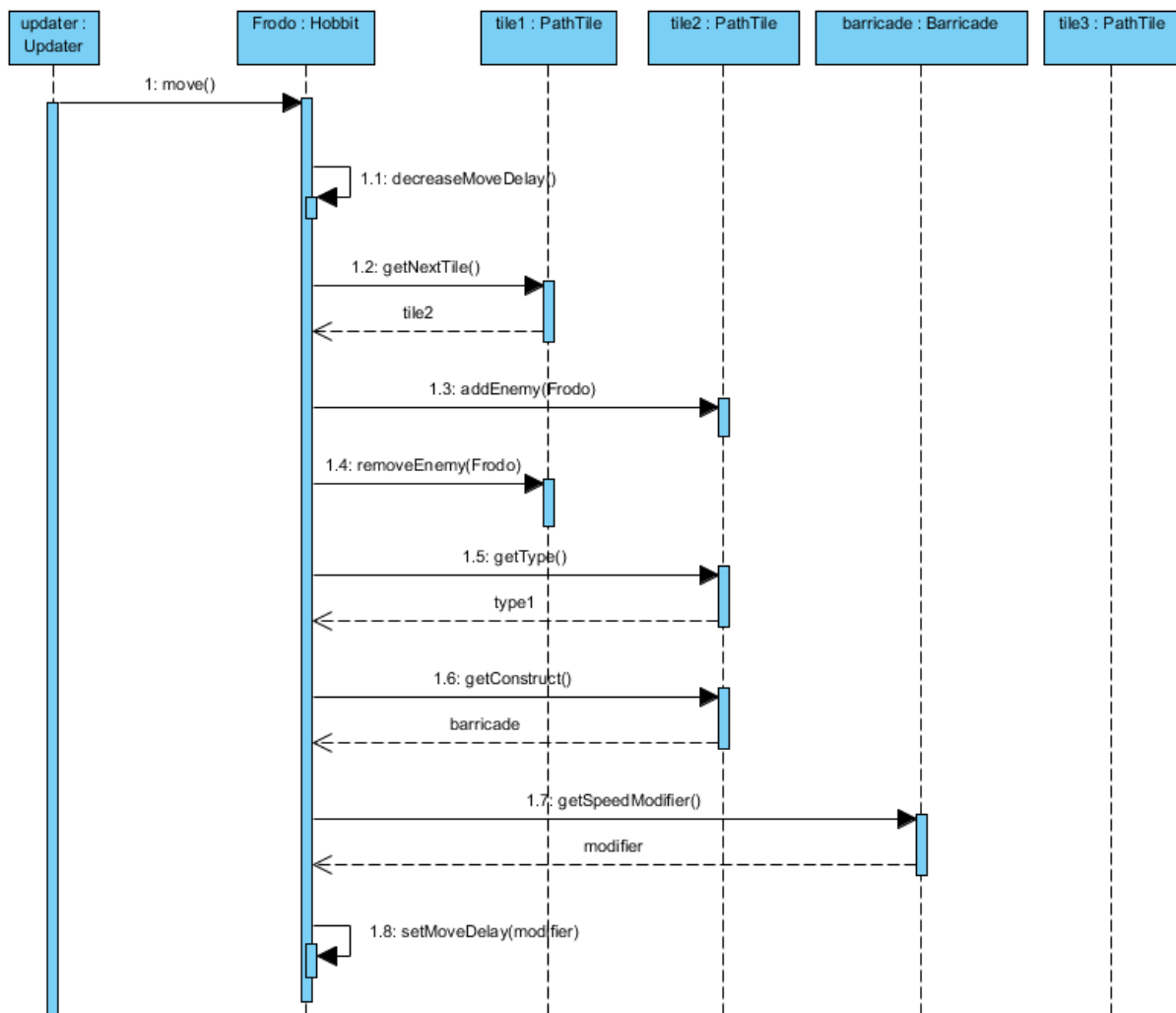
- Játék vereséggel ér véget

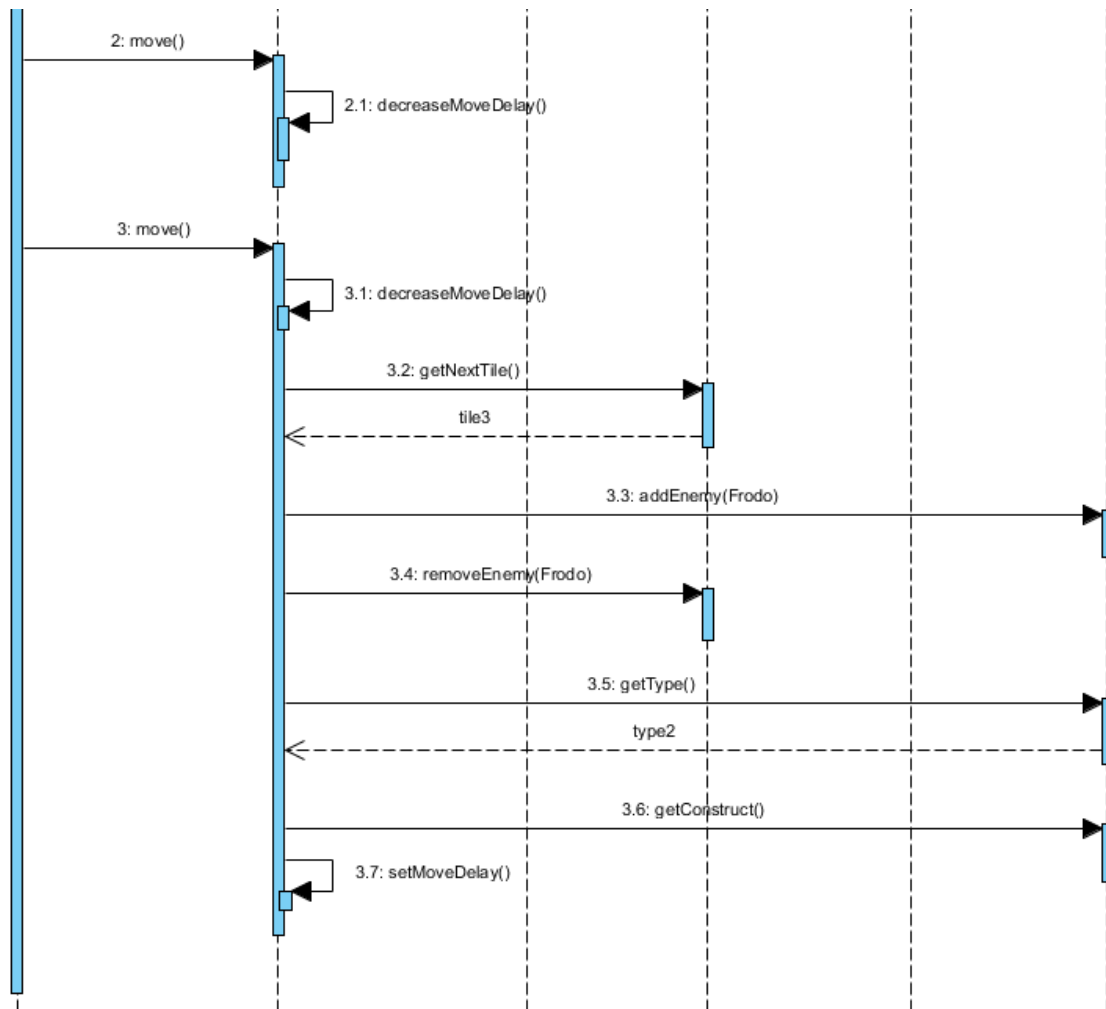
**Teszt eset #4**

Egy ellenség akadályra lép, majd késleltetve továbblép.

Tesztelendő use-case-ek:

- Lépés
- Lassítás



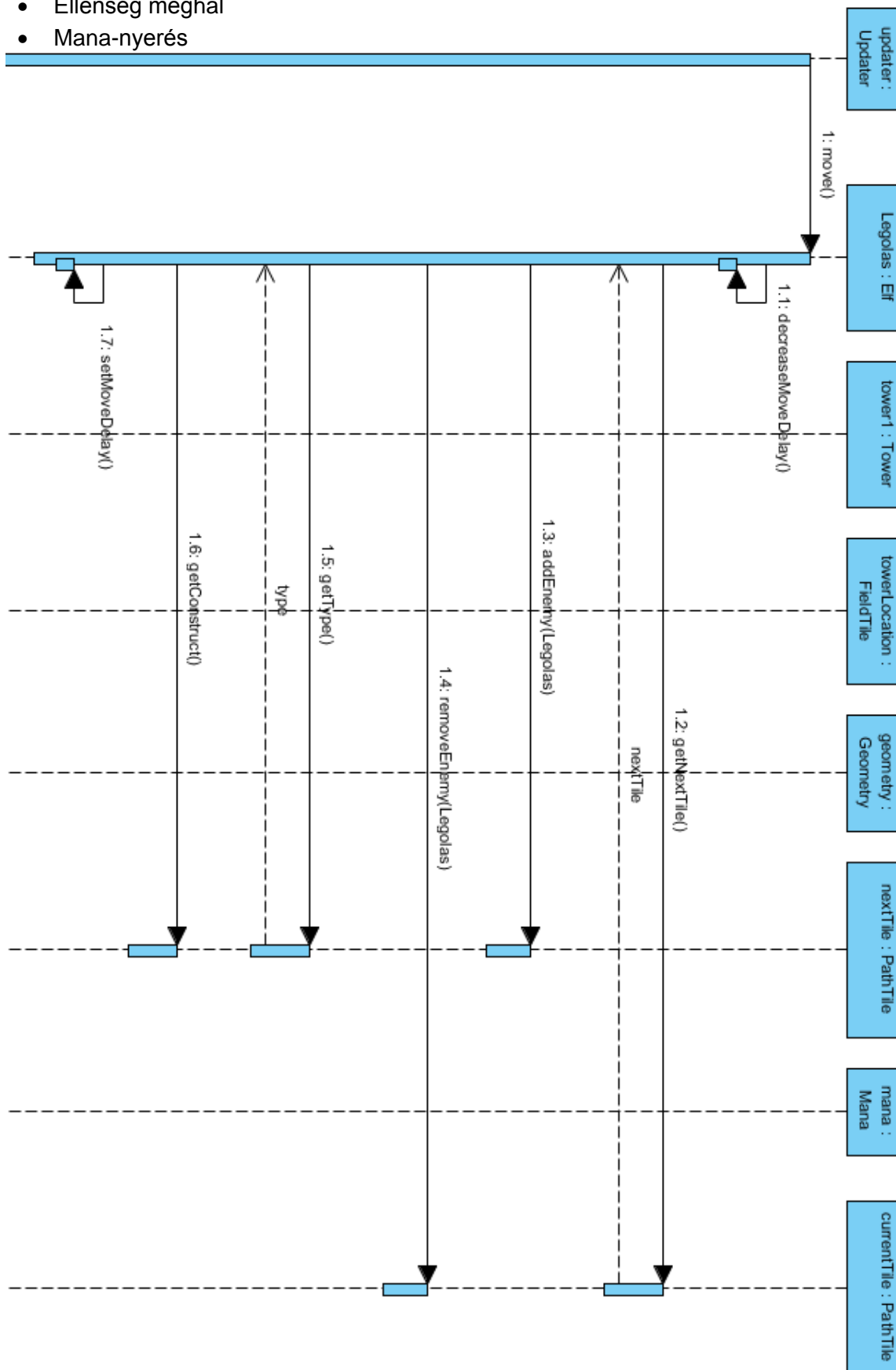


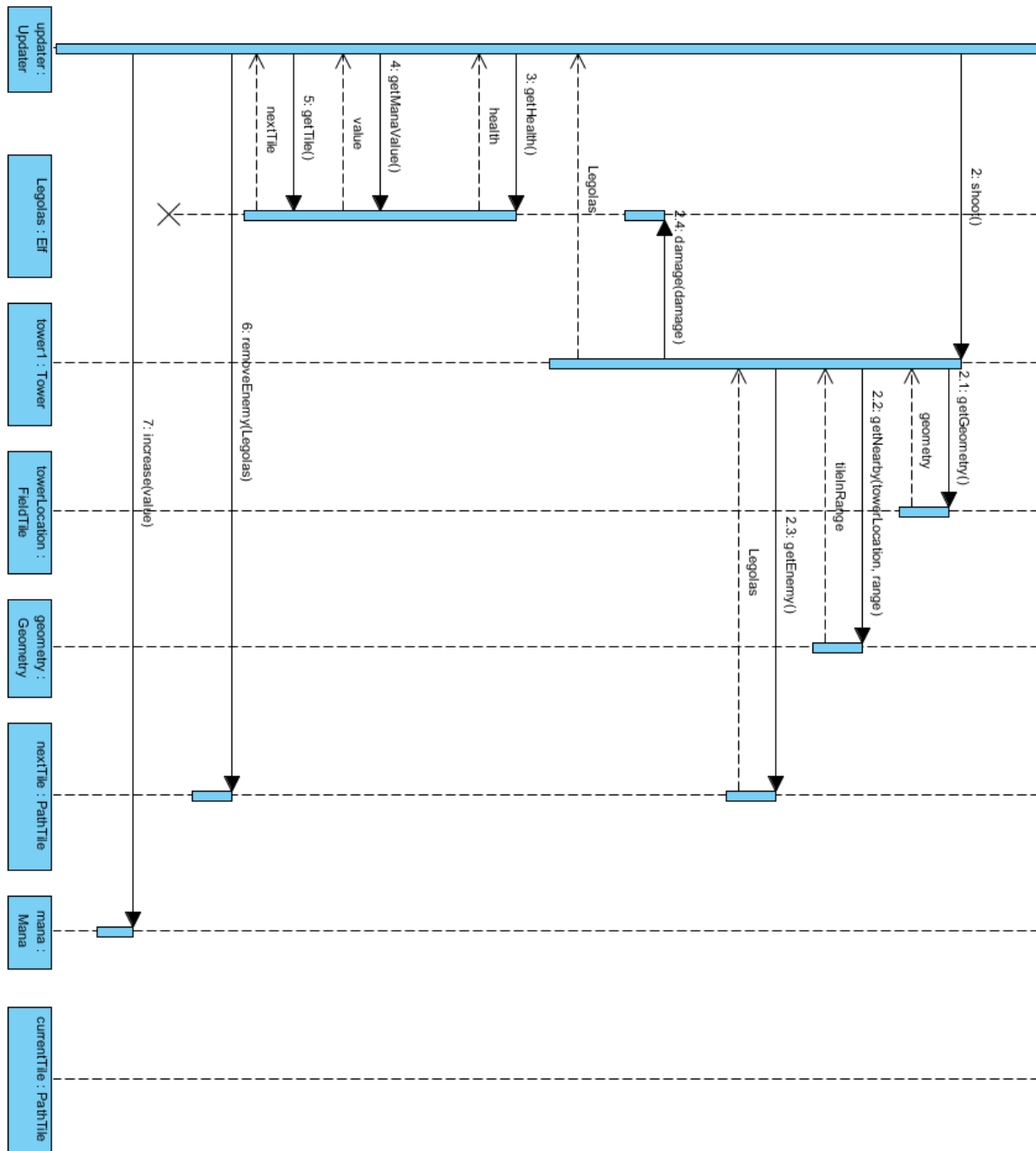
Teszt eset #5

Egy ellenség lép, majd rálő egy torony, és meghal.

Tesztelendő use-case-ek:

- Lövés
- Ellenség meghal
- Mana-nyerés



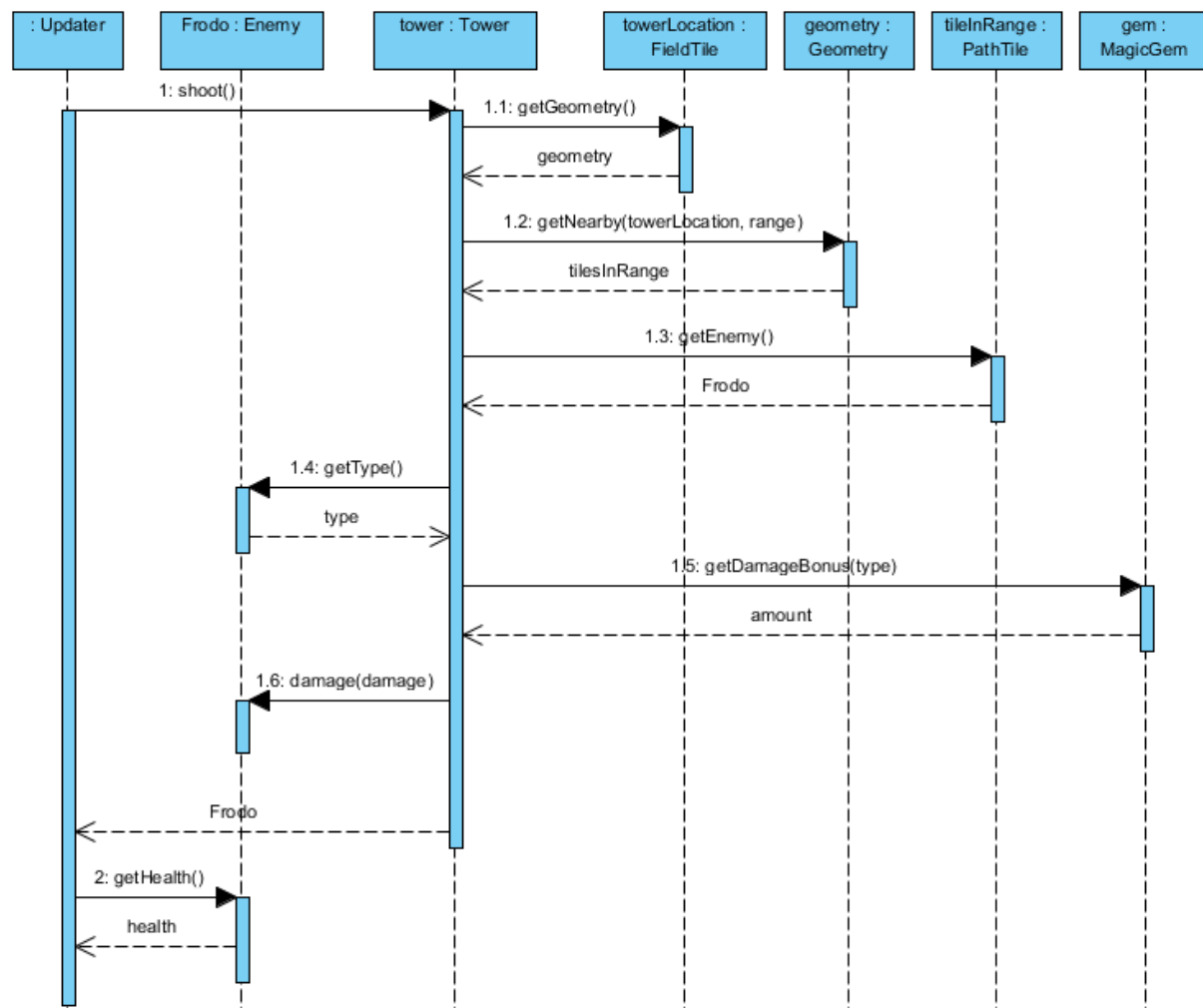


Teszt eset #6

Lövés varázskővel ellátott toronnyal.

Tesztelendő use-case-ek:

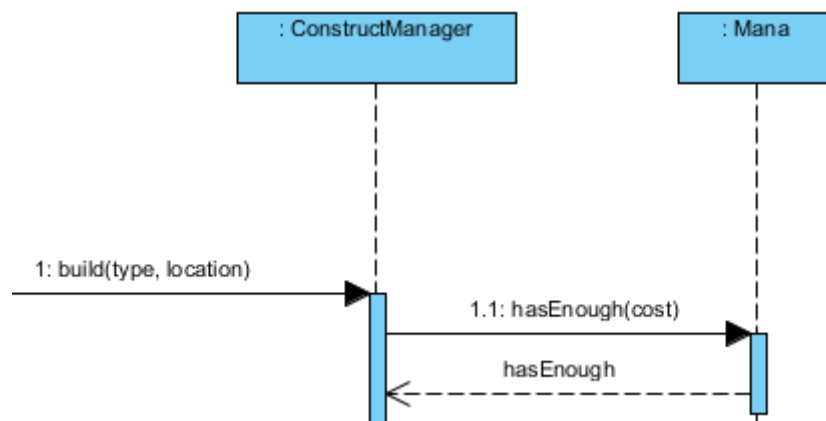
- Lövés varázskővel.

**Teszt eset #7**

A játékos építeni akar valamit, de nincs rá elég varázserő.

Tesztelendő use-case-ek:

- Nincs elég mana

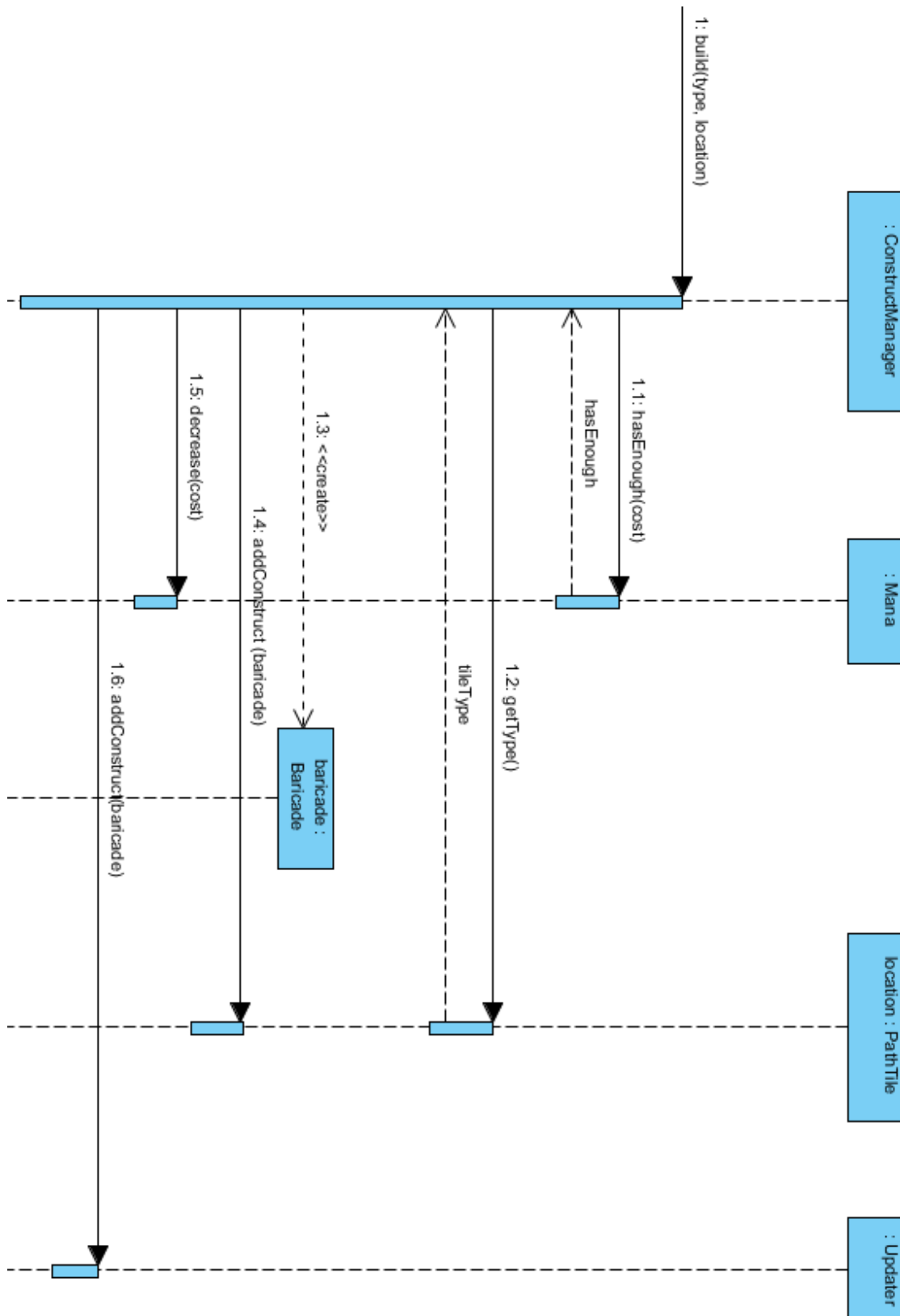


Teszt eset #8

A játékos épít egy akadályt.

Tesztelendő use-case-ek:

- Akadály-vásárlás
- Mana-költés

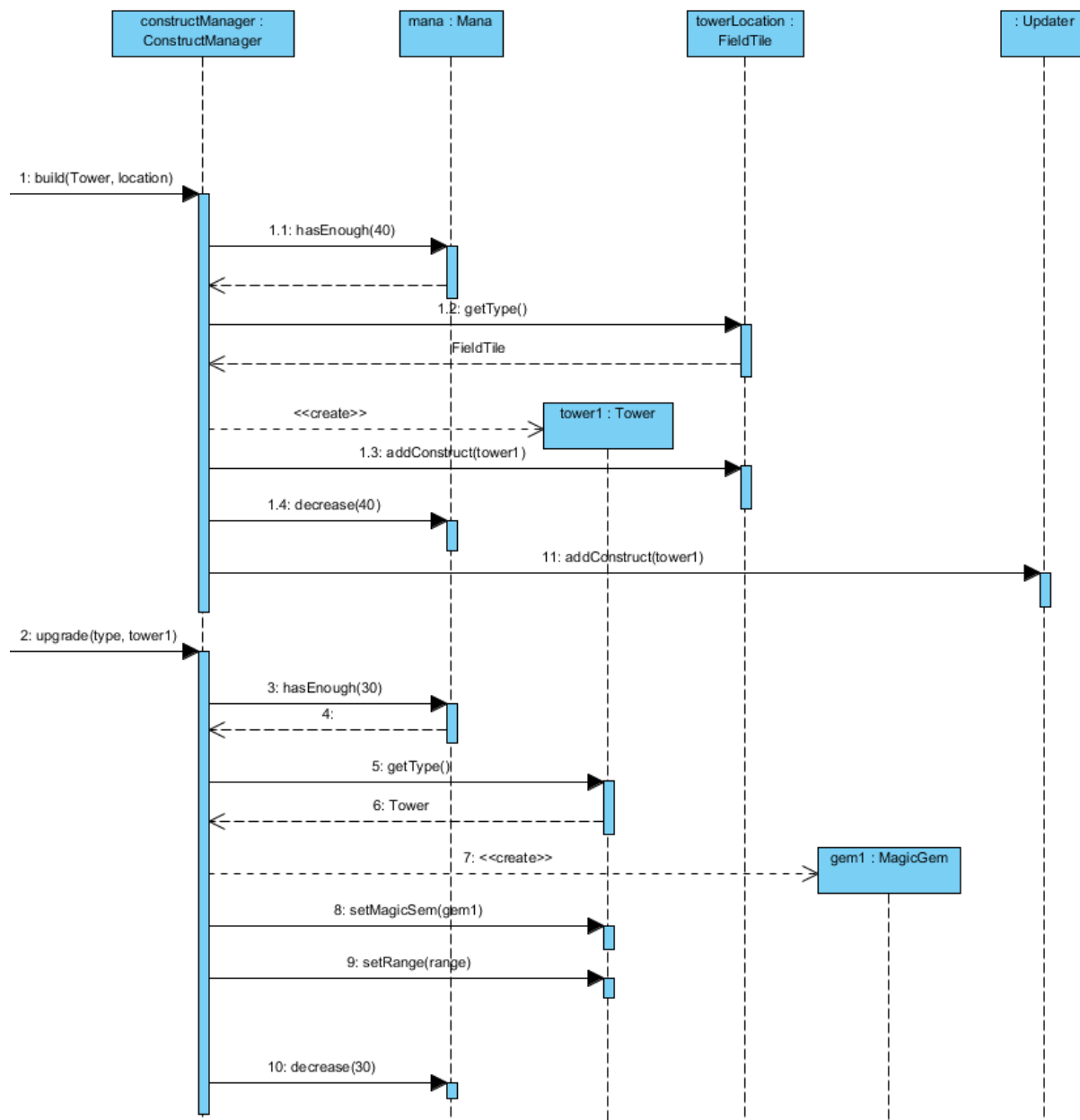


Teszt eset #9

A játékos épít egy tornyot, majd vásárol bele egy (hatótáv növelő) varázskövet.

Tesztelendő use-case-ek:

- Torony-vásárlás
- Varázskő-vásárlás

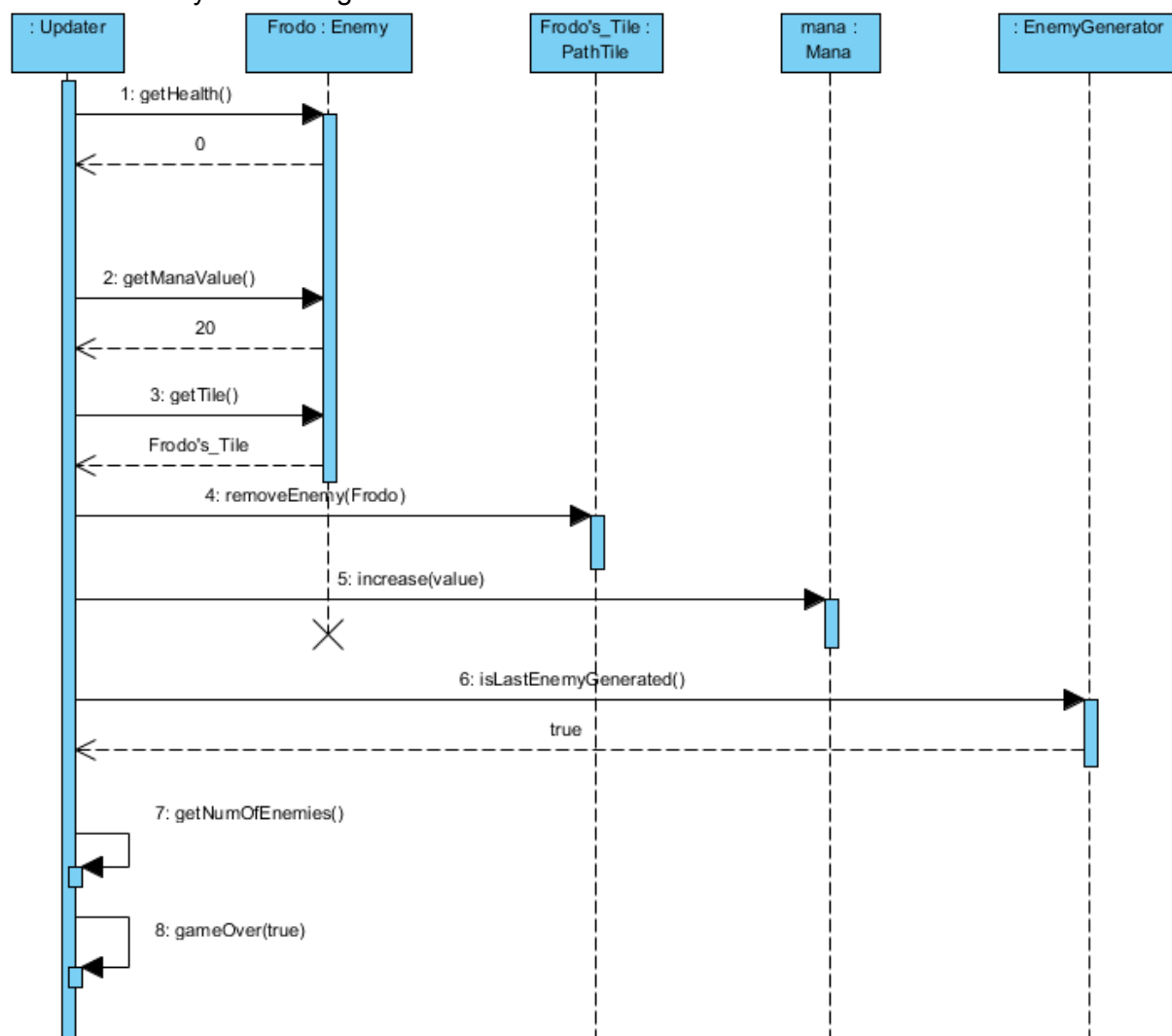


Teszt eset #10

Egy torony lőtt egy ellenségre, aki meghal, majd mivel ez volt az utolsó ellenség, győzünk.

Tesztelendő use-case-ek:

- Játék nyéréssel végződik



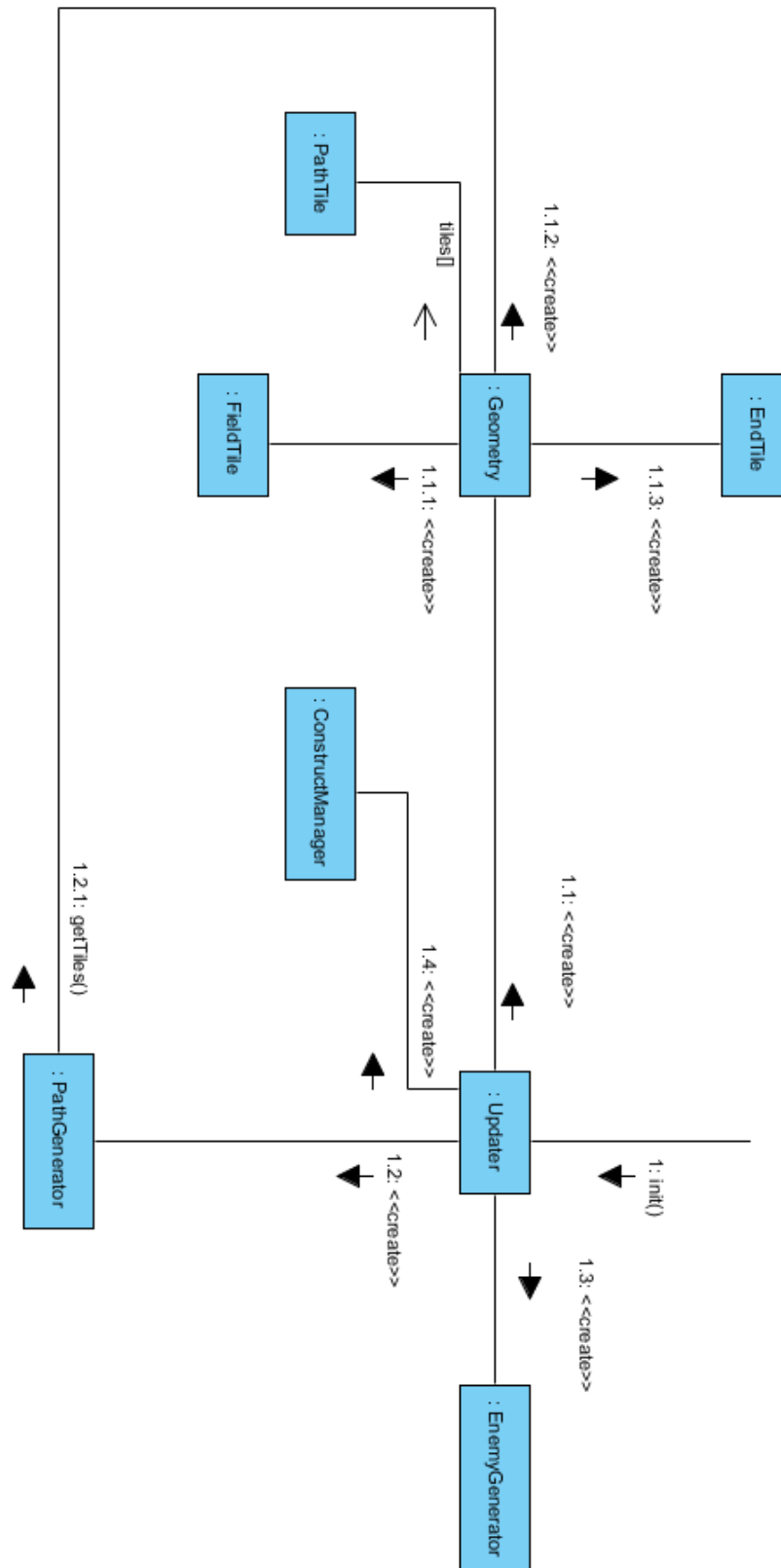
6.4 Kommunikációs diagramok

Teszt eset #1

Játék indítása.

Tesztelendő use-case-ek:

- Indítás

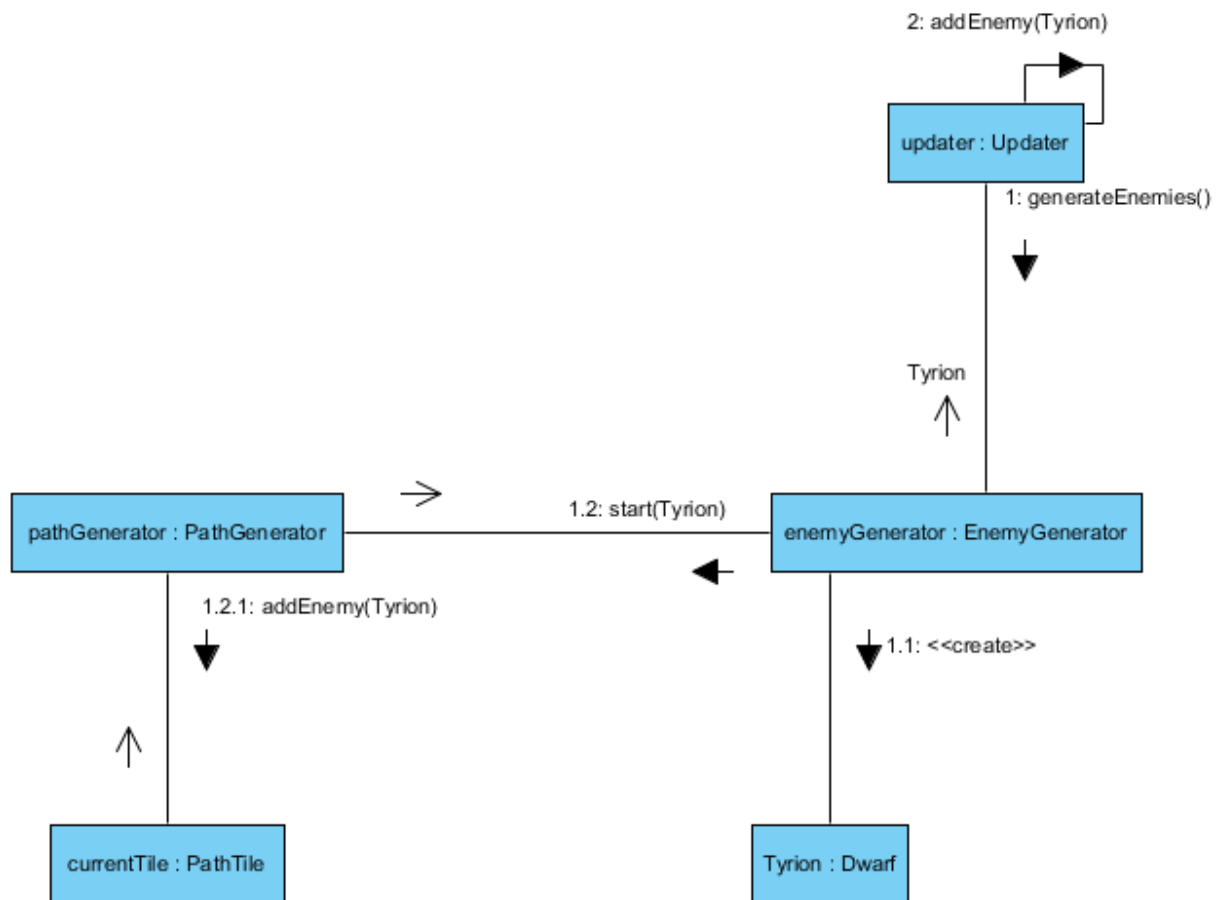


Teszt eset #2

Létrehozunk egy ellenséget.

Tesztelendő use-case-ek:

- Ellenség generálás

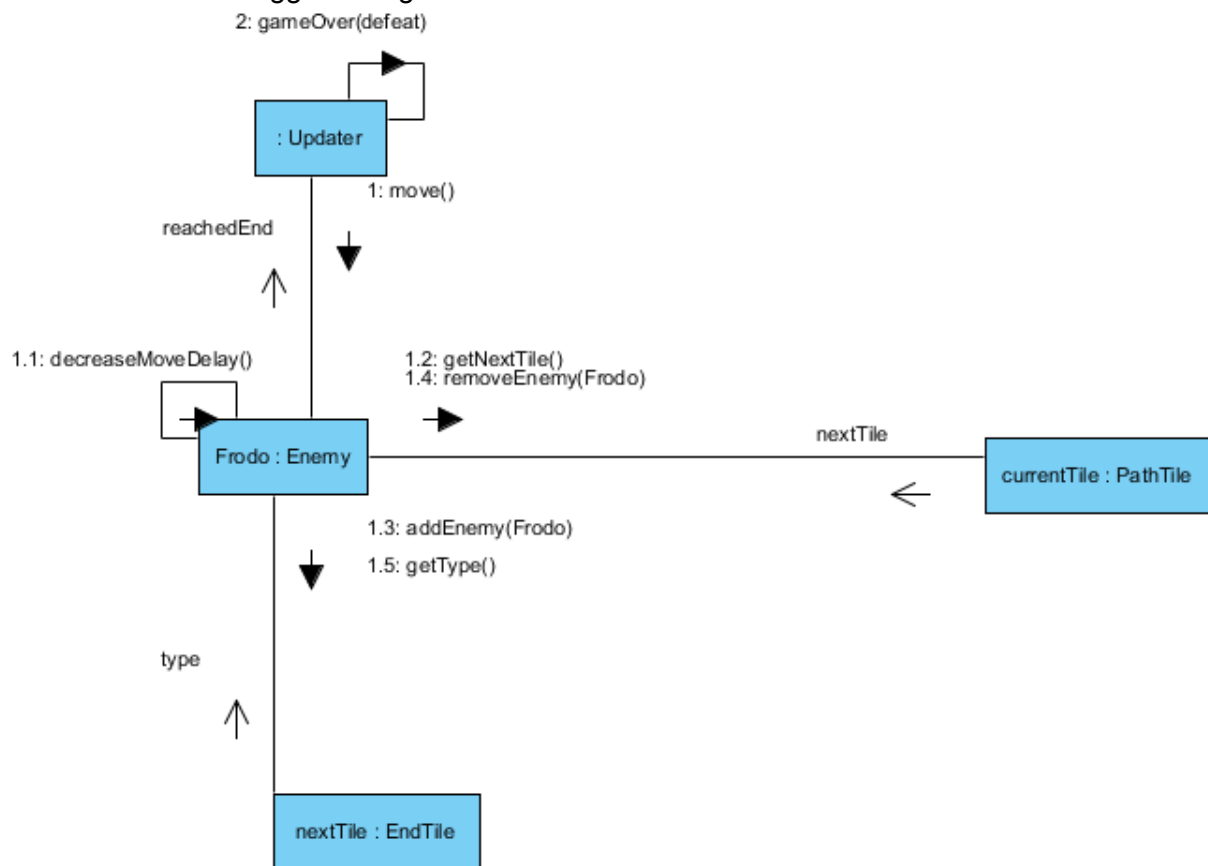


Teszt eset #3

Egy ellenség lép, majd mivel a Végzet hegyére lépett, veszítünk.

Tesztelendő use-case-ek:

- Játék vereséggel ér véget

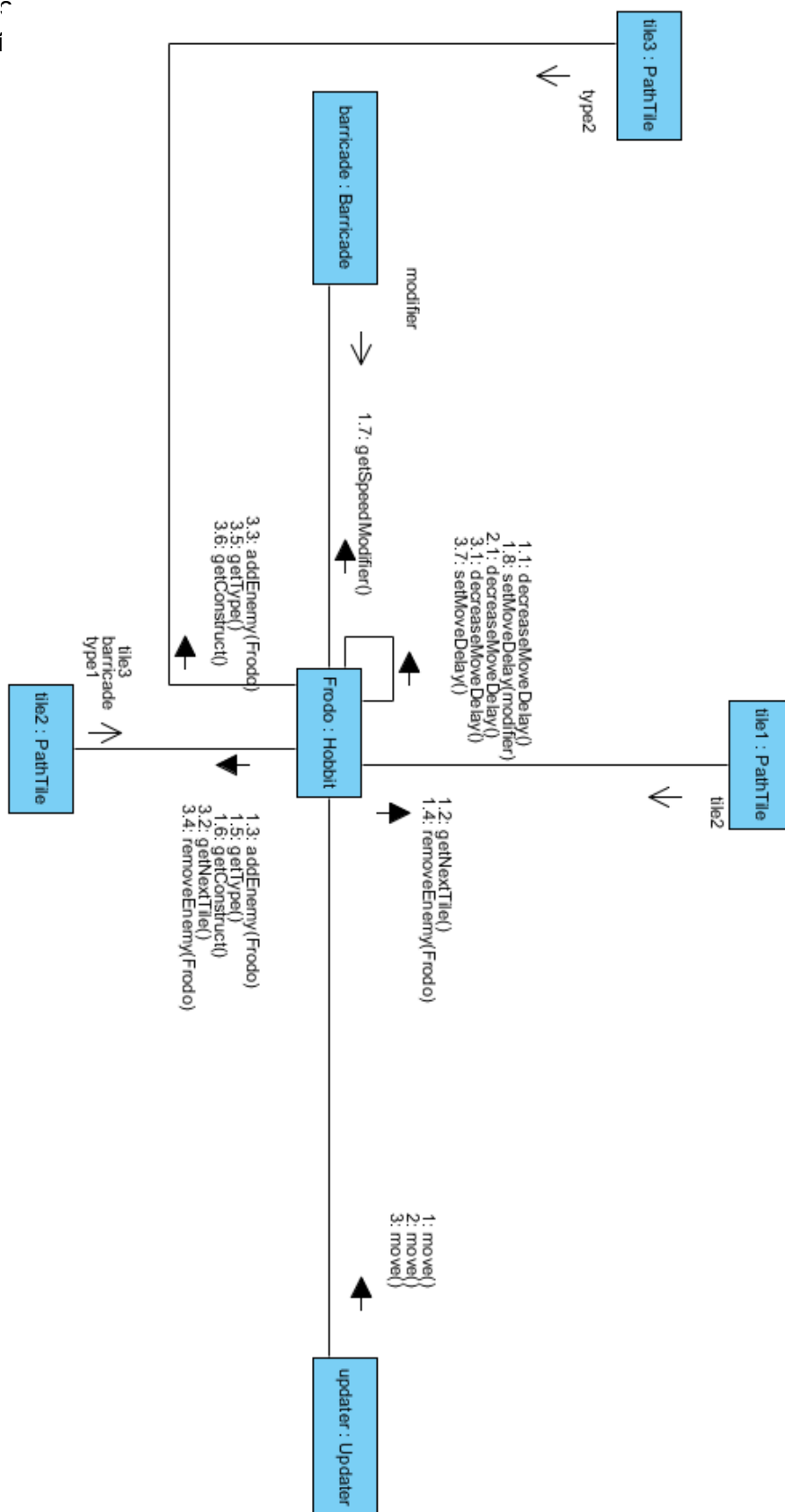


Teszteset #4

Egy ellenség akadályra lép, majd késleltetve továbblép.

Tesztelendő use-case-ek:

- Lépér
- Lassí

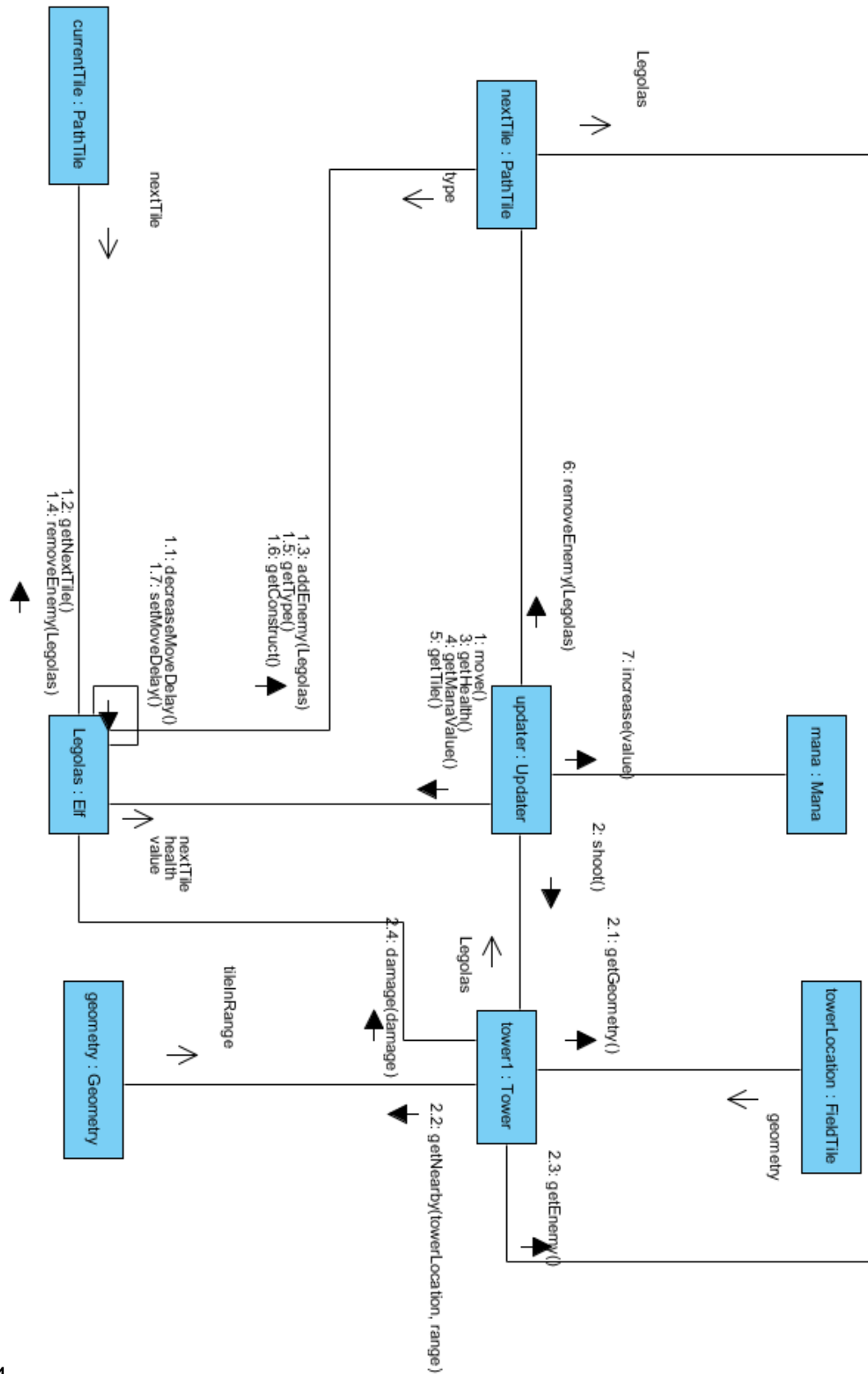


Teszt eset #5

Egy ellenség lép, majd rálő egy torony, és meghal.

Tesztelendő use-case-ek:

- Lövés
- Ellenség meghal
- Mana-nyerés

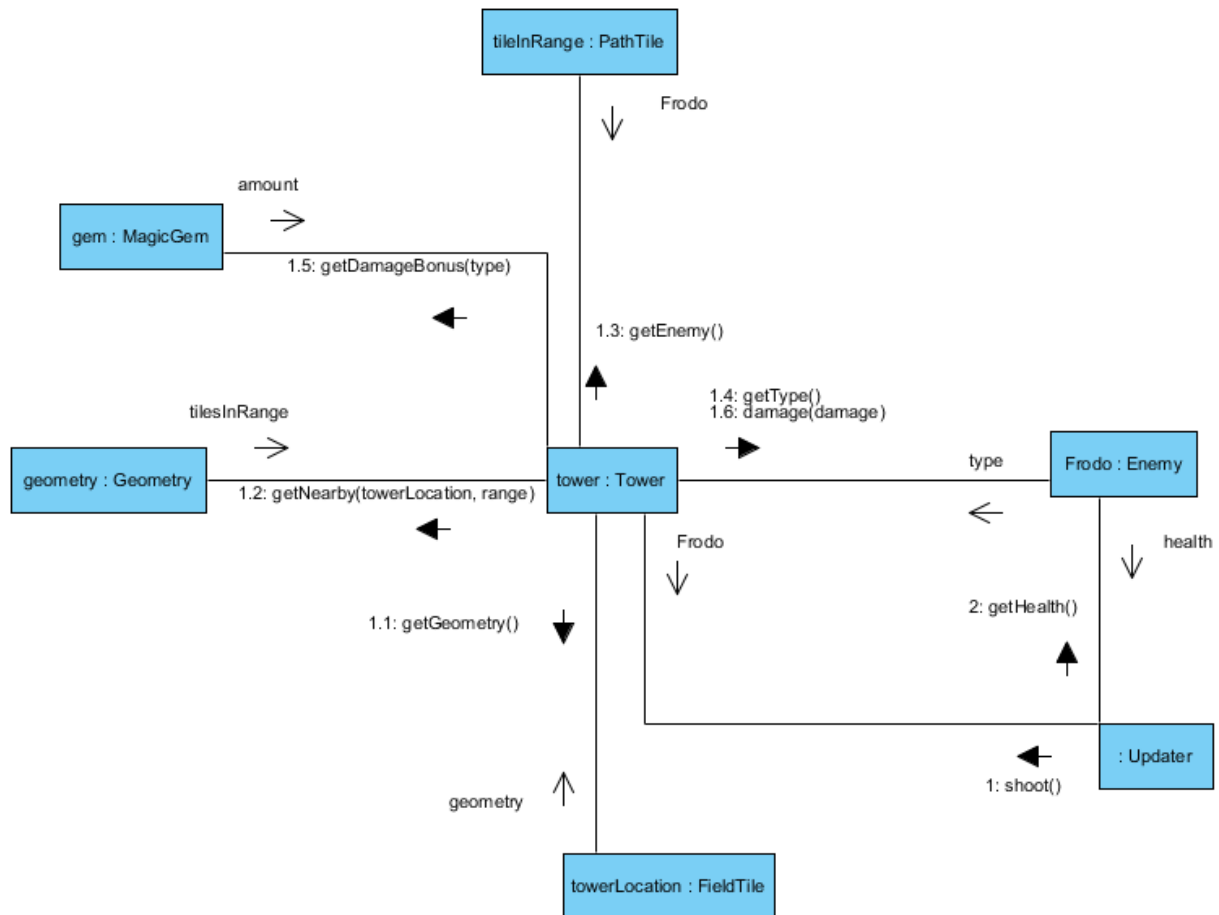


Teszt eset #6

Lövés varázskövel ellátott toronnyal.

Tesztelendő use-case-ek:

- Lövés varázskövel.

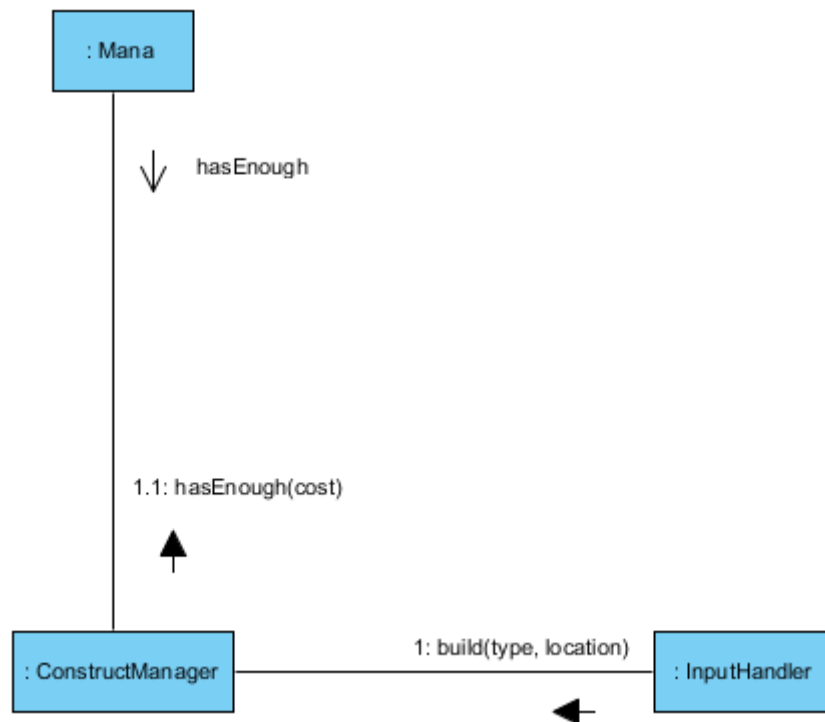


Teszt eset #7

A játékos építeni akar valamit, de nincs rá elég varázserő.

Tesztelendő use-case-ek:

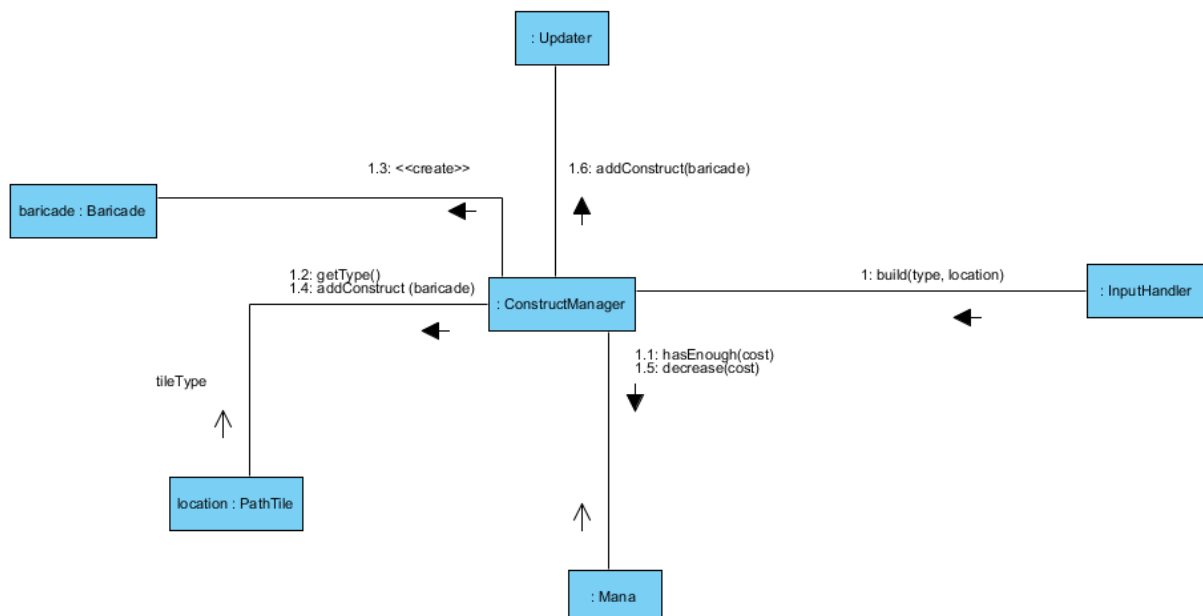
- Nincs elég mana

**Teszt eset #8**

A játékos épít egy akadályt.

Tesztelendő use-case-ek:

- Akadály-vásárlás
- Mana-költség

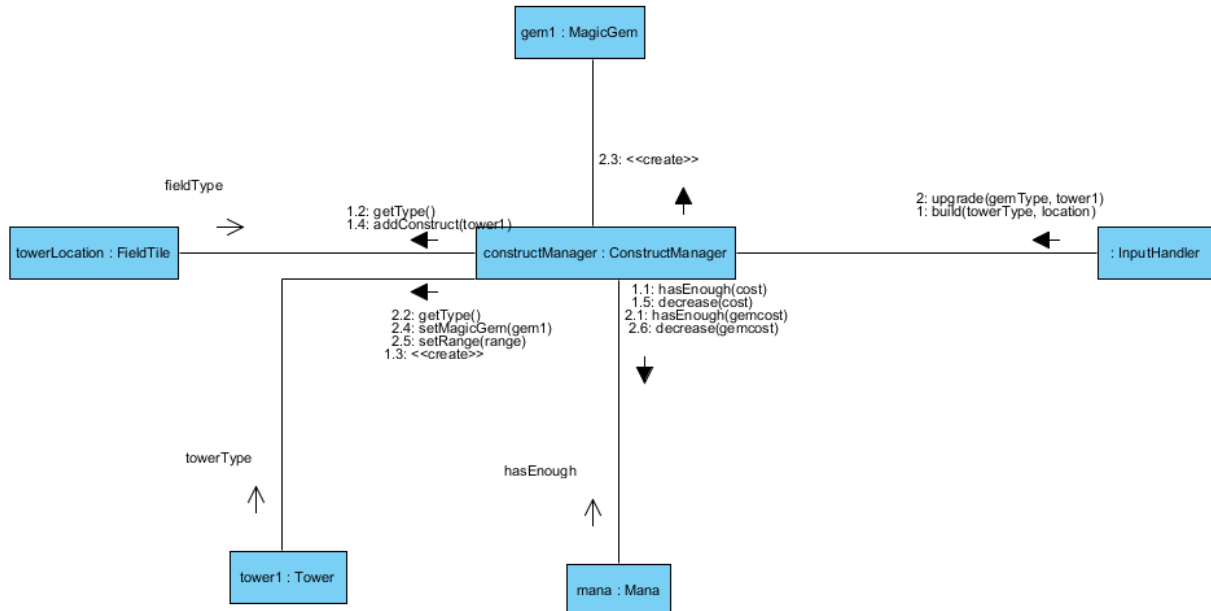


Teszt eset #9

A játékos épít egy tornyot, majd vásárol bele egy (hatótáv növelő) varázskövet.

Tesztelendő use-case-ek:

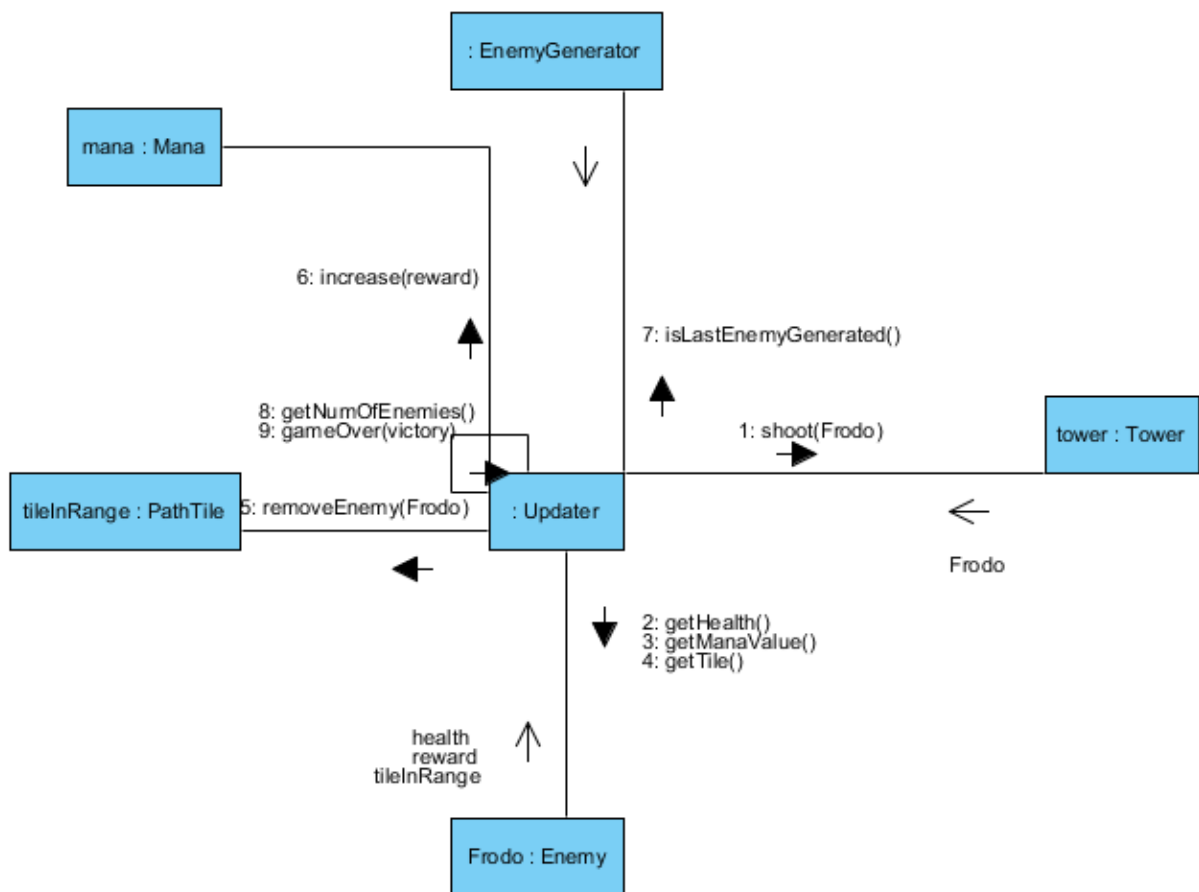
- Torony-vásárlás
- Varázskő-vásárlás

**Teszt eset #10**

Egy torony lő egy ellenségre, aki meghal, majd mivel ez volt az utolsó ellenség, győzünk.

Tesztelendő use-case-ek:

- Játék nyéréssel végződik



7. Szkeleton beadás

7.1 Fordítási és futtatási útmutató

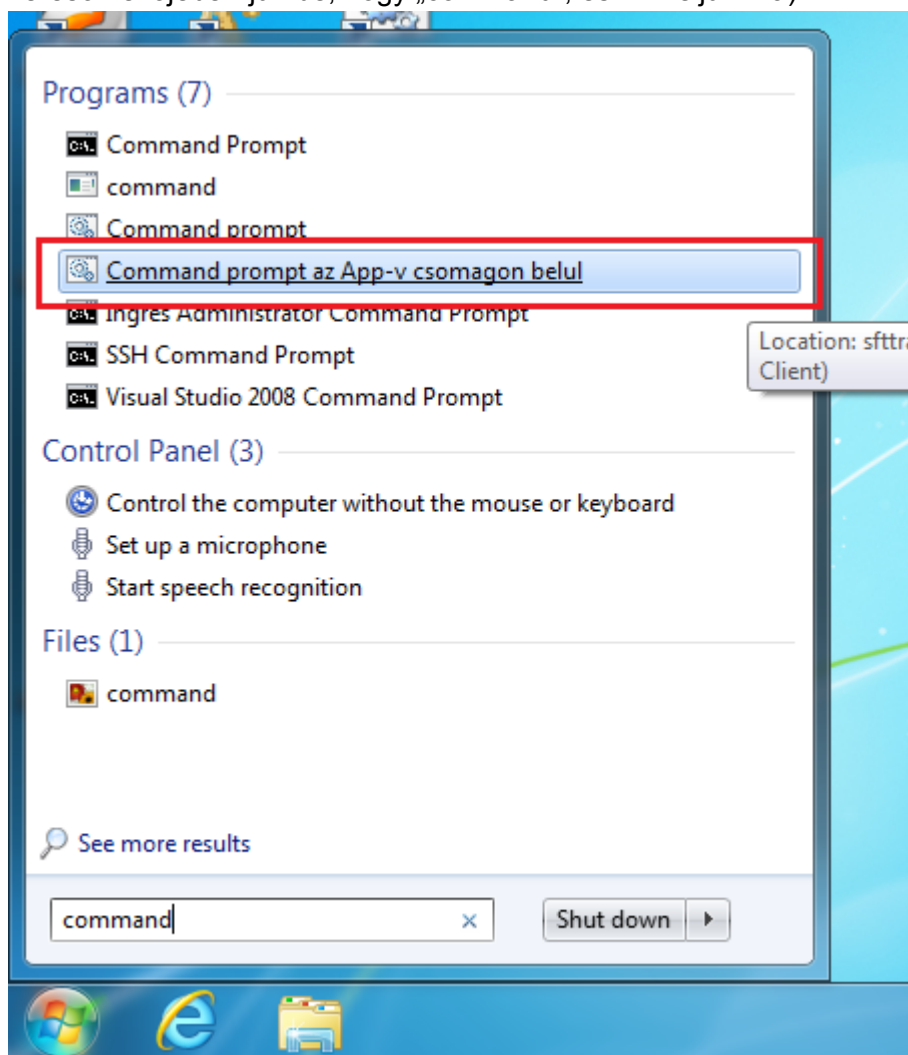
7.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Barricade.java	1 KB	2014.03.23.	Az akadályokért felelős osztály megvalósítása.
Construct.java	1 KB	2014.03.23.	Az épületek őssztályának megvalósítása.
ConstructManager.java	2 KB	2014.03.24.	Az épületek létrehozásáért és fejlesztéséért felelős osztály megvalósítása.
Dwarf.java	1 KB	2014.03.24.	A törpe típusú ellenségek kezeléséért felelős osztály megvalósítása.
Elf.java	1 KB	2014.03.23.	A tünde típusú ellenségek kezeléséért felelős osztály megvalósítása.
EndTile.java	1 KB	2014.03.23.	A Végzet Hegyét tároló csempe kezeléséért felelős osztály.
Enemy.java	3 KB	2014.03.24.	Az ellenségek őssztályának megvalósítása.
EnemyGenerator.java	1 KB	2014.03.23.	Az ellenségek létrehozásáért felelős osztály.
FieldTile.java	1 KB	2014.03.23.	A terepcellákért felelős osztály megvalósítása, ezekre lehet tornyot építeni.
Geometry.java	2 KB	2014.03.23.	A pálya geometriai szerkezetéért felelős osztály megvalósítása.
Hobbit.java	1 KB	2014.03.23.	A hobbit típusú ellenséget kezelő osztály megvalósítása.
Human.java	1 KB	2014.03.23.	Az ember típusú ellenséget kezelő osztály megvalósítása.
MagicGem.java	1 KB	2014.03.23.	A varázskövekért felelős osztály megvalósítása.
Mana.java	1 KB	2014.03.24.	A játékos varázserejének kezeléséért felelős osztály megvalósítása.

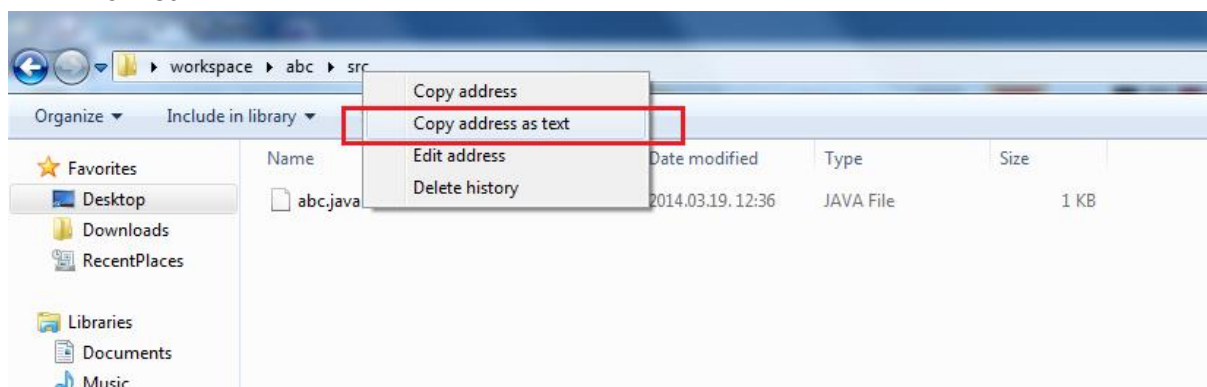
PathGenerator.java	1 KB	2014.03.23.	Az útvonalakat kezelő osztály megvalósítása.
PathTile.java	2 KB	2014.03.24.	Az útvonalcellákért felelős osztály megvalósítása, ezekre lehet akadályt építeni, és ellenségek helyzetét adja meg.
Tester.java	11 KB	2014.03.24.	A tesztesetek helyes lefutásáért felelős osztály megvalósítása.
Tile.java	1 KB	2014.03.24.	A csempék ösosztályának megvalósítása.
Tower.java	2 KB	2014.03.23.	A tornyokért felelős osztály megvalósítása.
Updater.java	4 KB	2014.03.24.	Az időkezelésért felelős osztály megvalósítása.

7.1.2 Fordítás

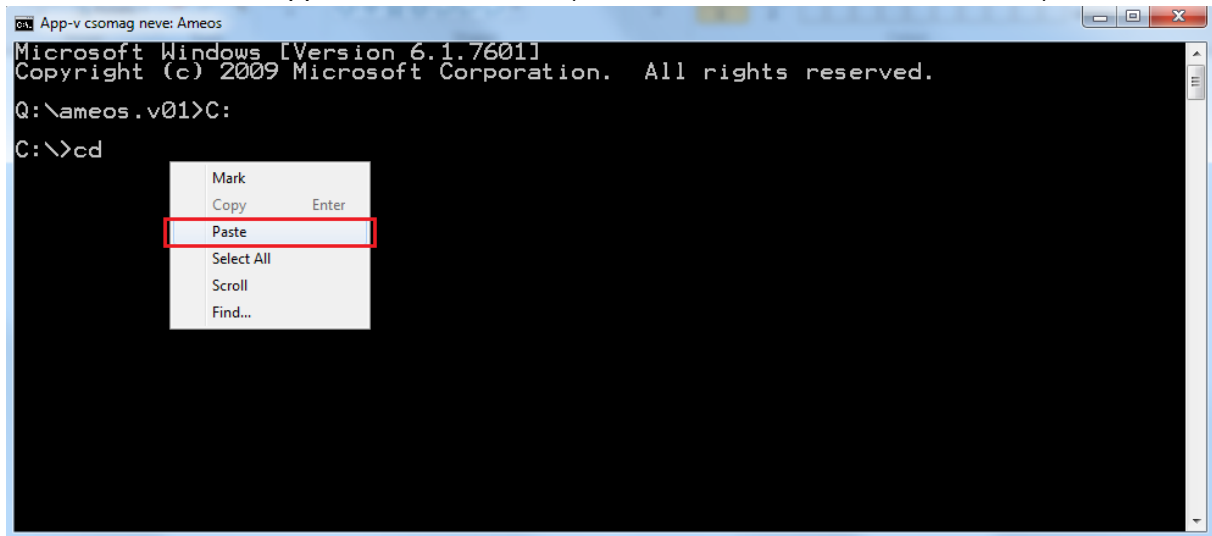
- Indítsunk egy Command Prompt-ot az App-v csomagon belül (A Start menü keresőmezőjébe írjuk be, hogy „command”, és klikkeljünk rá).



- Lépünk a command promptban abba a mappába, ahol a forráskód található. Ehhez először beléphetünk a mappába a Windows Intézőben és kimásolhatjuk a mappa címét.



- Ezt felhasználva a command promptban lépünk a forráskód mappájába. Először a meghajtó betűjelét írjuk be egy kettősponttal (C:), majd a cd parancs után kell illeszteni a mappa elérési útvonalát (amit most másoltunk ki az Intézőből).



- A sikeres mappaváltás után adjuk ki a következő parancsot:
`javac Tester.java`

7.1.3 Futtatás

Lépünk a mappába, ahol a fordítás végeztük! (A Fordítás 1-3. lépése ezt írja le). Ezután adjuk ki a következő parancsot, a <teszteset sorszáma> értéket megfelelően behelyettesítve:

```
java Tester <teszteset sorszáma>
```

Pl. az első teszteset lefuttatásához a következő parancsot kell kiadni:

```
java Tester 1
```

A teszt indítása után lefut a teszteset környezetének inicializálása, ez a teszt előkészületének tekinthető. Ezután fut le maga a teszt a szekvenciadiagramok szerint. Miután a teszt lefut, a program kilép, és a sorszámot lecserélve ugyanezzel a paranccsal indíthatjuk a következő tesztesetet. Összesen 10 teszteset van.

7.2 Értékelés

Tag neve	Munka százalékban
Apáti-Nagy Attila	18.0 %
Barabás Péter	21.1 %
Boczán Tamás	20.4 %
Csörgő Szabolcs	20.0 %
Révész Péter	20.5 %

8. Prototípus koncepciója

8.0 Változtatások a módosított feladatra

8.0.1 A változások leírása

1. Változtatás - Köd:

A tornyokra időnként köd ereszkedik, aminek következtében a látás erősen lecsökken. Ez hatással van a lövésre.

Megoldás:

A tornyokon mostantól beállítható, hogy lövésnél a torony hatótávát mennyivel változik. Ezt az Updater osztály állítja be, amikor a köd leereszkedik vagy felszáll.

Új metódus:

Tower osztályban: *setRangeModifier(int)*

2. Változtatás - Elágazások:

A játékosok által járható útvonalon lehetnek elágazások és becsatlakozások. Az elágazásokon az egyes játékosok véletlenszerűen mennek a különböző irányokba.

Megoldás:

Eddig minden Út-Csempe tárolta az úton utána következő 1 db csempét. Ehelyett most egy listát tárol a csempékről, amik utána következhetnek az úton. Amikor az ellenség lép, ezek közül választ egyet.

Lecserélt attribútum:

PathTile osztályban: *Tile* típusú *NextTile* lecserélve
Tile [] típusú *NexTiles* -ra.

Lecserélt metódus:

PathTile osztályban: *Tile* visszatérésű *getNextTile()* lecserélve
Tile[] visszatérésű *getNextTiles()*-ra.

3. Változtatás - Kettélövés:

A tornyokban elvértve lehetnek olyan lövedékek, amelyek az eltalált játékost kétféle vágják. A két játékos egymástól függetlenül él tovább, csökkentett életerővel.

Megoldás:

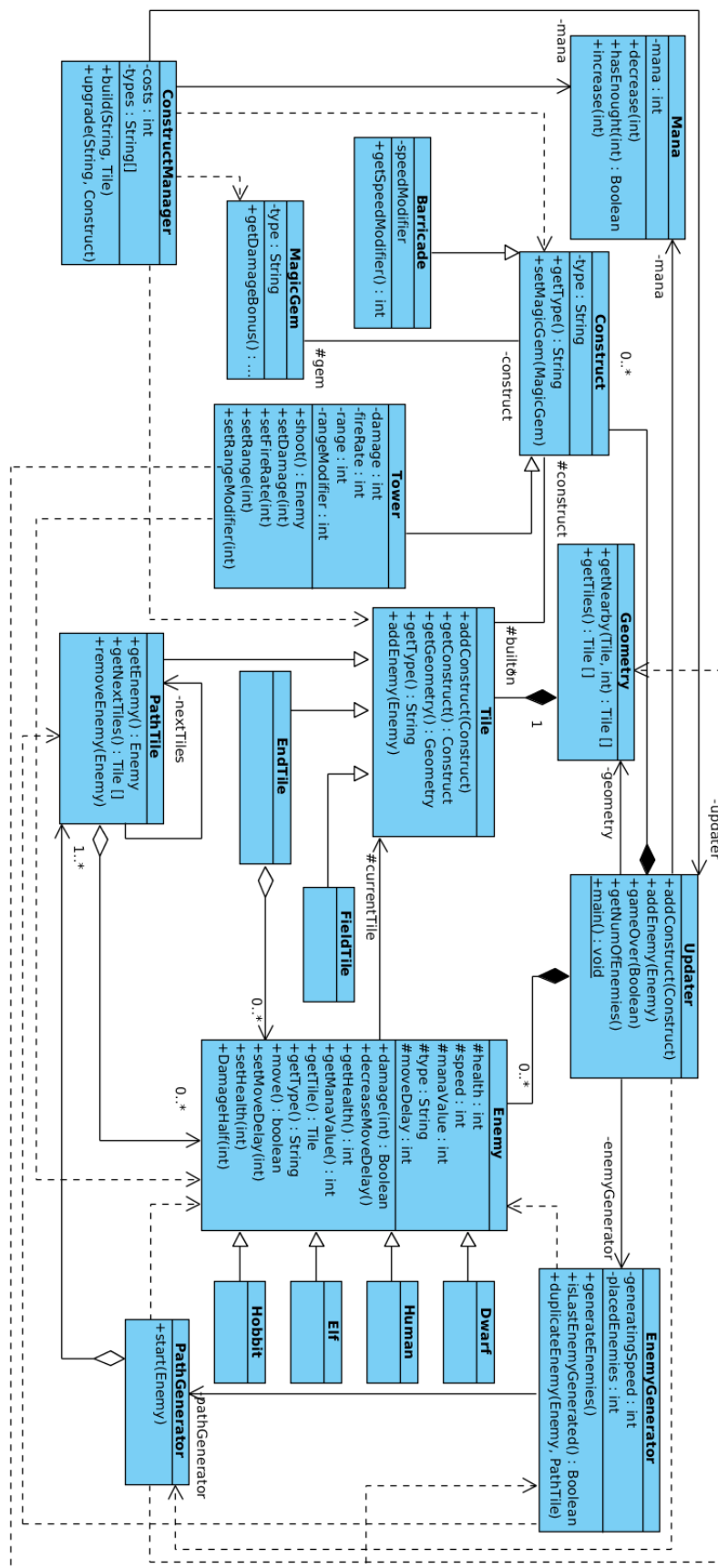
Amikor a torony kétféle lövedéket lő, akkor az ellenség életerejének felét elveszi, majd értesíti az EnemyGenerátort, hogy le kell másolni az ellenséget ugyanarra a mezőre. Az EnemyGenerator létrehoz egy ugyanolyan típusú és életerejű ellenséget ugyanazon a csempén.

Új metódusok:

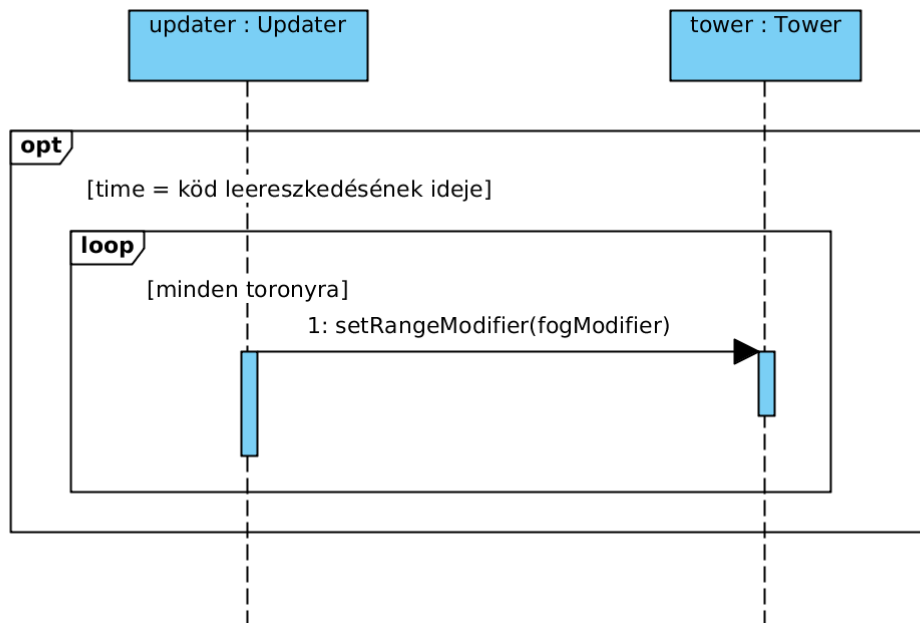
Enemy osztályban: *damageHalf()*
setHealth(int)

EnemyGenerator osztályban: *duplicateEnemy(Enemy, PathTile)*

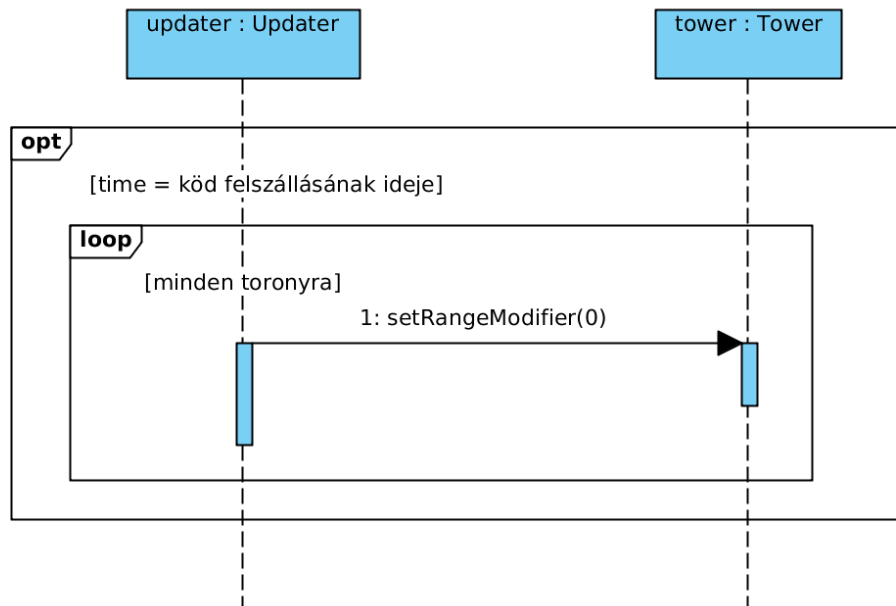
8.0.2 Megváltozott osztálydiagram



8.0.3 Megváltozott szekvencia diagramok

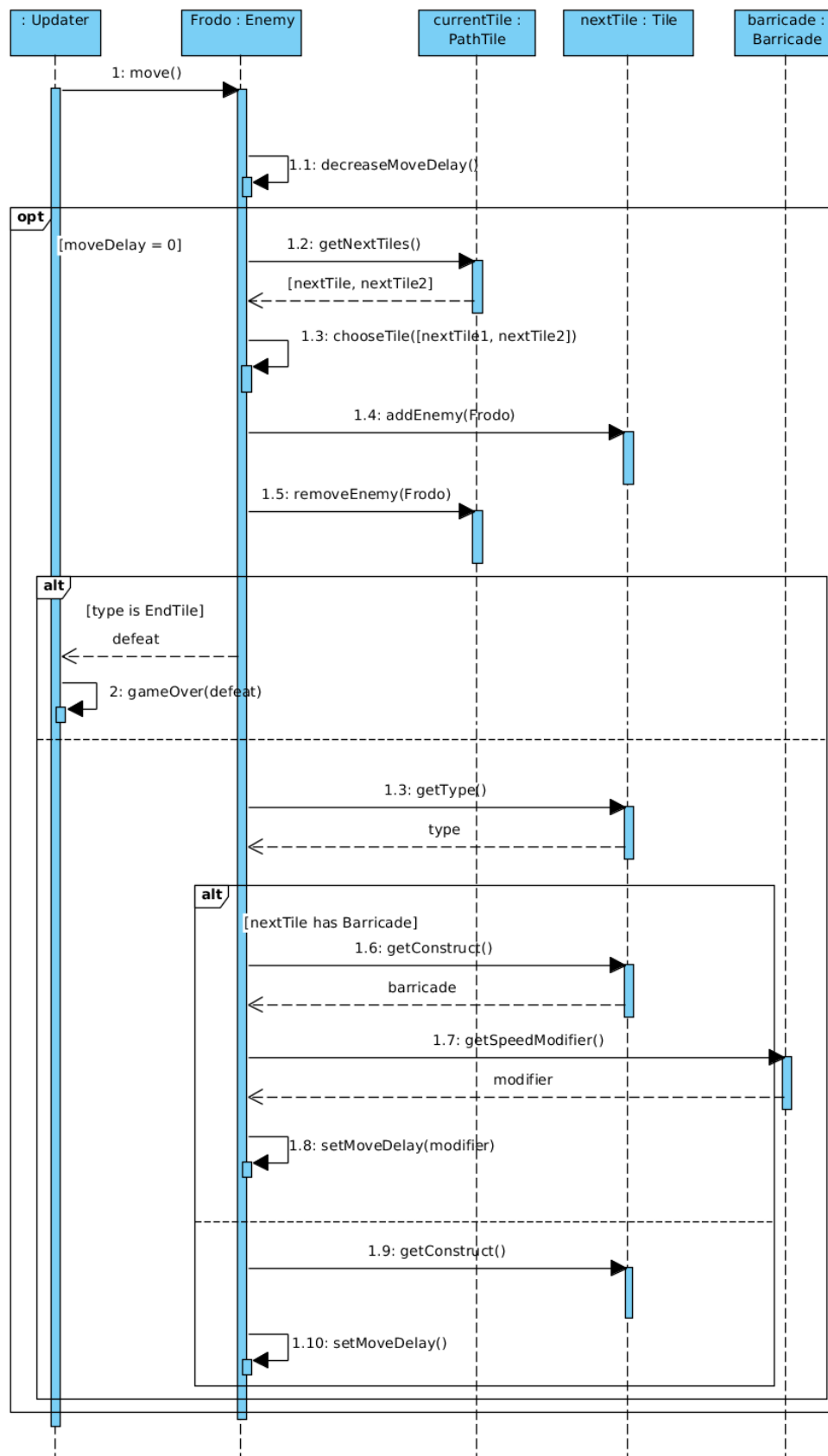


1. Kód leszáll:



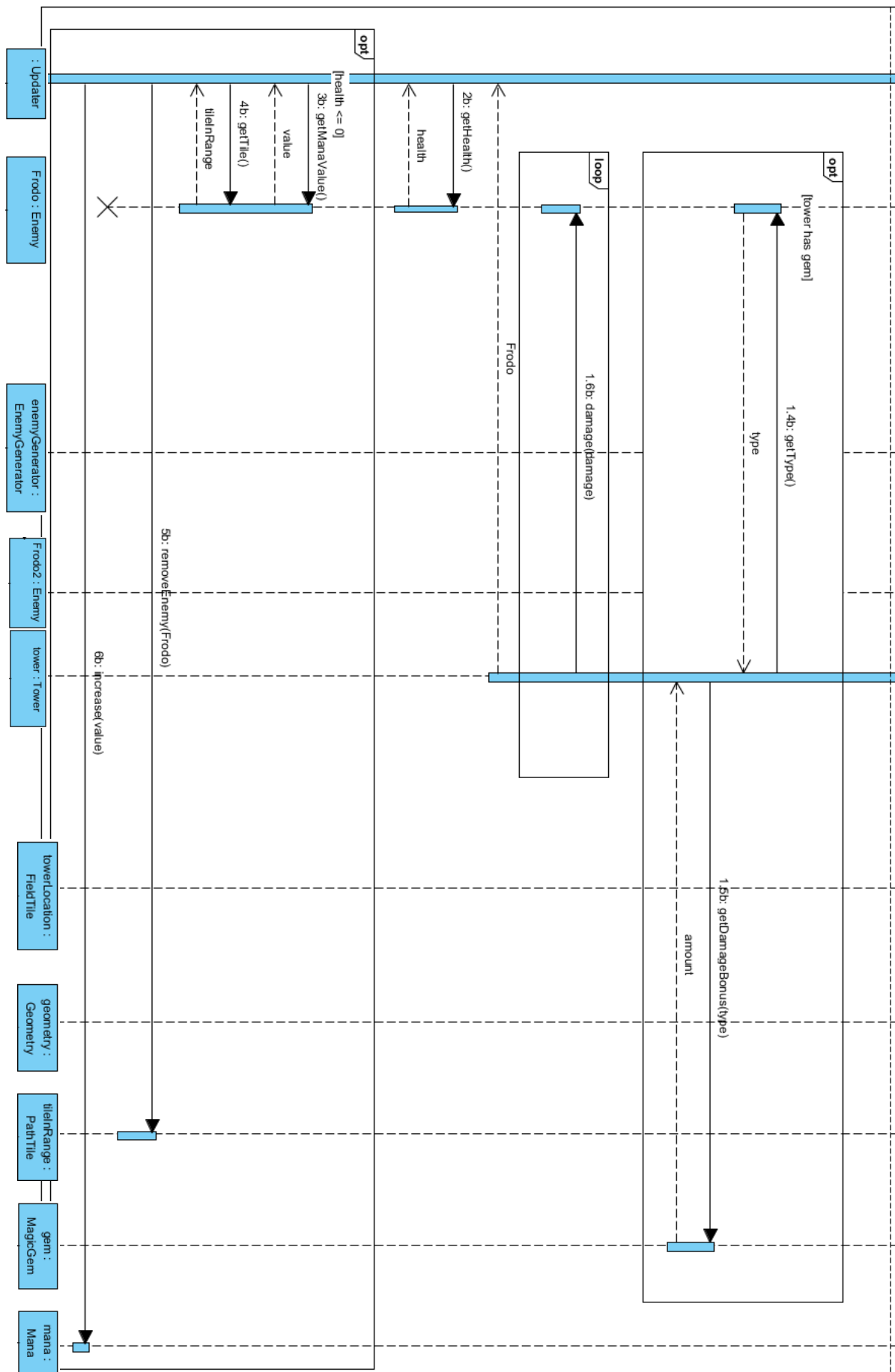
2. Kód felszáll:

3. Lépés (kiegészítve az elágazásokkal):



2014-05-14





8.1 Prototípus interface-definíciója

A prototípus program célja az, hogy az eddig ismertetett tervek szerint, és jól működik a program. A szoftver még nem tartalmaz grafikus felületet, és a végleges irányítás sem szerepel benne. A főbb objektumok már a végleges algoritmusokat implementálják.

Mivel a prototípus nem tartalmaz grafikus felületet, így kezelése konzolon keresztül történik karakteres formában. A beviteli – és kimeneti nyelv előre definiált, és a dokumentum további részében ismertetésre is kerül. A teszteléshez készültek előre megírt tesztesetek, amelyekhez a parancsokat fájlokból tudja kinyerni a program. Azonban ha felhasználó tudja milyen kimenetet vár az adott bementre, ön maga is tesztelheti. Az előre megírt tesztesetekkel szemben támasztott fontos követelmény, hogy a program összes kritikus pontját lefedjék, és akárhányszor lefuttatva őket konzisztensen ugyanazt az eredményt adják. A reprodukálhatóság alapfeltétele pedig az, hogy kivesszük a játékból a véletlen faktort. Bár ez teszi a játékot élvezetessé, a tesztelés szempontjából elengedhetetlen az, hogy mindig a várt eredményt kapjuk.

8.1.1 Az interfész általános leírása

A felhasználóval a prototípus program a szabványos ki - és bementen kommunikál. A bemeneti és a kimeneti nyelv előre definiálva van a könnyebb kezelhetőség céljából. Parancsokat a szabványos be- és kimenetről kaphatja, vagy az előre elkészített parancsfájlból.

A pálya előre generált, és mindig a program futása előtt fájlból töltődik be. Amennyiben a szabványos bementen adtuk meg a parancsokat, úgy a szabványos kimeneten is kapjuk meg hozzá minden esetben a választ. Ha fájlból töltjük be a parancsokat az előre definiált tesztesethez, akkor fájlba is mentjük a futás eredményét.

8.1.2 Bemeneti nyelv

loadCommandFile <filename>

Végrehajtja a paraméterben kapott fájlban (txt) kapott utasításokat.

-*filename*: a bemeneti fájl elérési útvonala és neve.

beginWriteCommands <filename>

Fájlba menti az ez után kiadott parancsokat, egészen az *endWriteCommands* kulcsszóig.

-*filename*: a bemeneti fájl elérési útvonala és neve.

help

Kiírja az összes parancsot.

exit

Kilép a prototípusból.

simulate <count>

Lefuttat *count* számú játékciklust.
-*count*: a futtatandó ciklusok száma

setMana <value>

Beállítja a mana mennyiségét a megadott értékre.
-*value*: a kívánt mana mennyiség

Build <tileID> <type> <costsMana>

Hozzáadja a megadott típusú épületet a megadott azonosítójú cellához.
Megadja, hogy milyen sorszámmal érhető el ezután a torony.
-*tileID*: a csempe azonosítója, melyre építünk
-*type*: az épület típusa (tower vagy barricade)
-*costsMana*: kapcsoló, hogy az építés kerüljön-e varázserőbe. 1 esetén kerül varázserőbe, 0 esetén nem

Upgrade <constructID> <gemType> <costsMana>

Behelyezi a megadott típusú varázskövet a megadott azonosítójú épületbe.
-*constructID*: az épület azonosítója, melyet varázskővel látunk el
-*gemType*: a varázskő típusa
-*costsMana*: kapcsoló, hogy az építés kerüljön-e varázserőbe. 1 esetén kerül varázserőbe, 0 esetén nem

Shoot <tower> <critical>

A megnevezett torony lő. A *critical* paraméter dönti el, hogy a lövés felező lövés-e.
-*tower*: a torony azonosítója, melyet lövésre utasítunk
-*critical*: A lövés fajtáját adja meg. Ha '0', akkor nem felező, ha '1', akkor felező.
Hiányzó vagy ettől eltérő érték esetén véletlenszerűen dől el.

setFog <state>

Be- vagy kikapcsolja a ködöt.
-*state*: a köd állapota, 0: ki, 1:be

createMap <x> <y>

Létrehoz egy (csempékben) megadott méretű, üres, téglalap alapú pályát. $x*y$ csempe fér el egy pályán.
-*x*: a pálya vízszintes mérete
-*y*: a pálya függőleges mérete

addFieldTile <x> <y>

Lehelyez egy "fieldTile" típusú cellát a pálya megadott helyére.
-*x*: a pálya vízszintes koordinátája
-*y*: a pálya függőleges koordinátája

addPathTile <x> <y>

Lehelyez egy "pathTile" típusú cellát a pálya megadott helyére.
-*x*: a pálya vízszintes koordinátája
-*y*: a pálya függőleges koordinátája

addEndTile <x> <y>

Lehelyez egy "endTile" típusú cellát a pálya megadott helyére.

-x: a pálya vízszintes koordinátája

-y: a pálya függőleges koordinátája

addNextTile <sourceTileID> <destTileID>

Beállítja a megadott azonosítójú útvonal csempe után következő csempét. Többszöri hívás ugyanarra a csempére egy elágazást eredményez.

-sourceTileID: A kiinduló csempe azonosítója

-destTileID: A cél csempe azonosítója

setStartTile <tileID>

Beállítja a megadott azonosítójú csempét kezdőcsempének, azaz a generateEnemies parancs erre a csempére le fogja tudni tenni az ellenséget.

-tileID: A csempe azonosítója, melyre a generateEnemies ellenséget tehet le

getStatus

Kilistázza a játék összes objektumának állapotát. Ez egy teljes képet ad a program jelenlegi állásáról.

generateEnemies

Lefuttatja az "enemyGenerator" generáló függvényét a beállításainak megfelelően.

addEnemy <tileID> <type>

Hozzáad a játékhoz egy adott csempére egy adott fajtájú ellenséget.

-tileID: A csempe azonosítója, melyre az ellenséget tesszük le

-type: Létrehozott ellenség típusa

setRemainingEnemies <value>

Beállítja a még játék végéig legenerálandó ellenségek számát.

-value: a hátralévő ellensége száma

move <enemyID>

Meghívja az adott ellenségen a mozgás függvényét.

-enemyID: a mozgó ellenség azonosítója

generatePaths

Beállítja a pályán az útvonalakat.

8.1.3 Kimeneti nyelv

loadCommandFile <filename>

Kiírja a betöltés eredményét.

Lehetséges kimenetek:

1. <filename> betöltése sikeres
2. <filename> betöltése sikertelen

beginWriteCommands <filename>

Kiírja, hogy a fájlba írás megkezdődött, valamint jelzi, ha véget ért.

1. Fájlba írás megkezdődött.
2. Fájlba írás véget ért.
3. Hiba történt a fájlba íráskor!

help

Kiírja az összes elérhető parancsot, soronként egyet, utánuk a paramétereikkel.

pl: loadCommandFile <filename>

exit

Nincs kimenete.

simulate <count>

Kiírja, ha végzett a szimuláció a futással, valamint közli a lefuttatott ciklusok számát.

Kiírja, hogy melyik ciklusban melyik ellenségek lettek legenerálva, hogy melyik ciklusban melyik ellenségek honnan és hova léptek, illetve kiírja, hogy melyik torony melyik ellenségre lőtt, és az meghalt-e. Ezeket az információkat egy-egy sorba írja (időrendben), a szimuláció véget értenek és lefutott ciklusok számának jelzése után. Jelzi, ha a szimuláció közben a játék véget ért valamilyen eredménnyel és hogy ez hányadik ciklusban következett be.

Példa kimenetek:

1. A szimuláció véget ért. <count> ciklus futott le!
2. A szimuláció véget ért. <count> ciklus futott le!
 - A(z) 1. ciklusban generálódott az E1 azonosítójú ellenség.
 - A(z) 5. ciklusban generálódott az E2 azonosítójú ellenség.
3. A szimuláció véget ért. <count> ciklus futott le!
 - A(z) 3. ciklusban lépett az E5 azonosítójú ellenség a T1 celláról a T2-re.
 - A 3. ciklusban az E5 azonosítójú ellenség a végzet hegyére lépett, vesztettél!
4. A szimuláció véget ért. <count> ciklus futott le!
 - A(z) 1. ciklusban generálódott az E4 azonosítójú ellenség.
 - A(z) 3. ciklusban lőtt a T1 azonosítójú torony az E4 ellenségre, levéve tőle 15 életerőt.
 - A(z) 3. ciklusban lőtt a T2 azonosítójú torony az E4 ellenségre, mely meghalt, így a varázserőd 6-tal nőtt.
 - Az utolsó ellenség is meghalt a 2. ciklusban, nyertél!

setMana <value>

Lehetséges kimenete:

A varázserő sikeresen beállítva <value> értékre!

Build <tileID> <type> <costsMana>

Lehetséges kimenetek:

1. <tileID> csempére <type> épült.
2. <tileID> csempére <type> épült. 20 varázserőbe került.
3. Nincs elég varázserőd <type> építésére!
4. A megadott csempére nem helyezhető el az épület.

Upgrade <constructID> <gemType> <costsMana>

Lehetséges kimenetek:

1. <constructID> épületbe <gemType> varázskövet tettél.
2. <constructID> épületbe <gemType> varázskövet tettél. 10 varázserőbe került.
3. Nincs elég varázserőd <gemType> vásárlására!
4. A <gemType> varázskő nem helyezhető a <constructID> azonosítójú épületbe.
5. A megadott épület nem létezik.

Shoot <tower> <critical>

Lehetséges kimenetek:

1. <tower> nem lőtt, mert nem talált ellenséget.
2. <tower> lőtt, E1 azonosítójú ellenségre, levéve tőle 20 életerőt.
3. <tower> lőtt, E1 azonosítójú ellenségre, felező lövéssel, levéve tőle 30 életerőt.
Létrejött E4 ellenség (a T1 csempén, 20 életerővel. **[kell ez?]**)
4. A megadott torony nem létezik.

setFog <state>

Lehetséges kimenetek:

1. Kód bekapcsolva. Az összes torony hatótávja 0.7-szeresére csökkent.
2. Kód kikapcsolva. Az összes torony hatótávja eredeti értékre állt vissza.

createMap <x> <y>

Lehetséges kimenetek:

Sikeresen létrejött az <x>, <y> méretű pálya!

addFieldTile <x> <y>

Lehetséges kimenetek:

1. Sikeresen hozzáadtad a T1 azonosítójú terep csempét az <x>, <y> helyre.
2. Nincs ilyen hely a pályán!

addPathTile <x> <y>

Lehetséges kimenetek:

1. Sikeresen hozzáadtad a T1 azonosítójú út csempét az <x>, <y> helyre.
2. Nincs ilyen hely a pályán!

addEndTile <x> <y>

Lehetséges kimenetek:

1. Sikeresen hozzáadtad a T1 azonosítójú végzet hegye csempét az <x>, <y> helyre.
2. Már van egy végzet hegye csempe, T4 azonosítóval a 2, 3 helyen.
3. Nincs ilyen hely a pályán!

addNextTile <sourceTileID> <destTileID>

Lehetséges kimenete:

Hozzáadtad a <sourceTileID> azonosítójú útcsempéből elérhető útcsempékhez a <destTileID> csempét.

setStartTile <tileID>

Beállítja a megadott azonosítójú csempét kezdőcsempének, azaz a generateEnemies parancs erre a csempére le fogja tudni tenni az ellenséget.

-tileID: A csempe azonosítója, melyre a generateEnemies ellenséget tehet le

getStatus

Kilistázza először a pályát, mérettel együtt. Ezután a csempeazonosítókat típusukkal és helyükkel, illetve útcsempe esetén a vele szomszédos csempéket és, hogy generálható-e rá ellenség. Újabb listába teszi az összes épületazonosítót az épület típusával, benne lévő varázskővel és értékeivel együtt. Ezekután kilistázza az összes ellenségazonosítót, az ellenségek típusát, és értékeit. Végül közöl minden egyéb információt a felhasználóval.

Lehetséges kimenete:

Pálya: 5x5

Csempék:

T1 Típus: path	x:3	y:4	SzomszédCsempék: T3 Start: true
T2 Típus: field	x:4	y:4	
T3 Típus: path	x:5	y:5	SzomszédCsempék: Start: false

Épületek:

C1 Típus: tower Hely: T2 Kő: range Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5
HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus

C2 Típus: barricade Hely: T1 Kő: LassításMérték: 3

Ellenségek:

E1: Típus: elf Hely: T1 Életerő: 20 Sebesség: 1 KövetkezőLépés: 1 ciklus
Érték: 5

E2: Típus: dwarf Hely: T3 Életerő: 8 Sebesség: 4 KövetkezőLépés: 2 ciklus
Érték: 8

Varázserő: 172

Legenerálható ellenségek: 15

Legközelebbi generálás: 2 ciklus

Kód: nem

sgenerateEnemies

Lehetséges kimenetek:

1. <EnemyCount> számú ellenség generálódott!
2. Nem lehet ellenséget generálni!

addEnemy <tileID> <type>

Lehetséges kimenetek:

1. Sikeresen létrehozta az E1 azonosítójú "elf" típusú ellenséget a T1 csempén.
2. <tileID> nem utcsempé!
3. Nincs ilyen típusú ellenség!

setRemainingEnemies <value>

Lehetséges kimenete:

<value> számú ellenség vár még a legenerálásra!

move <enemyID>

Lehetséges kimenetek:

1. <enemyID> moveDelay értéke egyel csökkent.
2. <enemyID> lépett T1-ről T2-re. Legközelebb 5 szimulációs ciklus után lépne.
3. <enemyID> lépett T1-ről T2-re. T2-n akadály van, lelassult. Legközelebb 10 szimulációs ciklus után lépne.
4. <enemyID> nem tudott lépni.

generatePaths

Lehetséges kimenetek:

1. Nem sikerült létrehozni az útvonalakat
2. Az útvonalak létrehozása sikeres. Kezdőcsempéként használható utcsempék a T2, T4.

8.2 Összes részletes use-case

Use-case neve	Játék indítása
Rövid leírás	A felhasználó elindítja játékot.
Aktorok	Felhasználó
Forgatókönyv	A felhasználó elindítja játékot és a pálya betöltődik.

Use-case neve	Torony építés
Rövid leírás	A felhasználó tornyot épít.
Aktorok	Felhasználó
Forgatókönyv	A felhasználó egy kiválasztott terep csempére (FieldTile) egy tornyot épít, ha van elég varázsereje.

Use-case neve	Akadály építés
Rövid leírás	A felhasználó akadályt épít
Aktorok	Felhasználó
Forgatókönyv	A felhasználó egy kiválasztott út csempére (PathTile) egy akadályt épít, ha van elég varázsereje.

Use-case neve	Torony fejlesztése
Rövid leírás	A felhasználó egy tornyot fejleszt.
Aktorok	Felhasználó

Forgatókönyv	A felhasználó egy kiválasztott terep csempén (FieldTile) található tornyot fejleszt, ha van elég varázsereje.
---------------------	---

Use-case neve	Akadály fejlesztése
Rövid leírás	A felhasználó egy akadályt fejleszt.
Aktorok	Felhasználó
Forgatókönyv	A felhasználó egy kiválasztott út csempén (PathTile) található akadályt fejleszt, ha van elég varázsereje.

Use-case neve	Torony lövés
Rövid leírás	Torony lö
Aktorok	Felhasználó
Forgatókönyv	A torony a hatósugarában lévő ellenségek valamelyikére rálő.

Use-case neve	Ellenség lép
Rövid leírás	Az ellenség lép
Aktorok	Felhasználó
Forgatókönyv	Az ellenség a következő csempék valamelyikére rálép. Ha ez a csempe a Végzet Hegye a játék verességgel végződik.

Use-case neve	Akadály lassít
Rövid leírás	Akadály lassítja az ellenséget.
Aktorok	Felhasználó
Forgatókönyv	Az ellenség a következő csempén lévő akadályra lép és lelassul.

Use-case neve	Köd leereszkedik
Rövid leírás	Köd ereszkedik a pályára
Aktorok	Felhasználó
Forgatókönyv	Köd ereszkedik a pályára és a pályán lévő összes torony hatótávolsága egy bizonyos időre lecsökken

Use-case neve	Köd felszáll
Rövid leírás	Köd eltűnik a pályáról
Aktorok	Felhasználó
Forgatókönyv	A köd felszáll. A pályán lévő összes torony hatótávolsága a köd előtti értékre áll vissza.
Use-case neve	Ellenség meghal
Rövid leírás	Az ellenség életereje elfogy.
Aktorok	Felhasználó
Forgatókönyv	Az ellenséget meglövik, ezáltal az életereje elfogy és meghal, ezáltal a mana értéke megnő. Ha nincs több ellenség akkor a játék győzelemmel ér véget.

Use-case neve	Ellenség osztódik
Rövid leírás	Az ellenség osztódik
Aktorok	Felhasználó
Forgatókönyv	Az ellenség torony lövésének hatására osztódik. Az így létrejött ellenség és az eredeti ellenség csökkentett életerővel tovább él.

8.3 Tesztelési terv

Teszt-eset neve	Pálya betöltése
Rövid leírás	A pályát betöltjük egy adott fájlból.
Teszt célja	Leellenőrizzük, hogy a pálya és a rajta található fix elemek betöltődnek-e.

Teszt-eset neve	Torony építése
Rövid leírás	Tornyot szeretnénk építeni.
Teszt célja	A torony építés folyamatának ellenőrzése. A ConstructManager osztály build függvényének ellenőrzése torony esetén

Teszt-eset neve	Akadály építése
Rövid leírás	Az akadály épülését akarjuk.
Teszt célja	Akadály építés folyamatának ellenőrzése. A ConstructManager osztály build függvényének ellenőrzése akadály esetén

Teszt-eset neve	Torony fejlesztése
Rövid leírás	A torony felruházása varázskővel
Teszt célja	Az ConstructManager osztály upgrade függvényének ellenőrzése torony esetén.

Teszt-eset neve	Torony lő
Rövid leírás	A torony a hatótávolságán belül tartózkodó ellenségek valamelyikét kiválasztja és rálő.
Teszt célja	A Geometry osztály jól adja-e vissza a torony hatótávolságában lévő csempéket. A torony a megkapott csempékről kitudja-e választani egy ellenséget, amelyet meg is lő.

Teszt-eset neve	Akadály lassít
Rövid leírás	Az ellenség rálép egy út csempére (PathTile) amelyen akadály van, így az ellenség lelassul.
Teszt célja	Az akadály lassításának ellenőrzése.

Teszt-eset neve	Köd
Rövid leírás	Köd ereszkedik le a pályára majd egy bizonyos idő múlva felszáll.
Teszt célja	Köd leereszkedésének ellenőrzése. A tornyok hatótávolság csökkenésének tesztelése. Továbbá a köd felszállása esetén a megfelelő értékre áll-e vissza a tornyok hatótávolsága.

Teszt-eset neve	Ellenség halála
Rövid leírás	Az ellenség egy torony lövés miatt meghal
Teszt célja	Az ellenség halálának vizsgálata. A varázserő változás és a játék véget értének ellenőrzése.

Teszt-eset neve	Ellenség osztódik
Rövid leírás	Ellenség lövés hatására ketté válik
Teszt célja	Az ellenség osztódásának ellenőrzése, hogy az új ellenség jó helyre kerül-e, és az életereje kevesebb-e.

8.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

Az előre definiált tesztesetekhez tartozó parancsok sorozatánál minden esetben tudnunk kell előre pontosan sorról sorra, hogy milyen kimenetet kaptunk. A fájl kimenetét pedig fájlba írjuk szintén előre definiált formában. Így minden előre definiált tesztesethez készítünk egy fájlt, amely már a teszteset lefutása előtt, az előre definiált formában tartalmazza a kimenetet. Ezt a fájlt, és a tesztesetkor készített fájl könnyen összehasonlíthatjuk, akár 'kézi' módszerrel, de valamilyen egyszerű programmal is, amely két fájl tartalmát tudja összehasonlítani.

9. Részletes tervek

9.0 Módosított ki- és bemeneti nyelv

9.0.1 Bemeneti nyelv

loadCommandFile <filename>

Végrehajtja a paraméterben kapott fájlban (txt) kapott utasításokat.

-*filename*: a bemeneti fájl elérési útvonala és neve.

beginWriteCommands <filename>

Fájlba menti az ez után kiadott parancsokat, egészen az *endWriteCommands* kulcsszóig.

-*filename*: a bemeneti fájl elérési útvonala és neve.

help

Kiírja az összes parancsot.

exit

Kilép a prototípusból.

simulate <count>

Lefuttat *count* számú játékciklust.

-*count*: a futtatandó ciklusok száma

setMana <value>

Beállítja a mana mennyiségét a megadott értékre.

-*value*: a kívánt mana mennyiség

Build <tileID> <type> <costsMana>

Hozzáadja a megadott típusú épületet a megadott azonosítójú cellához.

Megadja, hogy milyen sorszámmal érhető el ezután a torony.

-*tileID*: a csempe azonosítója, melyre építünk

-*type*: az épület típusa (tower vagy barricade)

-*costsMana*: kapcsoló, hogy az építés kerüljön-e varázserőbe. 1 esetén kerül varázserőbe, 0 esetén nem

Upgrade <constructID> <gemType> <costsMana>

Behelyezi a megadott típusú varázskövet a megadott azonosítójú épületbe.

-*constructID*: az épület azonosítója, melyet varázskővel látunk el

-*gemType*: a varázskő típusa

-*costsMana*: kapcsoló, hogy az építés kerüljön-e varázserőbe. 1 esetén kerül varázserőbe, 0 esetén nem

Shoot <tower> <critical>

A megnevezett torony lő. A *critical* paraméter dönti el, hogy a lövés felező lövés-e.

-tower: a torony azonosítója, melyet lövésre utasítunk

-critical: A lövés fajtáját adja meg. Ha '0', akkor nem felező, ha '1', akkor felező.

Hiányzó vagy ettől eltérő érték esetén véletlenszerűen dől el.

setFog <state>

Be- vagy kikapcsolja a ködöt.

-state: a köd állapota, 0: ki, 1:be

createMap <x> <y>

Létrehoz egy (csempékben) megadott méretű, üres, téglalap alapú pályát. $x*y$ csempe fér el egy pályán.

-x: a pálya vízszintes mérete

-y: a pálya függőleges mérete

addFieldTile <x> <y>

Lehelyez egy "fieldTile" típusú cellát a pálya megadott helyére.

-x: a pálya vízszintes koordinátája

-y: a pálya függőleges koordinátája

addPathTile <x> <y>

Lehelyez egy "pathTile" típusú cellát a pálya megadott helyére.

-x: a pálya vízszintes koordinátája

-y: a pálya függőleges koordinátája

addEndTile <x> <y>

Lehelyez egy "endTile" típusú cellát a pálya megadott helyére.

-x: a pálya vízszintes koordinátája

-y: a pálya függőleges koordinátája

addNextTile <sourceTileID> <destTileID>

Beállítja a megadott azonosítójú útvonal csempe után következő csempét. Többszöri hívás ugyanarra a csempére egy elágazást eredményez.

-sourceTileID: A kiinduló csempe azonosítója

-destTileID: A cél csempe azonosítója

setStartTile <tileID>

Beállítja a megadott azonosítójú csempét kezdőcsempének, azaz a generateEnemies parancs erre a csempére le fogja tudni tenni az ellenséget.

-tileID: A csempe azonosítója, melyre a generateEnemies ellenséget tehet le

getStatus

Kilistázza a játék összes objektumának állapotát. Ez egy teljes képet ad a program jelenlegi állásáról.

addEnemy <tileID> <type>

Hozzáad a játékhoz egy adott csempére egy adott fajtájú ellenséget.

-*tileID*: A csempe azonosítója, melyre az ellenséget tesszük le

-*type*: Létrehozott ellenség típusa

setRemainingEnemies <value>

Beállítja a még játék végéig legenerálandó ellenségek számát.

-*value*: a hátralévő ellensége száma

move <enemyID>

Meghívja az adott ellenségen a mozgás függvényét.

-*enemyID*: a mozgó ellenség azonosítója

generatePaths

Beállítja a pályán az útvonalakat.

9.0.2 Kimeneti nyelv

loadCommandFile <filename>

Kiírja a betöltés eredményét.

Lehetséges kimenetek:

1. <filename> betöltése sikeres
2. <filename> betöltése sikertelen

beginWriteCommands <filename>

Kiírja, hogy a fájlba írás megkezdődött, valamint jelzi, ha véget ért.

1. Fájlba írás megkezdődött.
2. Fájlba írás véget ért.
3. Hiba történt a fájlba íráskor!

help

Kiírja az összes elérhető parancsot, soronként egyet, utánuk a paramétereikkel.

pl: loadCommandFile <filename>

exit

Nincs kimenete.

simulate <count>

Kiírja, ha végzett a szimuláció a futással, valamint közli a lefuttatott ciklusok számát.

Kiírja, hogy melyik ciklusban melyik ellenségek lettek legenerálva, hogy melyik ciklusban melyik ellenségek honnan és hova léptek, illetve kiírja, hogy melyik torony melyik ellenségre lőtt, és az meghalt-e. Ezeket az információkat egy-egy sorba írja (időrendben), a szimuláció véget értének és lefuttatott ciklusok számának jelzése után.

Jelzi, ha a szimuláció közben a játék véget ért valamilyen eredménnyel és hogy ez hányadik ciklusban következett be.

Példa kimenetek:

1. A szimuláció véget ért. <count> ciklus futott le!
2. A szimuláció véget ért. <count> ciklus futott le!
 - A(z) 1. ciklusban generálódott az E1 azonosítójú ellenség.
 - A(z) 5. ciklusban generálódott az E2 azonosítójú ellenség.
3. A szimuláció véget ért. <count> ciklus futott le!
 - A(z) 3. ciklusban lépett az E5 azonosítójú ellenség a T1 celláról a T2-re.
 - A 3. ciklusban az E5 azonosítójú ellenség a végzet hegyére lépett, vesztettél!
4. A szimuláció véget ért. <count> ciklus futott le!
 - A(z) 1. ciklusban generálódott az E4 azonosítójú ellenség.
 - A(z) 3. ciklusban lőtt a T1 azonosítójú torony az E4 ellenségre, levéve tőle 15 életerőt.
 - A(z) 3. ciklusban lőtt a T2 azonosítójú torony az E4 ellenségre, mely meghalt, így a varázserőd 6-tal nőtt.
 - Az utolsó ellenség is meghalt a 2. ciklusban, nyertél!

setMana <value>

Lehetséges kimenete:

A varázserő sikeresen beállítva <value> értékre!

Build <tileID> <type> <costsMana>

Lehetséges kimenetek:

1. <tileID> csempére <type> épült.
2. <tileID> csempére <type> épült. 20 varázserőbe került.
3. Nincs elég varázserőd <type> építésére!
 4. A megadott csempére nem helyezhető el az épület.

Upgrade <constructID> <gemType> <costsMana>

Lehetséges kimenetek:

1. <constructID> épületbe <gemType> varázskövet tettél.
2. <constructID> épületbe <gemType> varázskövet tettél. 10 varázserőbe került.
3. Nincs elég varázserőd <gemType> vásárlására!
4. A <gemType> varázskő nem helyezhető a <constructID> azonosítójú épületbe.
5. A megadott épület nem létezik.

Shoot <tower> <critical>

Lehetséges kimentek:

1. <tower> nem lőtt, mert nem talált ellenséget.
2. <tower> lőtt, E1 azonosítójú ellenségre, levéve tőle 20 életerőt.
3. <tower> lőtt, E1 azonosítójú ellenségre, felező lövéssel, levéve tőle 30 életerőt. Létrejött E4 ellenség.
4. A megadott torony nem létezik.

setFog <state>

Lehetséges kimenetek:

1. Köd bekapcsolva. Az összes torony hatótávja 0.7-szeresére csökkent.
2. Köd kikapcsolva. Az összes torony hatótávja eredeti értékre állt vissza.

createMap <x> <y>

Lehetséges kimenetek:

Sikeresen létrejött az <x>, <y> méretű pálya!

addFieldTile <x> <y>

Lehetséges kimenetek:

1. Sikeresen hozzáadtad a T1 azonosítójú terep csempét az <x>, <y> helyre.
2. Nincs ilyen hely a pályán!

addPathTile <x> <y>

Lehetséges kimenetek:

1. Sikeresen hozzáadtad a T1 azonosítójú út csempét az <x>, <y> helyre.
2. Nincs ilyen hely a pályán!

addEndTile <x> <y>

Lehetséges kimenetek:

1. Sikeresen hozzáadtad a T1 azonosítójú végzet hegye csempét az <x>, <y> helyre.
2. Már van egy végzet hegye csempe, T4 azonosítóval a 2, 3 helyen.
3. Nincs ilyen hely a pályán!

addNextTile <sourceTileID> <destTileID>

Lehetséges kimenete:

Hozzáadtad a <sourceTileID> azonosítójú útcsempéből elérhető útcsempékhez a <destTileID> csempét.

setStartTile <tileID>

Lehetséges kimenetek:

1. Sikeresen beállítottad a <tileID> azonosítójú útcsempét kezdőcsempének.
2. A megadott csempe nem útcsempe!

getStatus

Kilistázza először a pályát, mérettel együtt. Ezután a csempeazonosítókat típusukkal és helyükkel, illetve útcsempe esetén a vele szomszédos csempéket és, hogy generálható-e rá ellenség. Újabb listába teszi az összes épületazonosítót az épület típusával, benne lévő varázskövel és értékeivel együtt. Ezekután kilistázza az

összes ellenségazonosítót, az ellenségek típusát, és értékeit. Végül közöl minden egyéb információt a felhasználóval.

Lehetséges kimenete:

Pálya: 5x5

Csempék:

T1 Típus: path	x:3	y:4	SzomszédCsempék: T3 Start: true
T2 Típus: field	x:4	y:4	
T3 Típus: path	x:5	y:5	SzomszédCsempék: Start: false

Épületek:

C1 Típus: tower Hely: T2 Kő: range Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5
HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus

C2 Típus: barricade Hely: T1 Kő: LassításMérték: 3

Ellenségek:

E1: Típus: elf Hely: T1 Életerő: 20 Sebesség: 1 KövetkezőLépés: 1 ciklus
Érték: 5

E2: Típus: dwarf Hely: T3 Életerő: 8 Sebesség: 4 KövetkezőLépés: 2 ciklus
Érték: 8

Varázserő: 172

Legenerálható ellenségek: 15

Legközelebbi generálás: 2 ciklus

Kód: nem

addEnemy <tileID> <type>

Lehetséges kimenetek:

1. Sikeresen létrehoztad az E1 azonosítójú "elf" típusú ellenséget a T1 csempén.
2. <tileID> nem utcsempé!
3. Nincs ilyen típusú ellenség!

setRemainingEnemies <value>

Lehetséges kimenete:

<value> számú ellenség vár még a legenerálásra!

move <enemyID>

Lehetséges kimenetek:

1. <enemyID> moveDelay értéke egyel csökkent.
2. <enemyID> lépett T1-ről T2-re. Legközelebb 5 szimulációs ciklus után lépne.
3. <enemyID> lépett T1-ről T2-re. T2-n akadály van, lelassult. Legközelebb 10 szimulációs ciklus után lépne.
4. <enemyID> nem tudott lépni.

generatePaths

Lehetséges kimenetek:

1. Nem sikerült létrehozni az útvonalakat
2. Az útvonalak létrehozása sikeres. Kezdőcsempéként használható utcsempék a T2, T4.

9.1 Osztályok és metódusok tervei

9.1.1 Barricade

- **Felelősség**
 - adott értékkel lassítja a csempéjén áthaladó ellenségeket
 - varázskövet lehet beletenni
- **Ősosztályok**
Object ->Construct
- **Interfészek**
nincs
- **Attribútumok**
 - - **int speedModifier**: a lassítás mértéke
 - - **PathTile barricadeLocation**: az akadály által elfoglalt csempe
- **Metódusok**
 - + **int getSpeedModifier()**: visszatér a lassítás mértékével

9.1.2 Construct

- **Felelősség**
 - ezzel az absztrakt osztállyal lehetséges a Tower-ök és Barricade-ek egy tömbben tárolása.
- **Ősosztályok**
Object
- **Interfészek**
nincs
- **Attribútumok**
 - # **MagicGem gem**: a constructban található varázskő
 - - **String type**: a construct típusa, például torony vagy akadály
- **Metódusok**
 - + **String getType()**: az épület típusával tér vissza
 - + **void setMagicGem(MagicGem magicgem)**: beállítja a gem attribútumot a kapott értékre

9.1.3 ConstructManager

- **Felelősség**

- létrehozza a különböző épületeket
- létrehoz varázsköveket, melyeket bele tesz épületekbe
- megkérdezi, hogy van-e elég mana az építéshez/fejlesztéshez, és levonja a varázserőt az építés/fejlesztés után

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- - **String[] types**: egyes épület és varázskő típusokat tárolja
- - **int[] costs**: egyes épület és varázskő típusokhoz tartozó árakat tárolja
- - **Mana mana**: referencia a varázserőre
- - **Updater updater**: referencia az időkezelőre

- **Metódusok**

- **+ void build(String type, Tile location)**: type alapján létrehozza egy épület típus példányát. Megkérdezi a Mana-t, hogy van-e elég az építéshez, ha igen levonja, és létrehozza a Constructot. Ezek után beregisztrálja a kapott Tile-ra.
- **+ void upgrade(String type, Construct location)**: type alapján tesz az adott tile-on lévő constructba varázskövet. Megkérdezi a Mana-t, hogy van-e elég, ha igen levonja, és utána hozza létre a varázskövet. Beállítja a type-nak megfelelő attribútumát a constructnak.

9.1.4 Dwarf

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

- - **String type**: az osztály típusát jelzi tehát az hogy, dwarf

- **Metódusok**

csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül

9.1.5 Elf

- **Felelőssége**

- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

- - **String type**: az osztály típusát jelzi tehát az, hogy elf

- **Metódusok**

csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül

9.1.6 EndTile

- **Felelősség**

- ha rálép valaki veszítünk
- nem mehetnek róla tovább ellenségek

- **Ősosztályok**

Object->Tile

- **Interfészek**

nincs

- **Attribútumok**

- - **Enemy[] enemies**: rajta tartózkodó ellenségeket tárolja

- **Metódusok**

csak öröklött metódusok, melyeket nem definiál felül

9.1.7 Enemy

- **Felelősség**

- ezzel az absztrakt osztállyal lehetséges a különböző ellenségtípusok egy tömbben tárolása.
- le tudja vonni saját életerejét
- meg tudja adni az ő varázserőben vett értékét
- tudja hogy lépet-e már
- tudja melyik csempén áll

- **Össosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **# int health:** az ellenség életerejét jelenti
- **# int speed:** a lépes gyakoriságát mutatja
- **# int manaValue:** az a mana értéket jelenti melyet a megölésért kap a játékos
- **# int moveDelay:** megmondja mikor léphet az ellenség legközelebb
- **# String type:** az ellenség típusát adja meg mint például hobbit vagy dwarf
- **# Tile currentTile:** a csempe melyen az ellenség jelenleg tartózkodik
- **# EnemyGenerator enemyGenerator:** az ellenségeket generáló objektum

- **Metódusok**

- **+ void damage(int value):** a value értékével csökkenti az életerejét
- **+ void decreaseMoveDelay():** csökkenti a moveDelay attribútum értékét egyel
- **+ int getHealth():** az ellenség életerejével tér vissza
- **+ int getManaValue():** visszaadja, hogy mennyi manát ér a megölése
- **+ Tile getTile():** visszaadja a csempét melyen az ellenség jelenleg tartózkodik
- **+ String getType():** az ellenség típusával tér vissza
- **+ boolean move():** lépteti az ellenséget a currentTile-ről a currentTile nextTile-jára. Megnézi, hogy a végzet hegyére léptünk-e, ha igen akkor true-val tér vissza. Lépés után meghívja a setMoveDelay-t az esetleges akadálytól kapott lassítás mértékével
- **+ void setMoveDelay(int modifier):** beállítja a moveDelay-t a sebesség és a kapott modifier összegére
- **+ void damageHalf()** – Lefelezi saját maga aktuális életerejét, és szól az Enemy Generatornak, hogy duplikálja meg az ellenfelet az adott Tile-n.

9.1.8 EnemyGenerator

- **Felelősség**

- létrehozza az ellenségeket
- gondoskodik az idővel egyre nagyobb ellenséghordák biztosításáról

- tudja, hogy létre hozta-e az utolsó ellenséget
- ketté tud vágni egy ellenfelet

- **Össosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- - **int delay**: egy counter amely azt számolja mikor kell a következő ellenséget generálni.
- - **int generatingSpeed**: a jelenlegi gyakorisága egy ellenség lerakásának
- - **int generatedEnemies**: a lerakott ellenségek száma. Ha elér egy bizonyos értéket kinullázzuk, és dekrementáljuk a generatingSpeed-et, így biztosítva az egyre nagyobb ellenségsűrűséget
- - **int maxGeneratingSpeed**: a maximális ellenséglerakási sebesség
- - **int maxEnemies**: az összes játék során lerakandó ellenség
- - **PathGenerator pathGenerator**: referencia az útvonal generátorra
- - **Updater updater**: referencia az időkezelőre

- **Metódusok**

- + **void generateEnemies()**: ez a metódus hoz létre egy véletlenszerű fajta ellenséget, majd átadja a PathGenerator-nak, ami lerakja az egyik útvonalra
- + **boolean isLastEnemyGenerated()**: ha elértük a végleges generatingSpeed-et és az utolsó ellenséget is leraktuk, igazgal tér vissza. Egyébként hamis.
- + **duplicateEnemy(Enemy, PathTile)** – a kapott csempén a kapott ellenségből létrehoz még egy példányt, fele élettel, és a kapott ellenség életét is lefelezi.

9.1.9 FieldTile

- **Felelősség**

- el lehet rá helyezni tornyokat

- **Össosztályok**

Object->Tile

- **Interfészek**

nincs

- **Attribútumok**

- - **Construct construct**: a Tile-ra épített épületre mutató referencia.

- **Metódusok**

- **+ addConstruct(Construct construct):** a kapott épületet letárolja

9.1.10 Geometry

- **Felelősség**

- a csempék elrendezése
- meg tudja mondani, hogy adott csempétől adott sugarú körön belül melyik csempék vannak

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **- Tile[] tiles:** az összes csempét tároló attribútum
- **- int size_x:** az egyik dimenziója a pályának
- **- int size_y:** az egyik dimenziója a pályának

- **Metódusok**

- **+ Tile[] getNearby(Tile tile, int range):** egy csempe adott sugarú körön belüli csempéit adja vissza
- **+ Tile[] getTiles():** a tiles attribútumát adja vissza

9.1.11 Hobbit

- **Felelőssége**

- ismeri, és meg tudja adni saját típusát, életerejét, sebességét, varázserőben vett értékét
- le tudja vonni saját életerejét
- tud lépni

- **Ősosztályok**

Object->Enemy

- **Interfészek**

nincs

- **Attribútumok**

- **#String type:** az osztály típusát jelzi tehát az, hogy hobbit

- **Metódusok**

- *csak az Enemy osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül*

9.1.12 Human

- **Felelőssége**
 - ismeri, és meg tudja adni saját típusát, életerejét, sebességét, varázserőben vett értékét
 - le tudja vonni saját életerejét
 - tud lépni
- **Ősosztályok**
Object->Enemy
- **Interfészek**
nincs
- **Attribútumok**
 - **#String type:** az osztály típusát jelzi tehát az, hogy *human*
- **Metódusok**
 - csak az *Enemy* osztálytól örökölt metódusokat tartalmaz amiket nem definiál felül

9.1.13 MagicGem

- **Felelősség**
 - toronyba lehet elhelyezni, egyszerre csak egy darab lehet. Elhelyezéskor növelheti a tornyának valamely attribútumát
 - tudja, milyen ellenség ellen mennyi plusz sebzést ad
- **Ősosztályok**
Object
- **Interfészek**
nincs
- **Attribútumok**
 - **-String type:** a varázskő típusának neve
- **Metódusok**
 - **+int getDamageBonus(String type):** a paraméter és a saját type attribútuma alapján visszaadja, hogy mennyi plusz sebzést ad az adott ellenségtípusra

9.1.14 Mana

- **Felelősség**

- ismeri a jelenlegi varázserő mennyiségét
- meg tudja mondani, hogy van-e elég varázserő
- változtatni tudja a varázserő mennyiségét

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **-int mana:** jelenlegi varázserő mennyiség

- **Metódusok**

- **+boolean hasEnough(int amount):** megadja, hogy van-e paraméterként kapott mennyiségű varázserő
- **+void decrease(int value):** kivonja a jelenlegi varázserőből a kapott értéket
- **+void increase(int value):** hozzáadja a jelenlegi varázserőhöz a kapott értéket

9.1.15 PathGenerator

- **Felelősség**

- létre tudja hozni az útvonalakat, amin később ellenségek mehetnek végig
- rá tud helyezni egy ellenséget egy útvonal elejére

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **- Tile[] pathStarts:** útvonalak kezdőcsempéinek tömbje. Ezekre teheti le az ellenségeket
- **- EndTile endTile:** a végzet hegyének csempéje

- **Metódusok**

- **+void start(Enemy enemy):** leteszi a kapott ellenséget valamelyik kezdőcsempére.

9.1.16 PathTile

- **Felelősség**

- tartózkodhat rajta ellenség
- el lehet rá helyezni akadályokat
- tárolja az úton utána következő csempéket

- **Össztályok**

Object->Tile

- **Interfészek**

nincs

- **Attribútumok**

- **-Tile[] nextTiles:** az útvonalon lehetségesen következő csempék listája, ahova az ellenségek lépnek
- **-Enemy[] enemies:** a csempén tartózkodó ellenségeket tárolja

- **Metódusok**

- **+Enemy getEnemy():** visszaad egy, az ezen a csempén lévő ellenséget
- **+Tile[] getNextTiles():** visszaadja az útban utána következő csempét
- **+void removeEnemy(Enemy):** törli a megadott ellenséget a celláról

9.1.17 Tile

- **Felelősség**

- ezzel az absztrakt osztállyal lehetséges különböző típusú csempék kollekcióban tárolása

- **Össztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **# Construct construct:** a csempén lévő épület referenciája
- **# Geometry geometry:** a Tile-t tároló Geometry-re mutató referencia.
- **# String type:** a csempe típusa

- **Metódusok**

- **+void addConstruct(Construct construct):** a kapott épületet letárolja, ha lehet rajta
- **+void addEnemy(Enemy enemy):** a kapott ellenséget felveszi az ellenségek közé, ha lehet rajta
- **+Construct getConstruct():** visszaadja a csempén lévő épületet
- **+Geometry getGeometry():** visszaadja a csempét tároló Geometry-t

- **+String getType():** visszaadja a csempe típusát

9.1.18 Tower

- **Felelősség**

- tud a hatáskörén belüli csempéken tartózkodó ellenségekre lőni
- lehet fejleszteni különböző varázskövekkel

- **Ősosztályok**

Object->Construct

- **Interfészek**

nincs

- **Attribútumok**

- **-int damage:** a torony sebzése
- **-int fireRate:** tüzelési sebesség, azaz mennyit lő egy torony egyszerre
- **-int range:** a torony hatótávolsága
- **-int rangeModifier:** a torony hatótávát a környezet szerint módosító változó
- **-int shootDelay:** tárolja, hogy mennyi idő van hátra a következő lövésig
- **-FieldTile towerLocation:** a torony által elfoglalt csempe

- **Metódusok**

- **+Enemy shoot():** Lekérdezi a hatótávon belüli csempéket, majd kér valamelyikről egy ellenséget. Ha van varázskő a toronyban, megkérdezi milyen plusz sebzést biztosít a varázskő, és a saját értékéhez hozzáadva belesebzti azt az ellenségbe, annyszor, amennyi a fireRate attribútumának az értéke. Visszatér azzal az ellenséggel, akit meglőtt.
- **+void setDamage(int damage):** beállítja a damage attribútumot
- **+void setFireRate(int fireRate):** beállítja a fireRate attribútumot
- **+void setRange(int range):** beállítja a range attribútumot
- **+void setRangeModifier(int rangeModifier):** beállítja a hatótávot módosító változót

9.1.19 Updater

- **Felelősség**

- időszimuláció
- periodikus tevékenységek, például mozgás, lövés, ellenség generálás indítása

- **Ősosztályok**

Object

- **Interfészek**

nincs

- **Attribútumok**

- **-boolean isFoggy:** logikai érték arra, hogy van-e épp köd vagy nincs
- **-EnemyGenerator enemyGenerator:** az ellenségeket generáló osztályt tartalmazza
- **-Geometry geometry:** egy referencia a geometry osztályra
- **-Enemy[] enemies:** a játékban létező összes ellenséget tárolja
- **-Construct[] constructs:** a játékban létező összes épületet tárolja
- **-Mana mana:** A játékos jelenlegi varázserejét kezelő osztály
- **-Timer timer:** az időt számláló időzítő

- **Metódusok**

- **+addEnemy (Enemy enemy):** hozzáad egy ellenséget az enemies attribútumhoz
- **+addConstruct (Construct construct):** hozzáad egy épületet a constructs attribútumhoz
- **+void gameOver(boolean outcome):** véget vet a játéknak győzelemmel, ha igaz a paraméter, vereséggel ha hamis
- **+int getNumOfEnemies():** megszámolja az ellenségeket, és visszaadja az ellenségek számát
- **+static void main():** main metódus. Ebben jönnek létre az első objektumok, majd innen lépünk be a játék főciklusába

9.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

9.2.1 Pálya betöltése

- **Leírás**

A pályát betöltjük egy adott fájlból (létrejön egy pálya, feltöltjük terep- és út csempékkel, elhelyezünk egy végzet hegye csempét is, majd legenerálódnak az útvonalak)

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leellenőrizzük, hogy a pálya és a rajta található fix elemek betöltődnek-e. Rossz kimenet esetén a hiba a betöltött fájlban (loadmap.txt) vagy a betöltő függvényben (loadCommandFile) lehet. Ezek mellett lehetséges, hogy a pálya felépítésére szolgáló parancsok működése hibás.

- **Bemenet**

loadCommandFile loadmap.txt

- **Elvárt kimenet**

loadmap.txt betöltése sikeres

Sikeresen létrejött az 4, 4 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 0, 0 helyre.

Sikeresen hozzáadtad a T1 azonosítójú végzet hegye csempét az 0, 1 helyre.

Sikeresen hozzáadtad a T2 azonosítójú terep csempét az 0, 2 helyre.

Sikeresen hozzáadtad a T3 azonosítójú terep csempét az 0, 3 helyre.

Sikeresen hozzáadtad a T4 azonosítójú terep csempét az 1, 0 helyre.

Sikeresen hozzáadtad a T5 azonosítójú út csempét az 1, 1 helyre.

Sikeresen hozzáadtad a T6 azonosítójú terep csempét az 1, 2 helyre.

Sikeresen hozzáadtad a T7 azonosítójú terep csempét az 1, 3 helyre.

Sikeresen hozzáadtad a T8 azonosítójú terep csempét az 2, 0 helyre.

Sikeresen hozzáadtad a T9 azonosítójú út csempét az 2, 1 helyre.

Sikeresen hozzáadtad a T10 azonosítójú út csempét az 2, 2 helyre.

Sikeresen hozzáadtad a T11 azonosítójú terep csempét az 2, 3 helyre.

Sikeresen hozzáadtad a T12 azonosítójú terep csempét az 3, 0 helyre.

Sikeresen hozzáadtad a T13 azonosítójú terep csempét az 3, 1 helyre.

Sikeresen hozzáadtad a T14 azonosítójú út csempét az 3, 2 helyre.

Sikeresen hozzáadtad a T15 azonosítójú terep csempét az 3, 3 helyre.

Sikeresen beállítottad a T14 azonosítójú utcsempét kezdőcsempének.

Hozzáadtad a T14 azonosítójú utcsempéből elérhető utcsempékhez a T10 csempét.

Hozzáadtad a T10 azonosítójú utcsempéből elérhető utcsempékhez a T9 csempét.

Hozzáadtad a T9 azonosítójú utcsempéből elérhető utcsempékhez a T5 csempét.

Hozzáadtad a T5 azonosítójú utcsempéből elérhető utcsempékhez a T1 csempét.

9.2.2 Torony építése

- **Leírás**

Tornyot szeretnénk építeni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A torony építés folyamatának ellenőrzése. A ConstructManager osztály build függvényének ellenőrzése torony esetén, a varázserő levonódásának és ellenőrzésének tesztelése. Hiba felléphet a toronyt elhelyező függvény működésében, valamint az aktuálisan rendelkezésre álló és a szükséges varázserő összehasonlításában.

- **Bemenet**

```
createMap 2 2
addFieldTile 1 1
setMana 50
build T0 Tower 1
getStatus
```

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!
 Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 1, 1 helyre.
 A varázserő sikeresen beállítva 50 értékre!
 T0 csempére tower épült. 20 varázserőbe került.
 Pálya: 2x2
 Csempék:
 T0 Típus: field x:1 y:1
 Épületek:
 C0 Típus: tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5
 HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus
 Ellenségek:
 Varázserő: 30 Legenerálható ellenségek:0 Legközelebbi generálás:0
 Kód: nem

9.2.3 Akadály építése

- **Leírás**

Akadály épülését akarjuk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Akadály építés folyamatának ellenőrzése. A ConstructManager osztály build függvényének ellenőrzése akadály esetén

- **Bemenet**

```
createMap 2 2
addPathTile 1 1
setMana 50
build T0 Barricade 1
getStatus
```

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!
 Sikeresen hozzáadtad a T0 azonosítójú út csempét az 1, 1 helyre.
 A varázserő sikeresen beállítva 50 értékre!
 T0 csempére barricade épült. 20 varázserőbe került.

Pálya: 2x2

Csempék:

T0 Típus: path x:1 y:1 SzomszédCsempék: Start: false

Épületek:

C0 Típus: barricade Hely: T0 Kő: LassításMérték: 3

Ellenségek:

Varázserő: 30 Legenerálható ellenségek:0 Legközelebbi generálás:0

Kód: nem

9.2.4 Torony fejlesztése

- **Leírás**

A torony felruházása varázskövel

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ConstructManager osztály upgrade függvényének ellenőrzése torony esetén. Hiba a varázsső behelyezésében, valamint a torony tulajdonságainak módosulásában léphet fel.

- **Bemenet**

createMap 2 2

addFieldTile 1 1

setMana 50

build T0 Tower 0

upgrade C0 Range 1

getStatus

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 1, 1 helyre.

A varázserő sikeresen beállítva 50 értékre!

T0 csempére tower épült.

C0 épületbe range varázskövet tettél. 10 varázserőbe került.

Pálya: 2x2

Csempék:

T0 Típus: field x:1 y:1

Épületek:

C0 Típus: tower Hely: T0 Kő: range Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 7

HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus

Ellenségek:

Varázserő: 40 Legenerálható ellenségek:0 Legközelebbi generálás:0

Kód: nem

9.2.5 Torony lő

- **Leírás**

A torony a hatótávolságán belül tartózkodó ellenségek valamelyikét kiválasztja és rálő.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Geometry osztály jól adja-e vissza a torony hatótávolságában lévő csempéket. A torony a megkapott csempékről ki tudja-e választani egy ellenséget, amelyet meg is lő. Hiba lehet az elérhető ellenségeket megkereső, valamint az őket sebző függvényekben.

- **Bemenet**

```
createMap 3 3
addFieldTile 0 1
addPathTile 1 0
addPathTile 1 1
addPathTile 1 2
addNextTile T3 T2
addNextTile T2 T1
build T0 Tower 0
addEnemy T3 Dwarf
shoot C0 0
getStatus
```

- **Elvárt kimenet**

Sikeresen létrejött az 3, 3 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 0, 1 helyre.

Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 0 helyre.

Sikeresen hozzáadtad a T2 azonosítójú út csempét az 1, 1 helyre.

Sikeresen hozzáadtad a T3 azonosítójú út csempét az 1, 2 helyre.

Hozzáadtad a T3 azonosítójú útcsempéből elérhető útcsempékhez a T2 csempét.

Hozzáadtad a T2 azonosítójú útcsempéből elérhető útcsempékhez a T0 csempét.

T0 csempére tower épült.

Sikeresen létrehozta az E0 azonosítójú "dwarf" típusú ellenséget a T3 csempén.

C0 lőtt, E0 azonosítójú ellenségre, levéve tőle 20 életerőt.

Pálya: 3x3

Csempék:

T0 Típus: field	x:0	y:1	
T1 Típus: path	x:1	y:0	SzomszédCsempék: Start: false
T2 Típus: path	x:1	y:1	SzomszédCsempék: T0 Start: false
T3 Típus: path	x:1	y:2	SzomszédCsempék: T2 Start: false

Épületek:

C0 Típus: tower Hely: T1 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5
HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus

Ellenségek:

E0: Típus: dwarf Hely: T3 Életerő: 8 Sebesség: 4 KövetkezőLépés: 4 ciklus
Érték: 8

Varázserő 0 Legenerálható ellenségek: 0 Legközelebbi generálás: 0 ciklus

Kód: nem

9.2.6 Akadály lassít

- **Leírás**

Az ellenség rálép egy út csempére (PathTile) amelyen akadály van, így az ellenség lelassul.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az akadály lassításának ellenőrzése. Lehetséges hiba az lehet hogy az ellenség az akadályra való lépés utáni ciklusban tovább lép. Tovább az is hiba ha az akadályra való lépés utáni második ciklusban nem lép az ellenség.

- **Bemenet**

```
createMap 3 3
addPathTile 1 0
addPathTile 1 1
addPathTile 1 2
addNextTile T0 T1
addNextTile T1 T2
addEnemy T0 Elf
build T1 Barricade 0
move E0
move E0
move E0
```

- **Elvárt kimenet**

Sikeresen létrejött az 3, 3 méretű pálya!
 Sikeresen hozzáadtad a T0 azonosítójú út csempét az 1, 0 helyre.
 Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 1 helyre.
 Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 2 helyre.
 Hozzáadtad a T0 azonosítójú útcsempéből elérhető útcsempékhez a T1 csempét.
 Hozzáadtad a T1 azonosítójú útcsempéből elérhető útcsempékhez a T2 csempét
 Sikeresen létrehozta az E0 azonosítójú elf típusú ellenséget a T0 csempén
 T1 csempére barricade épült.
 E1 lépett T0-ről T1-re. T1-n akadály van, lelassult. Legközelebb 2 szimulációs ciklus után lépne.
 E1 moveDelay értéke egyel csökkent.
 E1 lépett T1-ről T2-re. Legközelebb 1 szimulációs ciklus után lépne.

9.2.7 Köd

- **Leírás**

Köd ereszkedik le a pályára majd egy bizonyos idő múlva felszáll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt célja a kód leereszkedésének ellenőrzése. Ezen belül is a tornyok hatótávolság változásának tesztelése. Lehetséges hiba lehet hogy a tornyok hatótávolsága nem változik a kód leszálltakor vagy nem áll vissza a megfelelő értékre miután felszállt.

- **Bemenet**

```
createMap 2 2
addFieldTile 0 0
addFieldTile 0 1
build T0 Tower 0
build T1 Tower 0
getStatus
```

setFog 1
getStatus
setFog 0
getStatus

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 0, 0 helyre.

Sikeresen hozzáadtad a T1 azonosítójú terep csempét az 0, 1 helyre.

T0 csempére tower épült.

T1 csempére tower épült.

Pálya: 2x2

Csempék:

T0 Típus: field x:0 y:0

T1 Típus: field x:0 y:1

Épületek:

C0 Típus: tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5

HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus

C1 Típus: tower Hely: T1 Kő: range Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5

HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus

Varázserő: 0 Legenerálható ellenségek: 0 Legközelebbi generálás: 0 ciklus

Kód: nem

Kód bekapcsolva. Az összes torony hatótávja 0.7-szeresére csökkent.

Pálya: 2x2

Csempék:

T0 Típus: field x:0 y:0

T1 Típus: field x:0 y:1

Épületek:

C0 Típus: tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5

HatótávSzorzó: 0.7 KövetkezőLövés: 2 ciklus

C1 Típus: tower Hely: T1 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5

HatótávSzorzó: 0.7 KövetkezőLövés: 2 ciklus

Varázserő: 0 Legenerálható ellenségek: 0 Legközelebbi generálás: 0 ciklus

Kód: igen

Kód kikapcsolva. Az összes torony hatótávja eredeti értékre állt vissza.

Pálya: 2x2

Csempék:

T0 Típus: field x:0 y:0

T1 Típus: field x:0 y:1

Épületek:

C0 Típus: tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5

HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus

C1 Típus: tower Hely: T1 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 5

HatótávSzorzó: 1 KövetkezőLövés: 2 ciklus

Varázserő: 0 Legenerálható ellenségek: 0 Legközelebbi generálás: 2 ciklus

Kód: nem

9.2.8 Ellenség halála

- **Leírás**

Az ellenség egy torony lövés miatt meghal.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ellenség halálának vizsgálata. A varázserő változás és a játék véget értének ellenőrzése. Hiba lehet az ha nem hal meg az ellenség , nem kapunk varázserőt vagy ha nem ér véget a játék.

- **Bemenet**

```
createMap 2 2
addPathTile 1 0
addPathTile 1 1
addNextTile T0 T1
addFieldTile 0 1
build T2 Tower 0
addEnemy T0 Elf
setRemainingEnemies 0
setMana 0
simulate 1
```

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú út csempét az 1, 0 helyre.

Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 1 helyre.

Hozzáadtad a T0 azonosítójú útcsempéből elérhető útcsempékhez a T1 csempét

Sikeresen hozzáadtad a T2 azonosítójú terep csempét az 0, 1 helyre.

T2 csempére tower épült.

Sikeresen létrehoztad az E0 azonosítójú elf típusú ellenséget a T0 csempén

0 számú ellenség vár még a legenerálásra!

A varázserő sikeresen beállítva 0 értékre

A szimuláció véget ért. 1 ciklus futott le!

- A(z) 1. ciklusban lépett az E0 azonosítójú ellenség a T0 celláról T1-re.

- A(z) 1. ciklusban lőtt a T0 azonosítójú torony az E0 ellenségre, mely meghalt, így a varázserőd 6-tal nőtt.

- Az utolsó ellenség is meghalt a 1.ciklusban, nyertél!

9.2.9 Ellenség osztódik

- **Leírás**

Ellenség lövés hatására kettő válik

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük hogy egy felező lövés hatására képes-e az ellenség ketté válni. Amennyiben az ellenség nem válik kettő ,vagy nem megfelelő életerővel , esetleg nem jó csempére rakódott le akkor a teszt hibának minősül.

- **Bemenet**

```
createMap 2 2
addPathTile 1 0
addPathTile 1 1
addNextTile T0 T1
addFieldTile 0 1
build T2 Tower 0
addEnemy T1 Dwarf
shoot C0 1
```

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú út csempét az 1, 0 helyre.

Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 1 helyre.

Hozzáadtad a T0 azonosítójú útcsempéből elérhető útcsempékhez a T1 csempét

Sikeresen hozzáadtad a T2 azonosítójú terep csempét az 0, 1 helyre.

T2 csempére tower épült.

Sikeresen létrehoztad az E0 azonosítójú "Dwarf" típusú ellenséget a T2 csempén

C0 lőtt, E0 azonosítójú ellenségre, felező lövéssel, levéve tőle 30 életerőt.

Létrejött E1 ellenség.

9.3 A tesztelést támogató programok tervei

A tesztelést egy egyszerű parancssori alkalmazás támogatja. A program két parancssori argumentumot vár: az első a megadott fájl, aminek a tartalma a megírt kimenet, a második argumentum pedig a tesztprogram által generált file.

Az összehasonlító program soronként értelmezi a két fájlt, melyben a két megfelelő sort összehasonlítja. Az összehasonlítás eredményét egy eredmény fájlba írjuk ki. Amennyiben a két sor megegyezik, úgy nem ír ki semmit. Ha a két sor nem egyezik meg, úgy a következő üzenetet kapjuk: 'A kapott kimenet nem egyezik meg a várt kimenettel!'. Ezután kiírja a két különböző sort egymás alá. A sorrend a megadott fájlsorrendet követi, tehát az első sor a megírt kimenet, a második pedig a tesztprogram által generált kimenet. Amennyiben a teszt lefut, úgy minden esetben a fájl végére kiírja, hogy a 'A teszt sikeresen lefutott'.

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

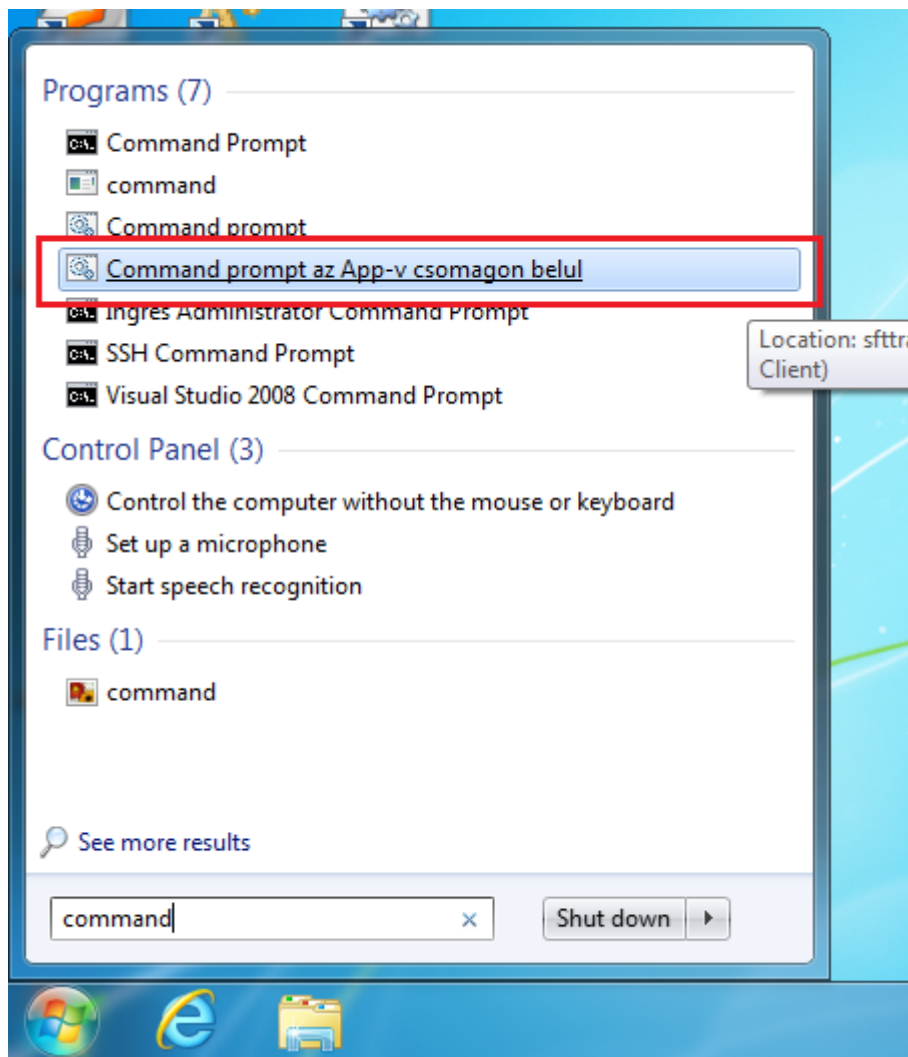
Fájl neve	Méret	Keletkezés ideje	Tartalom
Barricade.java	579 byte	2014.04.21. 21:30	Az akadályokért felelős osztály megvalósítása.
Construct.java	399 byte	2014.04.21. 21:30	Az épületek ősosztályának megvalósítása.
ConstructManager.java	2 691 byte	2014.04.21. 21:37	Az épületek létrehozásáért és fejlesztéséért felelős osztály megvalósítása.
Dwarf.java	289 byte	2014.04.21. 21:30	A törpe típusú ellenségek kezeléséért felelős osztály megvalósítása.
Elf.java	284 byte	2014.04.21. 21:30	A tünde típusú ellenségek kezeléséért felelős osztály megvalósítása.
EndTile.java	708 byte	2014.04.21. 21:30	A Végzet Hegyét tároló csempe kezeléséért felelős osztály.
Enemy.java	3 639 byte	2014.04.21. 21:30	Az ellenségek ősosztályának megvalósítása.
EnemyGenerator.java	3 739 byte	2014.04.21. 21:32	Az ellenségek létrehozásáért felelős osztály.
FieldTile.java	441 byte	2014.04.21. 21:30	A terepcellákért felelős osztály megvalósítása, ezekre lehet tornyot építeni.
Geometry.java	3 031 byte	2014.04.21. 21:30	A pálya geometriai szerkezetéért felelős osztály megvalósítása.
help.txt	3 843 byte	2014.04.22. 11:04	A beadható parancsok leírását tartalmazó fájl.
Hobbit.java	255 byte	2014.04.21. 21:30	A hobbit típusú ellenséget kezelő osztály megvalósítása.
Human.java	250 byte	2014.04.21. 21:30	Az ember típusú ellenséget kezelő osztály megvalósítása.
MagicGem.java	538 byte	2014.04.21. 21:30	A varázskövekért felelős osztály megvalósítása.

Mana.java	924 byte	2014.04.21. 21:30	A játékos varázserejének kezeléséért felelős osztály megvalósítása.
PathGenerator.java	909 byte	2014.04.21. 21:30	Az útvonalakat kezelő osztály megvalósítása.
PathTile.java	1 544 byte	2014.04.21. 21:30	Az útvonalcellákért felelős osztály megvalósítása, ezekre lehet akadályt építeni, és ellenségek helyzetét adja meg.
PrototypeController.java	18 760 byte	2014.04.21. 21:38	A tesztelésért felelős osztály megvalósítása.
test1.txt	392 byte	2014.04.22. 11:04	Első teszt eset parancsait tartalmazó fájl.
test1_elvart_kimenet.txt	1 674 byte	2014.04.22. 11:04	Első teszt eset parancsainak a kimenetét tartalmazó fájl.
test2.txt	72 byte	2014.04.22. 11:04	Második teszt eset parancsait tartalmazó fájl.
test2_elvart_kimenet.txt	520 byte	2014.04.22. 11:04	Második teszt eset parancsainak a kimenetét tartalmazó fájl.
test3.txt	75 byte	2014.04.22. 11:04	Harmadik teszt eset parancsait tartalmazó fájl.
test3_elvart_kimenet.txt	482 byte	2014.04.22. 11:04	Harmadik teszt eset parancsainak a kimenetét tartalmazó fájl.
test4.txt	92 byte	2014.04.22. 11:04	Negyedik teszt eset parancsait tartalmazó fájl.
test4_elvart_kimenet.txt	571 byte	2014.04.22. 11:04	Negyedik teszt eset parancsainak a kimenetét tartalmazó fájl.
test5.txt	180 byte	2014.04.22. 11:04	Ötödik teszt eset parancsait tartalmazó fájl.
test5_elvart_kimenet.txt	1 275 byte	2014.04.22. 11:04	Ötödik teszt eset parancsainak a kimenetét tartalmazó fájl.
test6.txt	168 byte	2014.04.22. 11:04	Hatodik teszt eset parancsait tartalmazó fájl.
test6_elvart_kimenet.txt	787 byte	2014.04.22. 11:04	Hatodik teszt eset parancsainak a kimenetét tartalmazó fájl.
test7.txt	138 byte	2014.04.22. 11:04	Hetedik teszt eset parancsait tartalmazó fájl.
test7_elvart_kimenet.txt	1 331 byte	2014.04.22. 11:04	Hetedik teszt eset parancsainak a kimenetét tartalmazó fájl.

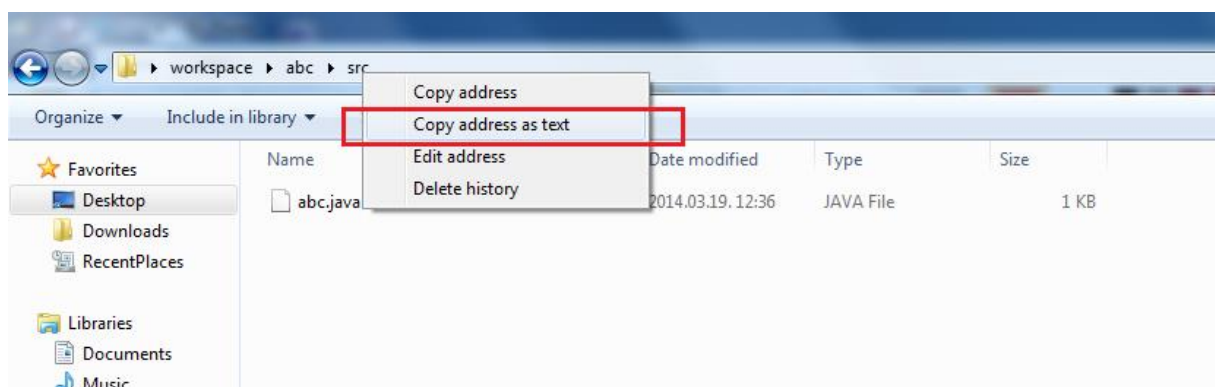
			tartalmazó fájl.
test8.txt	165 byte	2014.04.22. 11:04	Nyolcadik teszteset parancsait tartalmazó fájl.
test8_elvart_kimenet.txt	1 058 byte	2014.04.22. 11:04	Nyolcadik teszteset parancsainak a kimenetét tartalmazó fájl.
test9.txt	133 byte	2014.04.22. 11:04	Kilencedik teszteset parancsait tartalmazó fájl.
test9_elvart_kimenet.txt	589 byte	2014.04.22. 11:04	Kilencedik teszteset parancsainak a kimenetét tartalmazó fájl.
Tile.java	1 107 byte	2014.04.21. 21:30	A csempék oszostályának megvalósítása.
Tower.java	3 004 byte	2014.04.21. 21:30	A tornyokért felelős osztály megvalósítása.
Updater.java	6 919 byte	2014.04.21. 21:30	Az időkezelésért felelős osztály megvalósítása.

10.1.2 Fordítás

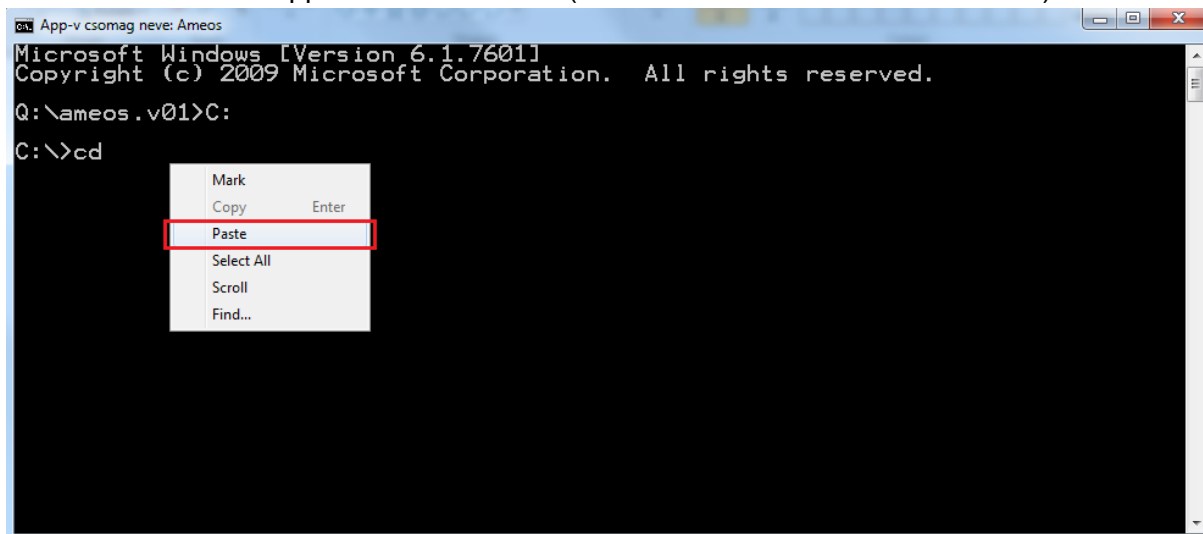
- Indítsunk egy Command Prompt-ot az App-v csomagon belül (A Start menü keresőmezőjébe írjuk be, hogy „command”, és klikkeljünk rá).



- Lépjünk a command promptban abba a mappába, ahol a forráskód található. Ehhez először beléphetünk a mappába a Windows Intézőben és kimásolhatjuk a mappa címét.



- Ezt felhasználva a command promptban lépünk a forráskód mappájába. Először a meghajtó betűjelét írjuk be egy kettősponttal (C:), majd a cd parancs után kell illeszteni a mappa elérési útvonalát (amit most másoltunk ki az Intézőből).



- A sikeres mappaváltás után adjuk ki a következő parancsot:
`javac *.java`

10.1.3 Futtatás

Lépünk a mappába, ahol a fordítás végeztük! (A Fordítás 1-3. lépése ezt írja le). Ezután adjuk ki a következő parancsot:

```
java PrototypeController
```

Ekkor a program elindul, várja a bemeneti nyelvnek megfelelő parancsokat a konzolról. Az előre megadott teszteseteket a loadCommandFile paranccsal tudjuk futtatni. Az egyes teszteseteket külön fájlokba tettük, számozva. Pl. az 1. tesztet a következő parancs futtatja:

```
loadCommandFile test1.txt
```

Ezután a test1.txt-ben lévő parancsok futnak ne. Ezután ha szeretnénk, további parancsokat adhatunk, de egy új teszt lefuttatása előtt szükséges egy exit hívás. Ilyenkor a program a test1_output.txt fájlba menti a kimenetet, és kilép.

```
exit
```

Két teszt futtatása között mindenképpen szükséges kilépni az **exit** paranccsal, hogy az elvárt eredményt kapjuk.

A tesztesetek elvárt kimenetei a test<sorszám>_elvar_t_kimenet.txt fájlokban találhatóak.

Tehát pl. a test1_elvar_t_kimenet.txt az első teszteset elvárt kimenetét tartalmazza. A futtatás során létrejött outputnak és a hozzá tartozó elvar_t_kimenet fájlak tartalmilag meg kell egyeznie.

10.2 Tesztek jegyzőkönyvei

10.2.1 Teszteset 1

Tesztelő neve	Barabás Péter
Teszt időpontja	2014.04.20

10.2.2 Teszteset 2

Tesztelő neve	Apáti – Nagy Attila
Teszt időpontja	2014.04.20

10.2.3 Teszteset 3

Tesztelő neve	Révész Péter
Teszt időpontja	2014.04.20

10.2.4 Teszteset 4

Tesztelő neve	Csörgő Szabolcs
Teszt időpontja	2014.04.20

10.2.5 Teszteset 5

Tesztelő neve	Boczán Tamás
---------------	--------------

Teszt időpontja	2014.04.21
Teszt eredménye	<u>java.lang.IndexOutOfBoundsException</u>
Lehetséges hibaok	Nem kezeltük le, hogy mi van akkor, ha a körülöttünk levő cellákon nincs ellenség.
Változtatások	A kérdéses eset megvizsgálása

10.2.6 Teszteset 6

Tesztelő neve	Révész Péter
Teszt időpontja	2014.04.20

10.2.7 Teszteset 7

Tesztelő neve	Barabás Péter
Teszt időpontja	2014.04.20
Teszt eredménye	A kód felszállta után a hatótáv nem állt vissza eredeti értékre.
Lehetséges hibaok	Implementációs hiba
Változtatások	A kód felszállta után a régi érték visszaállítása

10.2.8 Teszteset 8

Tesztelő neve	Boczán Tamás
Teszt időpontja	2014.04.20

10.2.9 Teszteset 9

Tesztelő neve	Révész Péter
Teszt időpontja	2014.04.21
Teszt eredménye	Az ellenség osztódik, de az életerő nem feleződik
Lehetséges hibaok	Rosszul implementált duplikáló függvény
Változtatások	Helyes implementálás

10.3 Értékelés

Tag neve	Munka százalékban
Apáti-Nagy Attila	18.8%
Barabás Péter	20.1%
Boczán Tamás	20.4%
Csörgő Szabolcs	19.2%
Révész Péter	21.5%

10.4 A tesztek részletes tervei módosítás

10.4.1 Teszteset 1 - Pálya betöltése

- **Leírás**

A pályát betöltjük egy adott fájlból (létrejön egy pálya, feltöltjük terep- és út csempékkel, elhelyezünk egy végzet hegye csempét is, majd legenerálódnak az útvonalak)

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leellenőrizzük, hogy a pálya és a rajta található fix elemek betöltődnek-e. Rossz kimenet esetén a hiba a betöltött fájlban (test1.txt) vagy a betöltő függvényben (loadCommandFile) lehet. Ezek mellett lehetséges, hogy a pálya felépítésére szolgáló parancsok működése hibás.

- **Bemenet**

```
createMap 4 4
addFieldTile 0 0
addEndTile 0 1
addFieldTile 0 2
addFieldTile 0 3
addFieldTile 1 0
addPathTile 1 1
addFieldTile 1 2
addFieldTile 1 3
addFieldTile 2 0
addPathTile 2 1
addPathTile 2 2
addFieldTile 2 3
addFieldTile 3 0
addFieldTile 3 1
addPathTile 3 2
addFieldTile 3 3
setStartTile T14
addNextTile T14 T10
addNextTile T10 T9
addNextTile T9 T5
addNextTile T5 T1
```

- **Elvárt kimenet**

Sikeresen létrejött az 4, 4 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 0, 0 helyre.

Sikeresen hozzáadtad a T1 azonosítójú végzet hegye csempét az 0, 1 helyre.

Sikeresen hozzáadtad a T2 azonosítójú terep csempét az 0, 2 helyre.

Sikeresen hozzáadtad a T3 azonosítójú terep csempét az 0, 3 helyre.

Sikeresen hozzáadtad a T4 azonosítójú terep csempét az 1, 0 helyre.

Sikeresen hozzáadtad a T5 azonosítójú út csempét az 1, 1 helyre.

Sikeresen hozzáadtad a T6 azonosítójú terep csempét az 1, 2 helyre.

Sikeresen hozzáadtad a T7 azonosítójú terep csempét az 1, 3 helyre.

Sikeresen hozzáadtad a T8 azonosítójú terep csempét az 2, 0 helyre.
 Sikeresen hozzáadtad a T9 azonosítójú út csempét az 2, 1 helyre.
 Sikeresen hozzáadtad a T10 azonosítójú út csempét az 2, 2 helyre.
 Sikeresen hozzáadtad a T11 azonosítójú terep csempét az 2, 3 helyre.
 Sikeresen hozzáadtad a T12 azonosítójú terep csempét az 3, 0 helyre.
 Sikeresen hozzáadtad a T13 azonosítójú terep csempét az 3, 1 helyre.
 Sikeresen hozzáadtad a T14 azonosítójú út csempét az 3, 2 helyre.
 Sikeresen hozzáadtad a T15 azonosítójú terep csempét az 3, 3 helyre.
 Sikeresen beállítottad a T14 azonosítójú útcsempét kezdőcsempének.
 Hozzáadtad a T14 azonosítójú útcsempéből elérhető útcsempékhez a T10 csempét.
 Hozzáadtad a T10 azonosítójú útcsempéből elérhető útcsempékhez a T9 csempét.
 Hozzáadtad a T9 azonosítójú útcsempéből elérhető útcsempékhez a T5 csempét.
 Hozzáadtad a T5 azonosítójú útcsempéből elérhető útcsempékhez a T1 csempét.

10.4.2 Teszteset 2 - Torony építése

- **Leírás**

Tornyot szeretnénk építeni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A torony építés folyamatának ellenőrzése. A ConstructManager osztály build függvényének ellenőrzése torony esetén, a varázserő levonódásának és ellenőrzésének tesztelése. Hiba felléphet a toronyt elhelyező függvény működésében, valamint az aktuálisan rendelkezésre álló és a szükséges varázserő összehasonlításában.

- **Bemenet**

```
createMap 2 2
addFieldTile 1 1
setMana 80
build T0 tower 1
getStatus
```

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 1, 1 helyre.

A varázserő sikeresen beállítva 80 értékre!

T0 csempére tower épült. 50 varázserőbe került.

Pálya: 2x2

Csempék:

T0 FieldTile x: 1 y: 1

Épületek:

C0 Tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 3
 HatótávSzorzó: 1.0 KövetkezőLövés: 3 ciklus

Ellenségek:

Varázserő: 30 LegenerálhatóEllenségek: 30 LegközelebbiGenerálás: 999

Kód: nem

10.4.3 Teszteset 3 - Akadály építése

- **Leírás**

Akadály épülését akarjuk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Akadály építés folyamatának ellenőrzése. A ConstructManager osztály build függvényének ellenőrzése akadály esetén

- **Bemenet**

createMap 2 2
addPathTile 1 1
setMana 50
build T0 barricade 1
getStatus

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú út csempét az 1, 1 helyre.

A varázserő sikeresen beállítva 50 értékre!

T0 csempére barricade épült. 30 varázserőbe került.

Pálya: 2x2

Csempék:

T0 PathTile x: 1 y: 1 Szomszéd Csempék: Start: false

Épületek:

C0 Barricade Hely: T0 Kő: LassításMértéke: 1

Ellenségek:

Varázserő: 20 LegenerálhatóEllenségek: 30 LegközelebbiGenerálás: 999

Kód: nem

10.4.4 Teszteset 4 - Torony fejlesztése

- **Leírás**

A torony felruházása varázskövel

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ConstructManager osztály upgrade függvényének ellenőrzése torony esetén. Hiba a varázsső behelyezésében, valamint a torony tulajdonságainak módosulásában léphet fel.

- **Bemenet**

createMap 2 2
addFieldTile 1 1
setMana 50
build T0 tower 0
upgrade C0 range 1
getStatus

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 1, 1 helyre.

A varázserő sikeresen beállítva 50 értékre!

T0 csempére tower épült.

C0 épületbe range varázskövet tettél. 20 varázserőbe került.

Pálya: 2x2

Csempék:

T0 FieldTile x: 1 y: 1

Épületek:

C0 Tower Hely: T0 Kő: range Sebzés: 20 TüzelésiSebesség: 3

Hatótáv: 20 HatótávSzorzó: 1.0 KövetkezőLövés: 3 ciklus

Ellenségek:

Varázserő: 30 LegenerálhatóEllenségek: 30 LegközelebbiGenerálás: 999

Kód: nem

10.4.5 Teszteset 5 - Torony lő

- **Leírás**

A torony a hatótávolságán belül tartózkodó ellenségek valamelyikét kiválasztja és rálő.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Geometry osztály jól adja-e vissza a torony hatótávolságában lévő csempéket. A torony a megkapott csempékről ki tudja-e választani egy ellenséget, amelyet meg is lő. Hiba lehet az elérhető ellenségeket megkereső, valamint az őket sebző függvényekben.

- **Bemenet**

createMap 3 3

addFieldTile 0 1

addPathTile 1 0

addPathTile 1 1

addPathTile 1 2

addNextTile T3 T2

addNextTile T2 T1

build T0 tower 0

addEnemy T3 dwarf

shoot C0 0

getStatus

- **Elvárt kimenet**

Sikeresen létrejött az 3, 3 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 0, 1 helyre.

Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 0 helyre.

Sikeresen hozzáadtad a T2 azonosítójú út csempét az 1, 1 helyre.

Sikeresen hozzáadtad a T3 azonosítójú út csempét az 1, 2 helyre.

Hozzáadtad a T3 azonosítójú utcsempéből elérhető utcsempékhez a T2 csempét.

Hozzáadtad a T2 azonosítójú utcsempéből elérhető utcsempékhez a T1 csempét.

T0 csempére tower épült.

Sikeresen létrehozta az E0 azonosítójú "dwarf" típusú ellenséget a T3 csempén.

C0 lőtt, E0 azonosítójú ellenségre, levéve tőle 20 életerőt.

Pálya: 3x3

Csempék:

T0 FieldTile	x: 0 y: 1	
T1 PathTile	x: 1 y: 0	Szomszéd Csempék: Start: false
T2 PathTile	x: 1 y: 1	Szomszéd Csempék: T1 Start: false
T3 PathTile	x: 1 y: 2	Szomszéd Csempék: T2 Start: false

Épületek:

C0 Tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 3
 HatótávSzorzó: 1.0 KövetkezőLövés: 3 ciklus

Ellenségek:

E0 dwarf Típus: dwarf Hely: T3 Életerő: 60 Sebesség: 6
 KövetkezőLépés: 0 ciklus Érték: 8

Varázserő: 0 LegenerálhatóEllenségek: 30 LegközelebbiGenerálás: 999

Kód: nem

10.4.6 Teszteset 6 - Akadály lassít

- **Leírás**

Az ellenség rálép egy út csempére (PathTile) amelyen akadály van, így az ellenség lelassul.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az akadály lassításának ellenőrzése. Lehetséges hiba az lehet hogy az ellenség az akadályra való lépés utáni ciklusban tovább lép. Tovább az is hiba ha az akadályra való lépés utáni második ciklusban nem lép az ellenség.

- **Bemenet**

```
createMap 3 3
addPathTile 1 0
addPathTile 1 1
addPathTile 1 2
addNextTile T0 T1
addNextTile T1 T2
addEnemy T0 elf
build T1 barricade 0
move E0
move E0
move E0
move E0
```

- **Elvárt kimenet**

Sikeresen létrejött az 3, 3 méretű pálya!
 Sikeresen hozzáadtad a T0 azonosítójú út csempét az 1, 0 helyre.
 Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 1 helyre.
 Sikeresen hozzáadtad a T2 azonosítójú út csempét az 1, 2 helyre.
 Hozzáadtad a T0 azonosítójú útcsempéből elérhető útcsempékhez a T1 csempét.
 Hozzáadtad a T1 azonosítójú útcsempéből elérhető útcsempékhez a T2 csempét.
 Sikeresen létrehozta az E0 azonosítójú "elf" típusú ellenséget a T0 csempén.
 T1 csempére barricade épült.

E0 lépett T0-ről T1-re. T1-n akadály van, lelassult. Legközelebb 2 szimulációs ciklus után lépne.

E0 moveDelay értéke egyel csökkent.

E0 lépett T1-ről T2-re. Legközelebb 1 szimulációs ciklus után lépne.

10.4.7 Teszteset 7 - Kód

- **Leírás**

Kód ereszkedik le a pályára majd egy bizonyos idő múlva felszáll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt célja a kód leereszkedésének ellenőrzése. Ezen belül is a tornyok hatótávolság változásának tesztelése. Lehetséges hiba lehet hogy a tornyok hatótávolsága nem változik a kód leszálltakor vagy nem áll vissza a megfelelő értékre miután felszállt.

- **Bemenet**

```
createMap 2 2  
addFieldTile 0 0  
addFieldTile 0 1  
build T0 tower 0  
build T1 tower 1  
getStatus  
setFog 1  
getStatus  
setFog 0  
getStatus
```

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú terep csempét az 0, 0 helyre.

Sikeresen hozzáadtad a T1 azonosítójú terep csempét az 0, 1 helyre.

T0 csempére tower épült.

Pálya: 2x2

Csempék:

T0 FieldTile x: 0 y: 0

T1 FieldTile x: 0 y: 1

Épületek:

C0 Tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 3

HatótávSzorzó: 1.0 KövetkezőLövés: 3 ciklus

Ellenségek:

Varázserő: 0 LegenerálhatóEllenségek: 30

LegközelebbiGenerálás: 999

Kód: nem

Kód bekapcsolva. Az összes torony hatótávja 0.7-szeresére csökkent.

Pálya: 2x2

Csempék:

T0 FieldTile x: 0 y: 0

T1 FieldTile x: 0 y: 1

Épületek:

C0 Tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 3
 HatótávSzorzó: 0.7 KövetkezőLövés: 3 ciklus

Ellenségek:

Varázserő: 0 LegenerálhatóEllenségek: 30 LegközelebbiGenerálás: 999

Kód: igen

Kód kikapcsolva. Az összes torony hatótávja eredeti értékre állt vissza.

Pálya: 2x2

Csempék:

T0 FieldTile x: 0 y: 0

T1 FieldTile x: 0 y: 1

Épületek:

C0 Tower Hely: T0 Kő: Sebzés: 20 TüzelésiSebesség: 3 Hatótáv: 3
 HatótávSzorzó: 1.0 KövetkezőLövés: 3 ciklus

Ellenségek:

Varázserő: 0 LegenerálhatóEllenségek: 30 LegközelebbiGenerálás: 999

Kód: nem

10.4.8 Teszteset 8 - Ellenség halála

- **Leírás**

Az ellenség egy torony lövés miatt meghal.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ellenség halálának vizsgálata. A varázserő változás és a játék véget értének ellenőrzése. Hiba lehet az ha nem hal meg az ellenség , nem kapunk varázserőt vagy ha nem ér véget a játék.

- **Bemenet**

```
createMap 2 2
addPathTile 1 0
addPathTile 1 1
addNextTile T0 T1
addFieldTile 0 1
build T2 tower 0
addEnemy T0 elf
setRemainingEnemies 0
setMana 0
simulate 3
```

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú út csempét az 1, 0 helyre.

Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 1 helyre.

Hozzáadtad a T0 azonosítójú útcsempéből elérhető útcsempékhez a T1 csempét.

Sikeresen hozzáadtad a T2 azonosítójú terep csempét az 0, 1 helyre.

T2 csempére tower épült.

Sikeresen létrehozta az E0 azonosítójú "elf" típusú ellenséget a T0 csempén.

0 számú ellenség vár még a legenerálásra!

A varázserő sikeresen beállítva 0 értékre!

A szimuláció véget ért. 3 ciklus futott le!

- A(z) 1. ciklusban lépett az E0 azonosítójú ellenség a T0 celláról T1-re.
- A(z) 2. ciklusban nem tudott lépni az E0 azonosítójú ellenség a T1 celláról.
- A(z) 3. ciklusban nem tudott lépni az E0 azonosítójú ellenség a T1 celláról.
- A(z) 3. ciklusban lőtt a C0 azonosítójú torony az E0 ellenségre, mely meghalt, így a varázserő 5-al nőtt.
- Az utolsó ellenség is meghalt a 3. ciklusban, nyertél!

10.4.9 Teszteset 9 - Ellenség osztódik

- **Leírás**

Ellenség lövés hatására kettő válik

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük hogy egy felező lövés hatására képes-e az ellenség ketté válni. Amennyiben az ellenség nem válik kettő ,vagy nem megfelelő életerővel , esetleg nem jó csempére rakódott le akkor a teszt hibának minősül.

- **Bemenet**

```
createMap 2 2
addPathTile 1 0
addPathTile 1 1
addNextTile T0 T1
addFieldTile 0 1
build T2 tower 0
addEnemy T1 dwarf
shoot C0 1
```

- **Elvárt kimenet**

Sikeresen létrejött az 2, 2 méretű pálya!

Sikeresen hozzáadtad a T0 azonosítójú út csempét az 1, 0 helyre.

Sikeresen hozzáadtad a T1 azonosítójú út csempét az 1, 1 helyre.

Hozzáadtad a T0 azonosítójú útcsempéből elérhető útcsempékhez a T1 csempét.

Sikeresen hozzáadtad a T2 azonosítójú terep csempét az 0, 1 helyre.

T2 csempére tower épült.

Sikeresen létrehozta az E0 azonosítójú "dwarf" típusú ellenséget a T1 csempén.

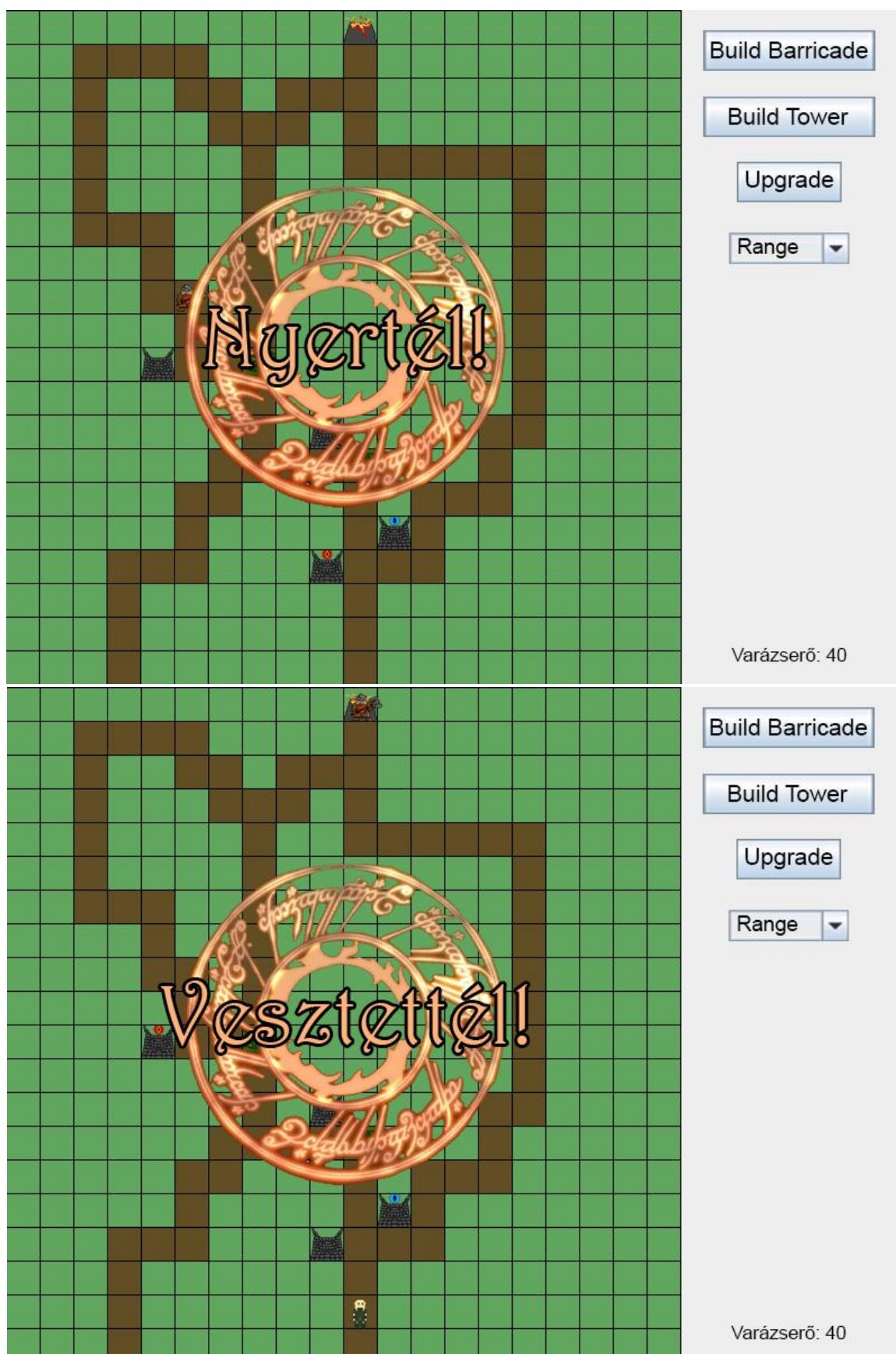
C0 lőtt, E0 azonosítójú ellenségre, felező lövéssel, levéve tőle 40 életerőt. Létrejött E1 ellenség.

11. Grafikus felület specifikációja

11.1 A grafikus interfész

A grafikus felület tervezésekor az átláthatóság és a könnyű kezelhetőség volt a fő szempont. A program indításakor rögtön a játékban találjuk magunkat. A játék irányítása egérrel történik. Az ablak nagy részében a pályát láthatjuk az aktuális játékelemekkel, jobb oldalt pedig a parancsok kiadásához használható gombokat és a rendelkezésünkre álló varázserőt láthatjuk. Három gombunk van, ezek segítségével tudunk tornyot és akadályt építeni, valamint fejleszteni. Fejlesztéskor az épületbe helyezendő varázskő típusát a gomb alatt található legördülő listából választhatjuk ki.



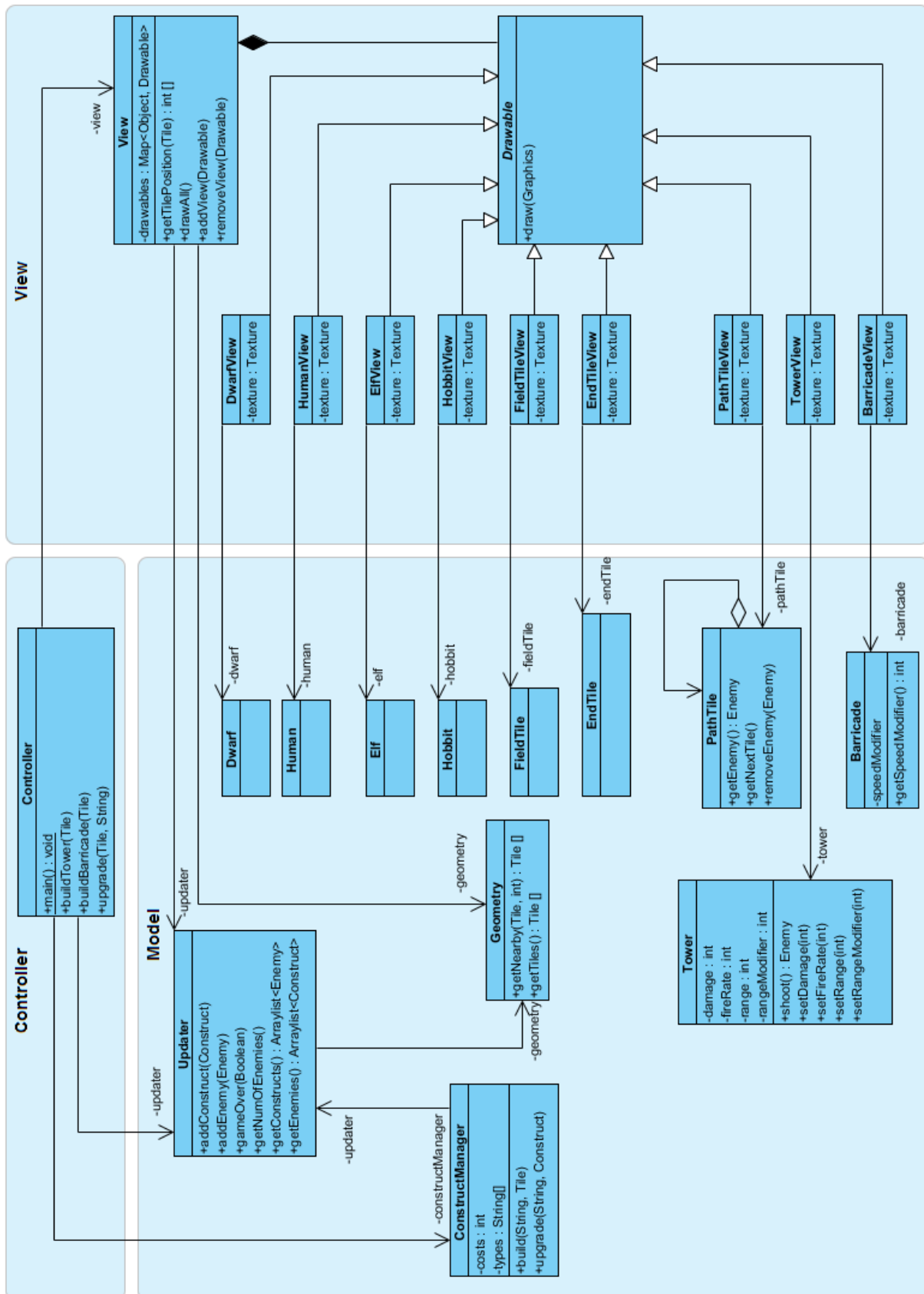


11.2 *A grafikus rendszer architektúrája*

11.2.1 A felület működési elve

A felület PULL elven működik. Rajzoláskor a felület (view) lekérdezi a modelltől a rajzolandó objektumok állapotát, majd a kapott adatok alapján megjeleníti azokat. A modellben a megjelenítésért felelős részek elkülönülésének érdekében a megjelenítendő osztályok minden példányához tartozik egy-egy kirajzoló (drawable) osztálypéldány, melyek példányait a felület tárolja.

11.2.2 A felület osztály-struktúrája



11.3 A grafikus objektumok felsorolása

11.3.1 BarricadeView

- **Felelősség**

Felelős az akadályok kirajzolásáért.

- **Ősosztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- - **texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzolja az entitást a texture textúrával.

11.3.2 Controller

- **Felelősség**

Inicializálja a programot és kezeli az inputokat. Tartalmazza a main metódust, ez a program belépési pontja.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- -**constructManager: ConstructManager** Referencia az építésért felelős objektumra.
- -**updater: Updater** Az időszimulációt kezelő osztály.
- -**view: View** A játékot megjelenítő view.

- **Metódusok**

- **+buildBarricade(Tile): void** Feldolgozza az akadály építő gomb megnyomásának eseményét. Ha a csempe megfelelő típusú, és nincs rajta épület, akkor meghívja a constructManager build metódusát.
- **+buildTower(Tile): void** Feldolgozza a torony építő gomb megnyomásának eseményét. Ha a csempe megfelelő típusú, és nincs rajta épület, akkor meghívja a constructManager build metódusát.
- **+main(): void** A program belépési pontja. Felépíti a program menüjéhez szükséges részeket.

- **+upgrade(Tile, String): void** Feldolgozza a fejlesztés gomb megnyomásának eseményét. Ha az épület megfelelő típusú, és nincs benne varázskő, akkor meghívja a constructManager upgrade metódusát.

11.3.3 Drawable

- **Felelősség**

Absztrakt osztály, mely összefogja a rajzoló objektumokat.

- **Össztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **#view: View** Referencia az őt tartalmazó View-ra.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzol egy entitást a neki megfelelő módon.

11.3.4 DwarfView

- **Felelősség**

Felelős a törpök kirajzolásáért.

- **Össztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- **- texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzolja az entitást a texture textúrával.

11.3.5 ElfView

- **Felelősség**

Felelős a tündék kirajzolásáért.

- **Ősosztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- - **texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzolja az entitást a texture textúrával.

11.3.6 EndTileView

- **Felelősség**

Felelős a végzet hegye csempe kirajzolásáért.

- **Ősosztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- - **texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzolja az entitást a texture textúrával.

11.3.7 FieldTileView

- **Felelősség**

Felelős a terepcsempék kirajzolásáért.

- **Ősosztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- - **texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

+draw(Graphics g): void Kirajzolja az entitást a texture textúrával.

11.3.8 HobbitView

- **Felelősség**

Felelős a hobbitok kirajzolásáért.

- **Össztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- - **texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzolja az entitást a texture textúrával.

11.3.9 HumanView

- **Felelősség**

Felelős az emberek kirajzolásáért.

- **Össztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- - **texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzolja az entitást a texture textúrával.

11.3.10 PathTileView

- **Felelősség**

Felelős az útvonalcsempék kirajzolásáért.

- **Ősosztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- - **texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzolja az entitást a texture textúrával.

11.3.11 TowerView

- **Felelősség**

Felelős a tornyok kirajzolásáért.

- **Ősosztályok**

Drawable

- **Interfészek**

Nincs

- **Attribútumok**

- - **texture: Texture** A kirajzolandó entitás textúrája.

- **Metódusok**

- **+draw(Graphics g): void** Kirajzolja az entitást a texture textúrával.

11.3.12 View

- **Felelősség**

Ez az osztály vezérli a grafikus megjelenést. Tárolja, létrehozza, és törli az összes entitást megjelenítő osztályt. Ütemezi a rajzolást, és rajzoláskor felszólítja a view-kat, hogy rajzoljanak.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

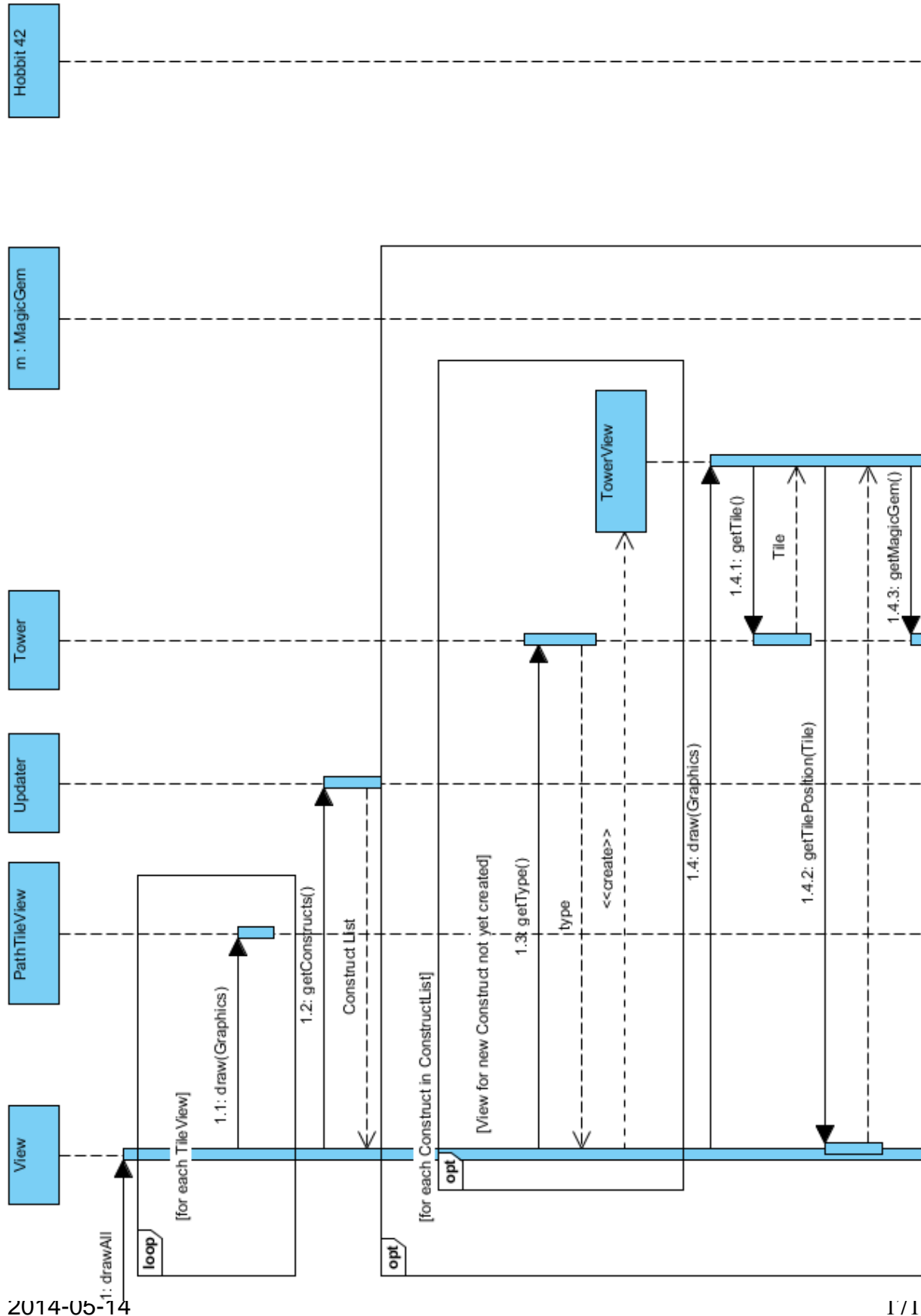
- -**geometry: Geometry** A pálya elrendezéséért felelős osztályra mutató referencia.

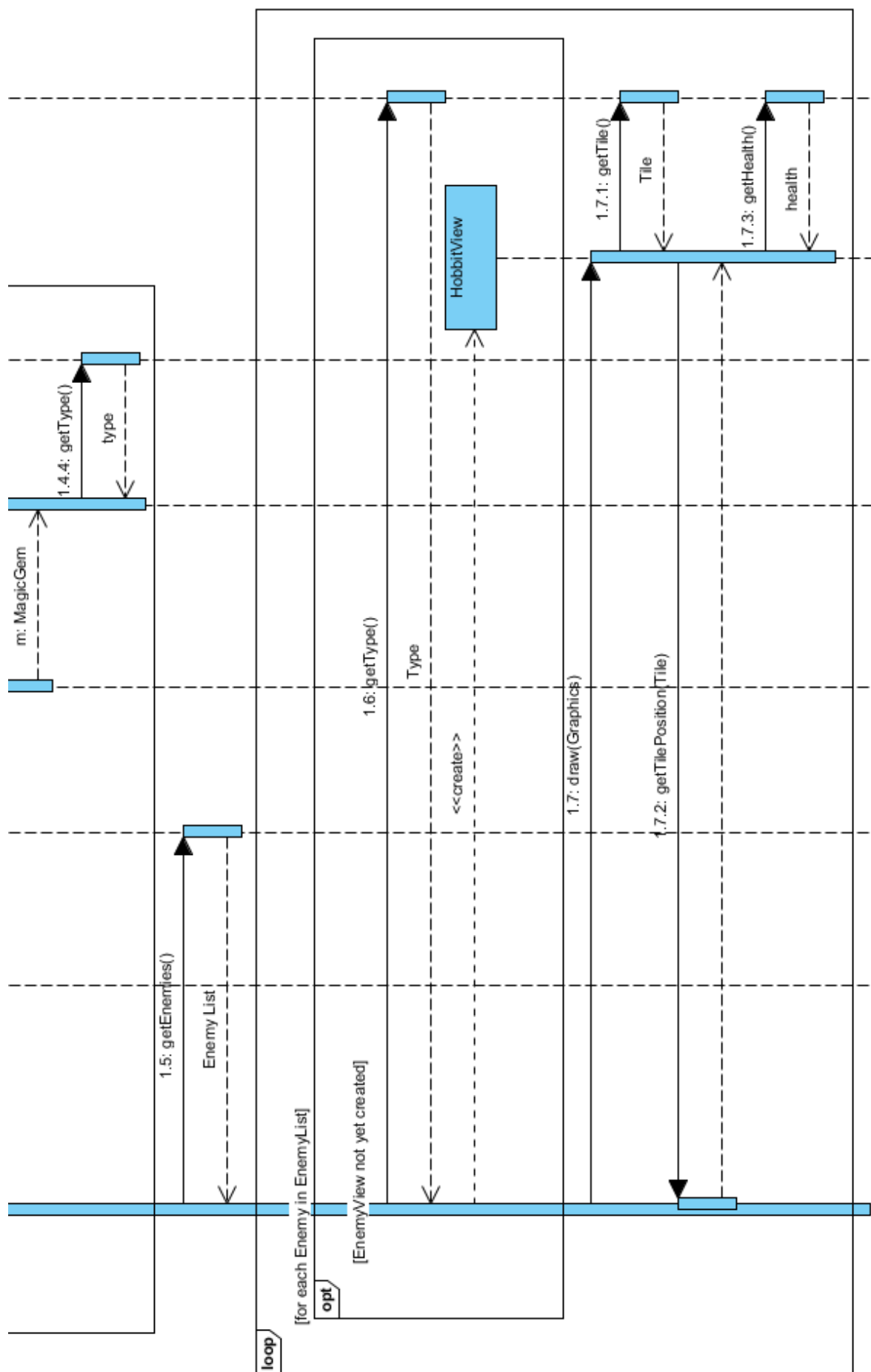
- **-drawables: ArrayList<Drawable>** Az egyes view-k kollekciója.
- **Metódusok**
 - **+addView(Drawable): void** Felvesz egy view-t a drawable-be.
 - **+drawAll(): void** Meghívja az összes view rajzoló metódusát.
 - **+getTilePosition(Tile): int[]** Adott csempe pixelkoordinátáit számolja és adja vissza.
 - **+removeView(Drawable): void** Eltávolít egy view-t a drawable-ből.

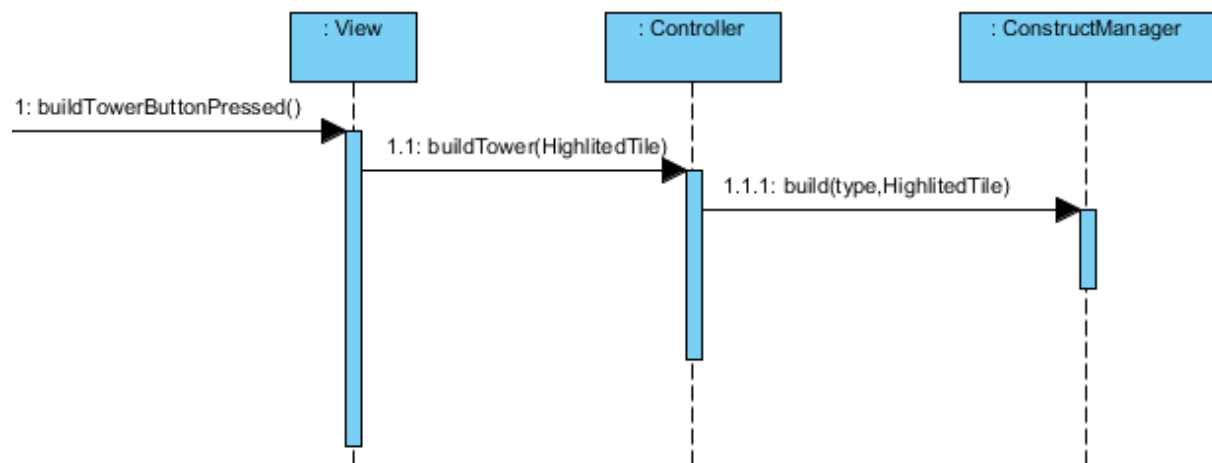
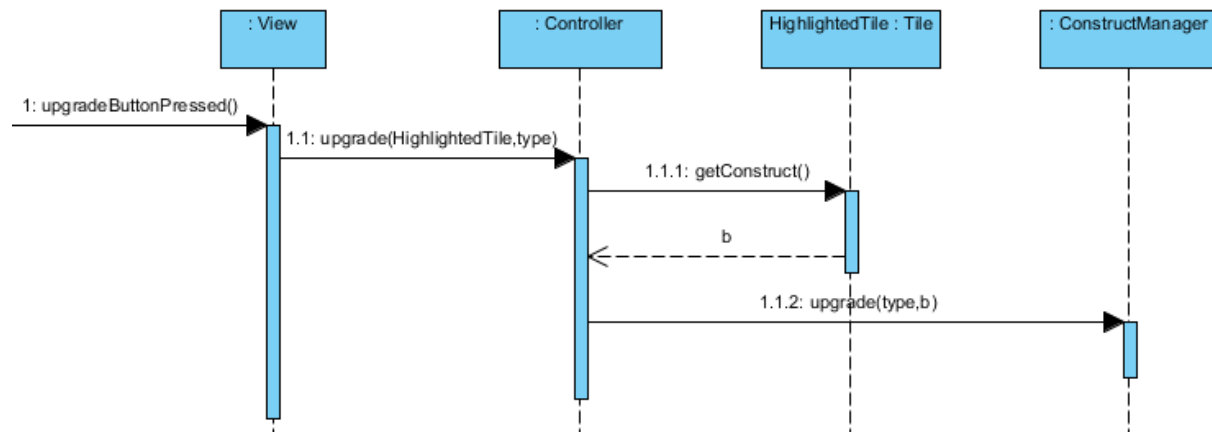
11.4 Kapcsolat az alkalmazói rendszerrel

A diagramok a Controller és a View komponensek működését mutatják be, a Model változatlanul működik.

Rajzolás:





Torony építése:**Fejlesztés:**

12. Grafikus változat beadása

12.1 Fordítási és futtatási útmutató

12.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
draw\BarricadeView.java	1235 byte	2014.05.12. 0:04	Az akadályokat rajzoló osztály megvalósítása.
draw\Controller.java	955 byte	2014.05.10. 22:58	Az inputot kezelő osztály megvalósítása.
draw\Drawable.java	1748 byte	2014.05.12. 0:04	A rajzoló osztályok absztrakt őssztálya.
draw\DwarfView.java	1186 byte	2014.05.12. 0:04	A törpöket rajzoló osztály megvalósítása.
draw\ElfView.java	1145 byte	2014.05.12. 0:04	A tündéket rajzoló osztály megvalósítása.
draw\EndTileView.java	945 byte	2014.05.12. 0:04	A végzet hegyét rajzoló osztály megvalósítása.
draw\FieldTileView.java	983 byte	2014.05.12. 0:04	A terep csempéket rajzoló osztály megvalósítása.
draw\HobbitView.java	1209 byte	2014.05.12. 0:04	A hobbitokat rajzoló osztály megvalósítása.
draw\HumanView.java	1177 byte	2014.05.12. 0:04	Az embereket rajzoló osztály megvalósítása.
draw\PathTileView.java	968 byte	2014.05.12. 0:04	A utcsempéket rajzoló osztály megvalósítása.
draw\TowerView.java	1743 byte	2014.05.11. 15:10	A tornyokat rajzoló osztály megvalósítása.
draw\View.java	12749 byte	2014.05.11. 19:44	Az ablakot rajzoló osztály megvalósítása.
game\Barricade.java	579 byte	2014.04.21. 21:30	Az akadályokért felelős osztály megvalósítása.
game\Construct.java	399 byte	2014.04.21. 21:30	Az épületek őssztályának megvalósítása.
game\ConstructManager.java	2 691 byte	2014.04.21. 21:37	Az épületek létrehozásáért és fejlesztéséért felelős osztály megvalósítása.
game\Dwarf.java	289 byte	2014.04.21. 21:30	A törpe típusú ellenségek kezeléséért felelős osztály megvalósítása.

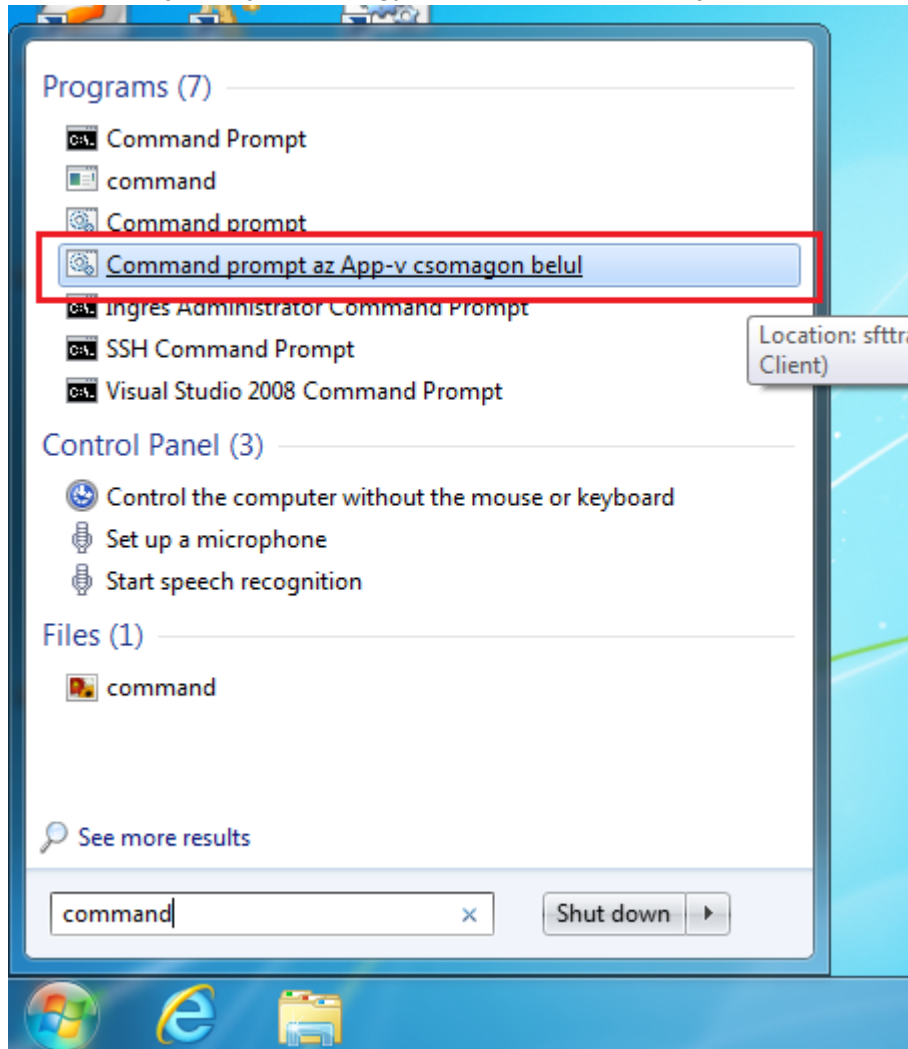
game\Elf.java	284 byte	2014.04.21. 21:30	A tünde típusú ellenségek kezeléséért felelős osztály megvalósítása.
game\EndTile.java	708 byte	2014.04.21. 21:30	A Végzet Hegyét tároló csempe kezeléséért felelős osztály.
game\Enemy.java	3 639 byte	2014.04.21. 21:30	Az ellenségek ősosztályának megvalósítása.
game\EnemyGenerator.java	3 739 byte	2014.04.21. 21:32	Az ellenségek létrehozásáért felelős osztály.
game\FieldTile.java	441 byte	2014.04.21. 21:30	A terepcellákért felelős osztály megvalósítása, ezekre lehet tornyot építeni.
game\Geometry.java	3 031 byte	2014.04.21. 21:30	A pálya geometriai szerkezetéért felelős osztály megvalósítása.
game\Hobbit.java	255 byte	2014.04.21. 21:30	A hobbit típusú ellenséget kezelő osztály megvalósítása.
game\Human.java	250 byte	2014.04.21. 21:30	Az ember típusú ellenséget kezelő osztály megvalósítása.
game\MagicGem.java	538 byte	2014.04.21. 21:30	A varázskövekért felelős osztály megvalósítása.
game\Mana.java	924 byte	2014.04.21. 21:30	A játékos varázserejének kezeléséért felelős osztály megvalósítása.
game\PathGenerator.java	909 byte	2014.04.21. 21:30	Az útvonalakat kezelő osztály megvalósítása.
game\PathTile.java	1 544 byte	2014.04.21. 21:30	Az útvonalcellákért felelős osztály megvalósítása, ezekre lehet akadályt építeni, és ellenségek helyzetét adja meg.
game\PrototypeController.java	18 760 byte	2014.04.21. 21:38	A tesztelésért felelős osztály megvalósítása.
game\Tile.java	1 107 byte	2014.04.21. 21:30	A csempék ősosztályának megvalósítása.
game\Tower.java	3 004 byte	2014.04.21. 21:30	A tornyokért felelős osztály megvalósítása.

game\Updater.java	6 919 byte	2014.04.21. 21:30	Az időkezelésért felelős osztály megvalósítása.
textures\constructs\barricade\barricade.png	21836 byte	2014.05.11 23:44	Akadály textúrája.
textures\constructs\barricade\barricade_gem.png	21812 byte	2014.05.11 23:44	Varázskővel ellátott akadály textúrája.
textures\constructs\tower\tower.png	17512 byte	2014.05.11 23:44	Torony textúrája.
textures\constructs\tower\tower_dwarf.png	17838 byte	2014.05.11 23:44	Törpe elleni varázskővel ellátott torony textúrája.
textures\constructs\tower\tower_elf.png	17843 byte	2014.05.11 23:44	Tünde elleni varázskővel ellátott torony textúrája.
textures\constructs\tower\tower_firerate.png	18319 byte	2014.05.11 23:44	Tüzelési sebesség varázskővel ellátott torony textúrája.
textures\constructs\tower\tower_hobbit.png	17845 byte	2014.05.11 23:44	Hobbit elleni varázskővel ellátott torony textúrája.
textures\constructs\tower\tower_human.png	17842 byte	2014.05.11 23:44	Ember elleni varázskővel ellátott torony textúrája.
textures\constructs\tower\tower_range.png	17840 byte	2014.05.11 23:44	Hatótáv növelő varázskővel ellátott torony textúrája.
textures\enemies\dwarf.png	16682 byte	2014.05.11 23:44	Törp textúrája.
textures\enemies\dwarf-damaged.png	4331 byte	2014.05.11 23:44	Sérült törp textúrája.
textures\enemies\elf.png	17355 byte	2014.05.11 23:44	Tünde textúrája.
textures\enemies\elf-damaged.png	4931 byte	2014.05.11 23:44	Sérült tünde textúrája.
textures\enemies\hobbit.png	17536 byte	2014.05.11 23:44	Hobbit textúrája.
textures\enemies\hobbit-damaged.png	5052 byte	2014.05.11 23:44	Sérült hobbit textúrája.
textures\enemies\human.png	17258 byte	2014.05.11 23:44	Ember textúrája.
textures\enemies\human-damaged.png	4697 byte	2014.05.11 23:44	Sérült ember textúrája.
textures\tiles\fieldTile.png	116235 byte	2014.05.11 23:44	Terepcsempe textúrája.
textures\tiles\mount_doom.png	5023 byte	2014.05.11 23:44	Végzet hegyének textúrája.
textures\tiles\mount_doom_transparent.png	1217 byte	2014.05.11 23:44	Végzet hegyének áttetsző textúrája.
textures\tiles\pathTile.png	131577 byte	2014.05.11 23:44	Útcsempe textúrája.

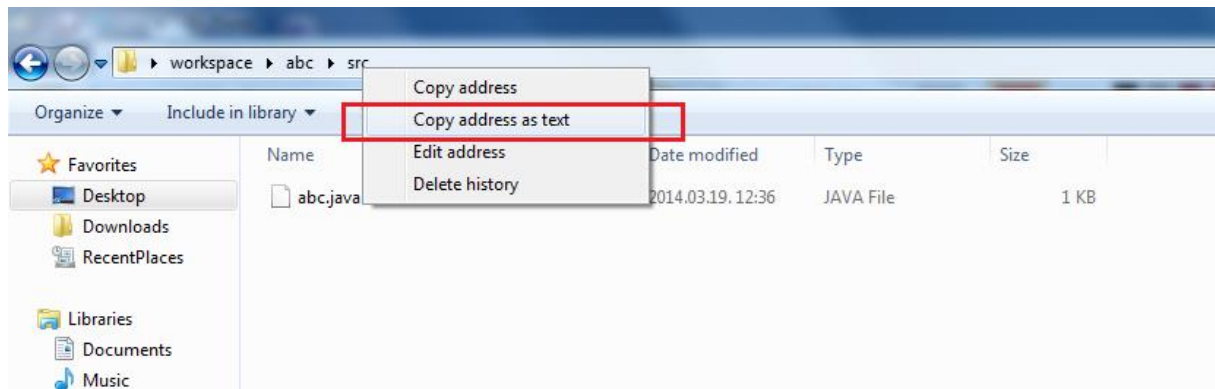
textures\defeat.png	146657 byte	2014.05.11 23:44	Vereséget jelző felirat textúrája.
textures\fog.png	125836 byte	2014.05.11 23:44	Köd textúrája.
textures\menubackground.jpg	23467 byte	2014.05.11 23:44	Kezelőpanel háttérének textúrája.
textures\victory.png	147005 byte	2014.05.11 23:44	Győzelmet jelző felirat textúrája.

12.1.2 Fordítás

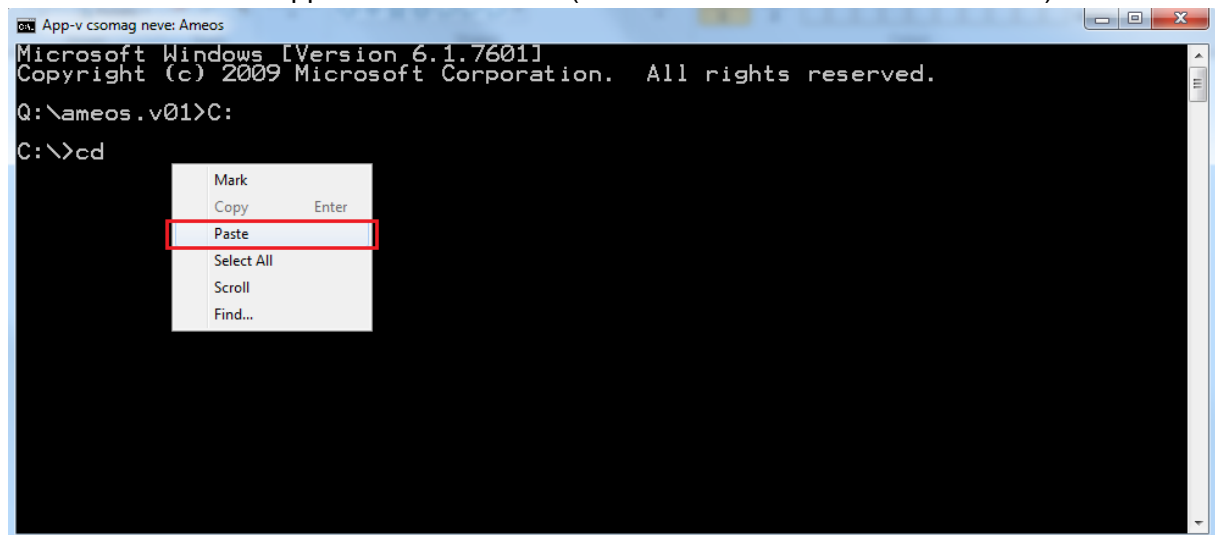
- Indítsunk egy Command Prompt-ot az App-v csomagon belül (A Start menü keresőmezőjébe írjuk be, hogy „command”, és klikkeljünk rá).



- Lépünk a command promptban a gyökérmappába, tehát ahol a draw, game és textures mappa van. Ehhez először beléphetünk a mappába a Windows Intézőben és kimásolhatjuk a mappa címét.



- Ezt felhasználva a command promptban lépünk a forráskód mappájába. Először a meghajtó betűjelét írjuk be egy kettősponttal (C:), majd a cd parancs után kell illeszteni a mappa elérési útvonalát (amit most másoltunk ki az Intézőből).



- A sikeres mappaváltás után adjuk ki a következő parancsot:
`javac -encoding utf8 draw/*.java game/*.java`

12.1.3 Futtatás

Lépünk a mappába, ahol a fordítás végeztük! (A Fordítás 1-3. lépése ezt írja le). Ezután adjuk ki a következő parancsot:

```
java draw/Controller
```

Ekkor a program elindul, és lehet játszani. Ki lehet bal klikkel választani bármelyik csempét. Jobb oldalt 3 gomb és egy legördülő menü van. A "Build Tower" gombbal lehet a kiválasztott csempére tornyot építeni, a "Build Barricade" gombbal lehet a kiválasztott csempére akadályt építeni. A torony 50 varázserő, az akadály 30. Ha a kiválasztott csempén épület van, akkor az "Upgrade" gombbal lehet bele varázskövet tenni. A varázskő típusa az aktuálisan kiválasztott eleme lesz a legördülő menünek. Az akadályba csak "Slow" típusút lehet tenni, a toronyba csak az összes többit. Tornyot csak üres, zöld csempére lehet tenni, akadályt csak üres utcsempére lehet tenni. Alul látszódik a jelenlegi varázserő értéke. Ha a pálya tetején lévő végzet hegyét eléri egy ellenség vesztettél. Ha nem maradt több ellenség, nyertél. Új játékot a fentebbi parancs újbóli kiadásával tudsz kezdeni, miután bezártad a játékot.

12.2 Értékelés

Tag neve	Munka százalékban
Apáti-Nagy Attila	19.4%
Barabás Péter	19.8%
Boczán Tamás	19.8%
Csörgő Szabolcs	19.6%
Révész Péter	21.4%

13. Összefoglalás

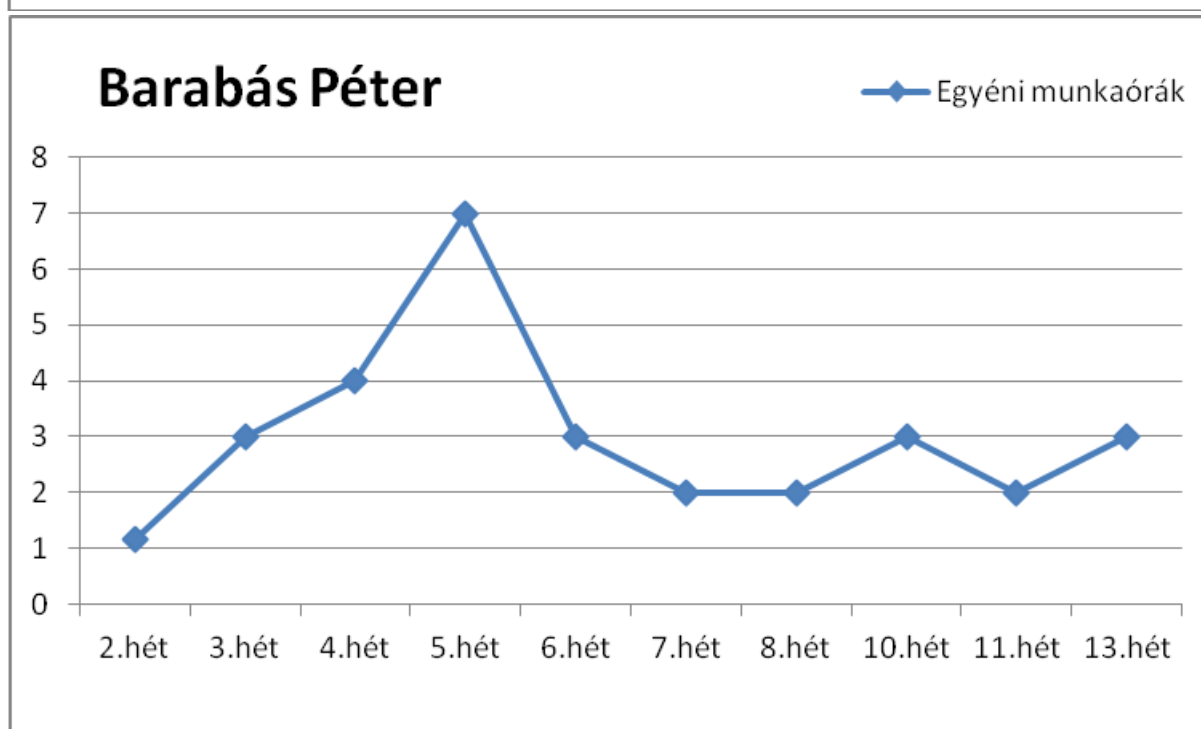
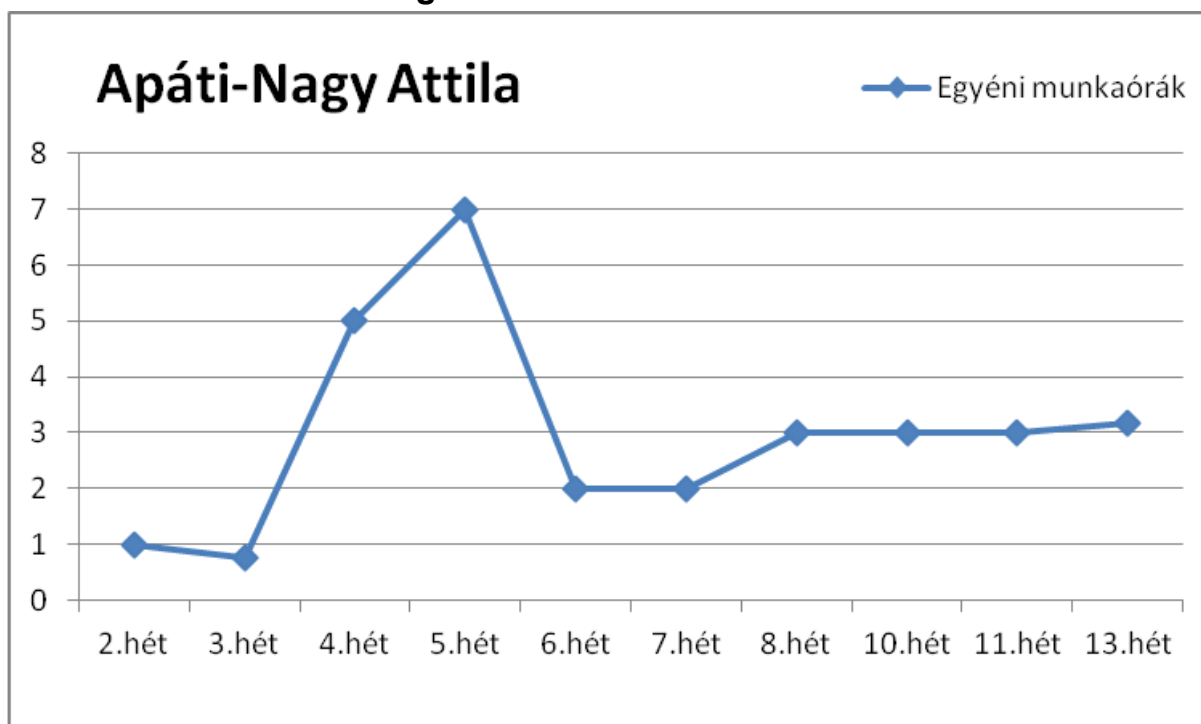
13.1 A projektre fordított összes munkaidő

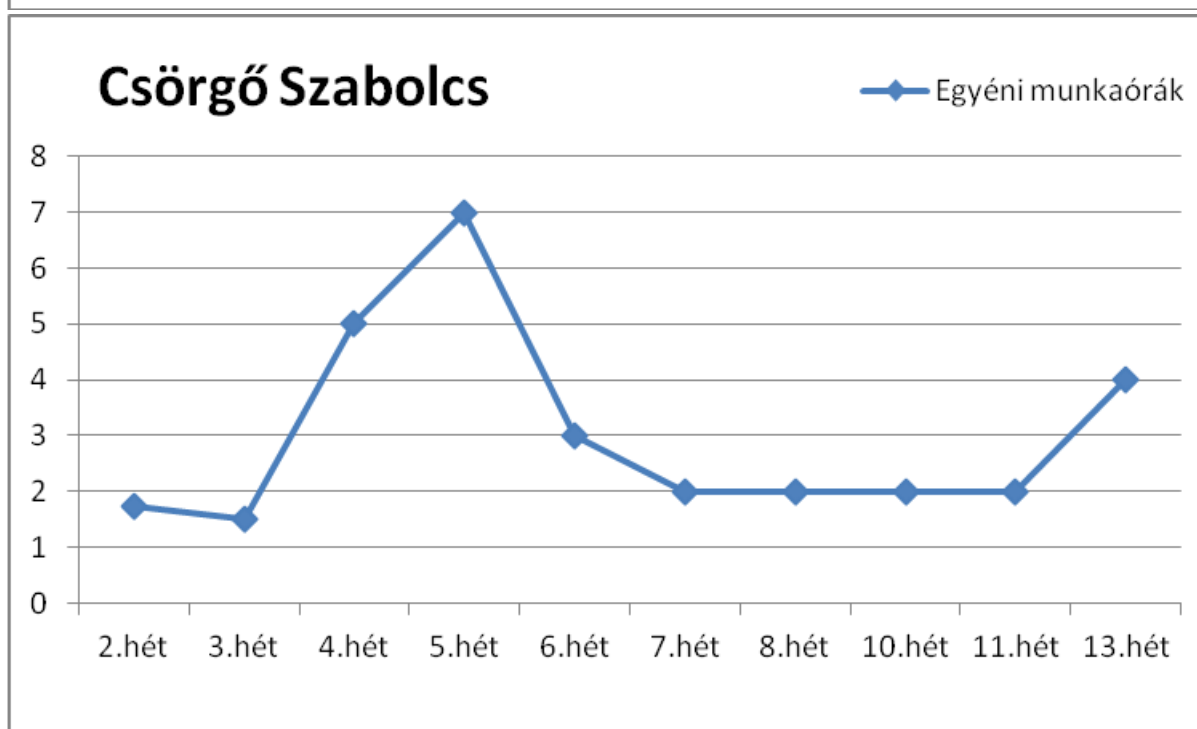
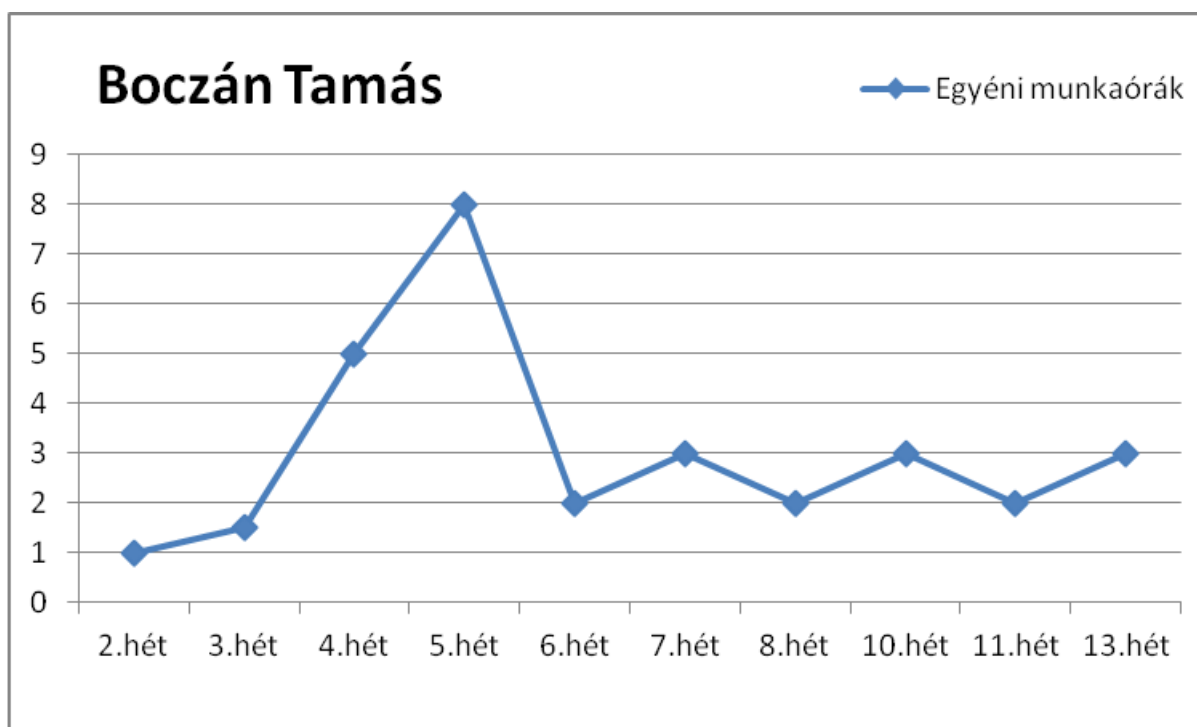
13.1.1 Egyéni munkaidő percben

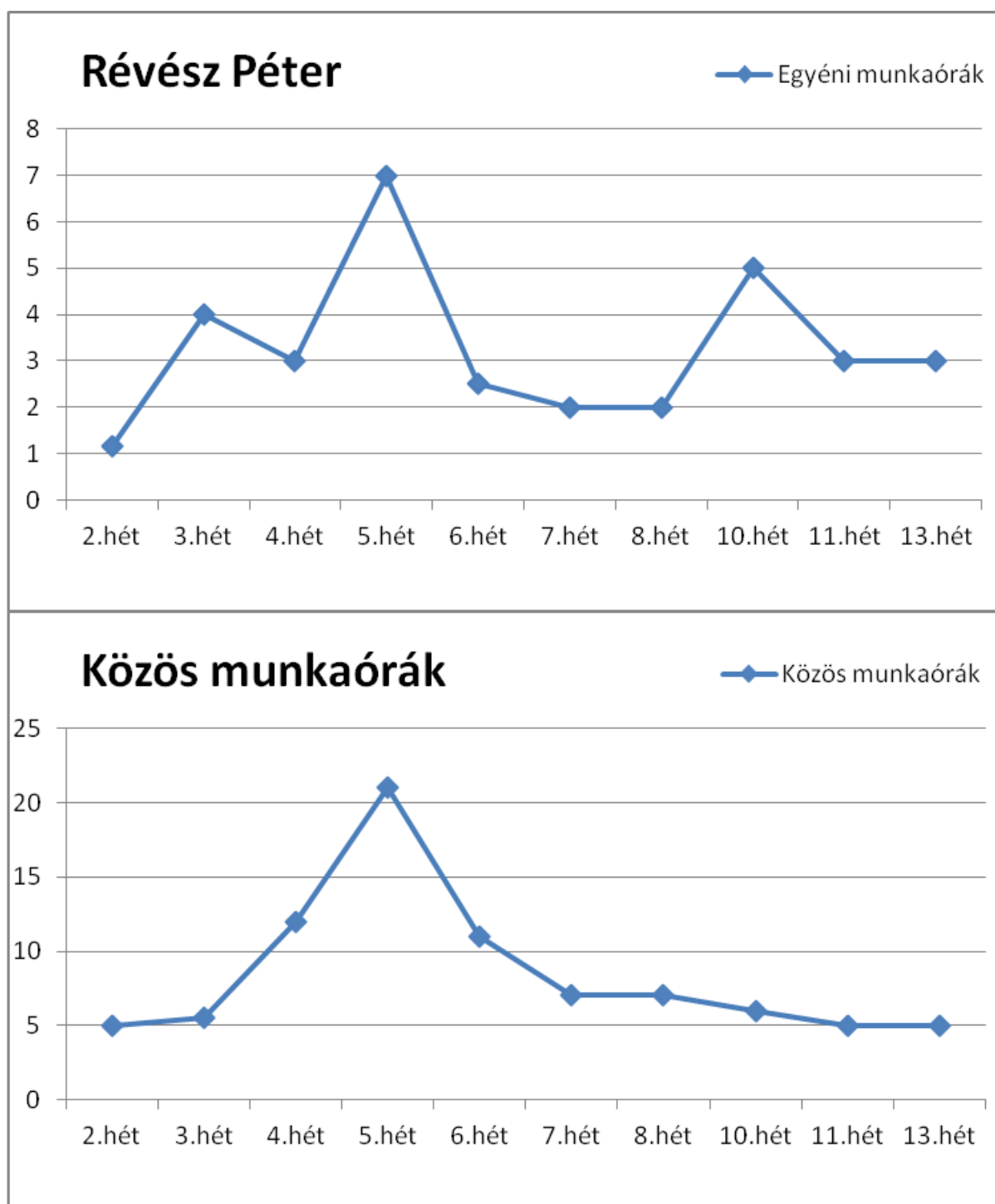
Név	Munka (perc)		Összesen
	Egyéni	Értekezlet	
Apáti-Nagy Attila	1795	5070	6865
Barabás Péter	1810	5070	6880
Boczán Tamás	1830	5070	7800
Csörgő Szabolcs	1815	5070	6885
Révész Péter	1960	5070	7030
Összesen	9210	25350	33660

13.1.2 Egyéni munkaidő órában

Név	Munka (óra)		Összesen
	Egyéni	Értekezlet	
Apáti-Nagy Attila	30	85	115
Barabás Péter	30	85	115
Boczán Tamás	31	85	116
Csörgő Szabolcs	30	85	115
Révész Péter	33	85	118
Összesen	153	422	561

13.1.3 Munkaidő diagramok





13.1.4 Végső összesítés:

Tag neve	Munka százalékban
Apáti-Nagy Attila	19.4%
Barabás Péter	19.8%
Boczán Tamás	19.8%
Csörgő Szabolcs	19.6%
Révész Péter	21.4%

13.2 A projekt során beadott forrásfájlok és azok sorainak a száma

13.2.1 Szkeleton

Fájl neve	Méret	Keletkezés ideje	Tartalom	Sorok száma
Barricade.java	1 KB	2014.03.23.	Az akadályokért felelős osztály megvalósítása.	14
Construct.java	1 KB	2014.03.23.	Az épületek ősosztályának megvalósítása.	13
ConstructManager.java	2 KB	2014.03.24.	Az épületek létrehozásáért és fejlesztéséért felelős osztály megvalósítása.	44
Dwarf.java	1 KB	2014.03.24.	A törpe típusú ellenségek kezeléséért felelős osztály megvalósítása.	6
Elf.java	1 KB	2014.03.23.	A tünde típusú ellenségek kezeléséért felelős osztály megvalósítása.	6
EndTile.java	1 KB	2014.03.23.	A Végzet Hegyét tároló csempe kezeléséért felelős osztály.	13
Enemy.java	3 KB	2014.03.24.	Az ellenségek ősosztályának megvalósítása.	72
EnemyGenerator.java	1 KB	2014.03.23.	Az ellenségek létrehozásáért felelős osztály.	21
FieldTile.java	1 KB	2014.03.23.	A terepcellákért felelős osztály megvalósítása, ezekre lehet tornyot építeni.	8
Geometry.java	2 KB	2014.03.23.	A pálya geometriai szerkezetéért felelős osztály megvalósítása.	23
Hobbit.java	1 KB	2014.03.23.	A hobbit típusú ellenséget kezelő osztály megvalósítása.	6
Human.java	1 KB	2014.03.23.	Az ember típusú ellenséget kezelő osztály megvalósítása.	6
MagicGem.java	1 KB	2014.03.23.	A varázskövekért felelős osztály megvalósítása.	14
Mana.java	1 KB	2014.03.24.	A játékos varázserejének kezeléséért felelős osztály megvalósítása.	17
PathGenerator.java	1 KB	2014.03.23.	Az útvonalakat kezelő osztály megvalósítása.	16
PathTile.java	2 KB	2014.03.24.	Az útvonalcellákért felelős osztály megvalósítása, ezekre lehet akadályt építeni, és	33

			ellenségek helyzetét adja meg.	
Tester.java	11 KB	2014.03.24.	A tesztesetek helyes lefutásáért felelős osztály megvalósítása.	282
Tile.java	1 KB	2014.03.24.	A csempék őszosztályának megvalósítása.	28
Tower.java	2 KB	2014.03.23.	A tornyokért felelős osztály megvalósítása.	39
Updater.java	4 KB	2014.03.24.	Az időkezelésért felelős osztály megvalósítása.	60
Összesen:				721

13.2.2 Prototípus

Fájl neve	Méret	Keletkezés ideje	Tartalom	Sorok száma
Barricade.java	579 byte	2014.04.21. 21:30	Az akadályokért felelős osztály megvalósítása.	15
Construct.java	399 byte	2014.04.21. 21:30	Az épületek őszosztályának megvalósítása.	10
ConstructManager.java	2 691 byte	2014.04.21. 21:37	Az épületek létrehozásáért és fejlesztéséért felelős osztály megvalósítása.	63
Dwarf.java	289 byte	2014.04.21. 21:30	A törpe típusú ellenségek kezeléséért felelős osztály megvalósítása.	9
Elf.java	284 byte	2014.04.21. 21:30	A tünde típusú ellenségek kezeléséért felelős osztály megvalósítása.	9
EndTile.java	708 byte	2014.04.21. 21:30	A Végzet Hegyét tároló csempe kezeléséért felelős osztály.	11
Enemy.java	3 639 byte	2014.04.21. 21:30	Az ellenségek őszosztályának megvalósítása.	75
EnemyGenerator.java	3 739 byte	2014.04.21. 21:32	Az ellenségek létrehozásáért felelős osztály.	79
FieldTile.java	441 byte	2014.04.21. 21:30	A terepcellákért felelős osztály megvalósítása, ezekre lehet tornyot építeni.	7
Geometry.java	3 031 byte	2014.04.21. 21:30	A pálya geometriai szerkezetéért felelős osztály megvalósítása.	59
Hobbit.java	255 byte	2014.04.21. 21:30	A hobbit típusú ellenséget kezelő osztály megvalósítása.	9
Human.java	250 byte	2014.04.21. 21:30	Az ember típusú ellenséget kezelő osztály megvalósítása.	9
MagicGem.java	538 byte	2014.04.21. 21:30	A varázskövekért felelős osztály megvalósítása.	12

Mana.java	924 byte	2014.04.21. 21:30	A játékos varázserejének kezeléséért felelős osztály megvalósítása.	18
PathGenerator.java	909 byte	2014.04.21. 21:30	Az útvonalakat kezelő osztály megvalósítása.	13
PathTile.java	1 544 byte	2014.04.21. 21:30	Az útvonalcellákért felelős osztály megvalósítása, ezekre lehet akadályt építeni, és ellenségek helyzetét adja meg.	28
PrototypeController.java	18 760 byte	2014.04.21. 21:38	A tesztelésért felelős osztály megvalósítása.	440
Tile.java	1 107 byte	2014.04.21. 21:30	A csempék ősosztályának megvalósítása.	29
Tower.java	3 004 byte	2014.04.21. 21:30	A tornyokért felelős osztály megvalósítása.	76
Updater.java	6 919 byte	2014.04.21. 21:30	Az időkezelésért felelős osztály megvalósítása.	128
Összesen:				1099

13.2.3 Grafikus változat

Fájl neve	Méret	Keletkezés ideje	Tartalom	Sorok száma
draw\BarricadeView.java	1235 byte	2014.05.12. 0:04	Az akadályokat rajzoló osztály megvalósítása.	19
draw\Controller.java	955 byte	2014.05.10. 22:58	Az inputot kezelő osztály megvalósítása.	28
draw\Drawable.java	1748 byte	2014.05.12. 0:04	A rajzoló osztályok absztrakt ősosztálya.	46
draw\DwarfView.java	1186 byte	2014.05.12. 0:04	A törpöket rajzoló osztály megvalósítása.	22
draw\ElfView.java	1145 byte	2014.05.12. 0:04	A tündéket rajzoló osztály megvalósítása.	22
draw\EndTileView.java	945 byte	2014.05.12. 0:04	A végzet hegyét rajzoló osztály megvalósítása.	17
draw\FieldTileView.java	983 byte	2014.05.12. 0:04	A terep csempéket rajzoló osztály megvalósítása.	17
draw\HobbitView.java	1209 byte	2014.05.12. 0:04	A hobbitokat rajzoló osztály megvalósítása.	22
draw\HumanView.java	1177 byte	2014.05.12. 0:04	Az embereket rajzoló osztály megvalósítása.	22
draw\PathTileView.java	968 byte	2014.05.12. 0:04	Az utcsempéket rajzoló osztály megvalósítása.	18
draw\TowerView.java	1743 byte	2014.05.11. 15:10	A tornyokat rajzoló osztály megvalósítása.	30

draw\View.java	12749 byte	2014.05.11. 19:44	Az ablakot rajzoló osztály megvalósítása.	258
game\Barricade.java	579 byte	2014.04.21. 21:30	Az akadályokért felelős osztály megvalósítása.	21
game\Construct.java	399 byte	2014.04.21. 21:30	Az épületek ősztyáának megvalósítása.	14
game\Construct Manager.java	2 691 byte	2014.04.21. 21:37	Az épületek létrehozásáért és fejlesztéséért felelős osztály megvalósítása.	66
game\Dwarf.java	289 byte	2014.04.21. 21:30	A törpe típusú ellenségek kezeléséért felelős osztály megvalósítása.	10
game\Elf.java	284 byte	2014.04.21. 21:30	A tünde típusú ellenségek kezeléséért felelős osztály megvalósítása.	10
game\EndTile.java	708 byte	2014.04.21. 21:30	A Végzet Hegyét tároló csempe kezeléséért felelős osztály.	12
game\Enemy.java	3 639 byte	2014.04.21. 21:30	Az ellenségek ősztyáának megvalósítása.	71
game\EnemyGenerator.java	3 739 byte	2014.04.21. 21:32	Az ellenségek létrehozásáért felelős osztály.	80
game\FieldTile.java	441 byte	2014.04.21. 21:30	A terepcellákért felelős osztály megvalósítása, ezekre lehet tornyot építeni.	8
game\Geometry.java	3 031 byte	2014.04.21. 21:30	A pálya geometriai szerkezetéért felelős osztály megvalósítása.	107
game\Hobbit.java	255 byte	2014.04.21. 21:30	A hobbit típusú ellenséget kezelő osztály megvalósítása.	10
game\Human.java	250 byte	2014.04.21. 21:30	Az ember típusú ellenséget kezelő osztály megvalósítása.	10
game\MagicGem.java	538 byte	2014.04.21. 21:30	A varázskövekért felelős osztály megvalósítása.	15
game\Mana.java	924 byte	2014.04.21. 21:30	A játékos varázserejének kezeléséért felelős osztály megvalósítása.	19
game\PathGenerator.java	909 byte	2014.04.21. 21:30	Az útvonalakat kezelő osztály megvalósítás.	183
game\PathTile.java	1 544 byte	2014.04.21. 21:30	Az útvonalcellákért felelős osztály megvalósítása, ezekre lehet akadályt építeni, és ellenségek helyzetét adja meg.	29
game\Tile.java	1 107 byte	2014.04.21. 21:30	A csempék ősztyáának megvalósítása.	30
game\Tower.java	3 004 byte	2014.04.21. 21:30	A tornyokért felelős osztály megvalósítása.	68

game\Updater.j ava	6 919 byte	2014.04.21. 21:30	Az időkezelésért felelős osztály megvalósítása.	106
Összesen:				1390

13.3 Projekt összegzés

13.3.1 Mit tanultak a projektből konkrétan és általában?

A feladat megoldása során végignézhettük egy program születését a tervezéstől egészen a megvalósításig. Alkalmazhattuk a Szoftvertechnológia tárgyból szerzett tudást ténylegesen a gyakorlatban, ezáltal sokkal jobban elmélyült bennünk a tudás. Láthattuk az egyes diagramok fontosságát, és azt is megtanultuk, hogy ha egy alkalmazás elég részletesen meg van tervezve, akkor az implementálás már tényleg csak 'téglapakolás'. Ami azonban talán a szakmai tudásnál is többet adott az a csoportmunka volt. Csapatunk egyik tagja sem dolgozott még komolyabb projecten csapatban, így mindnyájunknak komoly kihívást jelentett a feladat. A kommunikáció megszervezésével nem voltak különösebb gondok. Nagyon hatékonyá tudta tenni a munkát, hogy a gyűlések előre lefixált fix időpontban voltak, minden héten ugyanakkor. Így mindenki előre tudott vele számolni, és ha csak nem volt valami halaszthatatlan dolga, részt is tudott rajta venni. Csoportmunkánál nagyon fontos hogy a feladatok egyenlő mértékben legyenek elosztva, így senkinek se kelljen túlságosan sokat dolgozni a másikkal szemben. Pár leadandó után felfedeztük, hogy az egyes emberek miben jobbak, miben kevésbé jók, és ehhez mérten osztottuk el a feladatot. Van, aki kódoláshoz ért jobban, van, aki jól dokumentál, más a diagramok rajzolásában jeleskedik, és ezt felismerve igazán hatékonyá lehet tenni a munkát.

Tervezés folyamán voltak kisebb – nagyobb viták, ami más –más tervezői elképzelésekből fakadtak. Azonban a nagyobb viták tárgyáról utólag kiderült, hogy mind a két megoldás jó tervezői döntés lett volna. Ezáltal megtanulhattuk, hogy sokszor a másik fél véleménye is lehet ugyanolyan jó, mint a mienk, és ezzel a tapasztalattal talán sokkal toleránsabbak leszünk a másik véleménye iránt a jövőben.

Érzékelhető volt az is, hogy még a legegyszerűbbnek tűnő probléma is igazán bonyolult lesz, ha elkezdünk felette komolyabban gondolkodni, ezért soha nem szabad alulbecsülni semmilyen problémát.

Összességében tehát elmondható, hogy mind emberileg, mint szakmailag nagyon sokat fejlődtünk a félév során. Bár szerintünk a kreditszám nincs arányban a belefektetett munkával, egyikünk se bánta meg, hogy tárgyat hallgatta, hiszen a sok probléma megoldása által rengeteget tanult.

13.3.2 Mi volt a legnehezebb?

A legnagyobb nehézséget a feladat első, szkeleton része okozta. Már magában a tervezés se volt egyszerű feladat, hiszen minden részletre előre kellett gondolni, de a valódi problémát a csapat tapasztalatlansága okozta. Egyikünk se csinált még hasonlót ezelőtt és sok esetben nem tudtuk, hogy konkrétan mi is az elvárás az adott feladatban. Az objektumorientált szoftvertervezés elméleti háttére szinte ismeretlen volt még számunkra, így először mindenki a korábbi tárgyak során alkalmazott, gyakorlatorientált szemlélettel állt neki a tervezésnek, ami itt már nem volt célravezető.

13.3.3 Mi volt a legkönnyebb?

A grafikus felület elkészítése volt a legkönnyebb feladat, hiszen itt már működő programmal rendelkezünk, amihez csak a felületet kellett elkészíteni. Ennek felépítése pedig egyszerű és az eddig elkészített program által jól meghatározott volt. Rövid tervezés után már csak programozni kellett, ami mindenki számára szinte rutinszerűen ment. Továbbá a grafikus felület egyszerű és laza felépítése miatt könnyű volt a feladatok szétosztása is egymás között, ezeket a feladatokat pedig egymástól sokkal függetlenebbül tudtuk végezni, mint korábban a tervezési feladatok esetében.

13.3.4 Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?

A feladatok által ért pontszámok arányai lényegében igazodtak a beléjük fektetett idő arányával. Ami jelentősen eltért a befektetett munka mennyiségétől, az a tárgy kreditértéke. A beadandó feladatok elkészítése önmagában is sok időt vett igénybe, amit meetingek még tovább növeltek. A csapat és a tagok által külön-külön befektetett idő is jelentősen meghaladta azt az értéket, amit két kredit elméletileg igényelne. Emellett pedig a tárgy által lefedett tudásanyag is nagyobb és komplexebb, mint amennyi az ilyen kreditértékű tárgyaknál megszokott.

13.3.5 Milyen változtatási javaslatuk van?

Véleményünk szerint a tárgy ahhoz képest, hogy 2 kreditet ér ahhoz képest jóval több munka volt vele, mint a tárgy adatlapján leírt 60 óra/fő. A csapatból mindenki foglalkozott a tárgyal legalább 115 órát a félév során, ami majd nem eléri egy 4 kredites tárgyra fordítandó 120 órát, ezért a változtatási javaslatunk az lenne, hogy a tárgy kredit értéke ne 2 kredit legyen, hanem 3 vagy akár 4 kredit.

13.3.6 Milyen feladatot ajánlanának a projektre?

A táblás játék bemutatta az objektumorientált tervezés alapvető döntéseit, például a felelőségek megfelelő osztályhoz rendelését. Ötlet esetleges projektre: Lehetne olyan társasjátékhoz hasonló játék megírása a cél, amiben nem a gép az ellenfél, hanem a játékosok egymás ellen játszhatnak, minden körben sorban léphetnek egymás után. Ebben is lehetne valamilyen gép által generált változó, például dobókocka, vagy gépi ellenfelek.

14. Napló

2.hét

Kezdet	Időtartam	Résztevők	Leírás
2014.02.20. 17:00	2 óra	Apáti-Nagy Barabás Boczán Csörgő Révész	Feladatok kiosztása, megbeszélése. A feladat alapjainak kidolgozása.
2014.02.21. 18:00	1 óra 10 perc	Barabás	Tevékenység: kommunikációs és dokumentumkezelő technológiák beüzemelése, adatszótár elkészítése, gyűlésszervezés.
2014.02.21. 21:00	1 óra 10 perc	Révész	Tevékenység: Projekt terv elkészítése.
2014.02.22. 15:00	1 óra	Boczán	Tevékenység: Funkcionális követelmények és use-case-ek.
2014.02.22. 19:00	1 óra 45 perc	Csörgő	Tevékenység: Áttekintés: Funkciók rész megírása.
2014.02.22. 19:00	1 óra	Apáti-Nagy	Tevékenység: Bevezetés és általános áttekintés elkészítése.
2014.02.23. 17:00	3 óra	Apáti-Nagy Barabás Boczán Csörgő Révész	Egyéni részek közös átnézése. Hiányzó részek elkészítése.

3. hét

Kezdet	Időtartam	Résztevők	Leírás
2010.02.25. 17:40	2 óra	Apáti-Nagy Barabás Boczán Csörgő Révész	Értekezlet. Osztály struktúra első közelítése, feladatok kiosztása, kérdések összegyűjtése konzultációra
2014.02.26. 16:00	3 óra	Barabás	Tevékenység: Osztálydiagram megtervezése és elkészítése
2014.02.26. 16:00	4 óra	Révész	Tevékenység: Osztálydiagram megtervezése és objektum katalógus elkészítése. Osztályok leírásának elkészítése
2014.03.01. 18:00	1 óra 30 perc	Csörgő	Tevékenység: Szekvenciadiagramok rajzolása
2014.03.01. 19:00	1 óra 30 perc	Boczán	Tevékenység: Szekvenciadiagramok
2014.03.02. 12:00	45 perc	Apáti-Nagy	Tevékenység: State-chartok elkészítése
2014.03.02. 17:00	3 óra 30 perc	Apáti-Nagy Barabás Boczán Csörgő Révész	Értekezlet. Egyéni feladatok összehangolása, felépítésben talált hibák kijavítása.

4. hét

Kezdet	Időtartam	Résztevők	Leírás
2014.03.06. 17:00	4 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Értekezlet. Döntés: Révész elkészíti az osztálydiagramot, Boczán és Csörgő a szekvencia diagramokat, Apáti-Nagy és Barabás a formális és informális leírásokat.
2014.03.07. 18:00	5 óra	Boczán	Tevékenység: Szekvencia diagramok elkészítése
2014.03.08. 12:00	5 óra	Csörgő	Tevékenység: Szekvencia diagramok elkészítése
2014.03.08. 14:00	4 óra	Barabás	Tevékenység: Informális leírás elkészítése, eddigi munkák ellenőrzése, összehangolása
2014.03.08. 16:00	3 óra	Révész	Tevékenység: Osztálydiagram elkészítése
2014.03.08. 18:00	5 óra	Apáti-Nagy	Tevékenység: Formális leírás elkészítése
2014.03.09. 15:00	8 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Részfeladatok összehangolása, hibák kijavítása.

5. hét**Analízis modell 3.**

Kezdet	Időtartam	Résztevők	Leírás
2014.03.12. 17:00	4 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Értekezlet. Döntés: Révész elkészíti az osztálydiagramot, Boczán és Csörgő a szekvencia diagramokat, Apáti-Nagy és Barabás a formális és informális leírásokat.
2014.03.12. 21:00	3 óra	Boczán	Tevékenység: Szekvencia diagramok elkészítése
2014.03.13. 12:00	3 óra	Csörgő	Tevékenység: Szekvencia diagramok elkészítése
2014.03.13. 14:00	3 óra	Barabás	Tevékenység: Informális leírás elkészítése, eddigi munkák ellenőrzése, összehangolása
2014.03.14. 16:00	3 óra	Révész	Tevékenység: Osztálydiagram elkészítése
2014.03.14. 20:00	3 óra	Apáti-Nagy	Tevékenység: Formális leírás elkészítése
2014.03.15. 20:00	8 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Részfeladatok összehangolása, hibák kijavítása.

Szkeleton terv

Kezdet	Időtartam	Résztevők	Leírás
2014.03.13. 17:00	3 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Értekezlet. Döntés: Apáti-Nagy elkészíti a use-case-eket, Boczán és Barabás a teszteseteket és a szekvencia diagramokat. Révész és Csörgő elkészíti a szkeleton kezelő felületének tervét és a kommunikációs diagramokat.
2014.03.14. 16:00	4 óra	Apáti-Nagy	Tevékenység: Use-case diagram és use-case leírások elkészítése
2014.03.14. 21:00	5 óra	Boczán	Tevékenység: Tesztesetek definiálása, szekvencia diagramok elkészítése
2014.03.15. 15:00	4 óra	Barabás	Tevékenység: Szekvencia diagramok elkészítése
2014.03.15. 16:00	4 óra	Révész	Tevékenység: Kommunikációs diagramok elkészítése
2014.03.16. 10:00	4 óra	Csörgő	Tevékenység: A szkeleton kezelői felületének megtervezése, kommunikációs diagramok elkészítése
2014.03.16. 15:00	6 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Részfeladatok összehangolása, hibák kijavítása.

6. hét

Kezdet	Időtartam	Résztevők	Leírás
2014.03.20. 17:00	4 óra	Révész Boczán Apáti-Nagy Barabás Csörgő	Értekezlet. Döntés: Boczán elkészíti az 5., 10. teszteseteket, Révész elkészíti a tesztesetek keretrendszerét, és a 7., 8. teszteseteket. Apáti-Nagy elkészíti a 3., 4. teszteseteket. Barabás elkészíti a 1., 9. tesztesetet Csörgő elkészíti a 2., 6. teszteseteket.
2014.03.22. 17:00	2 óra	Boczán	Tevékenység: Tesztesetek implementálása
2014.03.22. 19:00	2 óra 30 perc	Révész	Tevékenység: Tesztesetek implementálása
2014.03.22. 22:00	3 óra	Barabás	Tevékenység: Tesztesetek implementálása
2014.03.23. 11:00	2 óra	Apáti-Nagy	Tevékenység: Tesztesetek implementálása
2014.03.23. 12:00	3 óra	Csörgő	Tevékenység: Tesztesetek implementálása. Fájllista, fordítási és futtatási útmutató elkészítése.
2014.03.23. 16:00	7 óra	Révész Boczán Apáti-Nagy Barabás Csörgő	Kódrészletek összehangolása, elnevezések és kiírások egységesítése, hibák kijavítása.

7. hét

Kezdet	Időtartam	Résztevők	Leírás
2014.03.27. 17:00	4 óra	Révész Boczán Apáti-Nagy Barabás Csörgő	Értekezlet. Döntés: Boczán elkészíti a változtatások dokumentálását, Révész és Barabás elkészíti a be- és kimeneti nyelvet. Apáti-Nagy elkészíti a Use-Case-eket és a teszteseteket. Csörgő elkészíti a definíciókat és leírásokat.
2014.03.28. 17:00	2 óra	Révész	Tevékenység: Bemeneti nyelv specifikációja
2014.03.29. 18:00	3 óra	Boczán	Tevékenység: Változtatások dokumentálása
2014.03.29. 20:00	2 óra	Barabás	Tevékenység: Kimeneti nyelv specifikációja
2014.03.30. 7:00	2 óra	Apáti-Nagy	Tevékenység: Use-Case-ek és tesztesetek elkészítése
2014.03.30. 10:00	2 óra	Csörgő	Tevékenység: Interface definíció, leírás és segédprogramok specifikálása
2014.03.30. 16:00	3 óra	Révész Boczán Apáti-Nagy Barabás Csörgő	Feladatok összeszerkesztése, ellenőrzése.

8. hét

Kezdet	Időtartam	Résztvevők	Leírás
2014.04.03. 17:00	3 óra	Révész Boczán Apáti-Nagy Barabás Csörgő	Értekezlet. Döntés: Boczán és Csörgő elkészíti az osztályok terveit, Csörgő elkészíti a tesztelést támogató programok terveit. Révész és Apáti-Nagy elkészíti a tesztek részletes terveit, Barabás elkészíti a módosított be- és kimeneti nyelvet.
2014.04.04. 17.30	2 óra	Barabás	Tevékenység: Ki- és bemeneti nyelv módosítása
2014.04.05. 20:00	2 óra	Révész	Tevékenység: Tesztek részletes terveinek elkészítése
2014.04.05. 20:00	3 óra	Apáti-Nagy	Tevékenység: Tesztek részletes terveinek elkészítése
2014.04.06. 10:00	2 óra	Csörgő	Tevékenység: Tesztelést támogató programok terveinek elkészítése, osztályok és metódusainak terveinek elkészítése
2014.04.06. 12:00	2 óra	Boczán	Tevékenység: Osztályok és metódusainak terveinek elkészítése
2014.04.06. 17:00	4 óra	Révész Boczán Apáti-Nagy Barabás Csörgő	Feladatok összeszerkesztése, ellenőrzése.

10. hét

Kezdet	Időtartam	Résztevők	Leírás
2014.04.10. 17:00	2 óra	Révész Boczán Apáti-Nagy Barabás Csörgő	Értekezlet. Döntés: Boczán elkészíti az Updatert és EnemyGeneratort, Révész elkészíti a PrototypeControllert és az Enemy osztályokat. Apáti-Nagy elkészíti a Geometry és Tile osztályokat, Barabás elkészíti a ConstructManagert és PathGeneratort, Csörgő elkészíti a Construct osztályokat és a MagicGem osztályt. Csörgő elkészíti a "Prototípus beadása" dokumentumot.
2014.04.14. 17.30	3 óra	Barabás	Tevékenység: Osztályok implementálása, részletes tervek módosításának elkészítése
2014.04.16. 22:00	5 óra	Révész	Tevékenység: Osztályok implementálása
2014.04.17. 21:00	3 óra	Apáti-Nagy	Tevékenység: Osztályok implementálása
2014.04.17. 18:00	2 óra	Csörgő	Tevékenység: Osztályok implementálása, dokumentáció részleges elkészítése
2014.04.18. 16:00	3 óra	Boczán	Tevékenység: Osztályok implementálása
2014.04.21. 17:00	4 óra	Révész Boczán Apáti-Nagy Barabás Csörgő	Feladatok összeszerkesztése, ellenőrzése. Dokumentáció befejezése, beadandók tesztelése.

11. hét

Kezdet	Időtartam	Résztevők	Leírás
2014.04.24. 17:00	2 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Értekezlet. Döntés: Apáti-Nagy elkészíti a struktúradiagramot, Boczán és Csörgő a szekvencia diagramokat. Révész elkészíti a grafikus interfészt és architektúrát, Barabás elkészíti a grafikus objektumok felsorolását.
2014.04.25. 14:00	2 óra	Boczán	Tevékenység: Szekvencia diagramok elkészítése
2014.04.25. 19:00	2 óra	Csörgő	Tevékenység: Szekvencia diagramok elkészítése
2014.04.26. 12:00	2 óra	Barabás	Tevékenység: A grafikus objektumok felsorolásának elkészítése
2014.04.26. 17:00	3 óra	Révész	Tevékenység: A grafikus interfész és architektúra elkészítése
2014.04.27. 11:00	3 óra	Apáti-Nagy	Tevékenység: Struktúradiagram elkészítése
2014.04.27. 16:00	3 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Részfeladatok összehangolása, ellenőrzése.

13. hét

Kezdet	Időtartam	Résztevők	Leírás
2014.05.01. 17:00	2 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Értekezlet. Döntés: Apáti-Nagy és Csörgő elkészíti a rajzoló osztályokat, Boczán elkészíti a Controller és Drawable osztályokat. Révész elkészíti a View osztályt, Barabás elkészíti a dokumentációt.
2014.05.04. 16:00	3 óra	Révész	Tevékenység: View implementálása
2014.05.07. 19:00	3 óra	Boczán	Tevékenység: Controller, Drawable implementálása
2014.05.08. 16:00	3 óra 10 perc	Apáti-Nagy	Tevékenység: ElfView, DwarfView, HobbitView, HumanView, EndTileView elkészítése
2014.05.08. 16:00	4 óra	Csörgő	Tevékenység: FieldTileView, PathTileView, TowerView BarricadeView elkészítése
2014.05.18. 12:00	3 óra	Barabás	Tevékenység: Dokumentáció elkészítése
2014.05.18. 16:00	3 óra	Révész Boczán Csörgő Apáti-Nagy Barabás	Részfeladatok összehangolása, ellenőrzése. Játékprogram tesztelése.