

Összefoglalás

52 – Nagymama lekvárjai

Konzulens:

Dr. Goldschmidt Balázs

Csapattagok

Lipták Levente
Szepes Nóra
Varga Gergő
Sebők Márton

HKTMJ1
N2WIN2
KG3LTA
G78SPI

liptaklevente@gmail.com
lordblendi@gmail.com
gvarga45@gmail.com
sebokmarton@gmail.com

2015-04-14

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

Ennek a dokumentumnak a célja, hogy a félév során a Szoftver Laboratórium 4 tárgy keretein belül készülő projekt megvalósulását részletesen leírja, bemutassa. A dokumentumban kitérünk a projekt elkészítésének minden egyes szakaszára. A fejlesztőcsapat kitűzött célja, hogy egy olyan programot hozzon létre, mely a specifikáció szerint van elkészítve, és hibátlanul fut a felsorolt erőforrásokkal rendelkező környezetben.

2.1.2 Szakterület

A szoftver egy szórakozásra szánt stratégiai játék, melynek célja a játékos logikai és stratégiai készségének a fejlesztése.

2.1.3 Definíciók, rövidítések

- **MVC** – Modell-View-Controller, Modell-nézet-vezérlő. A szoftvertetvezésben használatos szerkezeti minta. Elkülöníti az adatok elérését és az üzleti logikát az adatok megjelenítésétől és a felhasználói interakciótól egy közbülső összetevő, a vezérlő bevezetésével.
- **UML** – Unified Modeling Language. Szabványos, általános célú modellező nyelv, melynek segítségével szöveges és grafikus modelleket készíthetünk például a szereplőkről, folyamatokról, logikai összetevőkről és azok viselkedéséről, feladatáról.
- **RUP** – Rational Unified Process. Iteratív szoftverfejlesztési folyamat, mely a fejlesztés strukturálásáról, a projekt menedzselhetőségéről és a hibák kiszűréséből áll.
- **GIT** – Nyílt forráskódú, elosztott verziókezelő szoftver. Ennek segítségével tudjuk vezetni, hogy ki és milyen változásokat készített el a szoftver programkódjának írása során.
- **OOP** – Object-Oriented Programming, Objektumorientált Programozás. Egy programozási módszertan. Itt nem a műveletek megalkotása, hanem az egymással kapcsolatban álló programegységek hierarchiájának megtervezése áll a középpontban.
- **FPS** – Frame Per Second, egy szám, ami megmutatja, hogy másodpercenként hány képkocka jelenik meg.

2.1.4 Hivatkozások

- Fejlesztéshez használt szoftverek:
 - <http://www.eclipse.org/>
 - <https://bitbucket.org/>
 - <http://www.visual-paradigm.com/>
- Feladatkiírás és anyaggyűjtés:
 - <https://www.iit.bme.hu/~szoftlab4/>
 - <https://www.iit.bme.hu/~stuser/>
 - <http://directory.iit.bme.hu/belso/st/stbelso.html>
 - <http://hu.wikipedia.org/>
 - <http://en.wikipedia.org/>
- Kapcsolattartás, jelenlét és közös dokumentáció:

- <https://onedrive.live.com/>
- <http://doodle.com/>
- <https://groups.google.com/>
- <https://mail.google.com/>

2.1.5 Összefoglalás

A pontos terveink a dokumentum további részében olvashatóak.

Főbb fázisok a fejlesztés során:

- **Analízis modell** – minden osztály, interfész ismertetésre fog kerülni. Vagyis leírjuk ezeknek a kapcsolatait, attribútumaikat, metódusaikat. Ezenkívül diagramok, pl.: osztály, szekvencia, use-case segítségével fogjuk bemutatni a készítendő szoftver működését.
- **Szkeleton** – A szkeleton a program egy olyan verziója, melyben megtalálhatóak a végső rendszerben is szereplő objektumok. Ez esetünkben egy szöveges kimenettel működő verzió melyben a metódusok meghívhatóak. Futásuk során kiírják a nevüket, valamint meghívják azokat a metódusokat, melyeket a végrehajtás érdekében meg kell majd hívni a végleges verzióban.
- **Prototípus** – Azon verzió, melynek működése megegyezik a grafikus verzióval, viszont a grafikus megjelenés még hiányzik belőle. A működéshez szükséges metódusok már a végleges kódot tartalmazzák, kivéve azokat a részeket, amik a megjelenítéshez szükségesek.
- **Grafikus változat** – A prototípustól működésben nem eltérő változat. A két változat között a különbség egyedül a megjelenésben van. A grafikus változat során az eddig elkészített programhoz elkészítjük a program külalakját.

2.2 Áttekintés

2.2.1 Általános áttekintés

A rendszer az MVC architektúrát követi, miszerint a Model tartalmazza az alkalmazáslogikát (az adatokat és az ezeken végezhető műveleteket), a View felelős a megjelenítésért (az adatokat a Model-ből olvasva ki), a Controller pedig a felhasználói interakciókat kezeli és azokat továbbítja a Model-nek vagy a View-nak. Ennek megfelelően jól definiált interfész kell, hogy legyen az egyes részek között, hogy azok adott esetben cserélhetők legyenek, így biztosítva a program moduláris felépítését.

2.2.2 Funkciók

A gonosz emberek, tündék, törpök és hobbitok szövetséget kötnek, hogy elpusztítsák az Egy Gyűrűt a Végzet Hegyénél. Szerencsére csak Mordor földjén keresztül tudnak eljutni a hegyhez, így jószágos Szarumánnak lehetősége van védelmi tornyokat építeni, hogy segítsen megvédeni Szauron hatalmát. A játék célja annak megakadályozása, hogy a Gyűrű szövetségének tagjai közül bárki is eljusson a Végzet Hegyéhez. Egy ellenség akkor pusztul el, ha összességében megfelelő mértékű sebzést kap a tornyokból származó lövedékektől. A tornyok építéséhez Szarumánnak a varázserejét kell használnia. Szarumán akkor tud tornyot építeni, ha megfelelő mennyiségű varázsereje van hozzá. A varázsereje minden egyes elpusztított ember, tünde, törp vagy hobbit után bizonyos mértékben növekszik.

A Gyűrű szövetségének tagjai különböző utakon juthatnak el a Végzet Hegyéhez. Az utakról nem térhetnek le. Szarumán az utakra nem tud tornyot építeni, csak az utak mellé. Az utakra azonban tehet akadályokat, amik az akadály területén lassítják az ellenség haladását. A

tornyoknak van egy adott hatótávolsága és tüzelési gyakorisága. Szarumán a varázserejét arra is használhatja, hogy a tornyokat és akadályokat különböző varázskövekkel ruházza fel. A varázsköveknek több fajtája is létezik, és különböző hatásúak lehetnek. Egyes kövek növelhetik a tornyok hatótávolságát vagy tüzelési gyakoriságát, más kövek egy-egy típusú ellenség esetén megnövelik a lövedékek sebzési erejét.

A játék során az ellenségek folyamatosan jönnek. A játék elején ritkábban, később gyakrabban és nagyobb csoportokban, azonban számuk véges, előbb-utóbb elfogynak. A játék akkor ér véget, ha egy ellenség eljut a Végzet Hegyéhez, vagy ha már sikerült az összes ellenséget kiirtani. Az első esetben Szauron és Szarumán megsemmisül, utóbbi esetben fényes győzelmet aratnak és örökké uralni fogják a világot.

A játékos Szarumánt személyesíti meg. A pálya két dimenziós, négyzetrácsos szerkezetű, alapvetően kétféle cellát különböztetünk meg: az utakat és a domborzatot. Az ellenségek (emberek, tündék, törpök és hobbitok) az utakon tartózkodhatnak, egy cellán egyszerre többen is.

Az utak celláira még akadályok helyezhetők el, cellánként egy darab, ezek a rajtuk áthaladó ellenséget a következőképpen lassítják:

- az egyszerű akadály 50%-kal,
- a sárga varázskő multiplikatíván további 40%-kal,
- egy akadályra pontosan egy darab varázskő helyezhető el.

Az akadályokra tehát csak egyféle varázskő hatásos, ami a fenti módon a lassítást növeli.

Az egyik út cellában található a Végzet Hegye, erre akadály nem helyezhető el.

A pályát külön fájlban tároljuk, melyet a játék indításakor olvasunk be. Ebben az út, domborzat és Végzet Hegye „típusú” cellák vannak megkülönböztetve.

A domborzat celláira csak tornyok helyezhetők, cellánként egy darab. A tornyok működése a következőképpen történik:

- a tornyok adott gyakorisággal sebzik a hozzájuk legközelebb álló ellenséget, amennyiben az a hatótávolságon belül tartózkodik,
- ha több „legközelebbi” ellenség is van, ezek közül a torony pontosan egyet fog kiválasztani,
- a sebzés pillanatszerűen történik,
- a sebzés értéke a sebzett ellenség életpontjaiból kerül levonásra a tornyon elhelyezett varázsköveknek megfelelően.

A tornyok működését a következő varázskövek befolyásolják:

- a piros varázskő a hozzá tartozó fajnál megnöveli a sebzés értékét additívan az eredeti érték 40%-ával (tehát a fajok számának megfelelően négyféle piros varázskő létezik),
- a kék varázskő az aktuális hatótávolságot multiplikatíván 10%-kal növeli,
- a zöld varázskő a sebzési gyakoriságot növeli multiplikatíván 25%-kal, de a tornyok sebzési gyakorisága nem lehet nagyobb az eredeti érték 5-szörösénél,
- egy toronyra pontosan egy darab varázskő helyezhető el.

A tornyokat és akadályokat együttesen tereptárgyaknak nevezzük.

Az ellenségek életpontjai egész számokként tárolódnak (nem egész értékű sebzésnél kerekítésre kerülnek), ha 1 alá csökkennek, az ellenség halottnak tekintendő. Az ellenségek fajonként eltérő kezdő életponttal rendelkeznek.

Az ellenségek csökkenő időközönként, növekvő csoportokban érkeznek. A csoportok összetétele változó, egy csoportban több fajból is szerepelhetnek. Ezek a csoportok egy véletlenszerűen kiválasztott útvonal kezdőpontjáról indulnak.

A játék először az ellenségeket lépteti az útvonalakon (úgy, hogy azok a legrövidebb úton tartsanak a Végzet Hegye felé, az ellenségek átkelési sebességét egy cellán a rajta lévő akadály befolyásolja), majd végrehajtja a sebzést a tüzelésre kész tornyokból. Ezt a két műveletet ismétli és közben lehetőséget ad a felhasználói beavatkozásra.

A pontos számadatok a későbbi tesztelesek során változhatnak a jobb felhasználói élmény érdekében.

2.2.3 Felhasználók

A program egyszerre egy felhasználó interakcióit fogadja valós időben a View-ban specifikált módon.

2.2.4 Korlátozások

A program grafikus változata legalább 20 FPS képfrissítési sebességgel dolgozzon, a felhasználói beavatkozást azonnal képes legyen feldolgozni. A programnak a valós idejű játék érzetét kell keltenie.

2.2.5 Feltételezések, kapcsolatok

A Szoftver labor 4 tantárgyi oldaláról (<https://www.iit.bme.hu/~szoftlab4/>) származó információkat használtuk fel.

2.3 Követelmények

2.3.1 Funkcionális követelmények

#	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
F1	a játékos új játékot indíthat	tesztelés során kiértékelés	alapvető	csapat	új játék indítása	a pályához tartozó fájl betöltése és feldolgozása
F2	a játékos képes tereptárgyakat elhelyezni	tesztelés során kiértékelés	fontos	megren delő	torony elhelyezése, akadály elhelyezése	figyelembe kell venni, hogy a pálya adott cellájára helyezhet-e el adott tereptárgyat
F3	a játékos képes varázskövekkel ellátni a tereptárgyakat	tesztelés során kiértékelés	fontos	megren delő	varázskő elhelyezése toronyra, varázskő elhelyezése	figyelembe kell venni, hogy az adott tereptárgyra elhelyezhető-e

					akadályra	adott típusú varázskő, a varázskő hatását érvényesíteni kell a tereptárgyon
F4	a játékprogram képes az ellenségeket mozgatni és a tornyokból sebést leadni	tesztelés során kiértékelés	fontos	megrendelő	idő léptetése	a 2.2.2 Funkcióknál specifikált működés szerint

2.3.2 Erőforrásokkal kapcsolatos követelmények

#	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
E1	egységes fejlesztőkörnyezet	fejlesztés során bemutatás	fontos	csapat	JRE 1.7 és annak rendszerkövetelményei teljesüljenek
E2	verziókezelés	fejlesztés során bemutatás	opcionális	csapat	megkönnyíti a csapattagok munkájának nyomon követését, a fejlesztési menet átlátását
E3	online együttműködés	dokumentálás során bemutatás	opcionális	csapat	a dokumentáció megosztott szerkesztése a OneDrive rendszerben megkönnyíti az összefűzést

2.3.3 Átadással kapcsolatos követelmények

#	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
A1	USB-port vagy internetkapcsolat	átadás előtt bemutatás	alapvető	csapat	a termék eljuttatásához a célszámítógépre
A2	JRE 1.7 megléte és rendszerkövetelményeinek teljesülése	átadás előtt bemutatás	alapvető	csapat	http://docs.oracle.com/java-se/7/docs/webnotes/install/windows/windows-system-requirements.html

2.3.4 Egyéb nem funkcionális követelmények

#	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
N1	pontos dokumentáltság,	a dokumentációk leadásakor	alapvető	megrendelő	a terméknek követnie kell a

	modellhűség	kiértékelés, visszamenőlegesen is			dokumentációban specifikáltakat, hiszen a fejlesztést nagy mértékben megkönnyíti és az értékelés is ez alapján történik, a RUP módszertant használjuk
N2	modularitás	fejlesztés során kiértékelés	fontos	csapat	MVC architektúrában fejlesztünk
N3	kód kommentezése	fejlesztés során kiértékelés	fontos	csapat	Javadoc szintaktika használata
N4	hordozhatóság	fejlesztés során kiértékelés	opcionális	csapat	ne írjunk platformspecifikus kódot, így hagyjuk meg a más platformokon való futtatás lehetőségét

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	új játék indítása
Rövid leírás	a játékos új játékot indít
Aktorok	játékos
Forgatókönyv	a pályához tartozó fájl betöltése és feldolgozása

Use-case neve	torony elhelyezése
Rövid leírás	a játékos tornyot helyez el
Aktorok	játékos
Forgatókönyv	a játékos elhelyezi a tornyot az általa kiválasztott olyan domborzat cellába, ahol még nincs torony

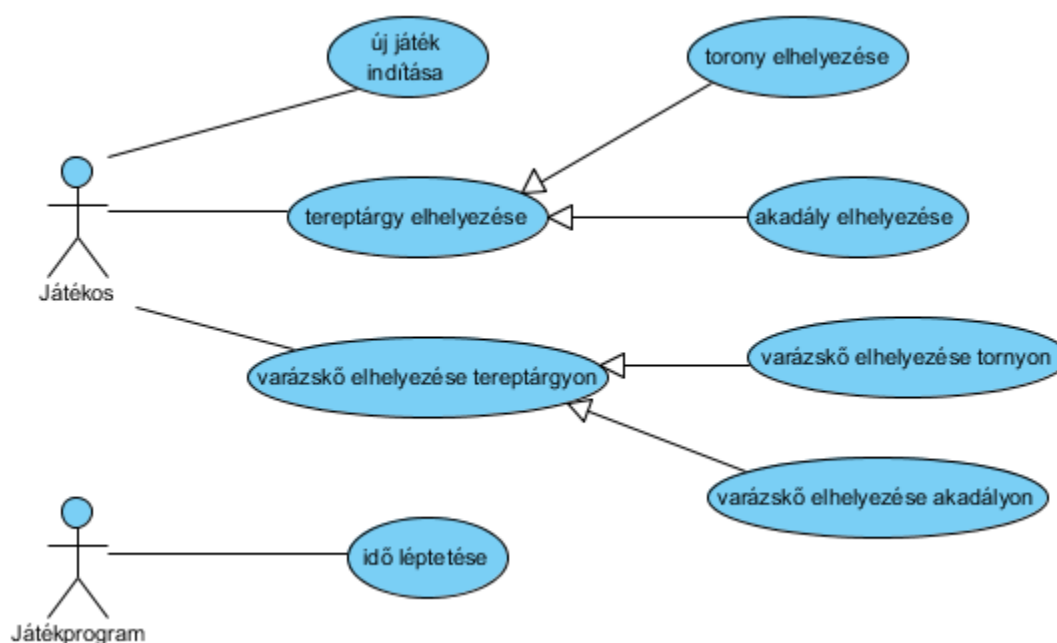
Use-case neve	akadály elhelyezése
Rövid leírás	a játékos akadályt helyez el
Aktorok	játékos
Forgatókönyv	a játékos elhelyezi az akadályt az általa kiválasztott olyan út cellára, ami nem a Végzet Hegye és ahol még nincs akadály

Use-case neve	varázskő elhelyezése toronyra
Rövid leírás	a játékos varázskövet helyez el egy meglévő toronyra
Aktorok	játékos
Forgatókönyv	a játékos elhelyezi az általa kiválasztott toronyra az általa kiválasztott piros, kék vagy zöld varázskövet, megváltoztatva ezzel a torony működését

Use-case neve	varázskő elhelyezése akadályra
Rövid leírás	a játékos varázskövet helyez el egy meglévő akadályra
Aktorok	játékos
Forgatókönyv	a játékos elhelyez egy sárga varázskövet az általa kiválasztott akadályra, megváltoztatva ezzel az akadály működését

Use-case neve	idő léptetése
Rövid leírás	a játékprogram végrehajtja az idő léptetésével járó feladatokat
Aktorok	játékprogram
Forgatókönyv	a játékprogram az ellenségeket mozgatja, illetve sebzést ad le a tornyokból a 2.2.2 Funkcióknál specifikált működés szerint

2.4.2 Use-case diagram



2.5 Szótár

Kifejezés	Magyarázat
játékos	A felhasználó.
ellenség	A térképen keresztülsétálni próbáló lények, feladatunk az ő elpusztításuk.
tünde	Ellenség faj.
törp	Ellenség faj.
ember	Ellenség faj.
hobbit	Ellenség faj.
faj	Négy ellenségtípust különböztetünk meg, melyek a történet szerint más fajhoz tartoznak és kissé különböző képességekkel bírnak. A négy faj: tünde, törpe, ember, hobbit.
Végzet Hegye	Az ellenségek célja ide eljutni, a mi célunk megakadályozni

egy gyűrű	Az ellenségek ezt akarják elpusztítani, csak a történet szempontjából fontos, a játékban nincs kitüntetett szerepe
Mordor földje	A pálya, amin keresztül az ellenségek el akarnak a végzet hegyéhez jutni.
Szarumán	A játékos az ő szerepében játszik.
védelmi torony	A játékos által építhető elem, mely sebzi az ellenségeket, így azok elpusztulnak, mire a védendő végzet hegyéhez érnének.
Szauron	A történet szerint Szarumán ura, a játékban nincs szerepe, csak a történetben.
Gyűrű szövetsége	Az ellenségek: tündék, törpök, emberek, és hobbitok.
út	Az egyik területtípus, melyre az ellenségek ráléphetnek.
domborzat	A másik területtípus, melyre az ellenségek nem léphetnek rá.
torony	A domborzatra építhető játékelem, ami sebzi az körülötte elhaladó ellenfelek közül az egyiket.
sebzés	A tornyok bizonyos időnként csökkentik a hozzájuk legközelebbi ellenség életpontjait, amennyiben azok a hatótávolságukon belül tartózkodnak. A csökkenés mértéke adott, de varázskövel módosítható.
tüzelés	A torony sebzése, más néven.
akadály	Az útra építhető játékelem, mely lassítja a rajta áthaladó ellenségeket.
tereptárgy	Torony és akadály közös néven.
varázskövek	Olyan játékelemek mellyel a tornyok és akadályok hatása erősíthető.
cella	A térkép a kezelhetőség érdekében egyenlő méretű, négyzet alakú részekre, cellákra van osztva, melyek vagy teljes részben út típusú, vagy teljes részben domborzat típusú területet fednek le.
piros varázskő	Egy toronyra helyezhető varázskő, mely a torony sebzését növeli.
kék varázskő	Egy toronyra helyezhető varázskő, mely a torony hatótávolságát növeli.
zöld varázskő	Egy toronyra helyezhető varázskő, mely a torony sebzési gyakoriságát növeli.
sárga varázskő	Egy akadályra helyezhető varázskő, mely az akadály hatásfokát növeli.
életpont	Az ellenségek egy pontszáma, mely kezdetben adott, fajonként különböző lehet, a tornyok sebzésével csökken, ha 0-ára (vagy az alá) csökkenne a sebzés által, az ellenfél elpusztul.

a játék vége	Ha egy ellenség eljut a Végzet Hegyéhez, a játékos veszít, ha minden ellenség elpusztul, a játékos nyer.
--------------	--

2.6 Projekt terv

2.6.1 Projekt lépései, határidők

A feladat elkészítése három fő részben történik:

- Szkeleton
 - határidő: 2014. március 24.
- Prototípus
 - határidő: 2014. április 22.
- Grafikus változat
 - határidő: 2014.május 12.

A Szoftver Labor 4 oldaláról az ütemterv.

	Dátum	Jelleg
1	febr. 12.	8:00 fakultatív eligazító
2	febr. 19.	konzultáció
3	febr. 26.	konzultáció
4	márc. 5.	konzultáció
5	márc. 12.	konzultáció
6	márc. 19.	konzultáció
7	márc. 26.	szkeleton bemutató - konzultáció
8	ápr. 2.	konzultáció
9	ápr. 9.	konzultáció
10	ápr. 16.	konzultáció
11	ápr. 23.	protó bemutató - konzultáció
12	ápr. 30.	munkaszünet
13	máj. 7.	konzultáció
14	máj. 14.	grafikus bemutató - konzultáció

A konzultációra minden tag elmegy, aki tud, de arra ügyelünk, hogy egy tag mindenképp ott legyen. A mérföldkövek nagy jelentőséggel bírnak, mivel itt más munkáját értékeljük, ezért rendkívül fontos, hogy mindenki megjelenjen.

A határidők a Szoftver Labor 4 oldaláról. Ezek betartása fontos, mert minden nap késéssel 10%-kal csökken a feladatra kapható pontszám.

	Dátum	Feladat
1	febr. 14.	14 h - csapatok regisztrációja
2	febr. 24.	Követelmény, projekt, funkcionalitás - beadás
3	márc. 3.	Analízis modell kidolgozása 1. - beadás
4	márc. 10.	Analízis modell kidolgozása 2. - beadás
5	márc. 17.	Szkeleton tervezése - beadás
6	márc. 24.	Szkeleton - beadás
7	márc. 31.	Prototípus koncepciója - beadás

8	ápr. 7.	Részletes tervek - beadás
9	ápr. 14.	
10	ápr. 22.	Prototípus - beadás
11	ápr. 28.	Grafikus felület specifikációja - beadás
12	máj. 5.	
13	máj. 12.	Grafikus változat - beadás
14	máj. 16.	Összefoglalás - beadás

A feladatok beadásának határideje minden hét hétfőn van, ezért már hétfővégre el kell készülnie a dokumentumoknak, hogy az átnézésre, esetleges javításokra, nyomtatásra legyen idő. A kinyomtatott dokumentumokat az I épület földszintjén kell átadni az IIT munkatársának 14:00 és 14:15 között. Késés esetén a konzulensnek vagy az IIT adminisztrációs irodájára kell leadni.

2.6.2 Fejlesztő csapat tagjai, feladatai

Arra törekszünk, hogy mindenki minden feladatkörből vegye ki a részét, mert úgy érezzük, hogy így tanulhatunk legtöbbet a projekt elkészítése során.

Név	Feladat
Lipták Levente	dokumentáció, kódolás, tesztelés
Szepes Nóra	dokumentáció, kódolás, tesztelés
Varga Gergő	dokumentáció, kódolás, tesztelés
Sebők Márton	csapatvezető, dokumentáció, kódolás, tesztelés

2.6.3 Felhasznált fejlesztői eszközök

- Eclipse – fejlesztés
- EGit - Git használata az Eclipse-ben
- Javadoc – kód dokumentálása
- Visual Paradigm for UML Community Edition – UML diagramok rajzolása
- Microsoft Word – dokumentáció szerkesztése
- Microsoft OneDrive – dokumentumok megosztása
- Bitbucket – kód megosztása

2.6.4 Kommunikáció

- Hetente egy személyes megbeszélés, ahol minden csapattag jelen van
- A csapatnak létre lett hozva egy Google Group, ami lényegében egy levelező lista, ahol bármikor üzenetet tudunk írni egymásnak
- E-mail, Facebook, Skype az egyéb megbeszélésekhez

2.6.5 Esetlegesen felmerülő problémák kezelése

- Csapattag az adott hétre lebetegszik vagy elutazik
 - A többi tag között egyenletesen elosztjuk a munkát, így minimalizálva a plusz terhet
- Csapattag eltűnik
 - A csapat szól a konzulensnek, hogy keresse meg az eltűnt tagot
 - A csapat tovább dolgozik, az eltűnt tag feladatát elosztja
- Csapattag számítógépe elromlik

- A csapat kölcsönad egy gépet neki
- A laborban található gépek is használhatók a munka folytatására
- Lecsúszás határidőről
 - A dokumentumokat minél előbb elkészítjük, hogy le lehessen adni
- OneDrive meghibásodik
 - Minden csapattagnak megvannak a dokumentumok, ezért ez könnyen orvosolható egy másik tárhely keresésével
- Kód nem fordul a laborban
 - Ennek megelőzése érdekében megpróbálunk még a bemutatás előtt bejutni a laborba, hogy a kódot tesztelhesük
 - Ha a laborban nem tudjuk tesztelni a kódot, akkor minél több környezetet kipróbálunk, így növelve a sikeres fordítás esélyét

2.7 Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.02.19. 18:00	1 óra	Sebők	Eclipse fejlesztőkörnyezet összeállítása, Git és Visual Paradigm integrálása, kezdő projekt elkészítése
2014.02.20. 18:00	2,5 óra	Lipták Sebők Szepes Varga	Első értekezlet. Döntések: jelen dokumentum elkészítésének felosztása (alább látható), alapelvek megfogalmazása.
2014.02.20. 21:30	2,5 óra	Szepes	2.1 Bevezetés elkészítése
2014.02.20. 22:00	2 óra	Lipták Sebők	2.2 Áttekintés elkészítése
2014.02.21. 16:00	2 óra	Sebők	2.3 Követelmények és 2.4 Lényeges use-case-ek elkészítése
2014.02.21. 17:00	2 óra	Varga	2.6 Projekt terv elkészítése
2014.02.21. 23:00	1 óra	Lipták	2.5 Szótár elkészítése

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Akadály

Egy olyan cella, amelyre rá tudnak lépni az ellenségek, de amíg rajta tartózkodnak, lassabban mozognak, vagyis időegységenként kevesebbet haladnak. Fel kell tudnia használni egy sárga követ. Tudnia kell, hogy hol található a térképen.

3.1.2 Domborzat

Egy olyan cella, amelyen az ellenségek nem tartózkodhatnak. Tornyot lehet rajta elhelyezni. Tudnia kell a helyét a térképen.

3.1.3 Ellenségek

A végzet hegye felé kell haladnia olyan cellákon, amelyre ráléphet. Amennyiben egy torony megsebzí, akkor az életpontjainak száma csökken. Amennyiben 1 alá csökken ez a szám, ezt jeleznie kell a játéknak, illetve varázserőt kell kapnia érte a játékosnak, ehhez tudnia kell a helyét a cellájában. 4 fajú ellenséget különböztetünk meg (ember, törpe, tünde, hobbit). Az ellenség tudja a saját fajtát, és fajonként más-más mennyiségű életponttal rendelkezik. Amennyiben eléri egy ellenég a végzet hegyét, jelez a Játéknak, hogy beért.

3.1.4 Játék

Az ő felelőssége hogy felépítse a pályát a játék elején, később pedig számon tartsa az ellenségeket, tornyokat és a pályát (annak celláit), illetve az ő hatására jönnek létre új ellenségek. Az ő utasítására sebeznek a tornyok, lépnek a játékosok, és amikor a játék véget ér, az ő dolga megjeleníteni az eredményét.

3.1.5 Kék kő

Tudnia kell, hogy mennyivel növeli egy torony hatótávolságát.

3.1.6 Piros kő

Tudnia kell, hogy melyik faj esetén mennyivel növeli a sebzés mértékét. A négy fajú ellenséghez négy piros követ különböztetünk meg, melyek közül mindegyik másakra ad sebzés bónuszt (tünde piros kő, törpe piros kő, hobbit piros kő, ember piros kő).

3.1.7 Sárga kő

Tudnia kell, hogy mennyivel tudja növelni egy akadály lassítását.

3.1.8 Torony

Tudnia kell a helyét a térképen, és ki kell tudnia számolni, hogy melyik ellenség milyen távolságban van tőle, azokat sebezni kell tudnia. A piros, kék és zöld köveket el kell tudnia tárolni (egyszerre csak egyet), és hatását fel kell tudnia használni.

3.1.9 Út

Egy olyan cella, amelyre az ellenségeknek rá kell tudnia lépni. Tudnia kell a helyét a térképen.

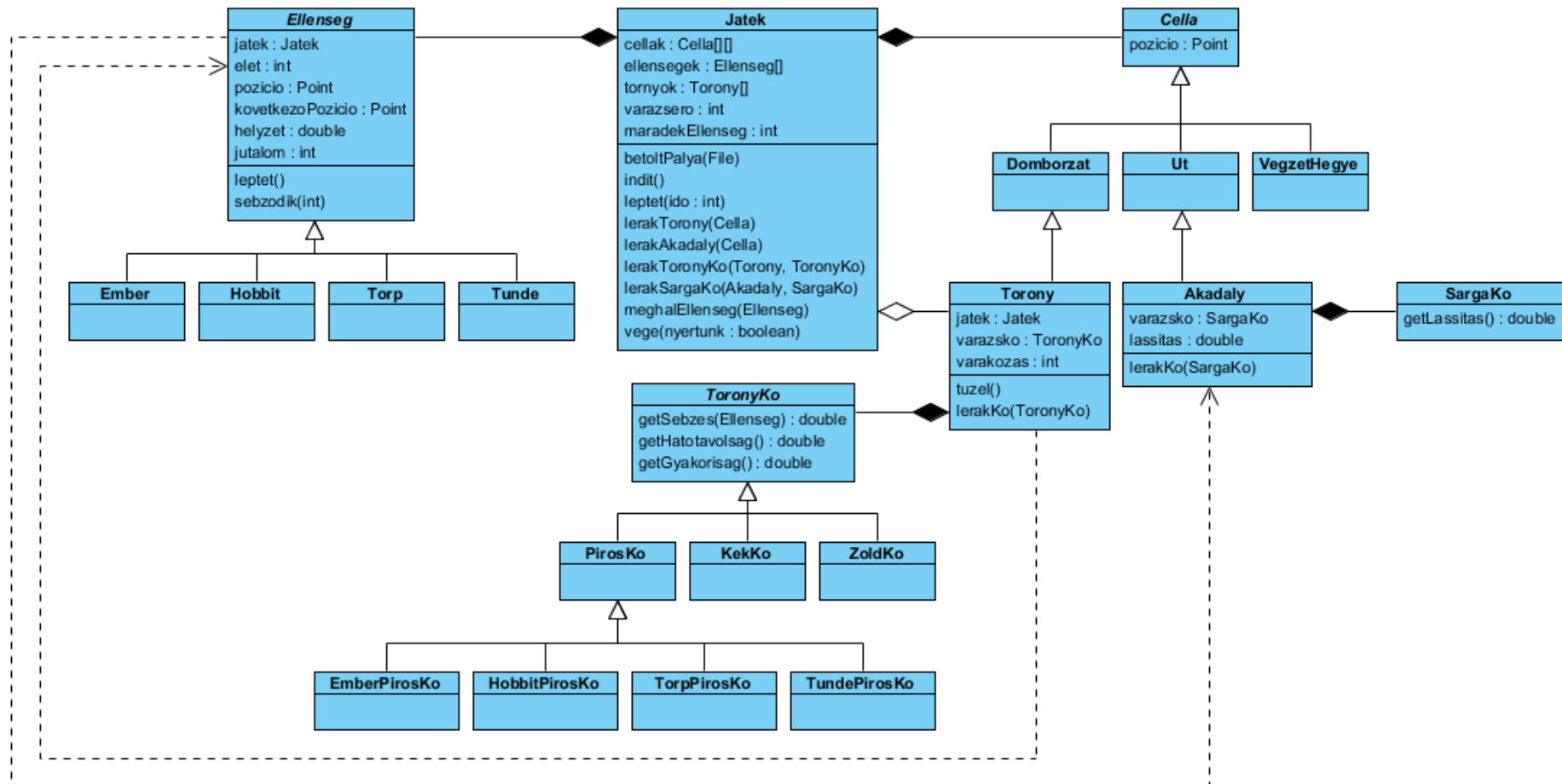
3.1.10 Végzet hegye

Egy olyan cella, amelyre rá kell tudnia lépni az ellenségeknek (nem szükséges hogy tudjanak rajta haladni), tudnia kell a helyét a térképen.

3.1.11 Zöldkő

Tudnia kell, hogy mennyivel növeli meg a torony sebzési gyakoriságát.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Akadály

- **Felelősség**

Útra építhető játékelem, mely lassítja a rajta áthaladó ellenségeket. Ezeket a játékos helyezi el az úton. Az akadály erősíthető sárga varázskővel, ilyen esetben nagyobb mértékkel lassítja az ellenségeket.

- **Ősosztályok**

Cella → Ut

- **Attribútumok**

- **lassitas:** double: Az ellenség sebességét ennyivel kell csökkenteni, amíg az akadályon áthalad.
- **varazsko:** SargaKo: ha ilyen varázskő van az akadályon, akkor jobban lassítja az ellenségeket.

- **Metódusok**

- **lerakKo(sargaKo SargaKo):** A sárga varázskő elhelyezése az akadályon.

3.3.2 Cella

- **Felelősség**

A térkép a kezelhetőség érdekében egyenlő méretű, négyzet alakú részekre, cellákra van osztva, melyek vagy teljes részben út típusú, vagy teljes részben domborzat típusú területet fednek le. Az úton haladnak az ellenségek, elhelyezhetők akadályok, míg a domborzaton tornyok helyezhetők el.

- **Attribútumok**

- **pozicio:** Point: meghatározza a cella elhelyezkedését a pályán, koordinátákat tárol.

3.3.3 Domborzat

- **Felelősség**

Azon területtípus, melyre az ellenségek nem léphetnek rá. Csak ezen a területen helyezhetők el a tornyok.

- **Ősosztályok**

Cella

3.3.4 Ellenség

- **Felelősség**

A térképen keresztül sétálni próbáló lények, feladatunk az ő elpusztításuk.

- **Attribútumok**

- **elet:** int: Mutatja, hogy mennyi élete van még az adott ellenségnek.

- **helyzet:** double Azt mondja meg, hogy egy cellán belül jelenleg hol van az ellenség. Így a mozgás folyamatosabban oldható meg.
- **jatek:** Jatek Referenciát tárol a statikus játék objektumra.
- **jutalom:** int Ha az ellenséget elpusztítja a játékos, akkor ennyi varázserővel gazdagodik.
- **kovetkezoPozicio:** Point: Meghatározza, hogy az ellenség melyik koordinátára fog lépni legközelebb.
- **pozicio:** Point: Meghatározza, hogy az ellenség melyik koordinátán helyezkedik el.
- **Metódusok**
 - **leptet():** Az ellenség lép egyet a Végzet Hegye felé vezető úton. Érzékeli, ha rálépett a Végzet Hegyére, ekkor ezt jelzi a jatek objektumnak.
 - **sebzodik(int sebzes):** Csökkenti az ellenség életét a kapott értékkel.

3.3.5 Ember

- **Felelősség**

Az egyik ellenség faj.

- **Ősosztályok**

Ellenseg

3.3.6 EmberPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak emberek ellen hatásos.

- **Ősosztályok**

ToronyKo→ PirosKo

3.3.7 Hobbit

- **Felelősség**

Az egyik ellenség faj.

- **Ősosztályok**

Ellenseg

3.3.8 HobbitPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak hobbitok ellen hatásos.

- **Ősosztályok**

ToronyKo→ PirosKo

3.3.9 Jatek

- **Felelősség**

Statikus objektum. Elsőként betölti a pályát és elindítja a játékot. Irányítja a játékot, vagyis lépteti az ellenségeket, újakat hoz be és sebez a tornyokkal. Amikor a játék véget ér, az ő dolga megjeleníteni az eredményét.

- **Attribútumok**

- **cellak:** Cella[][] A térképen lévő cellák egy kétdimenziós tömbben.
- **ellensegek:** Ellenseg[] A térképen lévő ellenségeket itt tároljuk.
- **maradekEllenseg: int** A játék során még ennyi ellenséget kell elpusztítani. Ha ez a szám elérte a nullát, akkor megnyertük a játékot.
- **tornyok:** Torony[] A térképen lévő tornyok referenciáját itt is tároljuk. Az egyszerű léptetés miatt van szükség rá.
- **varazsero: int** A játékos számára pénzegységként működik. Ebből építhetünk akadályt, tornyot és tehetünk varázskövet rájuk.

- **Metódusok**

- **betoltPalya(File f):** Játéktér betöltése a megadott fájlból.
- **indit():** A játékmenet elindítása.
- **leptet(int ido):** Az ellenségek léptetése, a tornyokból való tüzelés elindítása, új ellenségek behozása majd.
- **lerakAkadaly(Cella c):** Akadály elhelyezése a megadott cellára a játékos által, adott esetben hiba jelzése.
- **lerakSargaKo(Akadaly a, SargaKo sargaKo):** Sárga varázskő lerakása a megadott akadályra a játékos által, adott esetben hiba jelzése.
- **lerakTorony(Cella c):** Torony elhelyezése a megadott cellára a játékos által, adott esetben hiba jelzése.
- **lerakToronyKo(Torony torony, ToronyKo toronyKo):** Adott toronyra elhelyezi a megadott toronykövet a játékos által, adott esetben hiba jelzése.
- **meghalEllenseg(Ellenseg e):** Az ellenségek halálának kezelése. Ha az összes ellenség meghalt, a játék végének jelzése.
- **vege(boolean nyertunk):** A játék végének kezelése, eredmény kiírása.

3.3.10 KekKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony hatótávolságát növeli. Ezáltal olyan távolabbi ellenségeket is tud sebezni a torony, melyet a varázskő nélkül nem tudna.

- **Össztályok**

ToronyKo

3.3.11 PirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Ilyenkor egy ellenségen nagyobb mértékkel tud sebezni. A piros varázskőnek ellenség fajonként van speciális köve, mely csak arra a fajra van hatással.

- **Ősosztályok**

ToronyKo

3.3.12 SargaKo

- **Felelősség**

Egy akadályra helyezhető varázskő, mely az akadály hatásfokát növeli. Ezáltal nagyobb mértékben lassítja le az akadályon áthaladó ellenségeket.

- **Metódusok**

- **double getLassitas():** Megadja a rajta áthaladó ellenség sebességcsökkenésének értékét.

3.3.13 Torony

- **Felelősség**

A domborzatra építhető játékelem, ami sebzí azokat az ellenségeket, akik az ő hatótávolságába beleeső úton haladnak. A tornyokat a játékos helyezi el a pályán. Varázskővel lehet erősíteni őket, ezek hatótávolságot, a sebzés értékét vagy a sebzés gyakoriságát befolyásoló kövek.

- **Ősosztályok**

Cella → Domborzat

- **Attribútumok**

- **jatek:** Jatek Referenciát tárol a statikus játék objektumra.
- **varakozas:** int: Két sebzés közötti várakozás értéke.
- **varazsko:** ToronyKo: A tornyon elhelyezett varázskő.

- **Metódusok**

- **lerakKo(ToronyKo toronyKo):** Egy varázskő lerakása a toronyra.
- **tuzel():** Tüzelés a toronyból az ellenségekre.

3.3.14 ToronyKo

- **Felelősség**

Toronyra elhelyezhető varázskő. Ezek segítségével erősítheti a játékos a tornyokat. Három fajta varázskő létezik: az egyikkel megadott ellenséget a torony erősebben sebzí, a másikkal a sebzés gyakoriságát növeljük, a harmadikkal pedig a hatótávolságot. Egy tornyon egyszerre csak egy varázskő helyezhető el.

- **Metódusok**

- **double getGyakorisag()**: Megadja a toronynak a sebzési gyakoriságát.
- **double getHatotavolsag()**: Megadja a toronynak a hatótávolságát.
- **double getSebzes(Ellenseg e)**: Megadja a toronynak a sebzés értékét az adott ellenség típushoz.

3.3.15 Torp

- **Felelősség**

Az egyik ellenség faj.

- **Ősosztályok**

Ellenseg

3.3.16 TorpPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak törpök ellen hatásos.

- **Ősosztályok**

ToronyKo → PirosKo

3.3.17 Tunde

- **Felelősség**

Az egyik ellenség faj.

- **Ősosztályok**

Ellenseg

3.3.18 TundePirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak tündék ellen hatásos.

- **Ősosztályok**

ToronyKo → PirosKo

3.3.19 Ut

- **Felelősség**

Az a területtípus, melyen az ellenségek haladnak. Ide csak akadály helyezhető el, torony nem.

- **Ősosztályok**

Cella

- **Attribútumok**

- **lassitas**: double Az úton áthaladó ellenség esetleges lassítását befolyásolja.

3.3.20 VegzetHegye

- **Felelősség**

Az ellenségek célja ide eljutni, a mi célunk ezt megakadályozni. Ha egy ellenség erre a területre lép, a játék véget ér és a játékos veszít.

- **Ősosztályok**

Cella

3.3.21 ZoldKo

- **Felelősség**

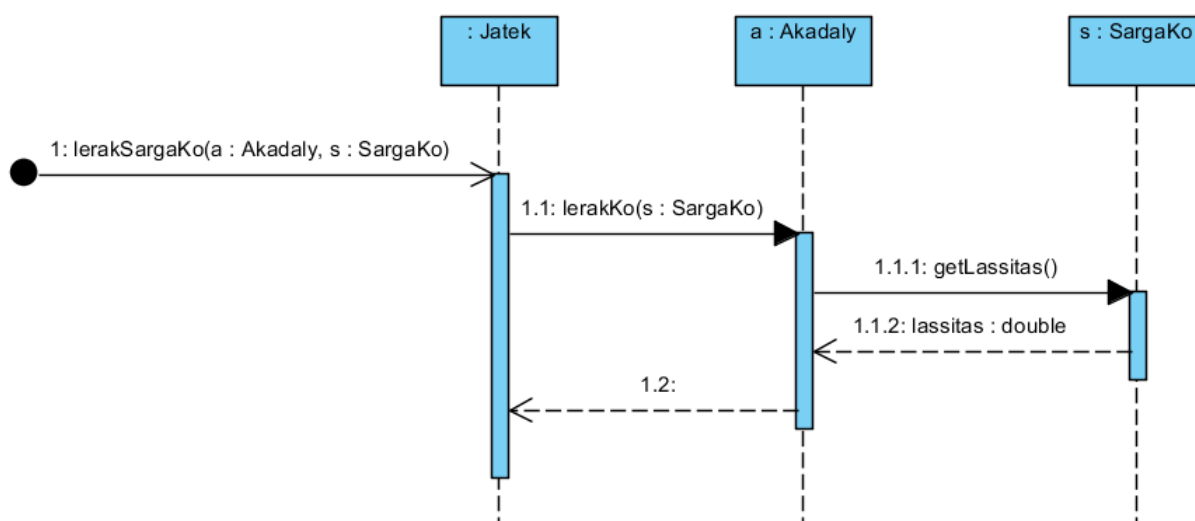
Egy toronyra helyezhető varázskő, mely a torony sebzési gyakoriságát növeli. Ilyenkor a két sebzés között eltelt idő csökken.

- **Ősosztályok**

ToronyKo

3.4 Szekvencia diagramok

3.4.1 4.1 Akadályra kő elhelyezése

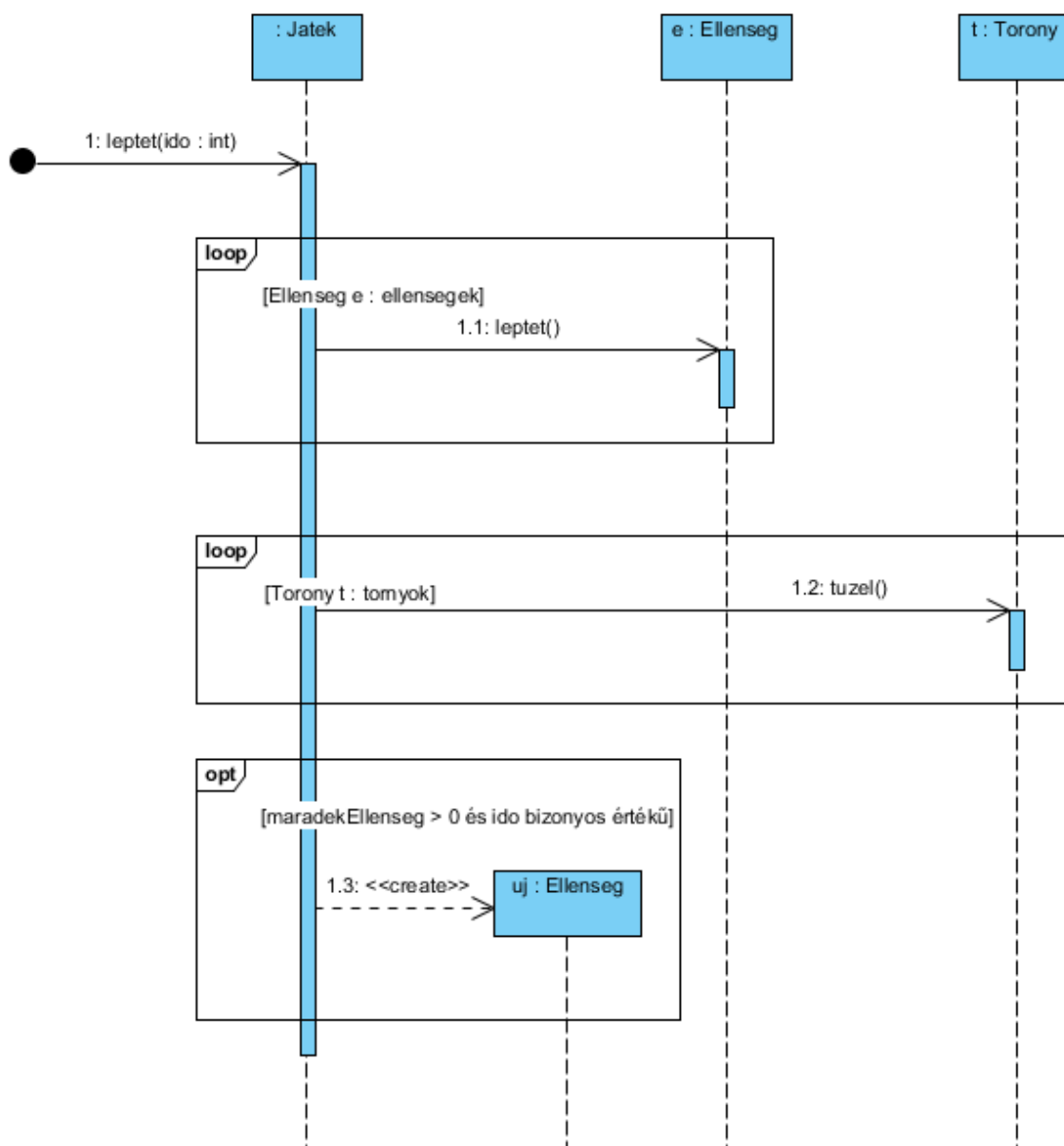


3.4.2 Toronyra kő elhelyezése

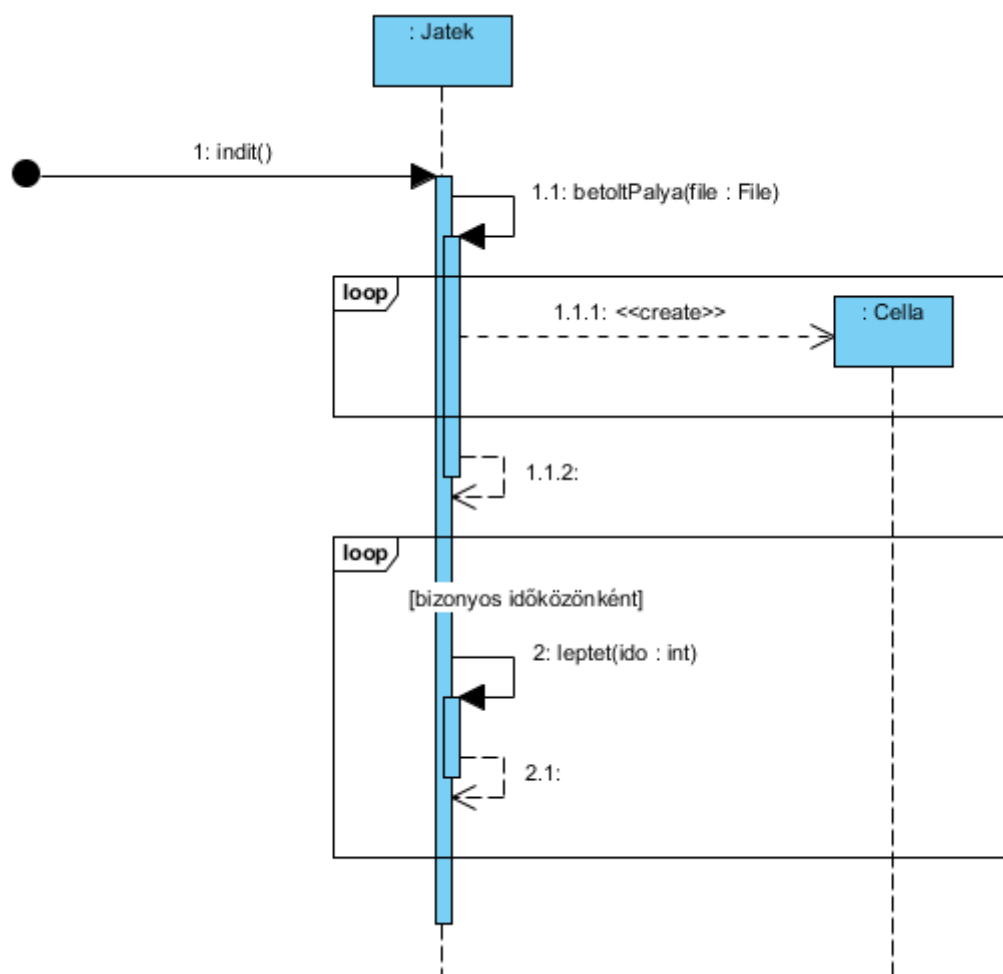
Hasonlóan a 3.4.1 Akadályra kő elhelyezése diagramhoz, a következő változtatásokkal, értelemszerűen:

- a : Akadaly objektum helyett t : Torony
- s : SargaKo objektum helyett k : ToronyKo
- 1: lerakToronyKo(t : Torony, k : ToronyKo)
- 1.1: lerakKo(k : ToronyKo)
- 1.1.1: getGyakorisag()
- 1.1.3: gyakorisag : double

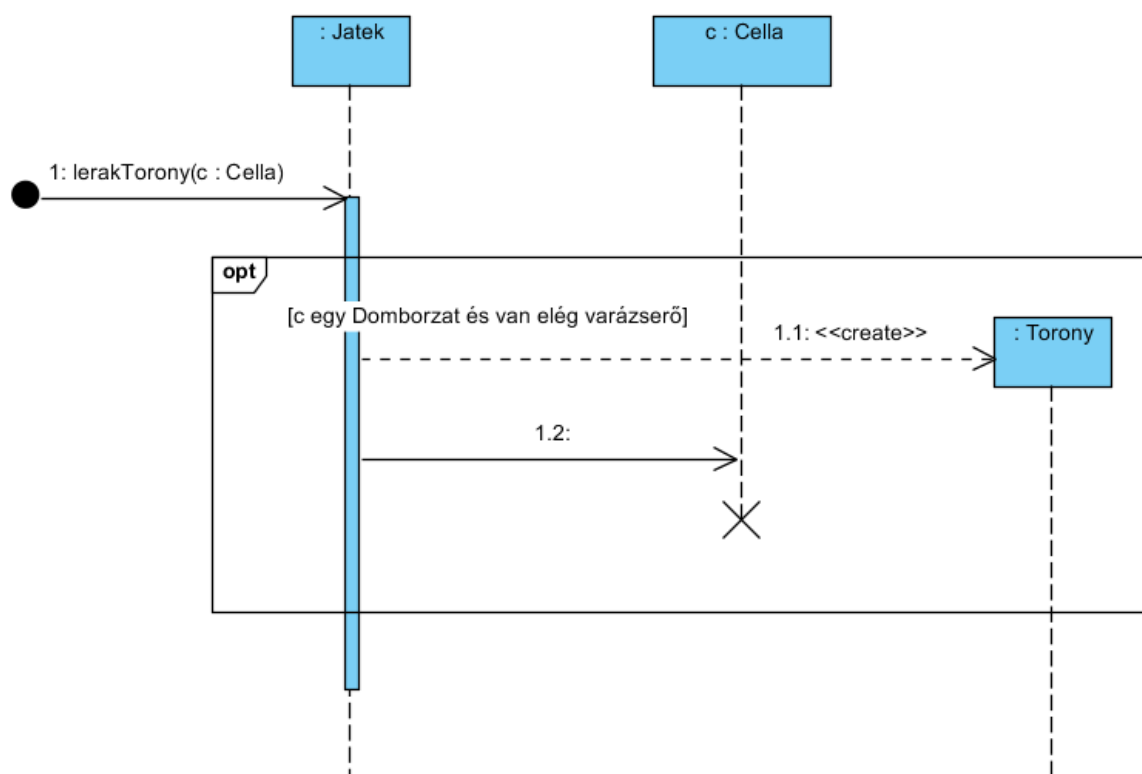
3.4.3 Léptetés



3.4.4 Játék indítása



3.4.5 Torony lerakása

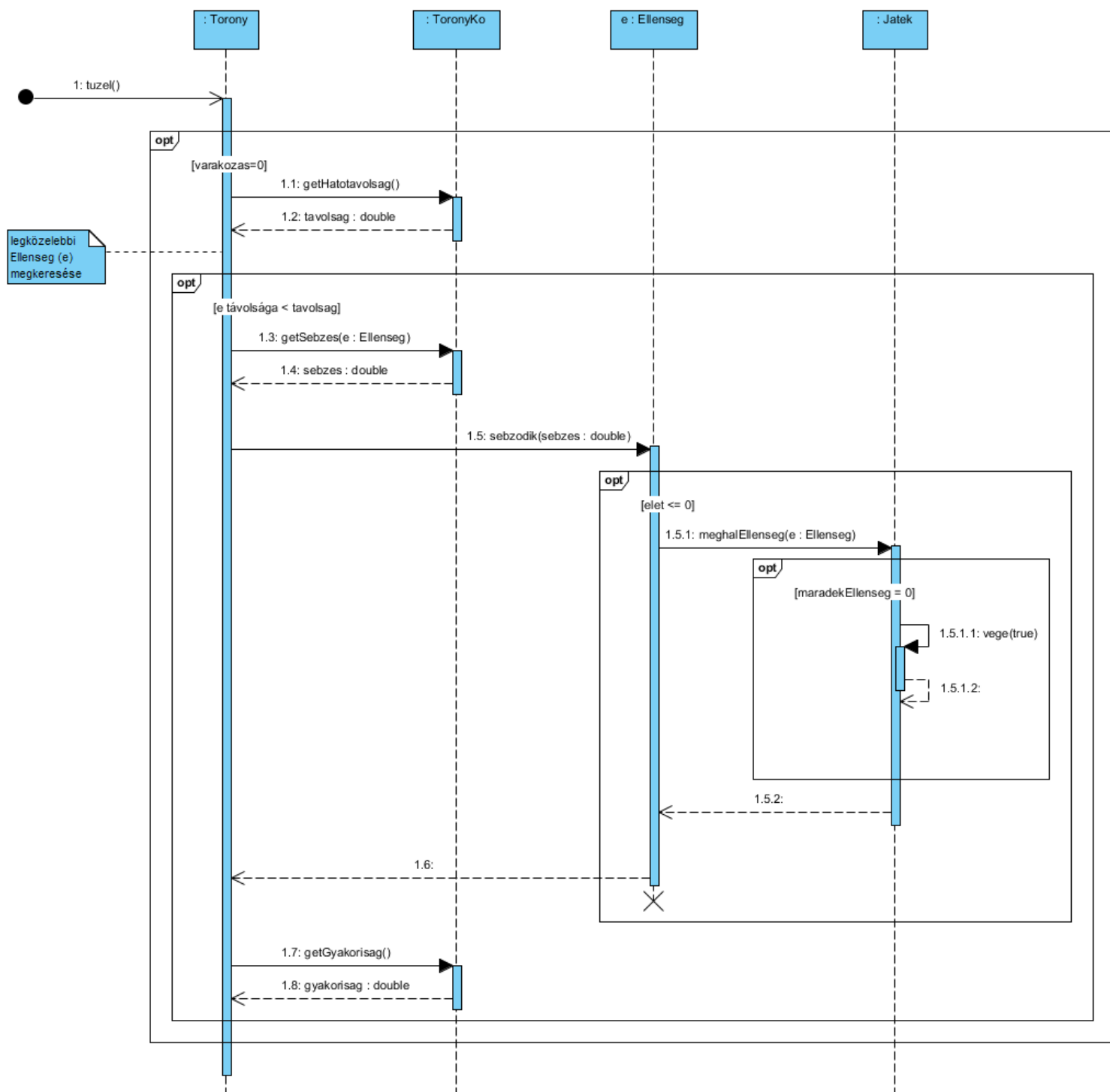


3.4.6 Akadály lerakása

Hasonlóan a 3.4.1 Akadályra kő elhelyezése diagramhoz, a következő változtatásokkal, értelemszerűen:

- : Torony objektum helyett : Akadaly
- opt akkor hajtódik végre, ha c egy Ut és van elég varázserő
- 1: lerakAkadaly(c : Cella)

3.4.7 Torony tüzel



3.5 State-chartok

Az objektumainkhoz nem indokolt state-chart készítése.

3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.02.28. 19:30	1 óra	Lipták Sebők Szepes Varga	Második értekezlet. Döntések: jelen dokumentum elkészítésének felosztása
2014.02.28. 20:00	1 óra	Sebők	3.2 Statikus struktúra diagramok felvázolása
2014.02.28. 21:00	2 óra	Lipták Sebők	3.2 Statikus struktúra diagramok elkészítése
2014.03.01. 10:00	2 óra	Szepes	3.3 Osztályok leírásának elkezdése a statikus diagram alapján
2014.03.01. 20:00	2 óra	Szepes	3.3 Osztályok leírásának kiegészítése új attribútumokkal és metódusokkal
2014.03.01. 20:00	2 óra	Varga	3.4 Szekvencia diagramok elkészítése
2014.03.01. 23:00	1 óra	Sebők Szepes	3.2 Statikus struktúra diagramok finomítása, ezek alapján 3.3 Osztályok leírásának kiegészítése
2014.03.02. 00:30	1 óra	Varga Sebők	3.4 Szekvencia diagramok javítása
2014.03.02. 14:00	1 óra	Lipták	3.1 Objektum katalógus elkészítése
2014.03.02. 17:30	1 óra	Varga	3.4 Szekvencia diagramok javítása, pontosítása

4. Analízis modell kidolgozása 2

4.1 Objektum katalógus

4.1.1 Akadály

Egy cellára lehet helyezni, melyre az áthaladó ellenségek így lassabban mozognak, vagyis időegységenként kevesebbet haladnak. Fel kell tudnia használni egy sárga követ.

4.1.2 Domborzat

Egy olyan cella, amelyen az ellenségek nem tartózkodhatnak. Tornyot lehet rajta elhelyezni. Tudnia kell a helyét a térképen.

4.1.3 Ellenségek

A végzet hegye felé kell haladnia olyan cellákon, amelyre ráléphet. Amennyiben egy torony megsebzí, akkor az életpontjainak száma csökken. Amennyiben 1 alá csökken ez a szám, ezt jeleznie kell a játéknak, illetve varázserőt kell kapnia érte a játékosnak, ehhez tudnia kell a helyét a cellájában. 4 fajú ellenséget különböztetünk meg (ember, törpe, tünde, hobbit). Az ellenség tudja a saját fajtát, és fajonként más-más mennyiségű életponttal rendelkezik. Amennyiben eléri egy ellenség a végzet hegyét, jelez a Játéknak, hogy beért.

4.1.4 Játék

Az ő felelőssége hogy felépítse a pályát a játék elején, később pedig számon tartsa az ellenségeket, tornyokat és a pályát (annak celláit), illetve az ő hatására jönnek létre új ellenségek. Az ő utasítására sebeznek a tornyok, lépnek a játékosok, és amikor a játék véget ér, az ő dolga megjeleníteni az eredményét.

4.1.5 Kék kő

Tudnia kell, hogy mennyivel növeli egy torony hatótávolságát.

4.1.6 Piros kő

Tudnia kell, hogy melyik faj esetén mennyivel növeli a sebzés mértékét. A négy fajú ellenséghez négy piros követ különböztetünk meg, melyek közül mindegyik másakra ad sebzés bónuszt (tünde piros kő, törpe piros kő, hobbit piros kő, ember piros kő).

4.1.7 Sárga kő

Tudnia kell, hogy mennyivel tudja növelni egy akadály lassítását.

4.1.8 Torony

Ki kell tudnia számolni, hogy melyik ellenség milyen távolságban van tőle, azokat sebezni kell tudnia. A piros, kék és zöld követ el kell tudnia tárolni (egyszerre csak egyet), és hatását fel kell tudnia használni.

4.1.9 Út

Egy olyan cella, amelyre az ellenségeknek rá kell tudnia lépni. Tudnia kell a helyét a térképen.

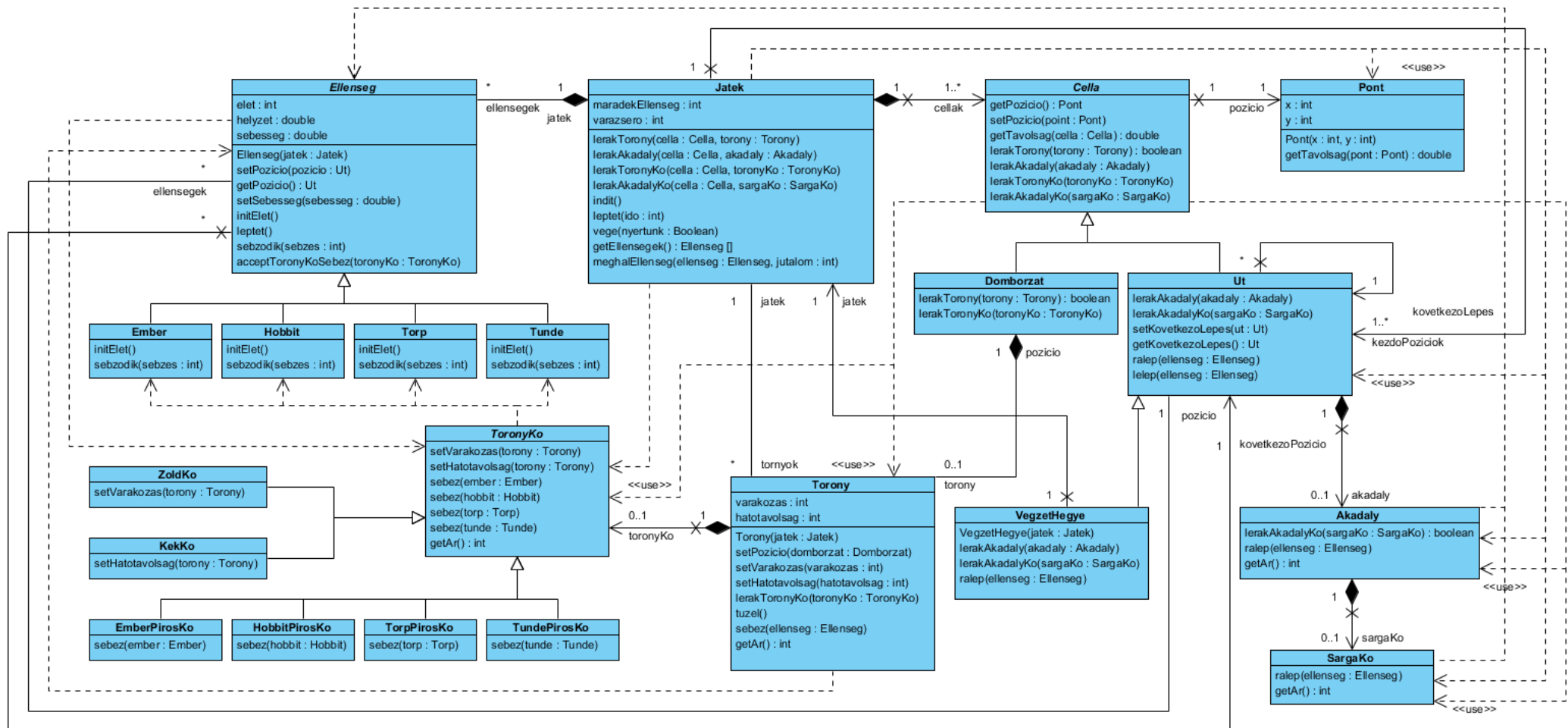
4.1.10 Végzet hegye

Egy olyan cella, amelyre rá kell tudnia lépni az ellenségeknek (nem szükséges hogy tudjanak rajta haladni), tudnia kell a helyét a térképen.

4.1.11 Zöld kő

Tudnia kell, hogy mennyivel növeli meg a torony sebzési gyakoriságát.

4.2 Statikus struktúra diagramok



A leszármazott osztályokon csak a felüldefiniált metódusok vannak feltüntetve, a többi értelemszerűen öröklí az ősétől.

4.3 Osztályok leírása

4.3.1 Akadály

- **Felelősség**

Útra építhető játékelem, mely lassítja a rajta áthaladó ellenségeket. Ezeket a játékos helyezi el az úton. Az akadály erősíthető sárga varázskövel, ilyen esetben nagyobb mértékkel lassítja az ellenségeket.

- **Attribútumok**

- **SargaKo sargaKo**: ha ilyen varázskő van az akadályon, akkor jobban lassítja az ellenséget.

- **Metódusok**

- **getAr()**: int: Visszaadja az építéshez szükséges árat.
- **boolean lerakAkadalyKo (SargaKo sargaKo)**: A sárga varázskő elhelyezése az akadályon. True visszatérési értékű, ha sikeres (azaz nem volt még az akadályon varázskő), false egyébként.
- **ralep(Ellenseg ellenseg)**: Akkor hívódik meg, amikor egy ellenség rálép az akadályra. Ha van rajta varázskő, akkor annak a segítségével állítja be az ellenség sebességét, különben pedig maga az akadály állítja be a sebességet.

4.3.2 Cella

- **Felelősség**

A térkép a kezelhetőség érdekében egyenlő méretű, négyzet alakú részekre, cellákra van osztva, melyek vagy teljes részben út típusú, vagy teljes részben domborzat típusú területet fednek le. Az úton haladnak az ellenségek, elhelyezhetők akadályok, míg a domborzaton tornyok helyezhetők el. Mivel cella helyett domborzat, út és végzet hegye fog szerepelni a térképen, ezért ez absztrakt osztály.

- **Attribútumok**

- **Pont pozicio**: meghatározza a cella elhelyezkedését a pályán, koordinátákat tárol.

- **Metódusok**

- **Pont getPozicio()**: visszaadja a cella pozícióját.
- **double getTavolsag(Cella cella)**: visszaadja a cella pozícióját a paraméterként kapott cellától
- **lerakAkadaly(Akadaly akadaly)**: lerak egy akadályt a cellára. Ezt a függvényt felülírja a domborzat és az út.
- **lerakAkadalyKo(SargaKo sargaKo)**: lerak egy akadálykövet a cellán lévő akadályra. Ezt a függvényt felülírja a domborzat és az út.
- **boolean lerakTorony(Torony torony)**: lerak egy tornyot a cellára és false értékkel visszatér. Ezt a függvényt felülírja a domborzat és az út.
- **lerakToronyKo(ToronyKo toronyKo)**: lerak egy toronykövet a cellán lévő toronyra. Ezt a függvényt felülírja a domborzat és az út.
- **setPozicio(pont Pont)**: beállítja a cella pozícióját.

4.3.3 Domborzat

- **Felelősség**

Azon területtípus, melyre az ellenségek nem léphetnek rá. Csak ezen a területen helyezhetők el a tornyok. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

Cella

- **Attribútumok**

- **Torony torony:** a domborzaton lévő tornyot tároljuk benne.

- **Metódusok**

- **boolean lerakTorony(Torony torony):** lerak egy tornyot a domborzatra. Felülírja a cella azonos nevű függvényét, true értékkel tér vissza, ha sikeres az elhelyezés (azaz nem volt még a domborzaton torony), egyébként false értékkel tér vissza.
- **lerakToronyKo(ToronyKo toronyKo):** lerak egy toronykövet a domborzaton lévő toronyra. Felülírja a cella azonos nevű függvényét, visszatér, ha nincs a domborzaton torony.

4.3.4 Ellenség

- **Felelősség**

A térképen keresztülsétálni próbáló lények, feladatunk az ő elpusztításuk. Absztrakt osztály, helyette csak ember, hobbit, törp és tünde lesz a pályán.

- **Attribútumok**

- **int elet:** Mutatja, hogy mennyi élete van még az adott ellenségnek.
- **double helyzet:** Azt mondja meg, hogy egy cellán belül jelenleg hol van az ellenség. Így a mozgás folyamatosabban oldható meg.
- **Jatek jatek:** Referenciát tárol a statikus játék objektumra.
- **Ut kovetkezoPozicio:** Meghatározza, hogy az ellenség melyik útra fog lépni legközelebb.
- **Ut pozicio:** Meghatározza, hogy az ellenség melyik úton helyezkedik el.
- **double sebesseg:** Meghatározza az ellenség sebességét

- **Metódusok**

- **acceptToronyKoSebez(ToronyKo toronyKo):** Sebzéshez szükséges. Ezáltal meg tudja hívni megfelelő leszármaztatott osztály sebez metódusát. (Visitor minta)
- **Ellenség(Jatek jatek):** létrehozza az ellenség objektumot.
- **Ut getPozicio():** Pozíció lekérdezése.
- **initElet():** Élet beállítása.
- **leptet():** Az ellenség lép egyet a Végzet Hegye felé vezető úton. Érzékeli, ha rálépett a Végzet Hegyére, ekkor ezt jelzi a jatek objektumnak.
- **sebzodik(int sebzes):** Csökkenti az ellenség életét a kapott értékkel.
- **setPozicio(Ut pozicio):** Pozíció beállítása.
- **setSebesseg(double sebesseg):** Sebesség beállítása.

4.3.5 Ember

- **Felelősség**

Az egyik ellenség faj. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

Ellenseg

- **Metódusok**

- **initElet()**: Felülírja az ellenség metódusát. Működése hasonló.
- **sebzodik(int sebzes)**: Felülírja az ellenség metódusát. Működése hasonló.

4.3.6 EmberPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak emberek ellen hatásos. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

ToronyKo

- **Metódusok**

- **sebez(Ember ember)**: Felülírja a toronykő metódusát. Működése hasonló.

4.3.7 Hobbit

- **Felelősség**

Az egyik ellenség faj. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

Ellenseg

- **Metódusok**

- **initElet()**: Felülírja az ellenség metódusát. Működése hasonló.
- **sebzodik(int sebzes)**: Felülírja az ellenség metódusát. Működése hasonló.

4.3.8 HobbitPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak hobbitok ellen hatásos. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

ToronyKo

- **Metódusok**

- **sebez(Hobbit hobbit)**: Felülírja a toronykő metódusát. Működése hasonló.

4.3.9 Jatek

- **Felelősség**

Statikus objektum. Elsőként betölti a pályát és elindítja a játékot. Irányítja a játékot, vagyis lépteti az ellenségeket, újakat hoz be és sebez a tornyokkal. Amikor a játék véget ér, az ő dolga megjeleníteni az eredményét.

- **Attribútumok**

- **Cella[] cellak**: A térképen lévő cellák egy kétdimenziós tömbben.
- **Ellenseg[] ellensegek**: A térképen lévő ellenségeket itt tároljuk.
- **Ut[] kezdőPoziciok**: Azon Ut objektumok, melyekről ellenség indulhat. A játék indításakor betöltjük ezeket, majd új ellenségek létrehozásakor innen választunk kezdő pozíciót.
- **int maradékEllenseg**: A játék során még ennyi ellenséget kell elpusztítani. Ha ez a szám elérte a nullát, akkor megnyertük a játékot.
- **Torony[] tornyok**: A térképen lévő tornyok referenciáját itt is tároljuk, a tüzelés miatt van rá szükség.
- **int varazsero**: A játékos számára pénzegységként működik. Ebből építhetünk akadályt, tornyot és tehetünk varázskövet rájuk.

- **Metódusok**

1. **Ellenseg[] getEllensegek()**: Ellenségek tömb visszaadása
2. **indit()**: A játékmenet elindítása.
3. **leptet(int ido)**: Az ellenségek léptetése, a tornyokból való tüzelés elindítása, új ellenségek behozása majd.
4. **lerakAkadaly(Cella cella, Akadaly akadaly)**: Akadály elhelyezése a megadott cellára a játékos által.
5. **lerakAkadalyKo(Cella cella, SargaKo sargaKo)**: A kapott sárga varázskő lerakása a megadott akadályra a játékos által.
6. **lerakTorony(Cella cella, Torony torony)**: Torony elhelyezése a megadott cellára a játékos által. Ha a cella lerakTorony(Torony torony) függvényétől true visszatérési értéket kap, a tornyot berakja a tornyok listájába.
7. **lerakToronyKo(Cella cella, ToronyKo toronyKo)**: Adott cellára elhelyezi a megadott toronykövet a játékos által.
8. **meghalEllenseg(Ellenseg ellenseg, int jutalom)**: Az ellenségek halálának kezelése. Ha az összes ellenség meghalt, a játék végének jelzése, egyébként a megfelelő jutalom jóváírása.
9. **vege(Boolean nyertunk)**: A játék végének kezelése, eredmény kiírása.

4.3.10 KekKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony hatótávolságát növeli. Ezáltal olyan távolabbi ellenségeket is tud sebezni a torony, melyet a varázskő nélkül nem tudna. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

ToronyKo

- **Metódusok**

- **setHatotavolsag(Torony torony):** Felülírja az ősosztály metódusát. Beállítja a kapott torony hatótávolságát.

4.3.11 Pont

- **Felelősség**

Saját koordináta osztályunk, melyet az elemek pozicionálásához használunk.

- **Attribútumok**

- **int x:** x koordináta értéke
- **int y:** y koordináta értéke

- **Metódusok**

- **double getTavolsag(Pont pont):** Visszaadja saját maga és a pont Pont közötti távolságot.
- **Pont(int x, int y):** Létrehozza a Pont objektumot.

4.3.12 SargaKo

- **Felelősség**

Egy akadályra helyezhető varázskő, mely az akadály hatásfokát növeli. Ezáltal nagyobb mértékben lassítja le az akadályon áthaladó ellenségeket.

- **Metódusok**

- **getAr():** int: Visszaadja az építéshez szükséges árat.
- **ralep(Ellenseg ellenseg):** Beállítja a paraméterként kapott ellenség sebességét.

4.3.13 Torony

- **Felelősség**

A domborzatra építhető játékelem, ami sebzí azokat az ellenségeket, akik az ő hatótávolságába beleeső úton haladnak. A tornyokat a játékos helyezi el a pályán. Varázskővel lehet erősíteni őket, ezek hatótávolságot, a sebzés értékét vagy a sebzés gyakoriságát befolyásoló kövek.

- **Attribútumok**
 - **int hatotavolsag**: Meghatározza a torony hatótávolságát.
 - **Jatek jatek**: Referenciát tárol a statikus játék objektumra.
 - **Domborzat pozicio**: Meghatározza melyik domborzaton helyezkedik el a torony.
 - **ToronyKo toronyKo**: A tornyon elhelyezett varázskő.
 - **int varakozas**: Két sebzés közötti varakozás értéke.
- **Metódusok**
 - **getAr()**: int: Visszaadja az építéshez szükséges árat.
 - **lerakToronyKo(ToronyKo toronyKo)**: Egy varázskő lerakása a toronyra, ha azon még nincs toronykő.
 - **sebez(Ellenseg ellenseg)**: Megsebz a paraméterként kapott ellenséget. Meghívja annak az acceptToronyKoSebez függvényét. Annak a leírása az ellenség osztály leírásában található.
 - **setHatotavolsag(int hatotavolsag)**: beállítja a hatótávolság értékét
 - **setPozicio(Domborzat domborzat)**: beállítja a pozíciót
 - **setVarakozas(int varakozas)**: beállítja a varakozás értékét
 - **Torony(Jatek jatek)**: létrehozza a torony objektumot.
 - **tuzel()**: Tüzelés a toronyból az ellenségekre.

4.3.14 ToronyKo

- **Felelősség**

Toronyra elhelyezhető varázskő. Ezek segítségével erősítheti a játékos a tornyokat. Három fajta varázskő létezik: az egyikkel megadott ellenséget a torony erősebben sebz, a másikkal a sebzés gyakoriságát növeljük, a harmadikkal pedig a hatótávolságot. Egy tornyon egyszerre csak egy varázskő helyezhető el. Absztrakt osztály, helyette zöldkövet, kékkövet illetve a piros kő négy változatát fogjuk használni.

- **Metódusok**

- **int getAr()**: Visszaadja az építéshez szükséges árat.
- **sebez(Ember ember)**: Megsebz a paraméterként kapott ellenség fajt.
- **sebez(Hobbit hobbit)**: Megsebz a paraméterként kapott ellenség fajt.
- **sebez(Torp torp)**: Megsebz a paraméterként kapott ellenség fajt.
- **sebez(Tunde tunde)**: Megsebz a paraméterként kapott ellenség fajt.
- **setHatotavolsag(Torony torony)**: Megadja a toronynak a hatótávolságát.
- **setVarakozas(Torony torony)**: Megadja a toronynak a sebzési gyakoriságát.

4.3.15 Torp

- **Felelősség**

Az egyik ellenség faj. Működése hasonlít az Ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

Ellenseg

- **Metódusok**

- **initElet()**: Felülírja az ellenség metódusát. Működése hasonló.
- **sebzodik(int sebzés)**: Felülírja az ellenség metódusát. Működése hasonló.

4.3.16 TorpPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak törpök ellen hatásos. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

ToronyKo

- **Metódusok**

- **sebez(Torp torp)**: Felülírja a toronykő metódusát. Működése hasonló.

4.3.17 Tunde

- **Felelősség**

Az egyik ellenség faj. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

Ellenseg

- **Metódusok**

- **initElet()**: Felülírja az ellenség metódusát. Működése hasonló.
- **sebzodik(int sebzés)**: Felülírja az ellenség metódusát. Működése hasonló.

4.3.18 TundePirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak tündék ellen hatásos. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

ToronyKo

- **Metódusok**

- **sebez(Tunde tunde)**: Felülírja a toronykő metódusát. Működése hasonló.

4.3.19 Ut

- **Felelősség**

Az a területtípus, melyen az ellenségek haladnak. Ide csak akadály helyezhető el, torony nem. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

Cella

- **Attribútumok**

- **Akadaly akadaly:** Az úton lévő akadály.
- **Ellenseg[] ellensegek:** Az adott úton lévő ellenségek összessége.
- **Ut kovetkezoLepes:** A végzet hegye felé vezető úton erre kell majd az ellenségeknek menni.

- **Metódusok**

- **lelep(Ellenseg ellenseg):** Akkor hívódik meg, amikor egy új ellenség lelép az útról. Ilyenkor az új ellenség kikerül az ellensegek tömbből.
- **lerakAkadaly(Akadaly akadaly):** Lerak egy akadályt az útra. Felüldefiniálja a cella azonos nevű függvényét. Ha még nincs rajta akadály, akkor elhelyezi a kapott akadályt, egyébként visszatér.
- **lerakAkadalyKo(SargaKo sargaKo):** Egy varázskő lerakása az akadályra, ha azon még nincs varázskő. Felüldefiniálja a cella azonos nevű függvényét. Visszatér, ha az úton nincs akadály.
- **raleb(Ellenseg ellenseg):** Akkor hívódik meg, amikor egy új ellenség lép rá az útra. ilyenkor az új ellenség belekerül az ellensegek tömbbe.
- **setKovetkezoLepes(Ut ut):** Beállítja a kovetkezoLepes változót.
- **Ut getKovetkezoLepes():** Visszaadja a kovetkezoLepes változót.

4.3.20 VégzetHegye

- **Felelősség**

Az ellenségek célja ide eljutni, a mi célunk ezt megakadályozni. Ha egy ellenség erre a területre lép, a játék véget ér és a játékos veszít. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

Cella → Út

- **Attribútumok**

- **Jatek jatek:** Referenciát tárol a statikus játék objektumra.

- **Metódusok**

- **lerakAkadaly(Akadaly akadaly):** Felüldefiniálja az ősosztály metódusát. Visszatér, ha akadályt próbálunk meg elhelyezni rajta.
- **lerakAkadalyKo(SargaKo sargaKo):** Felüldefiniálja az ősosztály metódusát. Visszatér, ha akadályt próbálunk meg elhelyezni rajta.

- **ralep(Ellenseg ellenseg)**: Felüldefiniálja az ősosztály metódusát. Ha egy ellenség rálép, a játék véget ér.
- **VegzetHegye(Jatek jatek)**: létrehozza a végzet hegye objektumot.

4.3.21 ZoldKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzési gyakoriságát növeli. Ilyenkor a két sebzés között eltelt idő csökken. Működése hasonlít az ősosztálya működésére. Lentebb csak a felüldefiniált metódusokat rögzítettem.

- **Ősosztályok**

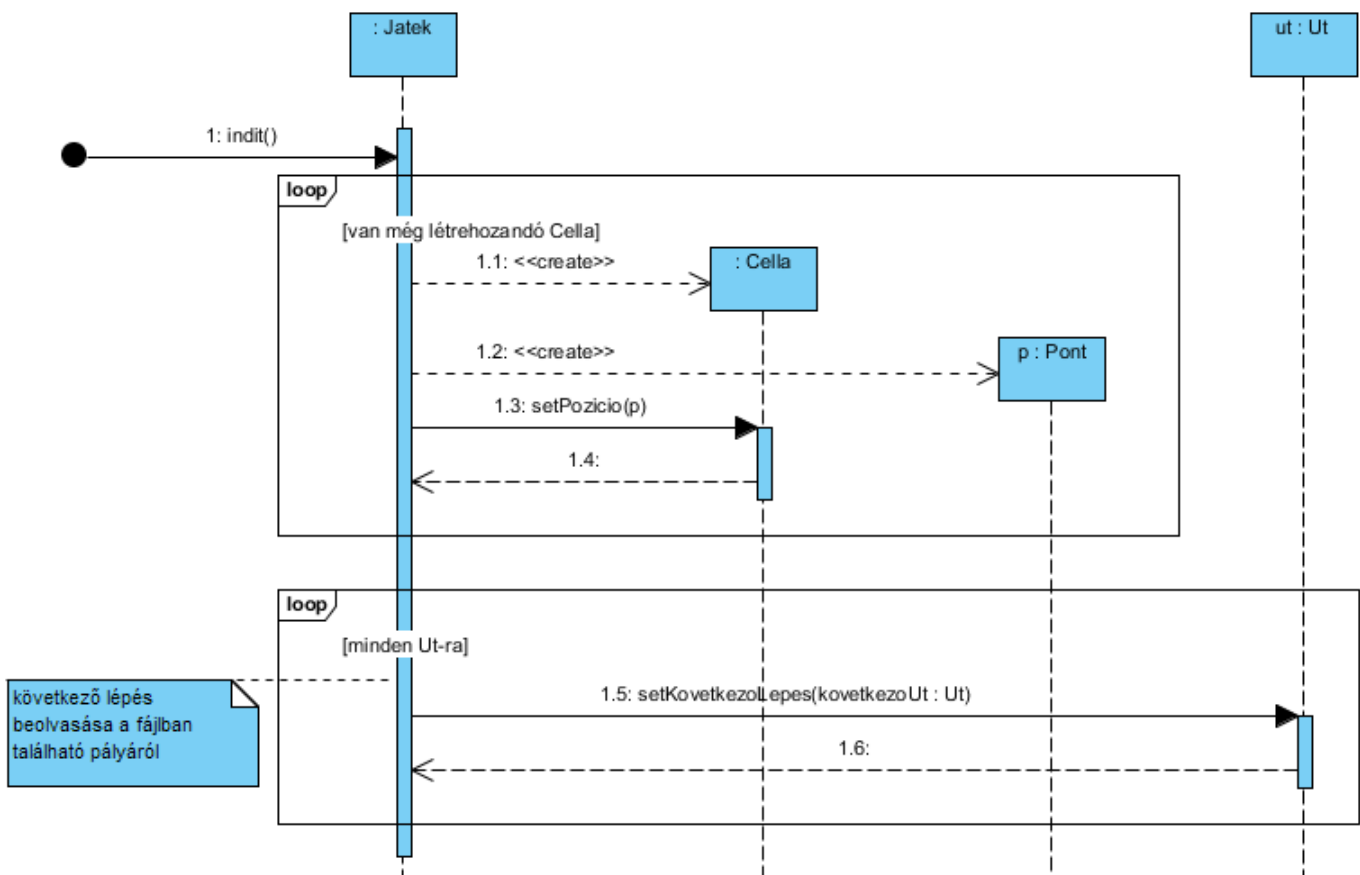
ToronyKo

- **Metódusok**

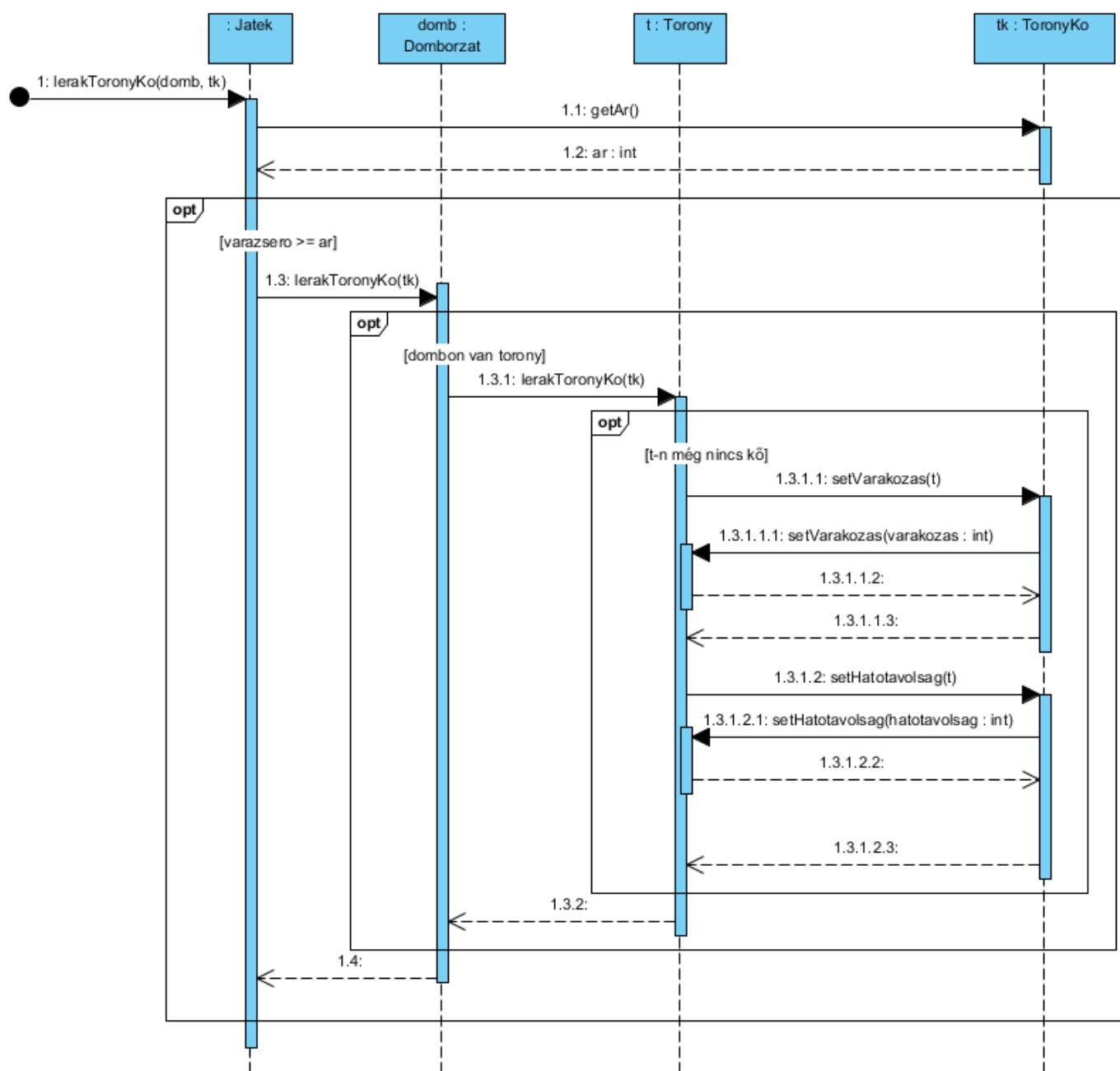
- **setVarakozas (Torony torony)**: Felülírja az ősosztály metódusát. Beállítja a kapott torony sebzésének gyakoriságát.

4.4 Szekvencia diagramok

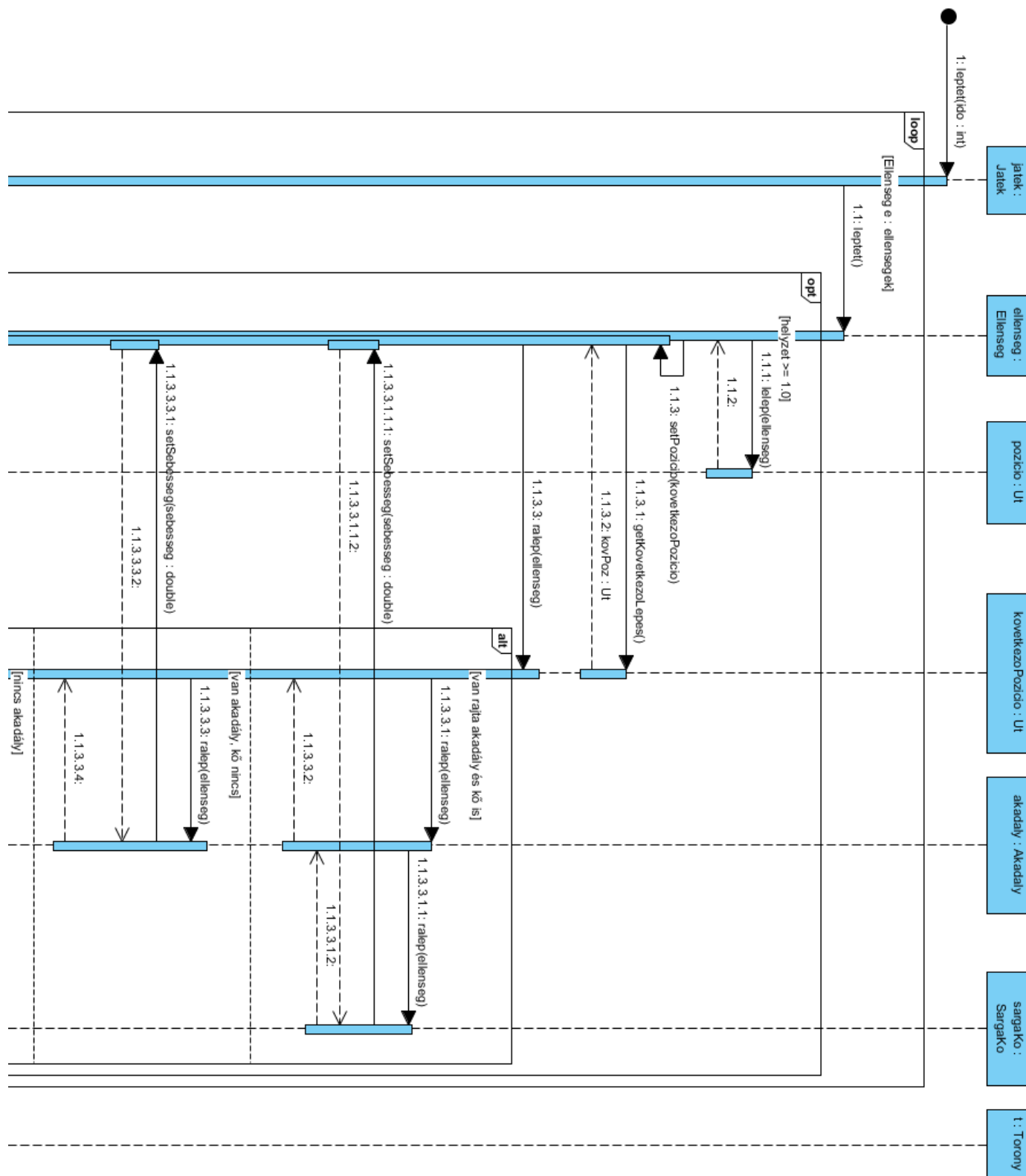
4.4.1 Játék indítása

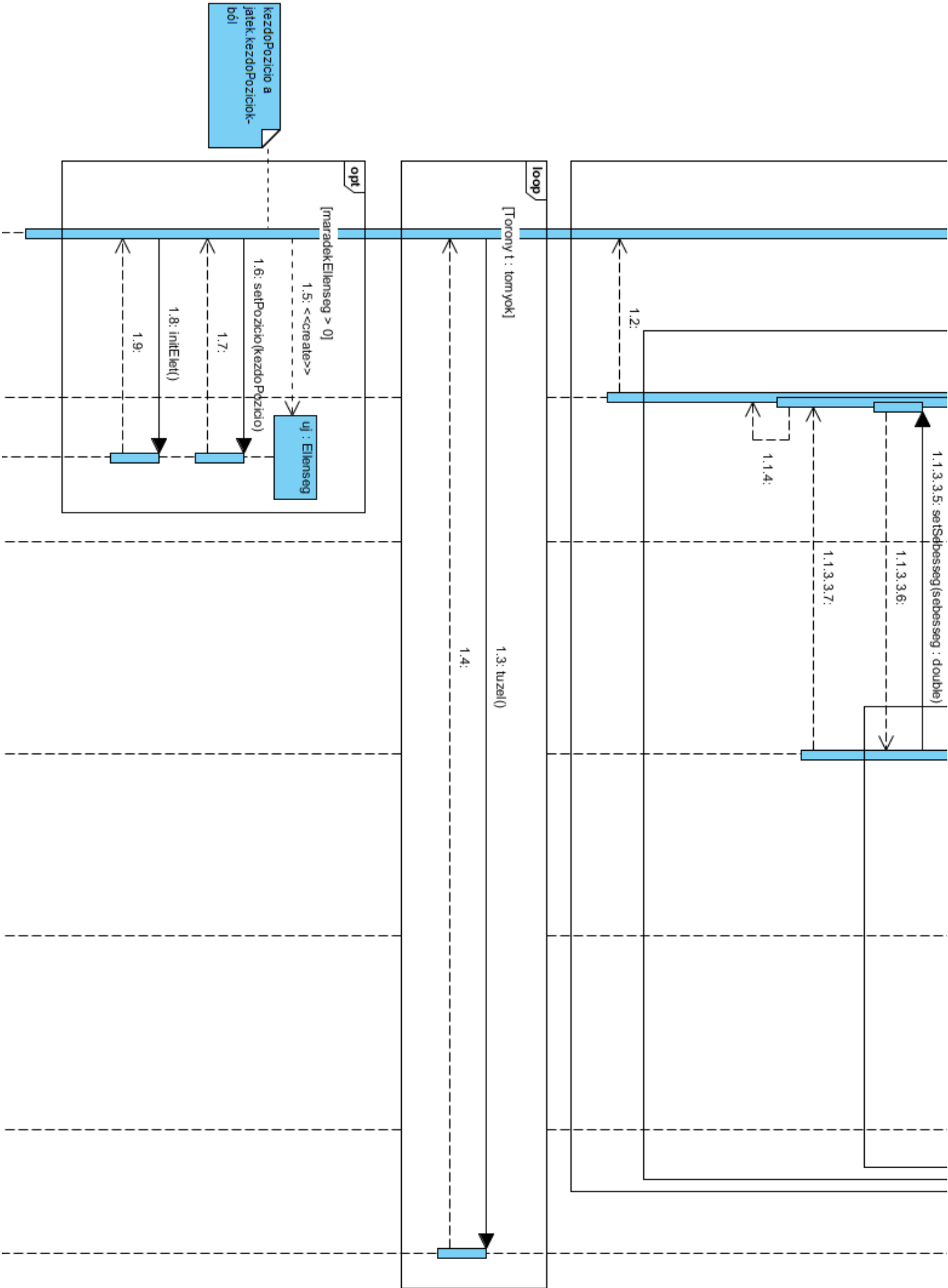


4.4.2 Toronykő lerakása domborzatra

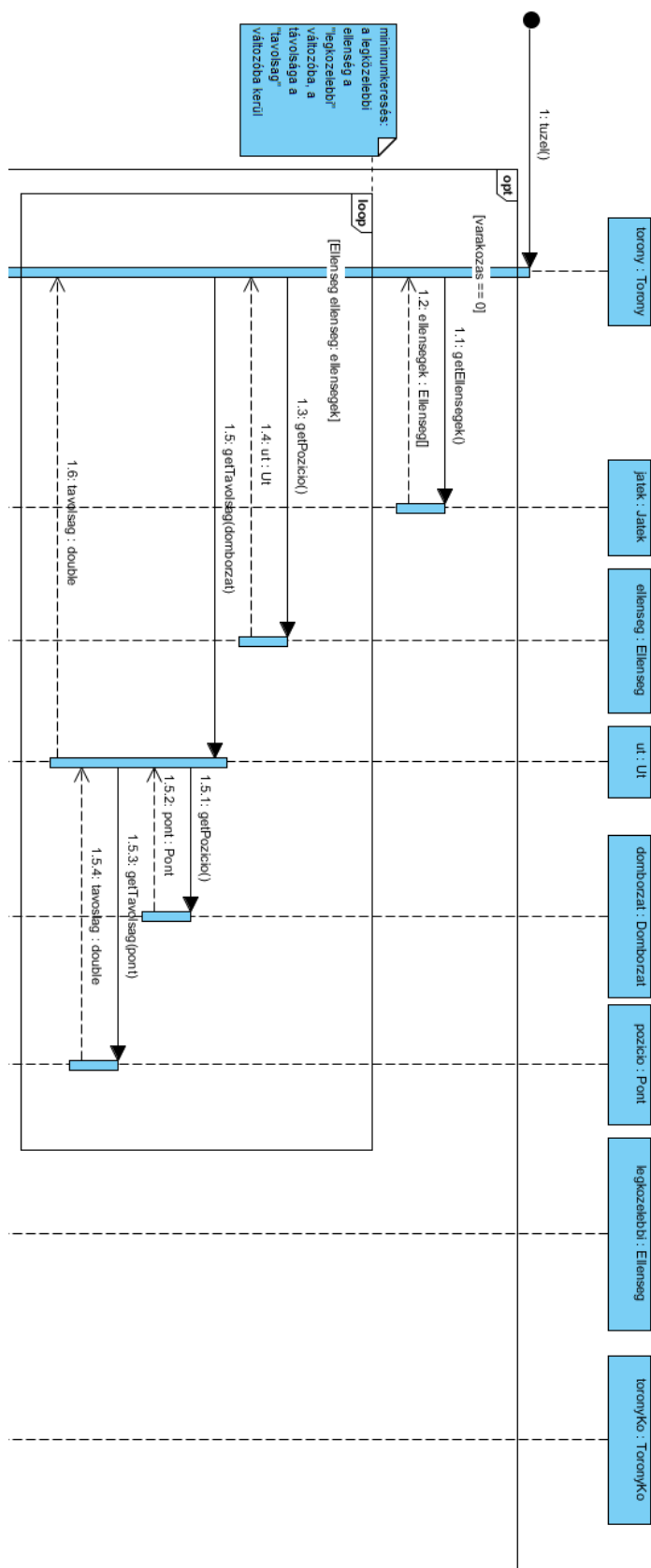


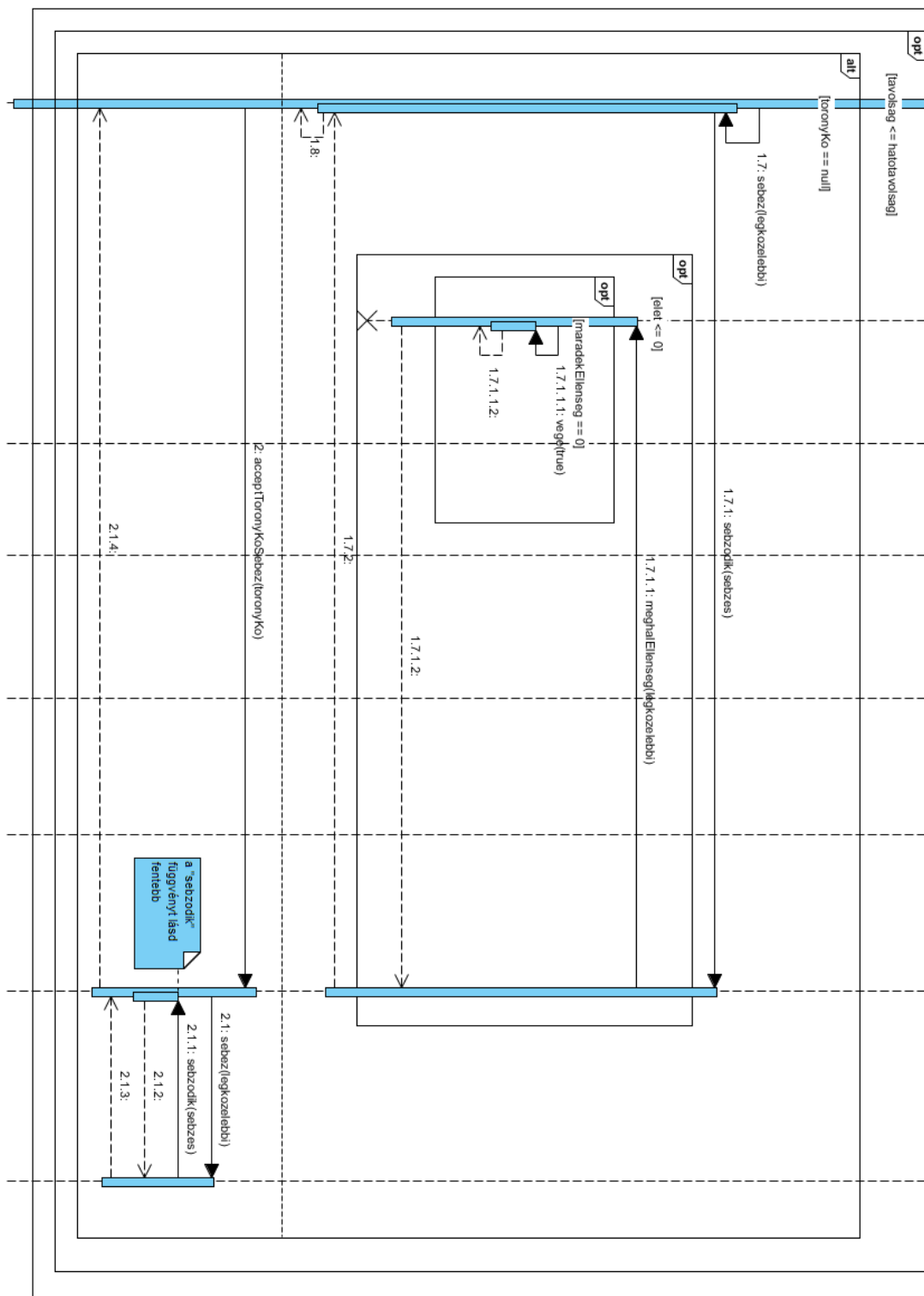
4.4.3 Játék léptetése (2 oldalas)



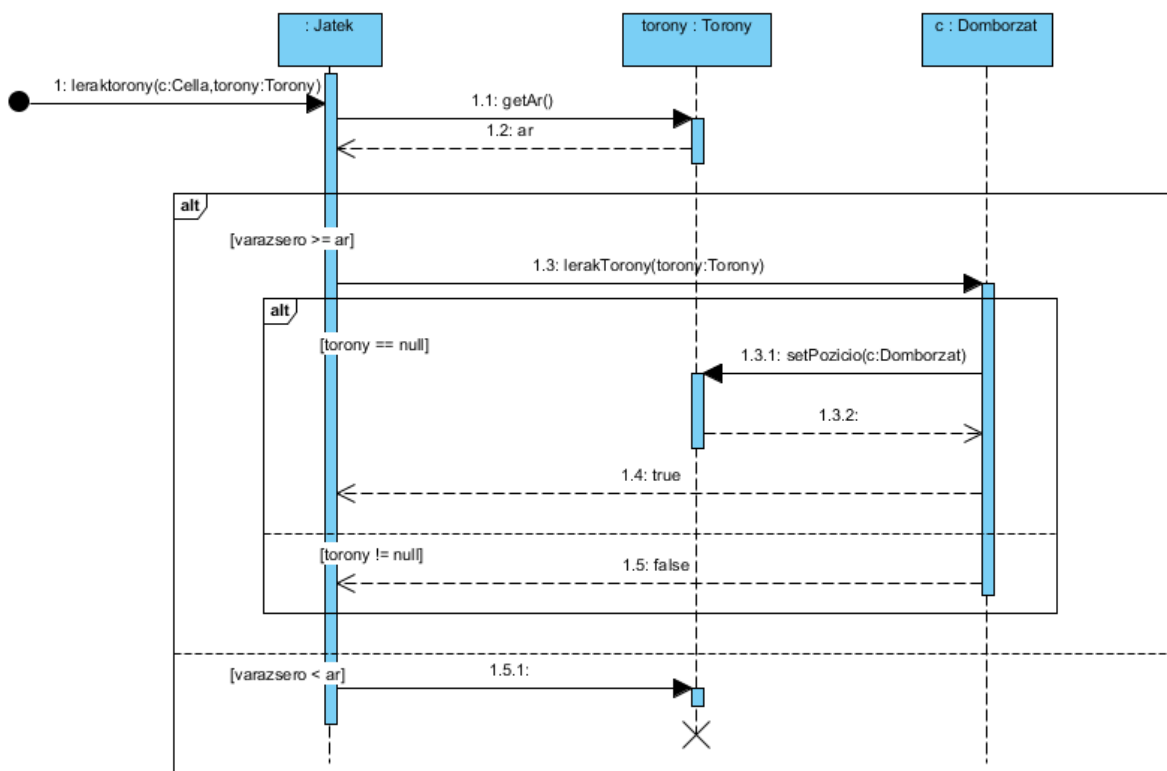


4.4.4 Torony tüzelése (2 oldalas)

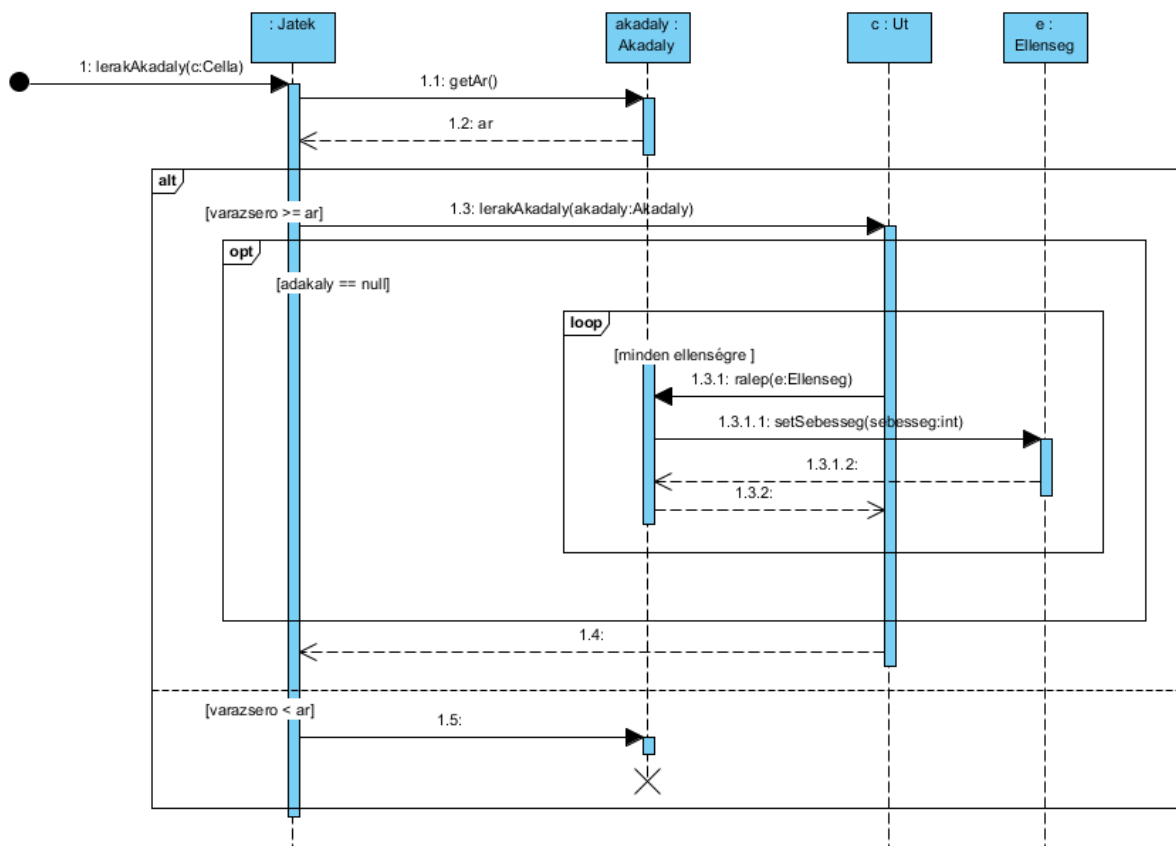




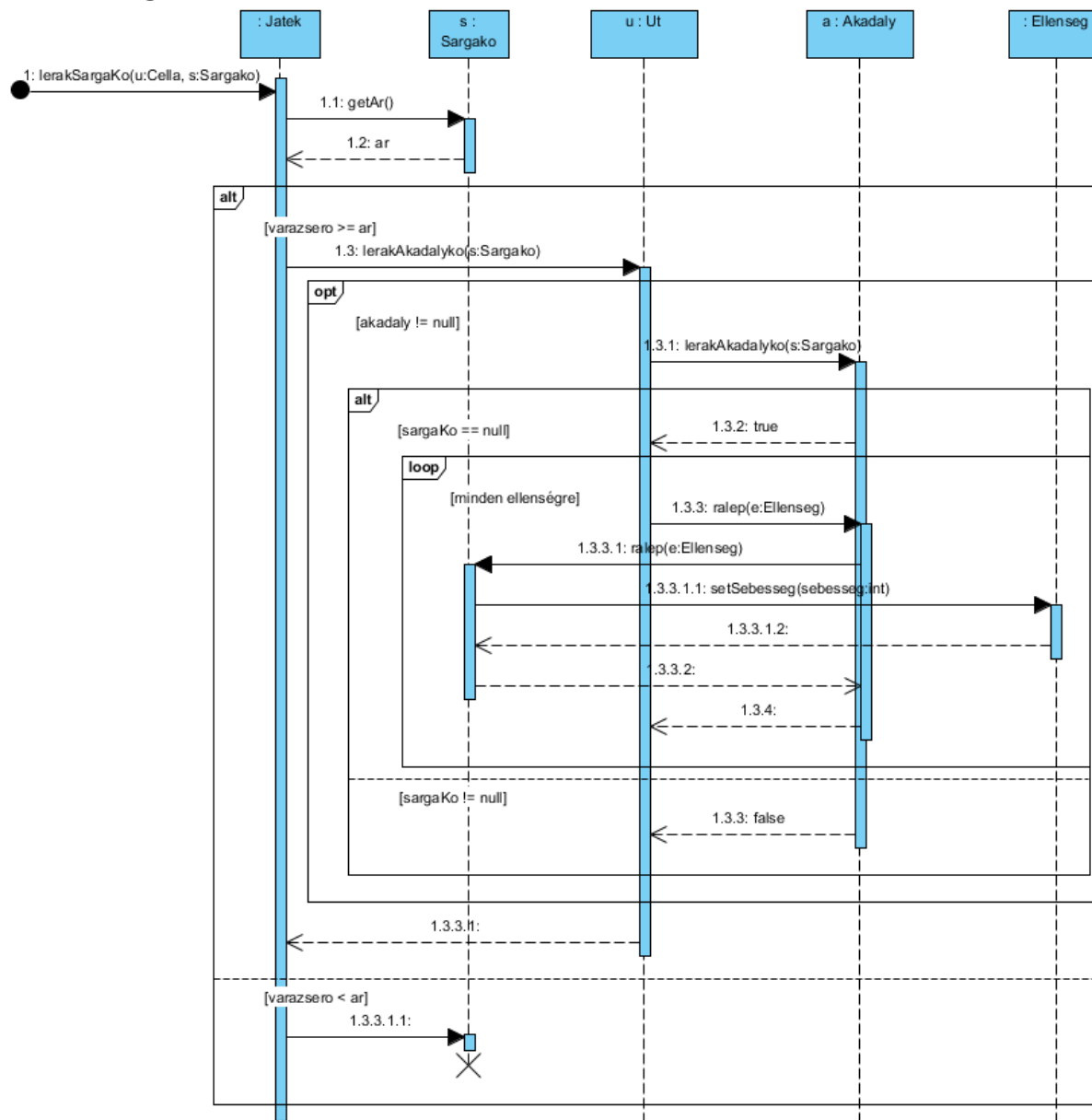
4.4.5 Torony lerakása domborzatra



4.4.6 Akadály lerakása útra



4.4.7 Sárga kő lerakása útra

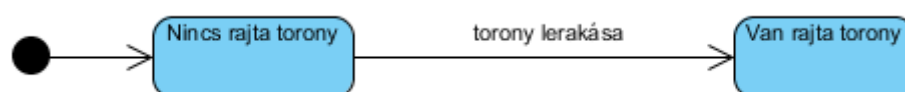


4.5 State-chartok

4.5.1 Akadaly



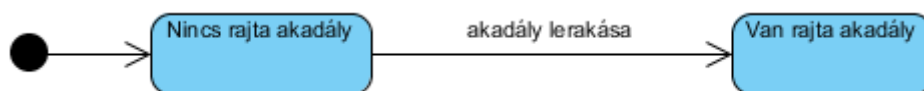
4.5.2 Domborzat



4.5.3 Torony



4.5.4 Út



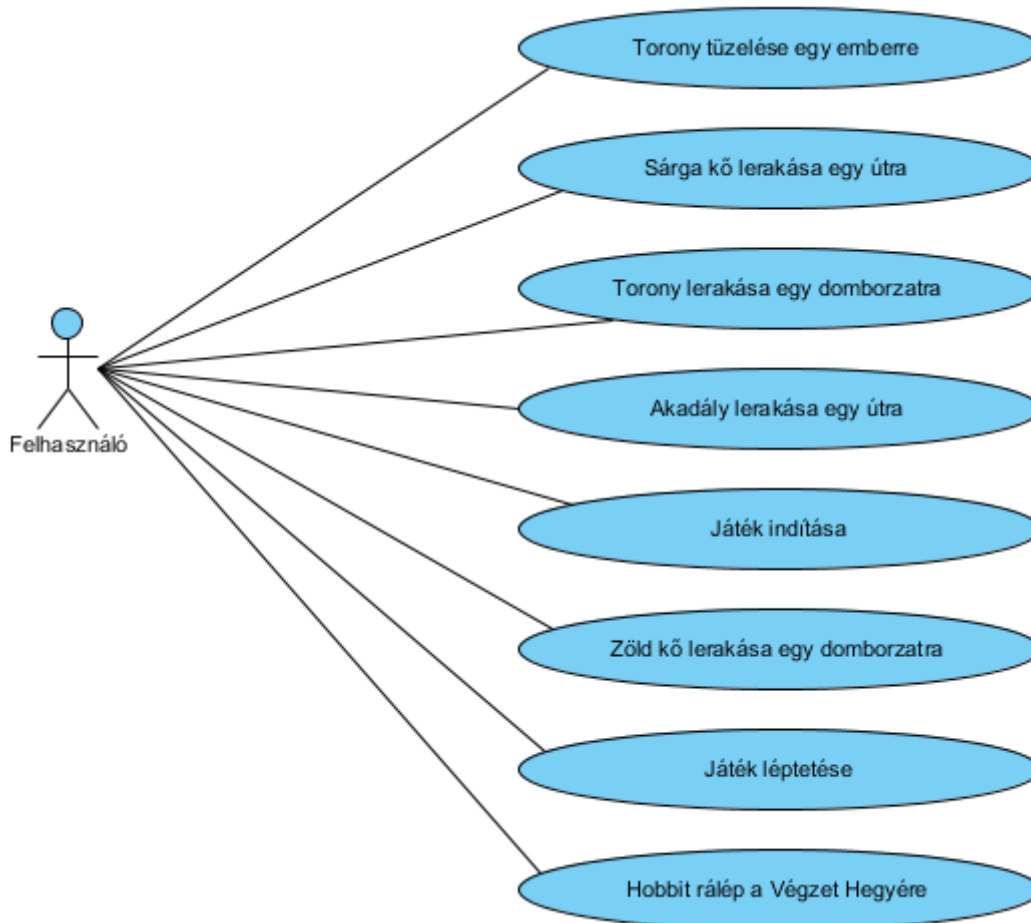
4.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.03.07. 19:00	2 óra	Lipták Sebők Szepes Varga	Harmadik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása
2014.03.08. 16:00	2 óra	Sebők	4.2 elkészítése
2014.03.09. 00:00	3 óra	Szepes	4.3 elkészítése, 4.2 apró hibáinak észrevétele
2014.03.09. 16:00	2 óra	Sebők	4.2 javítása, 4.4.4 elkészítése, 4.5 elkészítése
2014.03.09. 19:00	1.5 óra	Varga	4.4.1, 4.4.2, 4.4.3 elkészítése
2014.03.09. 23:00	1 óra	Varga	4.4 Javítások
2014.03.09. 23:30	4 óra	Lipták	4.1 javítása, 4.4.5, 4.4.6, 4.4.7 elkészítése
2014.03.10. 10:00	1 óra	Sebők	Javítások, készre formázás

5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ei

5.1.1 Use-case diagram



5.1.2 Use-case leírások

5.1.2.1 Torony tüzelése egy emberre

Use-case neve	Torony tüzelése egy emberre
Rövid leírás	A torony tüzel a legközelebbi ellenségre. Jelen esetben egy ember fajú ellenség van a játékban, így arra. Ugyanez a use-case érvényes a többi fajú ellenség sebzésére is.
Aktorok	Felhasználó
Forgatókönyv	Ha a torony tüzelésre kész, először kiválasztja a legközelebbi ellenséget. Ezt az olvashatóság érdekében egy külön diagramon jelöljük. Ha ez a legközelebbi ellenség hatótávolságon belül van, akkor a tornyon lévő kő, vagy annak hiányában maga a torony sebzést ad le rá. Ha az ellenség a sebzés következtében meghalt, ezt jelzi. Ha már nincs több ellenség (tehát az imént meghalt volt az utolsó), akkor vége a játéknak, nyertünk!

5.1.2.2 Sárga kő lerakása egy útra

Use-case neve	Sárga kő lerakása egy útra
Rövid leírás	A felhasználó lerak egy sárgakövet egy útra.
Aktorok	Felhasználó
Forgatókönyv	Ha a varázserő nagyobb, mint a sárga kő ára, az úton van már akadály, de azon még nincs sárga kő, akkor lerakjuk a sárga kövünket, különben a referenciáját elengedjük. Sikeres lerakáskor az adott úton lévő ellenségek sebességét beállítjuk (lassítjuk őket). Jelen esetben egy ember fajú ellenség tartózkodik az úton, a többi fajú ellenségnél is megegyezően történne a lassítás.

5.1.2.3 Torony lerakása egy domborzatra

Use-case neve	Torony lerakása egy domborzatra
Rövid leírás	A felhasználó lerak egy tornyot az egyik domborzatra.
Aktorok	Felhasználó
Forgatókönyv	A játék lekéri a torony árát. Amennyiben legalább annyi varázsereje van, és az adott domborzaton még nincs torony, akkor elhelyezi rajta. Sikeres elhelyezés esetén a játék hozzáadja az újonnan elhelyezett tornyot a toronylistájához.

5.1.2.4 Akadály lerakása egy útra

Use-case neve	Akadály lerakása egy útra
Rövid leírás	A felhasználó lerak egy akadályt az egyik útra.
Aktorok	Felhasználó
Forgatókönyv	A játék lekéri az akadály árát. Amennyiben legalább annyi varázsereje van, és az adott úton még nincs akadály, akkor elhelyezi rajta. Majd az akadály az összes az adott úton lévő ellenségnek (jelen esetben egy ember fajú ellenségnek) beállítja az új sebességét.

5.1.2.5 Játék indítása

Use-case neve	Játék indítása
Rövid leírás	Cellák létrehozása, pozíciójuk beállítása, az útcellákhoz az utánuk következő útcellák (tehát az, hogy hova kell onnan lépni az ellenségeknek) beállítása.
Aktorok	Felhasználó
Forgatókönyv	A játék jelen esetben létrehoz két utat és hozzájuk két pontot (melyek a koordinátájuknak felelnek meg), majd beállítja az utak pozícióját a pontokkal. Ezután a második utat beállítja az első út következő lépésének.

5.1.2.6 Zöld kő lerakása egy domborzatra

Use-case neve	Zöld kő lerakása egy domborzatra
Rövid leírás	Egy zöld kő lerakása egy toronyra. A többi, toronyra rakható kő lerakása is megegyezően történne.
Aktorok	Felhasználó
Forgatókönyv	A Játék megkérdezi a kőtől az árát. Ha van elegendő

	varázserő, a játék továbbadja a követ a domborzatnak. Ha a domborzaton van torony, a kő továbbítódik neki és ha még nincs köve, akkor elmenti magának. Ezután a kő beállítja a torony sebességét és hatótávolságát.
--	---

5.1.2.7 Játék léptetése

Use-case neve	Játék léptetése
Rövid leírás	Az ellenségek léptetése, tüzelés a tornyokkal, új ellenségek létrehozása és elhelyezése a pályán.
Aktorok	Felhasználó
Forgatókönyv	<p>Léptetjük az ellenségeket (jelen esetben egy hobbit fajú ellenség van a játékban, így azt). Ezt az olvashatóság érdekében egy külön diagramon jelöljük. A léptetés során az ellenség lelép arról az útcélláról, ahol áll és átlép az eltárolt következő útcéllára. Ezután eltárolja az újonnan következő útcéllát és beállításra kerül a sebessége annak megfelelően, hogy van-e akadály vagy sárga kő az új helyén, amire az imént lépett.</p> <p>Ezután tüzelünk a tornyokkal (jelen esetben egy darab torony van a játékban, így azzal). A tüzelés menete az 5.1.2.1 Torony tüzelése egy emberre use-case-ben van külön kifejtve, itt csak jelezzük.</p> <p>Végül pedig ha még szükséges, új ellenségeket hozunk létre és helyezünk el a pályán (jelen esetben egy ember fajú ellenséget).</p> <p>A többi fajú ellenségnél is megegyezően történne a léptetés, ill. a létrehozás.</p>

5.1.2.8 Hobbit rálép a Végzet Hegyére

Use-case neve	Hobbit rálép a Végzet Hegyére
Rövid leírás	<p>Egy hobbit fajú ellenség rálép a Végzet Hegyére, a játéknak vége (vesztettünk).</p> <p>Más fajú ellenségeknél a játék megegyező módon ér véget.</p>
Aktorok	Felhasználó
Forgatókönyv	Egy hobbit fajú ellenség rálép a Végzet Hegyére, a játéknak vége (vesztettünk).

5.2 A szkeleton kezelői felületének terve, dialógusok

A program indításkor egy sorszámozott listában felsorolja a támogatott use-case-eket, bekéri a felhasználótól a választott use-case sorszámát, lefuttatja azt, majd visszatér a listához. A kilépéshez külön listaelem áll rendelkezésre.

Egy use-case futtatása során az egyes kimenetek új sorokban, behúzásokkal strukturált formában jelennek meg. A kimenetek a következő típusúak lehetnek:

- függvényhívás kezdete
Az üzenet formátuma: „OBJEKTUM.FÜGGVÉNY hívás (PARAMÉTEREK)”, ahol az OBJEKTUM a hívott függvényhez tartozó objektum neve, a FÜGGVÉNY a hívott

függvény neve, a PARAMÉTEREK pedig a hívott függvényhez tartozó paraméterek, típusukkal együtt.

A behúzást egy tabulátorral növeljük a további kimeneteknél.

- függvényhívás vége

A behúzást egy tabulátorral csökkentjük ennél a sornál és a további kimeneteknél.

Az üzenet formátuma: „OBJEKTUM.FÜGGVÉNY visszatérés (VISSZATÉRÉSI_ÉRTÉK)”, ahol az OBJEKTUM a visszatérő függvényhez tartozó objektum neve, a FÜGGVÉNY a visszatérő függvény neve, a VISSZATÉRÉSI pedig a függvény visszatérési értéke, típussal együtt, void esetén elhagyásra kerül.

- objektum létrehozása (konstruktorhívás)

Az üzenet formátuma: „OBJEKTUM létrehozása (PARAMÉTEREK)”, ahol az OBJEKTUM a létrehozott objektum neve, a PARAMÉTEREK pedig a konstruktorához tartozó paraméterek, típusukkal együtt. A konstruktorból való visszatérést a szekvencia diagram jelöléseivel összhangban nem jelöljük.

- megjegyzés

Egyedi üzenet, mely megfelel a szekvencia diagramon jelölt megjegyzésnek.

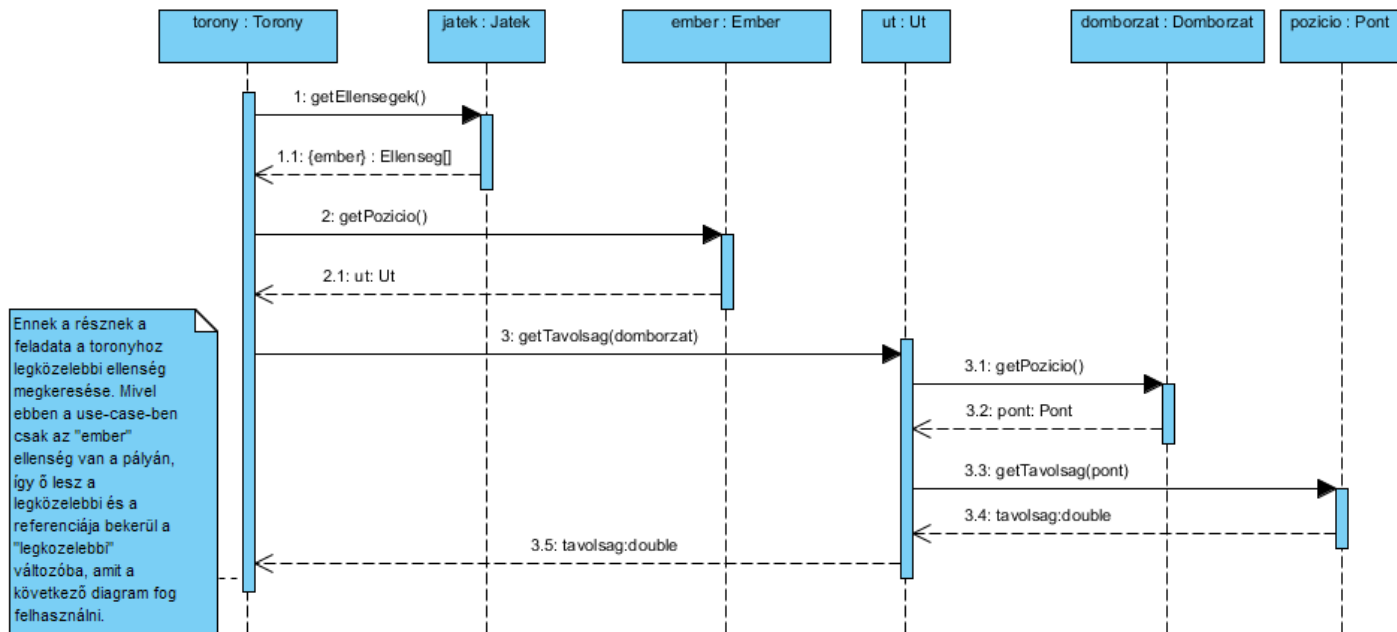
- választási lehetőség

Kérdés a felhasználóhoz arról, hogy a use-case egy elágazásában melyik ágon kíván továbbhaladni, vagy más paramétereikről. A lehetséges válaszlehetőségek a kérdés után egy zárójelben kerülnek felsorolásra. A kérdés megjelenítése után a program bekéri és értelmezi a felhasználó választát. Rosszul formált válasz esetén újra felteszi a kérdést, mindaddig, amíg jól formált választ nem kap.

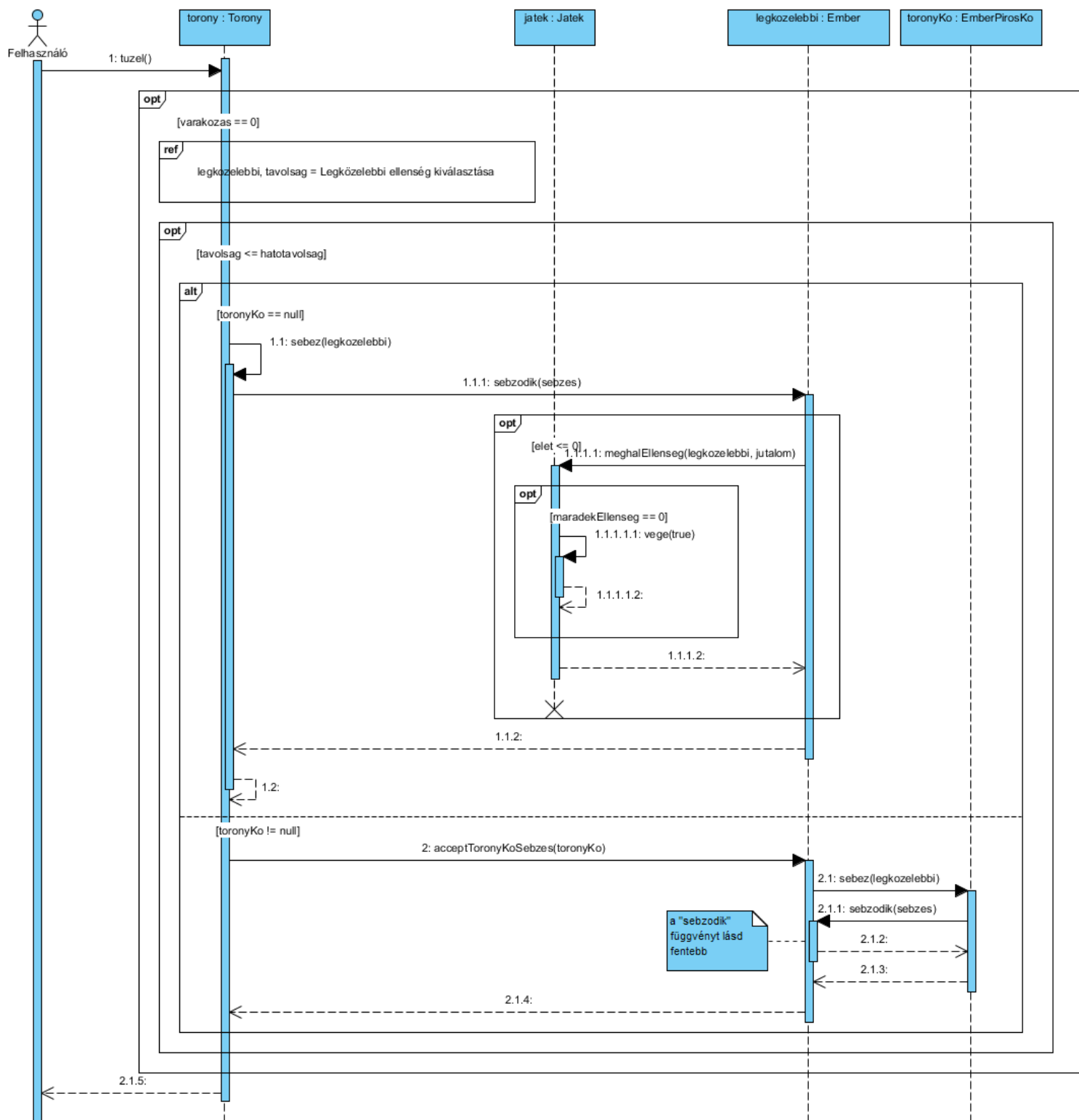
5.3 Szekvencia diagramok a belső működésre

5.3.1 Legközelebbi ellenség kiválasztása

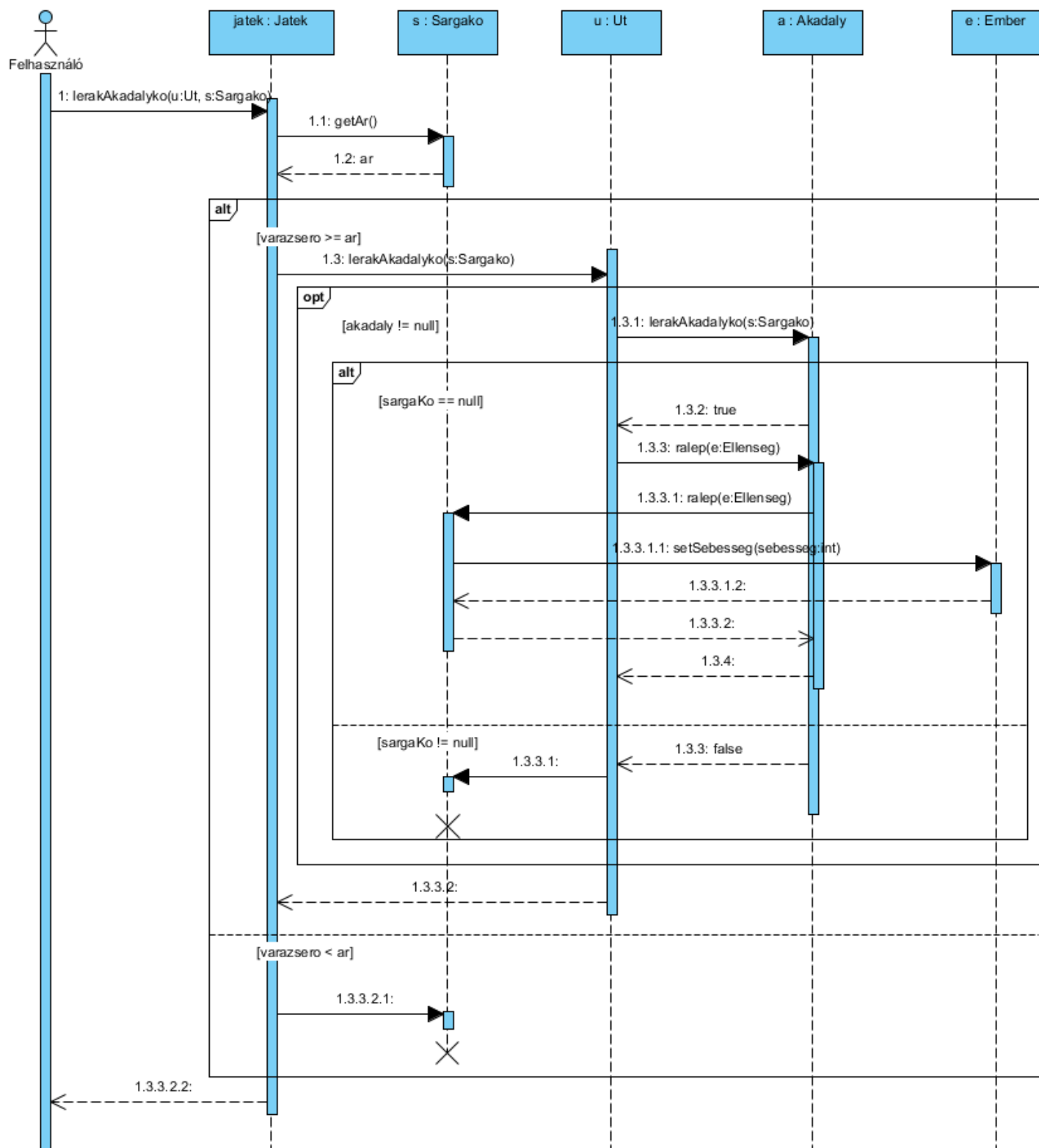
A következő (5.3.2 Torony tüzelése egy emberre) use-case-ben használjuk, a könnyebb olvashatóság érdekében lett különválasztva.



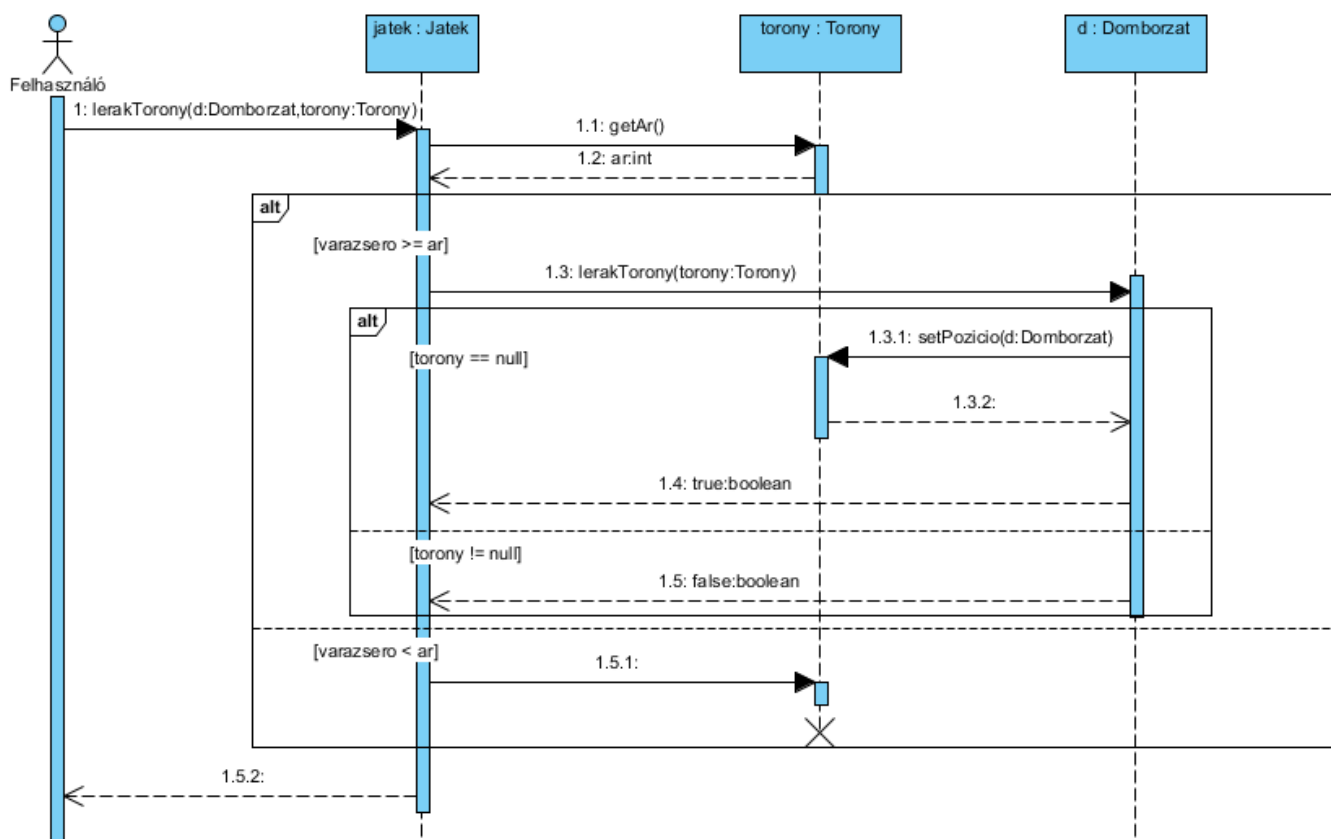
5.3.2 Torony tüzelése egy emberre



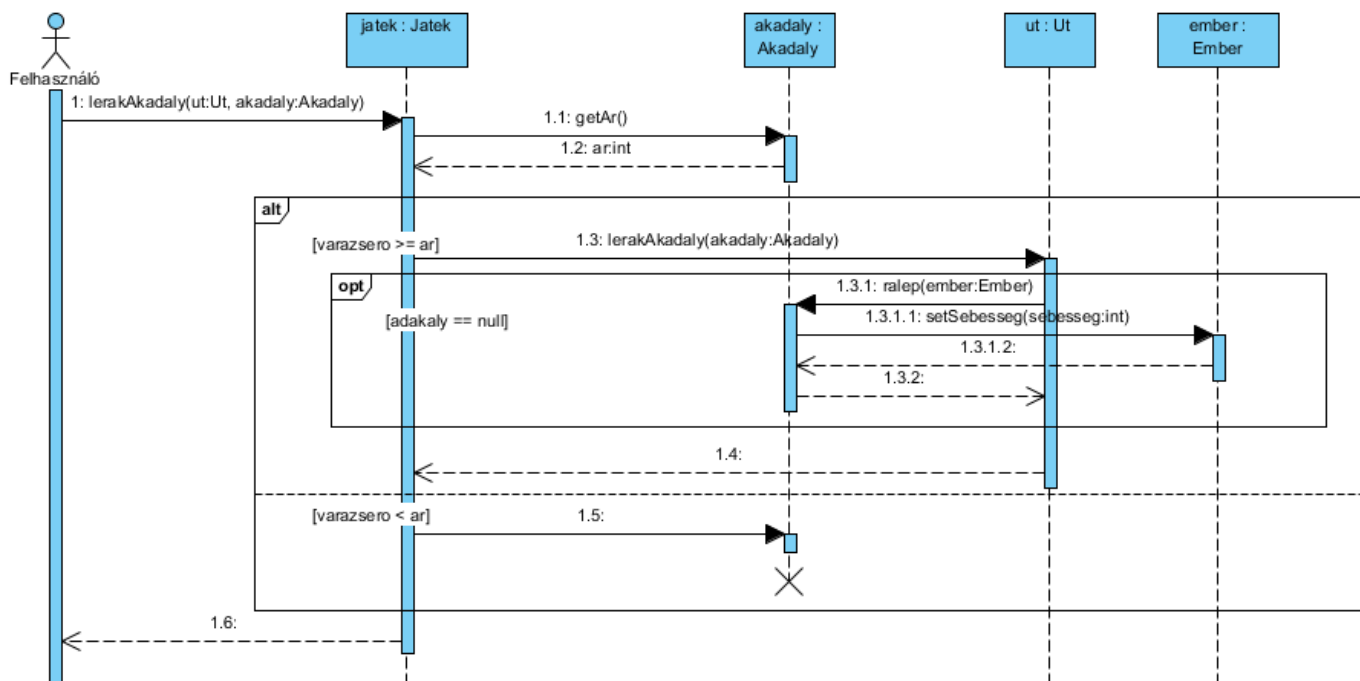
5.3.3 Sárgakő lerakása egy útra



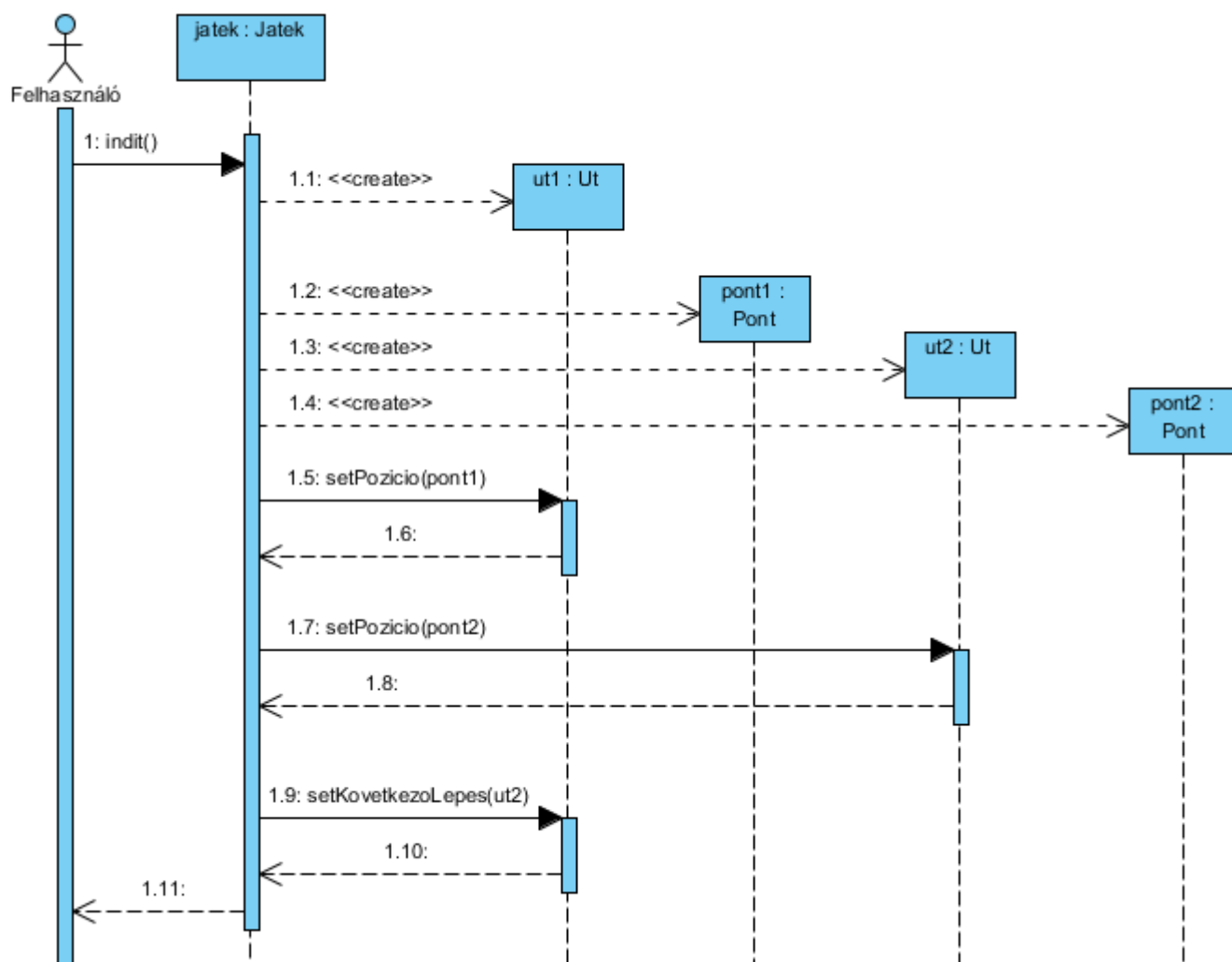
5.3.4 Torony lerakása egy domborzatra



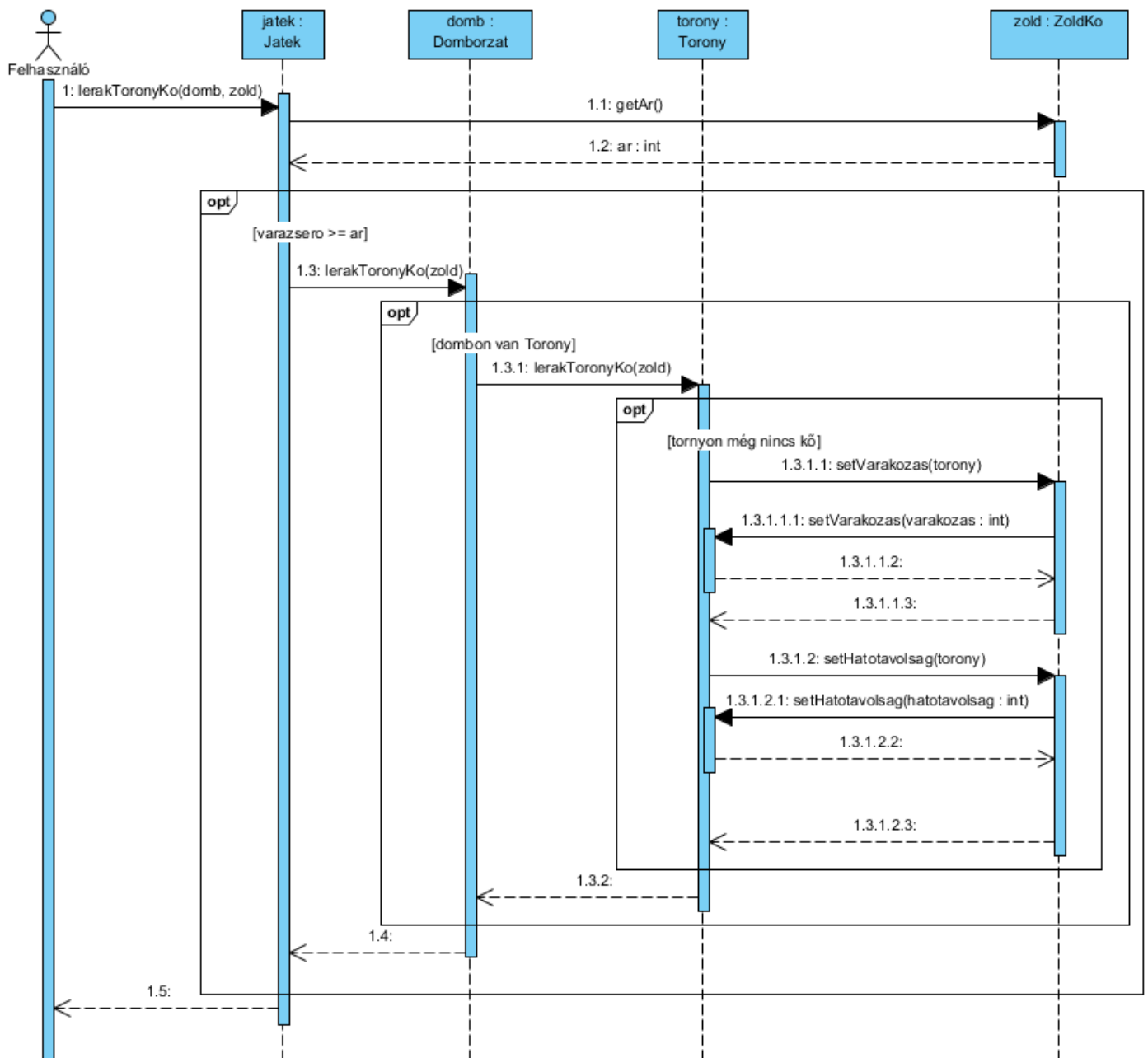
5.3.5 Akadály lerakása egy útra



5.3.6 Játék indítása

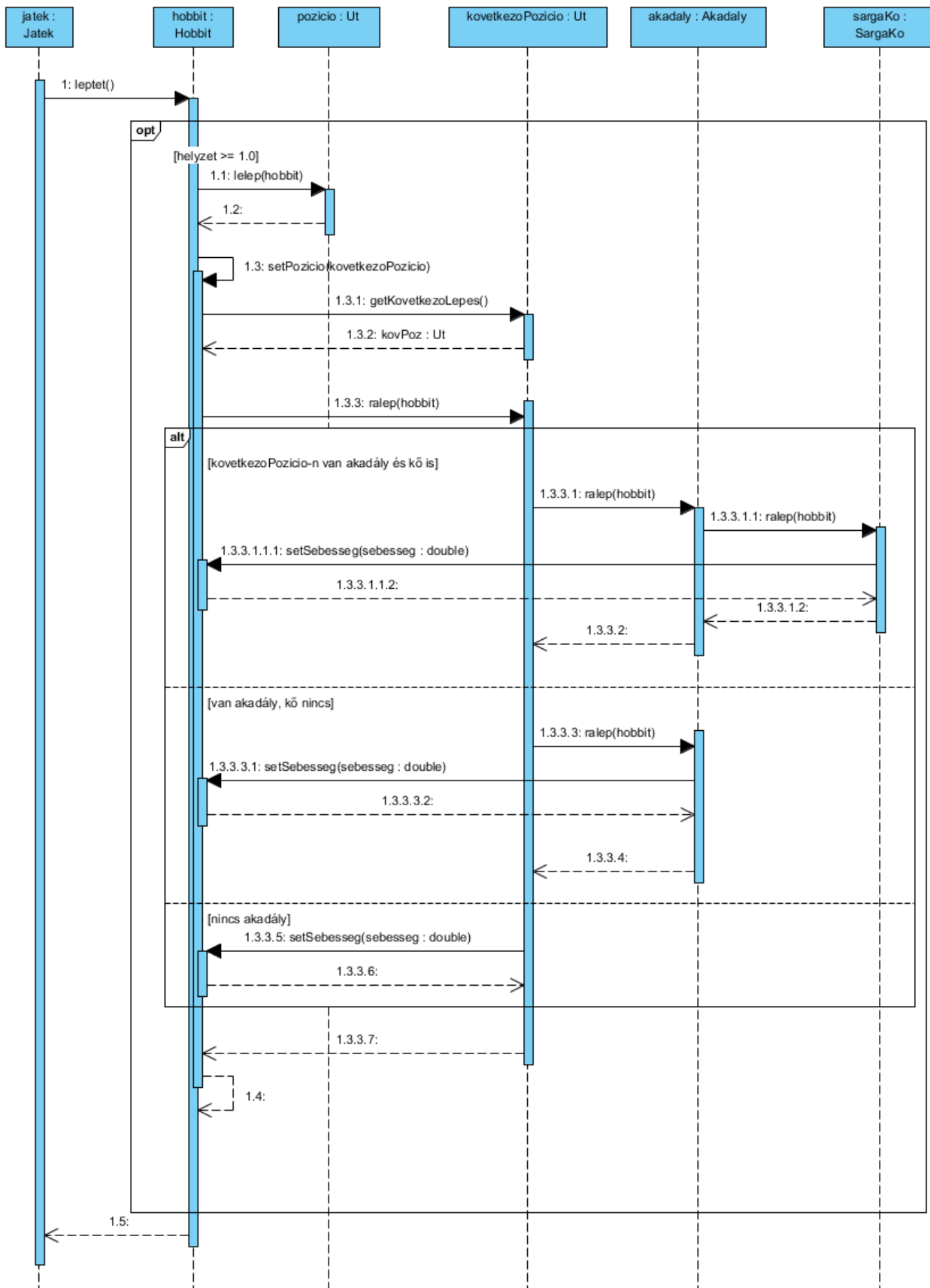


5.3.7 Zöld kő lerakása egy domborzatra

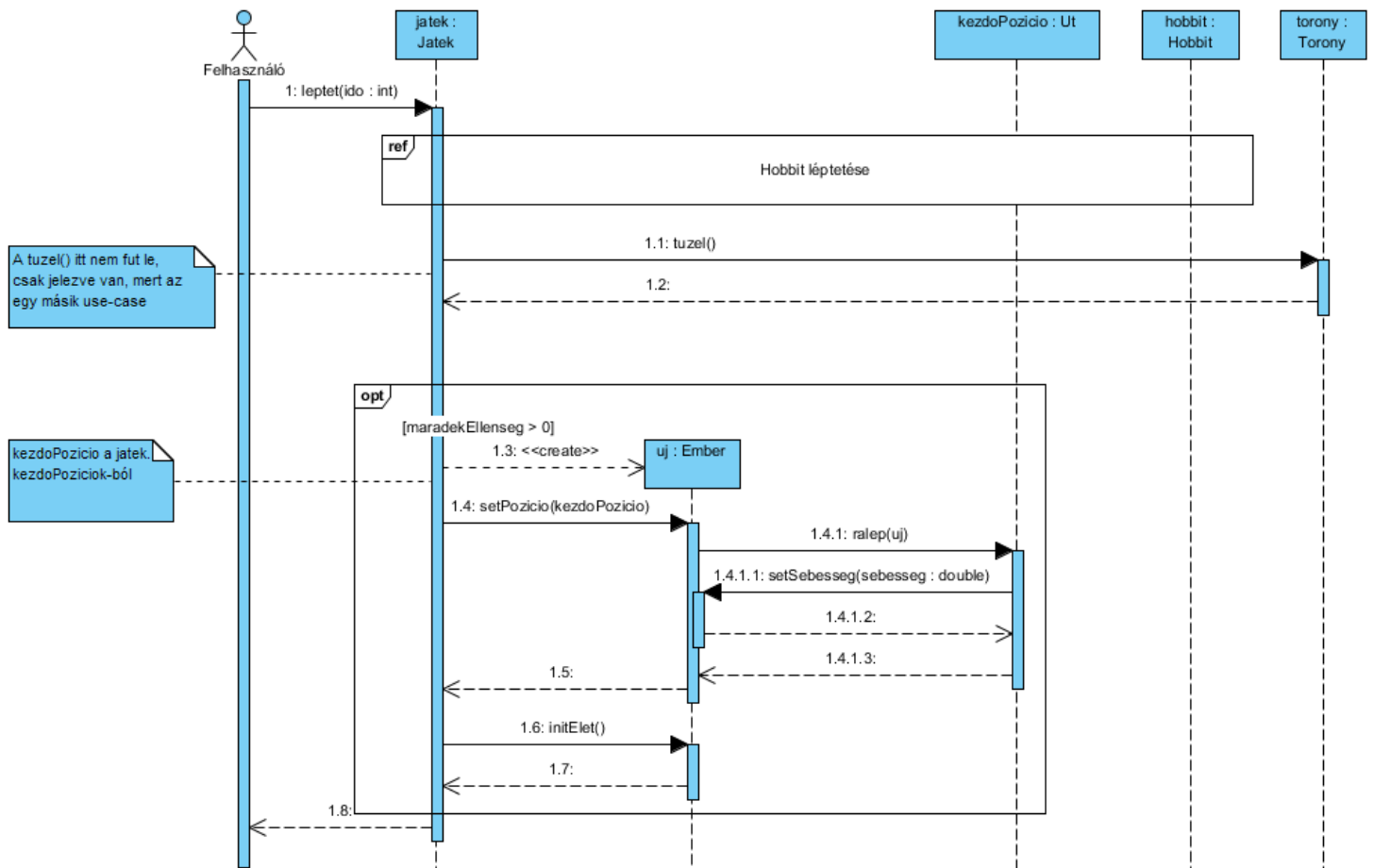


5.3.8 Hobbit léptetése

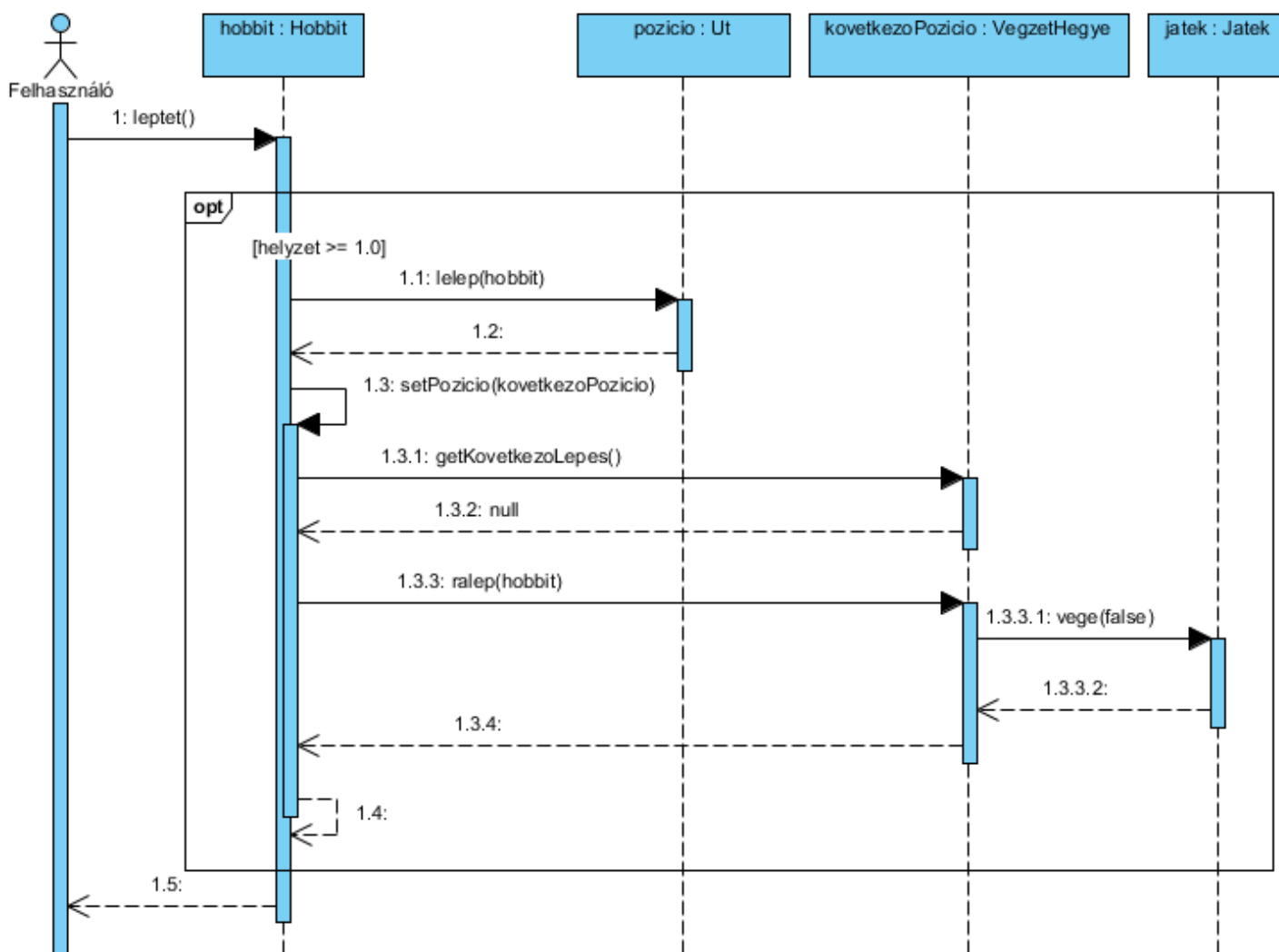
A következő (5.3.9 Játék léptetése) use-case-ben használjuk, a könnyebb olvashatóság érdekében lett különválasztva.



5.3.9 Játék léptetése

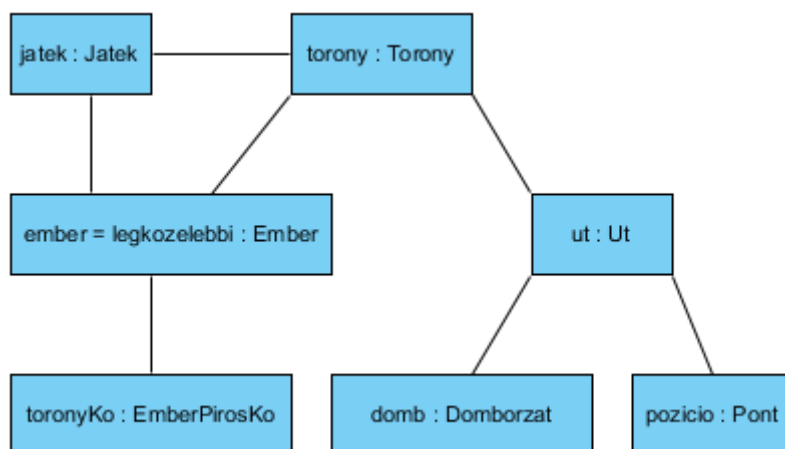


5.3.10 Hobbit rálép a Végzet Hegyére



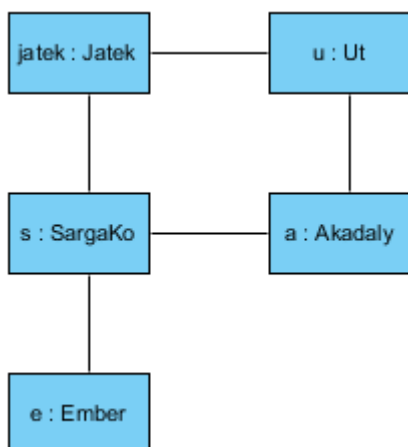
5.4 Kommunikációs diagramok

5.4.1.1 Torony tüzelése egy emberre

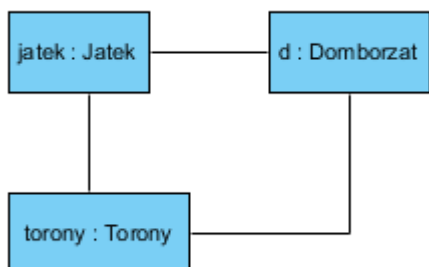


5.4.1.2 Sárga kő lerakása egy útra

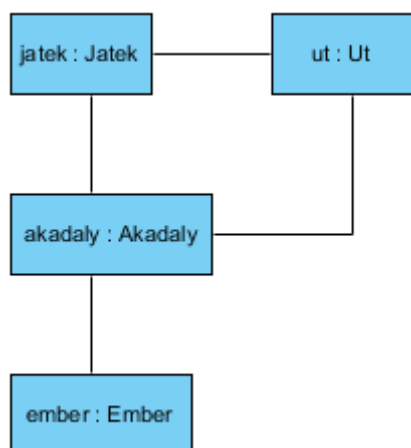
A referencia elengedését nem jelöljük a kommunikációs diagramon.



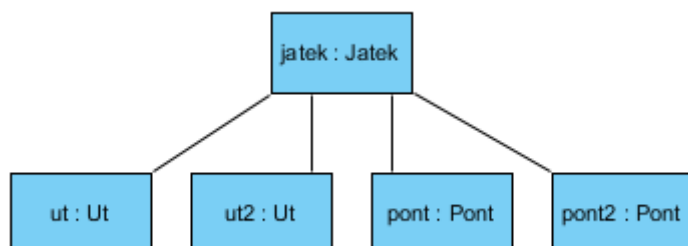
5.4.1.3 Torony lerakása egy domborzatra



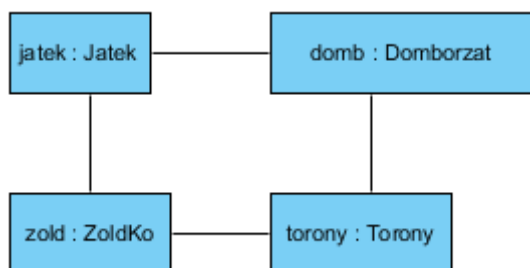
5.4.1.4 Akadály lerakása egy útra



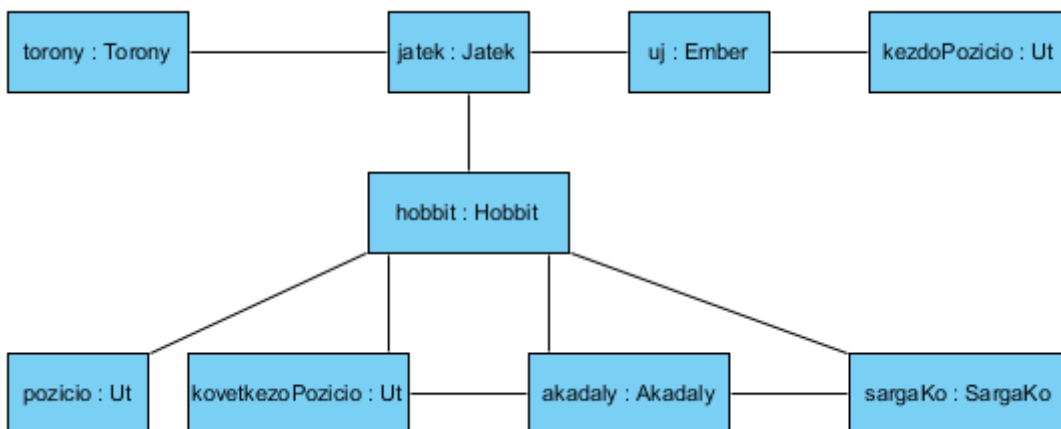
5.4.1.5 Játék indítása



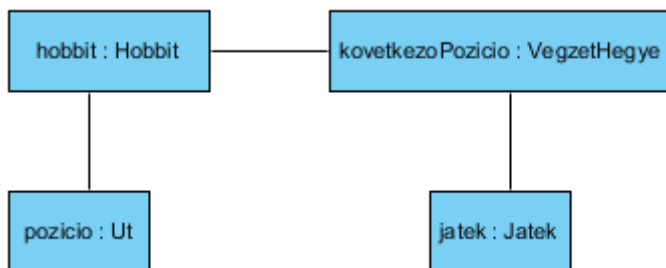
5.4.1.6 Zöld kő lerakása egy domborzatra



5.4.1.7 Játék léptetése



5.4.1.8 Hobbit rálép a Végzet Hegyére



5.4.1.9 Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.03.14. 19:00	0,5 óra	Lipták Sebők Szepes Varga	Negyedik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása.
2014.03.15. 18:30	2 óra	Szepes	5.3.1, 5.3.2, 5.3.3, 5.3.4 elkészítése
2014.03.16. 00:30	2 óra	Szepes	5.3.1, 5.3.2 javítása, use case leírások, 5.4.1.1, 5.4.1.2, 5.4.1.3 elkészítése
2014.03.16. 07:00	2,5 óra	Lipták	5.1.2.3, 5.1.2.4, 5.3.5, 5.3.6 elkészítése
2014.03.16. 11:30	2,5 óra	Varga	5.3.6, 5.3.7, 5.3.8, 5.3.9 elkészítése
2014.03.16. 15:30	2,5 óra	Varga	5.3.8, 5.3.9 javítása, 5.1.2.5, 5.1.2.6, 5.1.2.7, 5.1.2.8 elkészítése
2014.03.16. 20:30	1 óra	Lipták	Javítások, 5.4.1.4, 5.4.1.5 elkészítése
2014.03.16. 22:00	2 óra	Sebők	Többiek munkájának javítása, 5.1.1 Use-case diagram, 5.2 A szkeleton kezelői felületének terve, dialógusok, 5.1.2.8 Hobbit rálép a Végzet Hegyére, 5.3.10 Hobbit rálép a Végzet Hegyére elkészítése, 5.4 Kommunikációs diagramok kiegészítése

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Akadaly.java	1 960 B	2014.03.20. 17:02	Akadaly osztály
AkadalyLerakUseCase.java	594 B	2014.03.22. 22:41	5.3.5 Akadály lerakása egy útra use-case
Cella.java	2 017 B	2014.03.20. 17:02	Cella osztály
Domborzat.java	1 427 B	2014.03.20. 17:02	Domborzat osztály
Ellenseg.java	2 877 B	2014.03.20. 17:02	Ellenseg osztály
Ember.java	1 428 B	2014.03.20. 17:02	Ember osztály
EmberPirosKo.java	264 B	2014.03.20. 17:02	EmberPirosKo osztály
Hobbit.java	1 529 B	2014.03.20. 17:02	Hobbit osztály
HobbitLepteteseUseCase.java	839 B	2014.03.20. 17:02	5.3.8 Hobbit léptetése use-case
HobbitPirosKo.java	183 B	2014.03.23. 20:15	HobbitPirosKo osztály
HobbitRalepVegzetHegyereUseCase.java	615 B	2014.03.20. 17:02	5.3.10 Hobbit rálép a Végzet Hegyére use-case
Jatek.java	6 404 B	2014.03.22. 16:49	Jatek osztály
JatekInditasaUseCase.java	351 B	2014.03.20. 17:02	5.3.6 Játék indítása use-case
JatekLepteteseUseCase.java	1 128 B	2014.03.23. 20:15	5.3.9 Játék léptetése use-case
KekKo.java	185 B	2014.03.23. 20:15	KekKo osztály
KonzolSeged.java	5 220 B	2014.03.20. 17:02	KonzolSeged segédosztály
Main.java	670 B	2014.03.20. 18:59	Main osztály (program indulása)
PeldaUseCase.java	531 B	2014.03.20. 16:59	PeldaUseCase osztály
Pont.java	538 B	2014.03.20. 20:02	Pont osztály

SargaKo.java	873 B	2014.03.20. 17:02	SargaKo osztály
SargaKoLerakUseCase.java	625 B	2014.03.20. 17:02	5.3.3 Sárgakő lerakása egy útra use-case
Torony.java	3 987 B	2014.03.22. 16:49	Torony osztály
ToronyKo.java	1 511 B	2014.03.20. 17:02	ToronyKo osztály
ToronyLerakUseCase.java	533 B	2014.03.20. 17:02	5.3.4 Torony lerakása egy domborzatra use- case
ToronyTuzeleseUseCase.java	427 B	2014.03.22. 22:41	5.3.2 Torony tüzelése egy emberre use-case
Torp.java	688 B	2014.03.22. 22:41	Torp osztály
TorpPirosKo.java	173 B	2014.03.20. 17:02	TorpPirosKo osztály
Tunde.java	693 B	2014.03.20. 17:02	Tunde osztály
TundePirosKo.java	178 B	2014.03.20. 17:02	TundePirosKo osztály
UseCase.java	217 B	2014.03.20. 17:02	UseCase osztály
Ut.java	4 172 B	2014.03.20. 19:41	Ut osztály
VegzetHegye.java	1 555 B	2014.03.20. 17:02	VegzetHegye osztály
ZoldKo.java	418 B	2014.03.20. 17:02	ZoldKo osztály
ZoldKoLerakasaDomborzatraUseCase.java	572 B	2014.03.20. 17:02	5.3.7 Zöld kő lerakása egy domborzatra use- case

6.1.2 Fordítás

A követelményekben leírtaknak megfelelően a program fordításához egy olyan parancssori környezetre van szükség, melyben a Java 1.7-es fordítókörnyezet (**javac** program) elérhető.

(Egy parancssori ablakban a **javac** mappáját a PATH környezeti változóhoz a

```
set PATH=%PATH%;C:\Program Files\Java\jdk1.7.0_40\bin
```

paranccsal adhatjuk hozzá, az elérési útvonalat értelemszerűen átírva a saját környezetnek megfelelően.)

Egy ilyen parancssorban navigáljunk a projekt gyökérkönyvtárába és adjuk ki a

```
del bin\*.class
javac src\*.java -d bin
```

parancsokat.

6.1.3 Futtatás

A követelményekben leírtaknak megfelelően a program futtásához egy olyan parancssori környezetre van szükség, melyben a Java 1.7-es futtatókörnyezet (**java** program) elérhető.

(Egy parancssori ablakban a **java** mappáját a PATH környezeti változóhoz a

```
set PATH=%PATH%;C:\Program Files\Java\jre7\bin
```

paranccsal adhatjuk hozzá, az elérési útvonalat értelemszerűen átírva a saját környezetnek megfelelően.)

Egy ilyen parancssorban navigáljunk a projekt gyökérkönyvtárába és adjuk ki a

```
java -cp bin Main
```

parancsot.

6.2 Értékelés

Tag neve	Munka százalékban
Lipták Levente	25%
Szepes Nóra	25%
Varga Gergő	25%
Sebők Márton	25%

6.3 Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.03.17. 19:00	0,5 óra	Szepes	Osztály diagramból a program kódjának implementálása.
2014.03.19. 20:00	0,5 óra	Lipták Sebők Szepes Varga	Ötödik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása.
2014.03.20. 20:00	1 óra	Sebők	Szkeleton interfészének implementálása.
2014.03.22. 12:00	3 óra	Szepes	5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.10 use-casek, kommentelése, javítása
2014.03.22. 16:00	3 óra	Lipták	5.3.5, 5.3.6 use-casek implementálása,
2014.03.22. 22:00	4 óra	Varga	5.3.6, 5.3.7, 5.3.8, 5.3.9 use-casek implementálása
2014.03.23. 00:00	0,5 óra	Szepes	Fentebbi use-case diagramok hibáinak javítása
2014.03.23. 20:00	2 óra	Sebők	6.0 Skeleton tervének javításai, 6.1 Fordítási és futtatási útmutató elkészítése
2014.03.23. 20:00	0,5 óra	Varga	Javítások
2014.03.23. 23:00	1,5 óra	Lipták	Javítások, az elkészült részek mergelése

7. Prototípus koncepciója

7.0 Specifikáció változás

7.0.1 Köd

Bizonyos időközönként (melyet a tesztelés során állítunk be pontosan) az összes toronyra egyszerre köd ereszkedik, mely a tornyok hatótávolságát megfelezi. Ez az állapot egy torony következő tüzeléséig áll fenn (és arra érvényes), a tüzelés hatására az adott toronyról a köd „felszáll” és a hatótávolság rendes értékével számolunk tovább.

Változtatások (az új diagramokat lásd lentebb)

- az osztálydiagramon a Torony osztály a kodosit() metódussal bővült, mely a hatótávolság változó értékét felezi
- a 4.4.3 Játék léptetése szekvencia diagram bővült a kodosit() hívásával

7.0.2 Elágazások és becsatlakozások

A térképen meghatározott módon egyes utak elágazhatnak, azaz egyes útcellákról több másik útcellára is tovább lehet lépni az ellenségeknek. Ilyenkor véletlenszerűen döntjük el, hogy melyik irányba lépjenek. Az utak egymásba becsatlakozása további megfontolásokat nem igényel.

Változtatások (az új diagramokat lásd lentebb)

- az osztálydiagramon az Ut osztály setKovetkezoLepes(ut : Ut) metódusa módosult setKovetkezoLepesek(utak : Ut[]) -re
- a 4.4.1 Játék indítása szekvencia diagramon az ehhez tartozó hívás módosult
- az Ut osztály getKovetkezoLepes() : Ut függvényének belső működése kiegészült a véletlenszerű választással

7.0.3 Kettévágás

A tornyok és a toronykövek sebzéskor véletlenszerűen küldenek „kettévágós” és „nem kettévágós” lövedéket, ezt a sebzett ellenségnek jelzik. Ha az a sebzés hatására még életben maradt és kettévágós lövedéket kapott, akkor a feladat szerint kettévágja magát, majd ezt jelzi a játéknak, hogy az az új ellenség példányt felvehesse az ellenségek listájába.

Változtatások (az új diagramokat lásd lentebb)

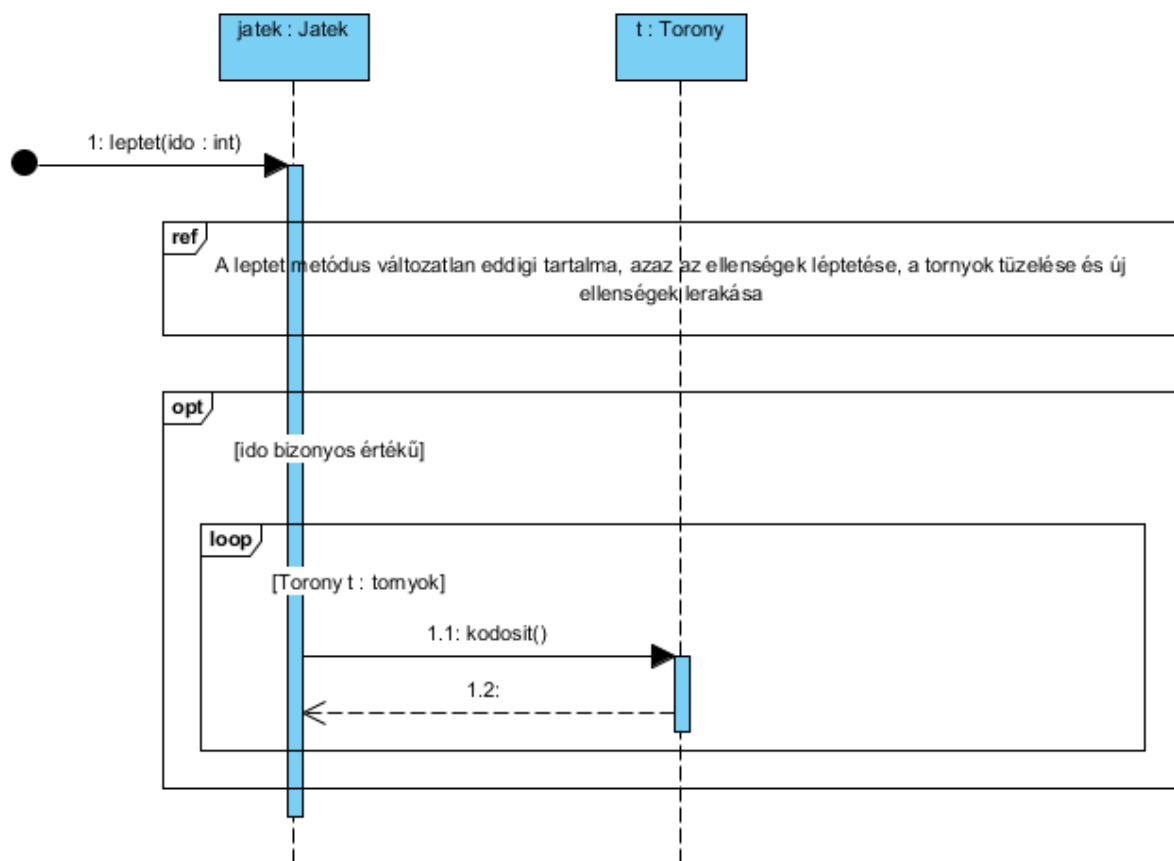
- az osztálydiagramon az Ellenseg osztálynak és az összes leszármazottjának a sebzodik(sebez : int) metódusa módosult sebzodik(sebez : int, kettevag : boolean)-ra,
- ezek az osztályok bővültek egy Ellenseg(jatek : Jatek, ellenseg : Ellenseg) konstruktorral (leszármazottak értelemszerűen), mely a paraméterként átadott ellenség változóit lemásolja
- a Jatek osztály az addEllenseg(ellenseg : Ellenseg) metódussal bővült
- a 4.4.4 Torony tüzelése szekvencia diagramon az ehhez tartozó hívás módosult, továbbá bővült az új ellenség létrehozásával és annak berakásával az ellenségek listájába
- a Torony osztály, a ToronyKo osztály, ill. annak leszármazottainak a sebez metódusa belső működésében kiegészült a véletlenszerű választással

2015-04-14

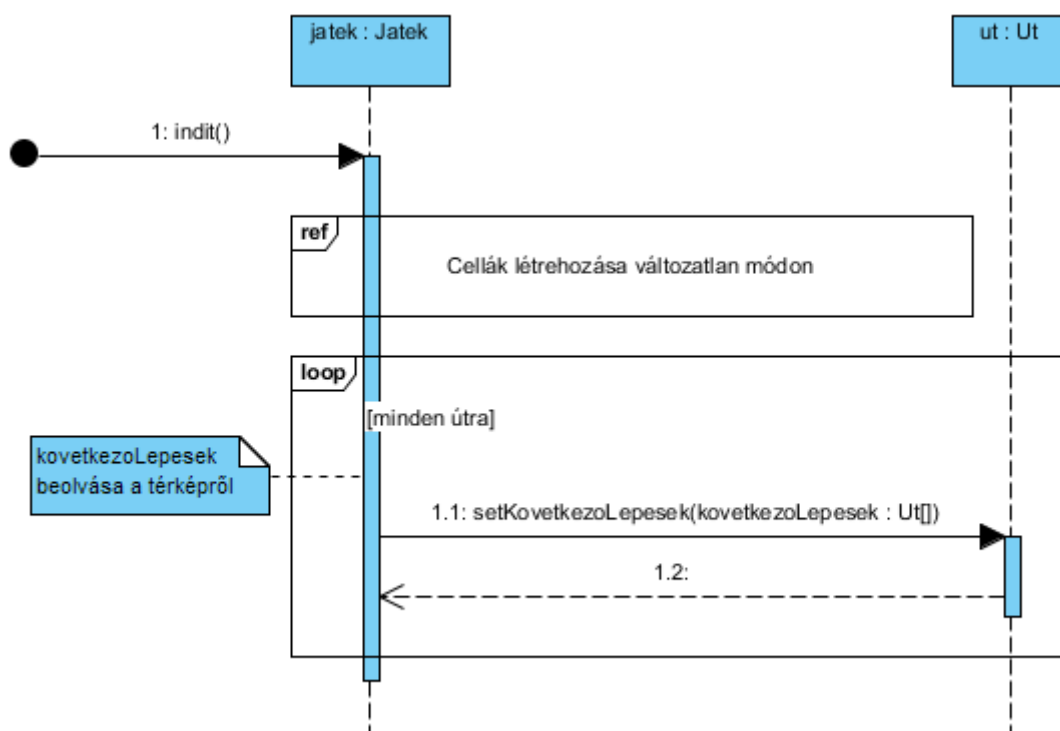


7.0.5 Módosult szekvencia diagramok

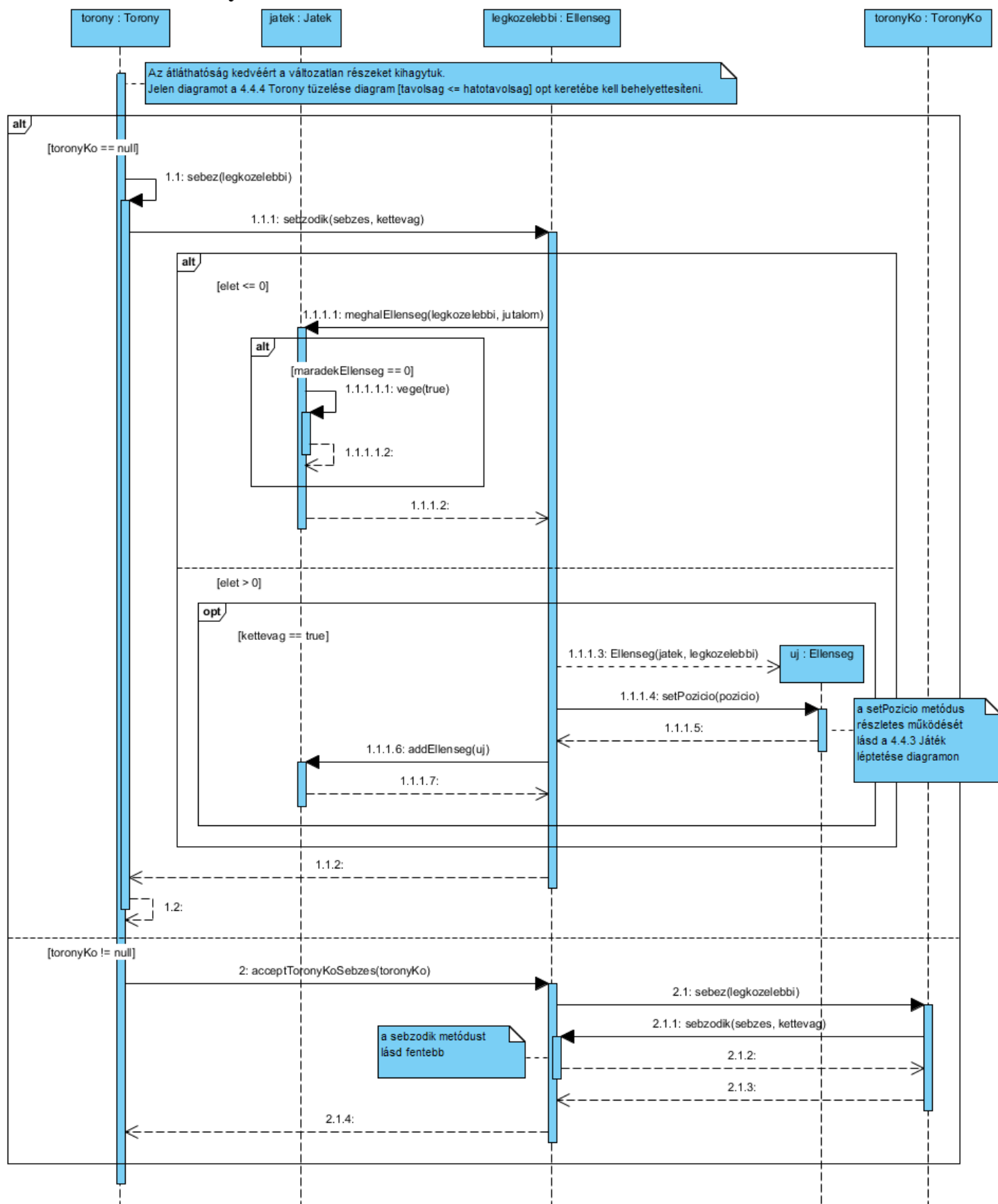
7.0.5.2 Játék léptetésének változásai



7.0.5.1 Játék indításának változásai



7.0.5.2 Torony tüzelésének változásai



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A prototípus a programunk grafikus interfész nélküli változata, amely már a végleges algoritmusokat tartalmazza. A felhasználó konzolos felületen keresztül kommunikálhat a prototípussal az alább definiált bemeneti és kimeneti nyelvek ismeretében.

A bemeneti parancsokat soronként kell megadni, ez lehetővé teszi a prototípus parancsonkénti, egyéni tesztelését is a konzolt interaktívan használva, és az előre elkészített tesztesetek fájlból való betöltését is.

Utóbbi esetben a teszteset parancsait tartalmazó fájlt a szabványos bemenetre irányítjuk, a szabványos kimenetet pedig egy új fájlba irányítjuk, melyet összehasonlítva az elvárt kimenetet tartalmazó, előre elkészített fájlokkal, ellenőrizhetjük a prototípus helyes működését.

A véletlenszerű elemek kezeléséhez a véletlen változók értékét kézzel is be lehet állítani, a többszálú működés szimulálását (ami jelen esetben a játék elemeinek automatikus, időzített léptetése) külön léptető parancs bevezetésével oldottuk meg.

A bemeneti és kimeneti nyelvben a pálya egyes celláira koordinátaikkal hivatkozunk, ezek számozása a szokásos mátrix-indexelést követi (sor, oszlop), a bal felső cella az (1, 1) koordinátájú.

Az egyes paramétereket <paraméterNév> formátummal jelöltük, ennek helyére kerül behelyettesítésre a paraméterNév nevű paraméter aktuális értéke.

7.1.2 Bemeneti nyelv

addTorony <x> <y>

Leírás: Egy torony lerakása az (x, y) koordinátájú cellára.

Opciók: x, y: Cella koordinátái.

addZoldKo <x> <y>

Leírás: Egy zöldkő lerakása az (x, y) koordinátájú cellán lévő toronyra.

Opciók: x, y: Cella koordinátái.

addKekKo <x> <y>

Leírás: Egy kékkő lerakása az (x, y) koordinátájú cellán lévő toronyra.

Opciók: x, y: Cella koordinátái.

addTorpPirosKo <x> <y>

Leírás: Egy törp piroskő lerakása az (x, y) koordinátájú cellán lévő toronyra.

Opciók: x, y: Cella koordinátái.

addHobbitPirosKo <x> <y>

Leírás: Egy hobbit piroskő lerakása az (x, y) koordinátájú cellán lévő toronyra.

Opciók: x, y: Cella koordinátái.

addTundePirosKo <x> <y>

Leírás: Egy tünde piroskő lerakása az (x, y) koordinátájú cellán lévő toronyra.

Opciók: x, y: Cella koordinátái.

addEmberPirosKo <x> <y>

Leírás: Egy ember piroskő lerakása az (x, y) koordinátájú cellán lévő toronyra.

Opciók: x, y: Cella koordinátái.

addAkadaly <x> <y>

Leírás: Egy akadály lerakása az (x, y) koordinátájú cellára.

Opciók: x, y: Cella koordinátái.

addSargaKo <x> <y>

Leírás: Egy sárgakő lerakása az (x, y) koordinátájú cellán lévő akadályra.

Opciók: x, y: Cella koordinátái.

tick <mennyi>

Leírás: Játék léptetése.

Opciók: <mennyi>: Ennyiszer lépteti egymás után a játékot.

indit <fájlNév>

Leírás: Pálya betöltése a fájlból, majd a játék elindítása.

Opciók: <fájlNév>: Ebből a fájlból olvassuk be a pályát.

A fájl formátuma a következő:

<pálya szélessége>,<pálya magassága>

<1. sor cellái sorfolytonosan, U - út, D - domborzat, V - Végzet Hegye>

<2. sor cellái sorfolytonosan, U - út, D - domborzat, V - Végzet Hegye>

...

<utolsó sor cellái sorfolytonosan, U - út, D - domborzat, V - Végzet Hegye>

<ellenségek kezdőpozícióinak száma>

<1. kezdőpozíció koordinátái <x>,<y> alakban>

<2. kezdőpozíció koordinátái <x>,<y> alakban>

...

<utolsó kezdőpozíció koordinátái <x>,<y> alakban>

<útcellák száma>

<1. útcelláról lehetséges lépések <út_x>,<út_y>;<x1>,<y1>;<x2>,<y2>;... alakban>

<2. útcelláról lehetséges lépések <út_x>,<út_y>;<x1>,<y1>;<x2>,<y2>;... alakban>

...

<utolsó útcelláról lehetséges lépések <út_x>,<út_y>;<x1>,<y1>;... alakban>

random <melyik> <on/off/auto>

Leírás: A játékban a véletlenszerű elemek be/kikapcsolása.

Opciók:

- **melyik:** Megmondja, hogy melyik elem véletlenszerűségét akarjuk átállítani. Értéke lehet **kettevagas** vagy **utvalasztas**. Ha **kettevagas**-t választunk, akkor azt mondhatjuk meg, hogy a lövedékek mikor válasszák ketté az ellenséget. Az **off** kapcsoló esetén ez soha nem történik meg, **on** esetén mindig, **auto** esetén véletlenszerűen. Az **utvalasztas** esetén az **auto** azt fejezi ki, hogy az ellenségek random választanak utat az elágazásoknál. Az **on** kapcsolóval mindig az első utat választják, az **off** kapcsolóval a legutolsót.
- **on/off/auto:** Az **on** kapcsoló esetén a véletlenszerű esemény mindig bekövetkezik, **off** esetén soha, **auto** esetén pedig véletlenszerűen.

quit

Leírás: Kilépés a programból

Opciók: -

7.1.3 Kimeneti nyelv

addTorony <x> <y>

Kiírja, hogy sikerült-e tornyot rakni a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet tornyot rakni
- Sikerult lerakni a tornyot az (<x>, <y>) koordinataju cellara.

addZoldKo <x> <y>

Kiírja, hogy sikerült-e lerakni a zöldkövet a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet zoldkovet rakni.
- Sikerult lerakni a zoldkovet az (<x>, <y>) koordinataju cellan levo toronyra.

addKekKo <x> <y>

Kiírja, hogy sikerült-e lerakni a kékkövet a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet kekkovet rakni.
- Sikerult lerakni a kekkovet az (<x>, <y>) koordinataju cellan levo toronyra.

addTorpPirosKo <x> <y>

Kiírja, hogy sikerült-e lerakni törp piroskövet a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet torp piroskovet rakni.
- Sikerult lerakni a torp piroskovet az (<x>, <y>) koordinataju cellan levo toronyra.

addHobbitPirosKo <x> <y>

Kiírja, hogy sikerült-e lerakni hobbit piroskövet a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet hobbit piroskovet rakni.
- Sikerult lerakni a hobbit piroskovet az (<x>, <y>) koordinataju cellan levo toronyra.

addTundePirosKo <x> <y>

Kiírja, hogy sikerült-e lerakni tünde piroskövet a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet tunde piroskovet rakni.
- Sikerult lerakni a tunde piroskovet az (<x>, <y>) koordinataju cellan levo toronyra.

addEmberPirosKo <x> <y>

Kiírja, hogy sikerült-e lerakni ember piroskövet a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet ember piroskovet rakni.
- Sikerult lerakni a ember piroskovet az (<x>, <y>) koordinataju cellan levo toronyra.

addAkadaly <x> <y>

Kiírja, hogy sikerült-e lerakni akadályt a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet akadalyt rakni.
- Sikerult lerakni az akadalyt az (<x>, <y>) koordinataju cellara.

addSargaKo <x> <y>

Kiírja, hogy sikerült-e lerakni sárgakövet a cellára.

Lehetséges kimenetek:

- Az (<x>, <y>) koordinataju cellara nem lehet sargakövet rakni.
- Sikerult lerakni a sargakövet az (<x>, <y>) koordinataju cellan levo akadalyra.

tick <mennyi>

Kiírja, hogy az objektumok mit léptek.

Lehetséges kimenetek, ezekből egymás után egyszerre több is megjelenhet:

- <ellenseg> ralepett az (<x>, <y>) cellara.
- <akadaly> lelassította <ellenseg>-et <x>-re
- <sargako> lelassította <ellenseg>-et <x>-re
- <torony> megsebezte <ellenseg>-et <x> sebzessel
- <toronyko> megsebezte <ellenseg>-et <x> sebzessel
- <ellenseg> meghalt
- <ellenseg> kettevagva
- jatek letrehozta <ellenseg>-et
- <x> varazseronk van
- a tornyokra kod szallt
- <torony>-rol a kod felszallt
- <ellenseg> ralepett a Vegzet Hegyere, vesztettunk
- minden ellenseg meghalt, nyertunk

indit <fájlNév>

Lehetséges kimenetek:

- <fájlNév> betoltese sikerult, jatek inditasa.
- <fájlNév> betoltese nem sikerult.

random <melyik> <on/off/auto>

Kimenet:

- <melyik> veletlenszerusegenek atallitasa.

quit

A program kilép, nincs kimenet.

7.2 Összes részletes use-case

Use-case neve	Torony lerakása
Rövid leírás	Egy torony hozzáadása a játékhoz.
Aktorok	Felhasználó
Forgatókönyv	Egy torony kerül a pályára, a megadott cellára. Ilyenkor meghívódik a Jatek addTorony metódusa.

Use-case neve	Zöld kő lerakása
Rövid leírás	Egy zöld kő hozzáadása a játékhoz.
Aktorok	Felhasználó
Forgatókönyv	A megadott cellára ráteszünk egy zöld toronykövet. Ilyenkor a Jatek addToronyKo metódusa hívódik meg.

Use-case neve	Kék kő lerakása
Rövid leírás	Egy kék kő hozzáadása a játékhoz.
Aktorok	Felhasználó
Forgatókönyv	A megadott cellára ráteszünk egy kék toronykövet. Ilyenkor a Jatek addToronyKo metódusa hívódik meg.

Use-case neve	Törp piros kő lerakása
Rövid leírás	Egy törp piros kő hozzáadása a játékhoz.
Aktorok	Felhasználó
Forgatókönyv	A megadott cellára ráteszünk egy piros toronykövet. Ilyenkor a Jatek addToronyKo metódusa hívódik meg.

Use-case neve	Hobbit piros kő lerakása
Rövid leírás	Egy hobbit piros kő hozzáadása a játékhoz.
Aktorok	Felhasználó
Forgatókönyv	A megadott cellára ráteszünk egy piros toronykövet. Ilyenkor a Jatek addToronyKo metódusa hívódik meg.

Use-case neve	Tünde piros kő lerakása
Rövid leírás	Egy tünde piros kő hozzáadása a játékhoz.
Aktorok	Felhasználó
Forgatókönyv	A megadott cellára ráteszünk egy piros toronykövet. Ilyenkor a Jatek addToronyKo metódusa hívódik meg.

Use-case neve	Ember piros kő lerakása
Rövid leírás	Egy ember piros kő hozzáadása a játékhoz.
Aktorok	Felhasználó

Forgatókönyv	A megadott cellára ráteszünk egy toronykövet. Ilyenkor a Jatek addToronyKo metódusa hívódik meg.
---------------------	--

Use-case neve	Akadály lerakása
Rövid leírás	Egy akadály hozzáadása a játékhoz.
Aktorok	Felhasználó
Forgatókönyv	Egy akadály kerül a pályára, a megadott cellára. Ilyenkor meghívódik a Jatek addAkadaly metódusa.

Use-case neve	Sárga kő lerakása
Rövid leírás	Egy sárga kő hozzáadása a játékhoz.
Aktorok	Felhasználó
Forgatókönyv	A megadott cellára ráteszünk egy sárga követ. Ilyenkor a Jatek addAkadalyKo metódusa hívódik meg.

Use-case neve	Léptetés
Rövid leírás	A játékban lezajlik egy időpillanat.
Aktorok	Felhasználó
Forgatókönyv	A játékon belül léptetjük az ellenségeket, tüzelünk a tornyokkal, időnként új ellenséget helyezünk el és ködöt eresztünk a tornyokra. A Jatek-on meghívódik a leptet metódus.

Use-case neve	Indítás
Rövid leírás	Elindítjuk a játékot.
Aktorok	Felhasználó
Forgatókönyv	Elindítjuk a játékot. Ilyenkor betöltjük a paraméterként kapott fájlból a térképet. A Jatek-on meghívódik az indit metódus.

Use-case neve	Random paraméterek beállítása
Rövid leírás	A megadott random paraméter kikapcsolása, bekapcsolása és auto módra állítása.
Aktorok	Felhasználó
Forgatókönyv	Beállíthatjuk, hogy a lövedék mindig kettévágja az ellenségeket (on), soha (off) vagy random (auto). Útválasztás esetében vagy mindig az elsőt választja az ellenség (on), mindig az utolsót (off) vagy randomot (auto).

Use-case neve	Kilépés
Rövid leírás	Kilépés a programból.
Aktorok	Felhasználó
Forgatókönyv	A programból való végleges kilépést biztosítja.

7.3 Tesztelési terv

Teszt-eset neve	Pálya betöltése
Rövid leírás	A pályát betöltjük adott fájlból
Teszt célja	A betöltött pályának meg kell egyeznie a fájl által leírttal.

Teszt-eset neve	Akadály lassítása
Rövid leírás	Megpróbálunk akadályt letenni útra is, domborzatra is, megmérjük a rajta áthaladó, és egy másik úton áthaladó ellenség sebességét.
Teszt célja	Ellenőrizzük, csak az utakra lehet akadályokat letenni, melyek csak a rajtuk áthaladó ellenségeket lassítják.

Teszt-eset neve	Torony lövése
Rövid leírás	Megpróbálunk tornyot letenni útra is, domborzatra is, ellenségeket teszünk a hatótávolságán belülre, különböző távolságra, illetve a hatótávolságán kívülre.
Teszt célja	Biztosítani, hogy a tornyokat csak a domborzatokra lehet letenni, azok a hozzájuk legközelebbi ellenséget sebzik, de csak akkor, ha hatótávolságán belül van.

Teszt-eset neve	Sárgakő hatása
Rövid leírás	Sárgaköveket próbálunk lerakni akadályokra, tornyokra, és üres cellákra, mérjük, az akadályon áthaladó ellenség sebességét, amire rátettük.
Teszt célja	Megvizsgálni, hogy csak az akadályokra helyezhetőek-e a sárgakövek, és növelik-e annak lassítását.

Teszt-eset neve	Piroskő hatása
Rövid leírás	Pirosköveket próbálunk lerakni akadályokra, tornyokra, és üres cellákra, mérjük, hogy mennyivel nő meg a torony sebzése, melyik fajra.
Teszt célja	Kipróbálni, hogy csak toronyra helyezhető-e piros kő, és a sebzése növekedik-e. Ellenőrizzük, hogy minden fajta piros kő faj specifikus, más fajra hatástalan kell, hogy legyen.

Teszt-eset neve	Kékkő hatása
Rövid leírás	Kékköveket próbálunk lerakni akadályokra, tornyokra, és üres cellákra, ellenőrizzük, hogy képes-e az eredeti hatótávolságon éppen kívül eső ellenséget sebezni.
Teszt célja	Biztosítani, hogy csak toronyra helyezhető kékkő, a torony hatótávolsága pedig megnő tőle.

Teszt-eset neve	Zöldkő hatása
Rövid leírás	Zöldköveket próbálunk lerakni akadályokra, tornyokra, és üres cellákra, mérjük, a torony sebzésének

	gyakoriságát.
Teszt célja	Megvizsgálni, hogy csak toronyra lehet-e zöldkövet tenni, a letett kő hatására pedig tényleg gyakrabban lő-e a torony.

Teszt-eset neve	Ellenség osztódik
Rövid leírás	Az ellenségeknek képesnek kell osztódnia, ha olyan sebzés éri.
Teszt célja	Megvizsgálni, hogy akkor és csak akkor osztódik-e az ellenség, amikor ilyen lövedék találja el, az osztódástól pedig pontosan 1 új ellenség keletkezik-e azonos fajjal.

Teszt-eset neve	Ellenség meghal
Rövid leírás	Ellenséget sebzünk több alkalommal, amíg el nem fogy az összes életpontja
Teszt célja	Biztosítani, hogy akkor, és csak akkor hal meg, ha legfeljebb annyi élete van, mint a sebzés mértéke. A halott ellenségtől az várjuk, hogy eltűnik a pályáról.

Teszt-eset neve	A toronyra kőd ereszkedik
Rövid leírás	A toronyra kőd ereszkedik, melytől a hatótávolság lecsökken, így az eredetileg hatótávolságon belüli ellenségeket nem tudja többé sebezni.
Teszt célja	Ellenőrizni, hogy csak toronyra ereszkedhet kőd, a hatótávolság pedig lecsökken tőle.

Teszt-eset neve	Elágazások
Rövid leírás	Az elágazáshoz vezetünk ellenségeket, melyeket más-más irányba terelünk.
Teszt célja	Ellenőrizni, hogy különböző irányokba el tudott indulni az arra küldött ellenség.

Teszt-eset neve	Játék elvesztése
Rövid leírás	Hagyjuk, hogy beérjen egy ellenség a végzet hegyére.
Teszt célja	Megvizsgálni, hogy mi történik, ha egy ellenség beér a végzet hegyére.

Teszt-eset neve	Játék megnyerése
Rövid leírás	Úgy alakítjuk a körülményeket, hogy minden ellenséget el tudjanak pusztítani a tornyok.
Teszt célja	Kipróbálni, hogy amikor már az összes ellenség elfogyott, hogyan jelzi a program azt, hogy nyertünk.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A tesztesetek futtatásakor a szabványos kimenetet fájlba kell irányítani, majd az így kapott fájlokat össze kell hasonlítani az előre megadott (és az elvárt kimenetet tartalmazó) fájlokkal. Ennek szkripteléséhez a diff parancssori eszközt (<http://gnuwin32.sourceforge.net/packages/diffutils.htm>) használjuk.

7.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2014.03.28. 19:00	0,5 óra	Lipták Sebők Szepes Varga	Hatodik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása.
2014.03.29. 19:45	0,5 óra	Szepes	7.2 Elkészítése.
2014.03.30. 04:00	1,5 óra	Lipták	7.3 Elkészítése
2014.03.30. 09:00	0,5 óra	Szepes	7.2 Hibáinak javítása.
2014.03.30. 20:00	2,5 óra	Sebők	7.0 Specifikáció változás, 7.1.1 Az interfész általános leírása, térképájl formátum, 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása elkészítése
2014.03.30. 21:45	0,5 óra	Szepes	7.2 Hibáinak javítása
2014.03.30. 22:00	2 óra	Varga	7.1.2, 7.1.3 elkészítése.
2014.03.30. 23:00	0,5 óra	Lipták	7.3 javítása, kiegészítése.

8. Részletes tervek

8.1 Osztályok és metódusok tervei.

8.1.1 Akadály

- **Felelősség**

Egy cellára lehet helyezni, melye az áthaladó ellenségek így lassabban mozognak, vagyis időegységenként kevesebbet haladnak. Fel kell tudnia használni egy sárga követ.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámából áll, a tesztelést segíti.
- **-sargaKo: SargaKo:** Az akadályon elhelyezett sárgakő.

- **Metódusok**

- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztalkimenetek kiírásához használjuk.
- **+boolean lerakAkadalyKo(SargaKo sargaKo):** Ha a sargaKo attribútuma üres, elmenti a paraméterül kapott sárgakövet és visszatért igaz értékkel. egyébként csak visszatér hamissal.
- **+void ralep(Ellenseg ellenseg):** A paraméterül kapott ellenség sebességét állítja be egy az eredetinel kisebb értékre. Ha nem tartalmaz a torony sárgakövet, egyébként meghívja az ő ralep(Ellenseg ellenseg) függvényét.
- **+int getAr():** Visszaadja az akadály letételéhez szükséges varázserőt.

8.1.2 Cella

- **Felelősség**

Tudnia kell a helyét a térképen. Le kell tudnia kezelni a torony , akadály, és különböző varázskövek lerakását meghívását (a heterogén kollekció miatt).

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **#pozicio: Point:** Megmutatja, hogy a térképen hol helyezkedik el a cella.

- **Metódusok**

- **+Pont getPozicio():** Visszaadja a cella helyét a térképen
- **+void setPozicio(Point point):** Beállítja a cella helyét a paraméterül kapott értékre.
- **+double getTavolsag(Cella cella):** Meghívja a pozicio attribútumának azonos nevű függvényét, majd visszaadja az így kapott értéket.
- **+boolean lerakTorony(Torony torony):** Visszatér hamis értékkel, itt nincs hasznos működése, csak a leszármazottaiban, a heterogén kollekció miatt szükséges.
- **+void lerakAkadaly(Akadaly akadaly):** Itt nincs hasznos működése, csak a leszármazottaiban, a heterogén kollekció miatt szükséges.
- **+void lerakToronyKo (ToronyKo toronyKo):** Itt nincs hasznos működése, csak a leszármazottaiban, a heterogén kollekció miatt szükséges.
- **+void lerakAkadalyKo (SargaKo sargaKo):** Itt nincs hasznos működése, csak a leszármazottaiban, a heterogén kollekció miatt szükséges.

8.1.3 Domborzat

- **Felelősség**

Egy olyan cella, amelyen az ellenségek nem tartózkodhatnak. Tornyot lehet rajta elhelyezni. Tudnia kell a helyét a térképen.

- **Ősosztályok**

Cella

- **Interfészek**

-

- **Attribútumok**

2. **-torony: Torony:** A domborzaton elhelyezett torony.

- **Metódusok**

3. **+boolean lerakTorony(Torony torony):** Ha a torony attribútuma még üres, akkor eltárolja a paraméterül kapott tornyot, és beállítja a pozícióját a sajátjára, és visszatér igaz értékkel. Ha a torony attribútum egy létező toronyra mutat már, akkor visszatér hamis értékkel.
4. **+void lerakToronyKo (ToronyKo toronyKo):** Ha a torony attribútuma nem üres, meghívja annak az azonos nevű függvényét.

8.1.4 Ellenség

- **Felelősség**

A pályán mozgó ellenségeket valósítja meg. Az ellenségek célja az, hogy eljussanak a Végzet Hegyéhez. Csak az Ut-akon tudnak mozogni. Ez egy absztrakt osztály.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **#elet: int:** Az ellenség életereje. Ha ez 1 alá csökken, akkor az ellenség meghal.
- **#helyzet: double:** Cellán belüli helyzet, értéke 0 és 1 közötti.
- **#jatek: Jatek:** Az ellenséget tartalmazó játék.
- **#kovetkezoPozicio: Ut:** Az ellenség erre lép tovább.
- **#pozicio: Ut:** Az ellenség jelenlegi helyzete.
- **#sebesseg: double:** Egy kör alatt az ellenség ennyit mozdul a cellán belül.

- **Metódusok**

- **+void acceptToronyKoSebez(ToronyKo toronyKo):** A torony ezt hívja meg, ha rálő az ellenségre és van rajta toronykő. Ekkor az ellenség meghívja a *kő sebez()* függvényét, saját magát átadva neki. Erre a *kő* megsebz az ellenséget, attól függően, hogy az milyen fajta.
- **+Ellenseg(Jatek jatek):** Ellenség konstruktora, ezt használja a játék akkor, amikor új ellenségeket rak le a pályára.
- **+Ellenseg(Jatek jatek, Ellenseg ellenseg):** Ezt a konstruktort egy ellenség akkor hívja meg, amikor azt egy lövedék kettévágta. Az új ellenség ugyanazon a cellán jön létre, amin az eredeti van, csökkentett életerővel.
- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+Ut getPozicio():** Visszaadja az ellenség jelenlegi pozícióját.
- **+void initElet():** Ellenség kezdő életerejének beállítása. Ez minden ellenség leszármazottnál más.
- **+void leptet():** Az ellenség lép egyet a cellán belül. Ha elérte jelenlegi cellájának a határát, akkor rálép a következőre.
- **+void sebzodik(int sebz, boolean kettevag):** Az ellenség életerejéből levonódik *sebz*-nyi. Ha ezáltal az *elet* 1 alá csökken, akkor meghal. Ha a *kettevag* true, akkor az ellenség kettéosztódik. Az új ellenség ugyanazon a cellán jön létre, mint amin az eredeti van, csökkentett életerővel. False esetén ez nem történik meg.
- **+void setPozicio(Ut pozicio):** Ellenség pozíciójának beállítása.
- **+void setSebesseg(double sebesseg):** Ellenség sebességének beállítása.

8.1.5 Ember

- **Felelősség**

Az egyik ellenség fajta.

- **Ősosztályok**

Ellenseg

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+void acceptToronyKoSebez(ToronyKo toronyKo):** A torony ezt hívja meg, ha rálő az emberre és van rajta toronykő. Ekkor az ember meghívja a *kő sebez* függvényét, saját magát átadva neki. Erre a *kő* megsebz az embert.
- **+void initElet():** Ember kezdő életerejének beállítása 1000-re.
- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebzodik(int sebzes, boolean kettevag):** Az ember életerejéből levonódik *sebzes*-nyi. Ha ezáltal az *elet* 1 alá csökken, akkor meghal. Ha a *kettevag* true, akkor az ember kettéosztódik. Az új ember ugyanazon a cellán jön létre, mint amin az eredeti van, csökkentett életerővel. False esetén ez nem történik meg.

8.1.6 EmberPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak emberek ellen hatásos.

- **Ősosztályok**

ToronyKo

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebez(Ember ember):** A paraméterként kapott embert sebz megadott (350) értékkel. Meghívja a kapott ember *acceptToronyKoSebez* függvényét. Paraméterként átadja önmagát.

8.1.7 Hobbit

- **Felelősség**

Az egyik ellenség fajta.

- **Ősosztályok**

Ellenség

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+void acceptToronyKoSebez(ToronyKo toronyKo):** A torony ezt hívja meg, ha rálő a hobbitra és van rajta toronykő. Ekkor a hobbit meghívja a kő *sebez* függvényét, saját magát átadva neki. Erre a kő megsebzí a hobbitot.
- **+void initElet():** Hobbit kezdő életerejének beállítása 1100-ra.
- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+sebzodik(int sebzes, boolean kettevag):** A hobbit életerejéből levonódik *sebzes*-nyi. Ha ezáltal az *elet* 1 alá csökken, akkor meghal. Ha a *kettevag* true, akkor a hobbit kettéosztódik. Az új hobbit ugyanazon a cellán jön létre, mint amin az eredeti van, csökkentett életerejével. False esetén ez nem történik meg.

8.1.8 HobbitPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak hobbitok ellen hatásos.

- **Ősosztályok**

ToronyKo

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebez(Hobbit hobbit):** A paraméterként kapott hobbitot sebz megadott (350) értékkel. Meghívja a kapott hobbit acceptToronyKoSebez függvényét. Paraméterként átadja önmagát.

8.1.9 Jatek

- **Felelősség**

Ez az osztály reprezentálja a játékot. Ebből csak egy objektum lesz. Elindítja a játékot, lépteti az ellenségeket és tornyokat, lerak köveket, tornyokat, akadályokat, létrehoz új ellenségeket, az elpusztultakat pedig eltakarítja. Számon tartja, hogy mennyi varázserőnk van.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **-cellak: Cella[][]:** A pálya cellái.
- **-ellensegek: List<Ellenseg>:** A pályán mozgó ellenségek.
- **-kezdőPoziciok: List<Ut>:** Az új ellenségeket a játék ezeken az utakon hozza létre.
- **-maradekEllenseg: int:** Ennyi ellenséget kell még lerakni. Ha ez nullára csökkent, akkor a játék nem hoz létre újabb ellenségeket.
- **-tornyok: List<Torony>:** A pályán lévő tornyok.
- **-varazsero: int:** Ennyi varázserőnk van, kezdő értéke 1000. Ebből lehet tornyokat, akadályokat, köveket venni, ha legalább annyi varázserőnk van, mint az áruk. Ellenségek megöléséért kapjuk. Értéke nem csökkenhet nulla alá.

- **Metódusok**

- **+void addEllenseg(Ellenseg ellenseg):** Ha egy ellenséget a torony kettévágott, akkor az meghívja ezt a függvényt, hogy a "klónját" a pályához adja.
- **+List<Ellenseg> getEllensegek():** Visszaadja az ellenségeket.
- **+void indit():** Betölti fájlból a pályát és elindítja a játékot.
- **+void leptet(int ido):** Egy körrel lépteti a játékot. Az ellenségek lépnek elsőnek, majd utánuk a tornyok. Ha a *maradekEllenseg* nagyobb, mint nulla, akkor a játék létrehoz egy új ellenséget. A kód periodikus leeresztését is végzi.
- **+void lerakAkadaly(Cella cella, Akadaly akadaly):** Megpróbál lerakni egy akadályt a megadott cellára.
- **+void lerakAkadalyKo(Cella cella, SargaKo sargaKo):** Megpróbál lerakni egy sárgakövet az adott cellára.
- **+void lerakTorony(Cella cella, Torony torony):** Megpróbál lerakni egy tornyot az adott cellára.

- **+void lerakToronyKo(Cella cella, ToronyKo toronyKo):** Megpróbál lerakni egy toronykövet az adott cellára.
- **+void meghalEllenseg(Ellenseg ellenseg, int jutalom):** Eltakarítja azt az ellenséget, amelyiknek az élete 1 alá csökkent, majd az érte járó jutalmat hozzáadja a *varazsero*-höz.
- **+void vege(boolean nyertunk):** A játék véget ért. A *nyertunk* akkor true, ha az összes ellenséget megöltük, ekkor a játékot megnyertük. Ha egy ellenség eljutott a Végzet Hegyéhez, akkor a *nyertunk* false, ebben az esetben veszítettünk.

8.1.10 KekKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony hatótávolságát növeli. Ezáltal olyan távolabbi ellenségeket is tud sebezni a torony, melyet a varázskő nélkül nem tudna.

- **Ősosztályok**

ToronyKo

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+String getObjectumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void setHatotavolsag(Torony torony):** A paraméterként kapott torony hatótávolságát növeli. Meghívja a Torony setHatotavolsag metódusát, és paraméterként átad 3,025-t.

8.1.11 Pont

- **Felelősség**

El kell tárolnia egy 2 dimenziós koordinátát, melyek a távolságát le kell tudnia mérni egy másik ugyanilyenhez képest.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **-x: int** A koordináta függőleges irányú összetevője.
- **-y: int** A koordináta vízszintes irányú összetevője.

- **Metódusok**

- **+Pont(int x, int y):** Konstruktor, amely azonnal beállítja a paraméterben kapott értékre a pontot.
- **+double getTavolsag(Pont pont):** Leméri a paraméterül kapott ponttól a távolságát, és visszaadja a kapott értéket.

8.1.12 SargaKo

- **Felelősség**

Tudnia kell, hogy mennyivel tudja növelni egy akadály lassítását.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+void ralep(Ellenseg ellenseg):** A paraméterül kapott ellenségnek meghívja a setSebesseg(sebesseg:int) függvényét, így annak a sebességét egy az eredetitől kisebb értékre állítja be.
- **+int getAr():** Visszaadja a Sárgakő letételéhez szükséges varázserőt.
- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.

8.1.13 Torony

- **Felelősség**

A domborzatra építhető játékelem, ami sebzi azokat az ellenségeket, akik az ő hatótávolságába beleeső utakon haladnak. A tornyokat a játékos helyezi el a pályán. Varázskövel lehet erősíteni őket, ezek hatótávolságot, a sebzés értékét vagy a sebzés gyakoriságát befolyásoló kövek.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

5. **-hatotavolsag: int:** Meghatározza a torony hatótávolságát.
6. **-jatek: Jatek:** Referenciát tárol a statikus játék objektumra.
 - **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
 - **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.
7. **-pozicio: Domborzat:** Meghatározza, hogy melyik domborzaton helyezkedik el a torony.
8. **-toronyKo: ToronyKo:** A tornyon elhelyezett varázskő. Ha nincs rajta, akkor az értéke null.
9. **-varakozas int:** Két sebzés között eltelt várakozás értéke.

- **Metódusok**

- **+int getAr():** Visszaadja az építéshez szükséges varázserőt, 400-at.
- **+void lerakToronyKo(ToronyKo toronyKo):** Lerakja a paraméterként kapott toronykövet a toronyra, ha azon még nincs toronykő.
- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebez(Ellenseg ellenseg):** Megsebz a paraméterként kapott ellenséget. Meghívja annak a sebzodik metódusát, és átad 250-et paraméterként.
- **+void setHatotavolsag(int hatotavolsag):** Beállítja a hatótávolság értékét a paraméterként kapott értékre.
- **+void setPozicio(Domborzat domborzat):** Beállítja a torony pozícióját a paraméterként kapott értékre.
- **+void setVarakozas(int varakozas):** Beállítja a várakozás értékét a paraméterként kapott értékre.
- **+void Torony(Jatek jatek):** Létrehozza a torony objektumot, és beállítja az osztály jatek attribútumát a paraméterként kapott jatek-ra.
- **+void tuzel():** Tüzelés a toronyból az ellenségekre. Ha a várakozás már 0, akkor lekéri a játéktól az ellenségek listáját. Azon végigiterálva megkeresi az ellenseg.getPozicio és az ut.getTavolsag metódusok segítségével a hozzá legközelebb lévő ellenséget. Ha nincs toronykő a toronyon, akkor meghívja a saját sebez metódusát, paraméterként pedig átadja a hozzá legközelebb lévő ellenséget. Ha van a tornyon toronykő, akkor meghívja az ellenseg acceptToronyKoSebzes függvényét, és paraméterként átadja a toronykövet.

8.1.14 ToronyKo

- **Felelősség**

Toronyra elhelyezhető varázskő. Ezek segítségével erősítheti a játékos a tornyokat. Három fajta varázskő létezik: az egyikkel megadott ellenséget a torony erősebben sebz, a másikkal a

sebzés gyakoriságát növeljük, a harmadikkal pedig a hatótávolságot. Egy tornyon egyszerre csak egy varázskő helyezhető el. Absztrakt osztály, helyette zöldkővet, kékkővet illetve a piros kő négy változatát fogjuk használni.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **+int getAr():** Visszaadja az építéshez szükséges varázserő mennyiségét.
- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebez(Ember ember):** Megsebz a paraméterként kapott embert. Meghívja annak a sebzodik metódusát 250 sebzéssel. A piroskövek felüldefiniálják.
- **+void sebez(Hobbit hobbit):** Megsebz a paraméterként kapott hobbitot. Meghívja annak a sebzodik metódusát 250 sebzéssel. A piroskövek felüldefiniálják.
- **+void sebez(Torp torp):** Megsebz a paraméterként kapott törpöt. Meghívja annak a sebzodik metódusát 250 sebzéssel. A piroskövek felüldefiniálják.
- **+void sebez(Tunde tunde):** Megsebz a paraméterként kapott tündét. Meghívja annak a sebzodik metódusát 250 sebzéssel. A piroskövek felüldefiniálják.
- **+void setHatotavolsag(Torony torony):** Megadja a toronynak a hatótávolságát. Meghívja annak a setHatotavolsag metódusát 2,75-ös hatótávolsággal. A kékkő felüldefiniálja.
- **+void setVarakozas(Torony torony):** Megadja a toronynak a sebzési gyakoriságát. Meghívja annak a setVarakozas metódusát 2 várakozási értékkel. A zöldkő felüldefiniálja.

8.1.15 Torp

- **Felelősség**

Az egyik ellenség fajta.

- **Ősosztályok**

Ellenseg

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+void acceptToronyKoSebez(ToronyKo toronyKo):** A torony ezt hívja meg, ha rálő a törpre és van rajta toronykő. Ekkor a törp meghívja a *kő sebez* függvényét, saját magát átadva neki. Erre a *kő* megsebzí az törpöt.
- **+void initElet():** Törp kezdő életerejének beállítása 1200-ra.
- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebzodik(int sebz, boolean kettevag):** A törp életerejéből levonódik *sebz*-nyi. Ha ezáltal az *elet* 1 alá csökken, akkor meghal. Ha a *kettevag* true, akkor a törp kettéosztódik. Az új törp ugyanazon a cellán jön létre, mint amin az eredeti van, csökkentett életterővel. False esetén ez nem történik meg.

8.1.16 TorpPirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak törpök ellen hatásos.

- **Ősosztályok**

ToronyKo

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebez(Torp torp):** A paraméterként kapott ellenséget sebzí megadott (350) értékkel. Meghívja a kapott törp *acceptToronyKoSebez* függvényét. Paraméterként átadja önmagát.

8.1.17 Tunde

- **Felelősség**

Az egyik ellenség fajta.

- **Ősosztályok**

Ellenseg

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámából áll, a tesztelést segíti.

- **Metódusok**

- **+void acceptToronyKoSebez(ToronyKo toronyKo):** A torony ezt hívja meg, ha rálő a tündére és van rajta toronykő. Ekkor a tünde meghívja a *kő sebez* függvényét, saját magát átadva neki. Erre a *kő* megsebzí a tündét.
- **+void initElet():** Tünde kezdő életerejének beállítása 1300-ra.
- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebzodik(int sebz, boolean kettevag):** A tünde életerejéből levonódik *sebz*-nyi. Ha ezáltal az *élet* 1 alá csökken, akkor meghal. Ha a *kettevag* true, akkor a tünde kettéosztódik. Az új tünde ugyanazon a cellán jön létre, mint amin az eredeti van, csökkentett életerővel. False esetén ez nem történik meg.

8.1.18 TundePirosKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzését növeli. Csak tündék ellen hatásos.

- **Ősosztályok**

ToronyKo

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámából áll, a tesztelést segíti.

- **Metódusok**

- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void sebez(Tunde tunde):** A paraméterként kapott ellenséget sebzí megadott (350) értékkel. Meghívja a kapott tünde *acceptToronyKoSebez* függvényét. Paraméterként átadja önmagát.

8.1.19 Ut

- **Felelősség**

Egy olyan cella, amelyre az ellenségeknek rá kell tudnia lépni. Tudnia kell a helyét a térképen, akadály helyezhető el rajta.

- **Ősosztályok**

Cella

- **Interfészek**

-

- **Attribútumok**

- **-akadaly: Akadaly:** Az úton elhelyezhető akadály.
- **-kovetkezoLepesek: List<Ut>:** Azokat az utakat jelöli, amelyeken az ellenség továbbhaladhat.
- **-ellensegek: List<Ellenseg>:** Az úton éppen áthaladó ellenségek tömbje.

- **Metódusok**

10. **+void lerakAkadaly(Akadaly akadaly):** Amennyiben még üres az akadaly attribútuma, beállítja a paraméterül kapott akadályt annak, és meghívja a `ralep(e:Ellenseg)` metódusát minden az ellensegek tömbben található ellenségre.
11. **+void lerakAkadalyKo(SargaKo sargaKo):** Amennyiben az akadaly attribútuma nem üres, meghívja annak az azonos nevű függvényét.
12. **+void setKovetkezoLepesek(List<Ut> utak):** Beállíthatjuk vele, hogy mely utakra lépjenek át az ellenségek.
13. **+Ut getKovetkezoLepes():** Visszaadja a következő utat amire az ellenségnek át kell lépnie.
14. **+void ralep(Ellenseg ellenseg):** Ha az akadaly attribútuma nem üres, meghívja az azonos nevű függvényét, egyébként meghívja az ellenség `setSebesseg(sebesseg:double)` metódusát, beállítva ezzel a sebességét.
15. **+void lelep(Ellenseg ellenseg):** Kitörli az ellensegek tömbjéből az a paraméterül kapott ellenséget.

8.1.20 VegzetHegye

- **Felelősség**

Egy olyan cella, amelyre rá kell tudnia lépni az ellenségeknek, tudnia kell a helyét a térképen.

- **Ősosztályok**

Cella → Ut

- **Interfészek**

-

- **Attribútumok**

- **-jatek: Jatek:** Az objektumot tartalmazó jatek.

- **Metódusok**

- **+VegzetHegye(Jatek jatek):** A konstruktora, melyben azonnal beállítjuk az őt tartalmazó jatek objektumot

- **+void lerakAkadaly(Akadaly akadaly):** Amennyiben még üres az akadaly attribútuma, beállítja a paraméterül kapott akadályt annak, és meghívja a `ralep(e:Ellenseg)` metódusát minden az ellensegek tömbben található ellenségre.
- **+void lerakAkadalyKo(SargaKo sargaKo):** Amennyiben az akadaly attribútuma nem üres, meghívja annak az azonos nevű függvényét.
- **+void ralep(Ellenseg ellenseg):** Ha az akadaly attribútuma nem üres, meghívja az azonos nevű függvényét, egyébként meghívja az `ellenseg.setSebesseg(sebesseg:double)` metódusát, beállítva ezzel az ellenség sebességét.

8.1.21 ZoldKo

- **Felelősség**

Egy toronyra helyezhető varázskő, mely a torony sebzési gyakoriságát növeli. Ilyenkor a két sebzés között eltelt idő csökken.

- **Ősosztályok**

ToronyKo

- **Interfészek**

-

- **Attribútumok**

- **-objektumDarabszam: int:** Az adott osztály példányainak száma, a tesztelést segíti.
- **-objektumAzonosito: String:** Az objektum azonosítója, mely a típusából és egy sorszámból áll, a tesztelést segíti.

- **Metódusok**

- **+String getObjektumAzonosito():** Visszaadja az objektum azonosítóját, kizárólag a tesztkimenetek kiírásához használjuk.
- **+void setVarakozas(Torony torony):** A paraméterként kapott torony hatótávolságát módosítja. Meghívja a Torony `setVarakozas` metódusát, és paraméterként átad 1,5-t.

8.2 A tesztek részletes tervei, leírásuk a teszt nyelven

8.2.1 Pálya betöltése

- **Leírás**
A pályát betöltjük adott fájlból
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy sikeres-e a pálya betöltése. A betöltött pályának meg kell egyeznie a fájl által leírttal.
- **Bemenet**
`indit palya.map.txt`
`quit`
- **Térképájl (palya.map.txt)**
`6, 2`
`UD`
`UD`
`UD`


```

UD
UD
VD
1
1,1
4
1,1;2,1
2,1;3,1
3,1;4,1
4,1;5,1
5,1;6,1

```

- **Elvárt kimenet**

palya.map.txt betöltése sikerült, játék indítása.

8.2.2 Akadály lassítása

- **Leírás**

Megpróbálunk akadályt letenni útra is, domborzatra is, megmérjük a rajta áthaladó, és egy másik úton áthaladó ellenség sebességét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, csak az utakra lehet akadályokat letenni, melyek csak a rajtuk áthaladó ellenségeket lassítják.

- **Bemenet**

```

indit palya.map.txt
tick 1
addAkadaly 1 2
addAkadaly 2 1
tick 3
quit

```

- **Térkép fájl (palya.map.txt)**

Lásd 8.2.1 Pálya betöltése.

- **Elvárt kimenet**

```

palya.map.txt betöltése sikerült, játék indítása.
1000 varazseronk van
játék létrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Az (1, 2) koordinataju cellara nem lehet akadályt rakni.
Sikerült lerakni az akadályt az (2, 1) koordinataju
cellara.
700 varazseronk van
Ember1 ralepett az (2, 1) cellara
Akadaly1 lelassította Ember1-et 1.5-re
700 varazseronk van
700 varazseronk van
Ember1 ralepett az (3, 1) cellara

```

8.2.3 Torony lövése

- **Leírás**

Megpróbálunk tornyot letenni útra is, domborzatra is, ellenségeket teszünk a hatótávolságán belülre, különböző távolságra, illetve a hatótávolságán kívülre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a tornyokat csak a domborzatokra lehet letenni, azok a hozzájuk legközelebbi ellenséget sebzik, de csak akkor, ha hatótávolságán belül van.

- **Bemenet**

```
indit torony.map.txt
random kettevagas off
tick 1
addTorony 1 1
addTorony 4 6
tick 6
quit
```

- **Térkép fájl (torony.map.txt)**

```
4,7
UUUUUUUU
DDDDDUU
DDDDDUU
DDDDDDV
2
1,1
1,7
11
1,1;1,2
1,2;1,3
1,3;1,4
1,4;1,5
1,5;1,6
1,6;2,6
1,7;2,7
2,6;3,6
2,7;3,7
3,6;3,7
3,7;4,7
```

- **Elvárt kimenet**

```
torony.map.txt betoltese sikerult, jatek inditasa.
kettevagas veletlenszerusegenek atallitasa.
1000 varazseronk van
jatek letrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Az (1, 1) koordinataju cellara nem lehet tornyot rakni
Sikerult lerakni a tornyot az (4, 6) koordinataju
cellara.
600 varazseronk van
Ember1 ralepett az (1, 2) cellara
600 varazseronk van
Ember1 ralepett az (1, 3) cellara
600 varazseronk van
Ember1 ralepett az (1, 4) cellara
600 varazseronk van
Ember1 ralepett az (1, 5) cellara
600 varazseronk van
Ember1 ralepett az (1, 6) cellara
```

```
jatek létrehozta Hobbit1-et
Hobbit1 ralepett az (1, 7) cellara
600 varazseronk van
Ember1 ralepett az (2, 6) cellara
Hobbit1 ralepett az (2, 7) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
```

8.2.4 Sárgakő hatása

- **Leírás**
Sárgaköveket próbálunk lerakni akadályokra, tornyokra, és üres cellákra, mérjük, az akadályon áthaladó ellenség sebességét, amire rátettük.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy csak az akadályokra helyezhetőek-e a sárgakövek, és növelik-e annak lassítását.
- **Bemenet**
indit palya.map.txt
tick 1
addAkadaly 2 1
addTorony 6 2
addSargaKo 1 1
addSargaKo 6 2
addSargaKo 2 1
tick 4
quit
- **Térkép fájl (palya.map.txt)**
Lásd 8.2.1 Pálya betöltése.
- **Elvárt kimenet**
palya.map.txt betoltese sikerult, jatek inditasa.
1000 varazseronk van
jatek létrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Sikerult lerakni az akadalyt az (2, 1) koordinataju cellara.
Sikerult lerakni a tornyot az (6, 2) koordinataju cellara.
Az (1, 1) koordinataju cellara nem lehet sargakövet rakni.
Az (6, 2) koordinataju cellara nem lehet sargakövet rakni.
Sikerult lerakni a sargakövet az (2, 1) koordinataju cellan levo akadalyra.
200 varazseronk van
Ember1 ralepett az (2, 1) cellara
SargaKo1 lelassította Ember1-et 2.1-re
200 varazseronk van
200 varazseronk van
200 varazseronk van
Ember1 ralepett az (3, 1) cellara

8.2.5 Piroskő hatása

- **Leírás**
Pirosköveket próbálunk lerakni akadályokra, tornyokra, és üres cellákra, mérjük, hogy mennyivel nő meg a torony sebzése, melyik fajra.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy csak toronyra helyezhető-e piros kő, és a sebzése növekedik-e. Ellenőrizzük, hogy minden fajta piros kő faj specifikus, más fajra hatástalan kell, hogy legyen.
- **Bemenet**

```
indit pirosko.map.txt
random kettevagas off
tick 1
addAkadaly 3 6
addTorony 1 7
addHobbitPirosKo 1 1
addHobbitPirosKo 3 6
addHobbitPirosKo 1 7
tick 6
quit
```
- **Térképájl (pirosko.map.txt)**

```
4,7
UUUUUUD
DDDDUUD
DDDDUUD
DDDDDVD
2
1,1
1,6
10
1,1;1,2
1,2;1,3
1,3;1,4
1,4;1,5
1,5;2,5
1,6;2,6
2,5;3,5
2,6;3,6
3,5;3,6
3,6;4,6
```
- **Elvárt kimenet**

```
pirosko.map.txt betoltese sikerult, jatek inditasa.
kettevagas veletlenszerusegenek atallitasa.
1000 varazseronk van
jatek letrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Sikerult lerakni az akadalyt az (3, 6) koordinataju
cellara.
Sikerult lerakni a tornyot az (1, 7) koordinataju
cellara.
Az (1, 1) koordinataju cellara nem lehet hobbit
```

```

piroskovet rakni.
Az (3, 6) koordinataju cellara nem lehet hobbit
piroskovet rakni.
Sikerult lerakni a hobbit piroskovet az (1, 7)
koordinataju cellan levo toronyra.
200 varazseronk van
Ember1 ralepett az (1, 2) cellara
200 varazseronk van
Ember1 ralepett az (1, 3) cellara
200 varazseronk van
Ember1 ralepett az (1, 4) cellara
200 varazseronk van
Ember1 ralepett az (1, 5) cellara
HobbitPirosKol megsebezte Ember1-et 250 sebzessel
200 varazseronk van
Ember1 ralepett az (2, 5) cellara
jatek letrehozta Hobbit1-et
Hobbit1 ralepett az (1, 6) cellara
200 varazseronk van
Ember1 ralepett az (3, 5) cellara
Hobbit1 ralepett az (2, 6) cellara
HobbitPirosKol megsebezte Hobbit1-et 350 sebzessel

```

8.2.6 Kékkő hatása

- **Leírás**
Kékköveket próbálunk lerakni akadályokra, tornyokra, és üres cellákra, ellenőrizzük, hogy képes-e az eredeti hatótávolságon éppen kívül eső ellenséget sebezni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy csak toronyra helyezhető kékkő, a torony hatótávolsága pedig megnő tőle.
- **Bemenet**

```

indit kekko.map.txt
random kettevagas off
tick 1
addAkadaly 3 1
addTorony 2 4
addKekKo 1 1
addKekKo 3 1
addKekKo 2 4
tick 1
quit

```
- **Térképájl (kekko.map.txt)**

```

4,4
UDDD
UDDD
UDDD
VDDD
1
1,1
3

```

```
1,1;2,1
2,1;3,1
3,1;4,1
```

- **Elvárt kimenet**

```
kekko.map.txt betoltese sikert, jatek inditasa.
kettevagag veletlenszerusegenek atallitasa.
1000 varazseronk van
jatek letrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Sikertult lerakni az akadalyt az (3, 1) koordinataju
cellara.
Sikertult lerakni a toronyot az (2, 4) koordinataju
cellara.
Az (1, 1) koordinataju cellara nem lehet hobbit
piroskovet rakni.
Az (3, 1) koordinataju cellara nem lehet hobbit
piroskovet rakni.
Sikertult lerakni a kekkovet az (2, 4) koordinataju cellan
levo toronyra.
200 varazseronk van
Ember1 ralepett az (2, 1) cellara
KekKo1 megsebezte Ember1-et 250 sebzessel
```

8.2.7 Zöldkő hatása

- **Leírás**

Zöldköveket próbálunk lerakni akadályokra, tornyokra, és üres cellákra, mérjük, a torony sebzésének gyakoriságát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy csak toronyra lehet-e zöldkővet tenni, a letett kő hatására pedig tényleg gyakrabban lő-e a torony.

- **Bemenet**

```
indit zoldko.map.txt
random kettevagag off
tick 1
addAkadaly 8 1
addTorony 4 2
addZoldKo 1 1
addZoldKo 8 1
addZoldKo 4 2
tick 4
quit
```

- **Térképájl (zoldko.map.txt)**

```
9,2
UD
UD
UD
UD
UD
UD
UD
UD
```

```

UD
VD
1
1,1
8
1,1;2,1
2,1;3,1
3,1;4,1
4,1;5,1
5,1;6,1
6,1;7,1
7,1;8,1
8,1;9,1

```

- **Elvárt kimenet**

```

zoldko.map.txt betoltese sikerult, jatek inditasa.
1000 varazseronk van
jatek létrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Sikerult lerakni az akadalyt az (8, 1) koordinataju
cellara.
Sikerult lerakni a tornyot az (4, 2) koordinataju
cellara.
Az (1, 1) koordinataju cellara nem lehet zoldkovet rakni.
Az (8, 1) koordinataju cellara nem lehet zoldkovet rakni.
Sikerult lerakni a zoldkovet az (4, 2) koordinataju
cellan levo toronyra.
200 varazseronk van
Ember1 ralepett az (2, 1) cellara
ZoldKo1 megsebezte Ember1-et 250 sebzessel
200 varazseronk van
Ember1 ralepett az (3, 1) cellara
200 varazseronk van
ZoldKo1 megsebezte Ember1-et 250 sebzessel
Ember1 ralepett az (4, 1) cellara
200 varazseronk van
Ember1 ralepett az (5, 1) cellara
ZoldKo1 megsebezte Ember1-et 250 sebzessel

```

8.2.8 Ellenség osztódik

- **Leírás**
Az ellenségeknek képesnek kell osztódnia, ha olyan sebzés éri.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy akkor és csak akkor osztódik-e az ellenség, amikor ilyen lövedék találja el, az osztódástól pedig pontosan 1 új ellenség keletkezik-e azonos fajjal.
- **Bemenet**
indit zoldko.map.txt
random kettevagass off
tick 1
addTorony 4 2
tick 1

```
random kettevagas on
tick 2
quit
```

- **Térkép fájl (zoldko.map.txt)**

Lásd 8.2.7 Zöldkő hatása.

- **Elvárt kimenet**

```
zoldko.map.txt betoltese sikerult, jatek inditasa.
kettevagas veletlenszerusegenek atallitasa.
1000 varazseronk van
jatek letrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Sikerult lerakni a tornyot az (4, 2) koordinataju
cellara.
600 varazseronk van
Ember1 ralepett az (2, 1) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
kettevagas veletlenszerusegenek atallitasa.
600 varazseronk van
Ember1 ralepett az (3, 1) cellara
600 varazseronk van
Ember1 ralepett az (4, 1) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
Ember1 kettevagva
```

8.2.9 Ellenség meghal

- **Leírás**

Ellenséget sebzünk több alkalommal, amíg el nem fogy az összes életpontja.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy akkor, és csak akkor hal meg, ha legfeljebb annyi élete van, mint a sebzés mértéke. A halott ellenségtől az várjuk, hogy eltűnik a pályáról.

- **Bemenet**

```
indit meghal.map.txt
random kettevagas off
tick 1
addTorony 4 2
addTorony 6 2
tick 6
quit
```

- **Térkép fájl (meghal.map.txt)**

```
7,3
UDD
UDD
UUU
DDU
UUU
UDD
UUV
1
1,1
12
```



```

1,1;2,1
2,1;3,1
3,1;3,2
3,2;3,3
3,3;4,3
4,3;5,3
5,1;6,1
5,2;5,1
5,3;5,2
6,1;7,1
7,1;7,2
7,2;7,3

```

- **Elvárt kimenet**

```

meghal.map.txt betöltése sikerült, játék indítása.
kettevagas veletlenszerűségének atallítása.
1000 varazseronk van
játék létrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Sikerült lerakni a tornyot az (4, 2) koordinataju
cellara.
Sikerült lerakni a tornyot az (6, 2) koordinataju
cellara.
200 varazseronk van
Ember1 ralepett az (2, 1) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
200 varazseronk van
Ember1 ralepett az (3, 1) cellara
200 varazseronk van
Ember1 ralepett az (3, 2) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
200 varazseronk van
Ember1 ralepett az (3, 3) cellara
200 varazseronk van
Ember1 ralepett az (4, 3) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
Torony2 megsebezte Ember1-et 250 sebzessel
Ember1 meghalt
játék létrehozta Ember2-et
Ember2 ralepett az (1, 1) cellara

```

8.2.10 A toronyra kód ereszkedik

- **Leírás**

A toronyra kód ereszkedik, melytől a hatótávolság lecsökken, így az eredetileg hatótávolságon belüli ellenségeket nem tudja többé sebezni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizni, hogy csak toronyra ereszkedhet kód, a hatótávolság pedig lecsökken tőle.

- **Bemenet**

```

indit kod.map.txt
random kettevagas off
tick 1

```

```
addTorony 6 2
tick 13
quit
```

- **Térkép fájl (kod.map.txt)**

```
7,5
UUUUU
UUUUU
UDDDD
UDDDD
UDDDD
UDDDD
UDDDD
VDDDD
1
1,1
14
1,1;1,2
1,2;1,3
1,3;1,4
1,4;1,5
1,5;2,5
2,2;2,1
2,3;2,2
2,4;2,3
2,5;2,4
2,1;3,1
3,1;4,1
4,1;5,1
5,1;6,1
6,1;7,1
```

- **Elvárt kimenet**

```
kod.map.txt betöltése sikerült, játék indítása.
kettevagás veletlenszerűségnek atallítása.
1000 varazseronk van
játék létrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Sikerült lerakni a tornyot az (4, 2) koordinataju
cellara.
600 varazseronk van
Ember1 ralepett az (1, 2) cellara
600 varazseronk van
Ember1 ralepett az (1, 3) cellara
600 varazseronk van
Ember1 ralepett az (1, 4) cellara
600 varazseronk van
Ember1 ralepett az (1, 5) cellara
600 varazseronk van
Ember1 ralepett az (2, 5) cellara
játék létrehozta Ember2-et
Ember2 ralepett az (1, 1) cellara
600 varazseronk van
Ember1 ralepett az (2, 4) cellara
```

```

Ember2 ralepett az (1, 2) cellara
jatek letrehozta Ember3-et
Ember3 ralepett az (1, 1) cellara
600 varazseronk van
Ember1 ralepett az (2, 3) cellara
Ember2 ralepett az (1, 3) cellara
Ember3 ralepett az (1, 2) cellara
600 varazseronk van
Ember1 ralepett az (2, 2) cellara
Ember2 ralepett az (1, 4) cellara
Ember3 ralepett az (1, 3) cellara
600 varazseronk van
Ember1 ralepett az (2, 1) cellara
Ember2 ralepett az (1, 5) cellara
Ember3 ralepett az (1, 4) cellara
a tornyokra kod szallt
600 varazseronk van
Ember1 ralepett az (3, 1) cellara
Ember2 ralepett az (2, 5) cellara
Ember3 ralepett az (1, 5) cellara
600 varazseronk van
Ember1 ralepett az (4, 1) cellara
Ember2 ralepett az (2, 4) cellara
Ember3 ralepett az (2, 5) cellara
600 varazseronk van
Ember1 ralepett az (5, 1) cellara
Ember2 ralepett az (2, 3) cellara
Ember3 ralepett az (2, 4) cellara
600 varazseronk van
Ember1 ralepett az (6, 1) cellara
Ember2 ralepett az (2, 2) cellara
Ember3 ralepett az (2, 3) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
Torony1-rol a kod felszallt

```

8.2.11 Elágazások

- **Leírás**
Az elágazáshoz vezetünk ellenségeket, melyeket más-más irányba terelünk.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy különböző irányokba el tudott indulni az arra küldött ellenség.
- **Bemenet**
indit elagazas.map.txt
random utvalasztas off
tick 2
random utvalasztas on
tick 1
quit
- **Térkép fájl (elagazas.map.txt)**
4,2
UU

```

UU
UU
VD
1
1,1
6
1,1;1,2;2,1
1,2;2,2
2,1;2,2;3,1
2,2;3,2
3,1;4,1
3,2;3,1

```

- **Elvárt kimenet**

```

elagazas.map.txt betoltese sikerult, jatek inditasa.
utvalasztas veletlenszerusegenek atallitasa.
1000 varazseronk van
jatek létrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
1000 varazseronk van
Ember1 ralepett az (2, 1) cellara
utvalasztas veletlenszerusegenek atallitasa.
1000 varazseronk van
Ember1 ralepett az (2, 2) cellara

```

8.2.12 Játék elvesztése

- **Leírás**

Hagyjuk, hogy beérjen egy ellenség a végzet hegyére.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy elvesztjük-e a játékot, ha egy ellenség beér a végzet hegyére.

- **Bemenet**

```

indit kekko.map.txt
tick 4
quit

```

- **Térkép fájl (kekko.map.txt)**

Lásd 8.2.6 Kékkő hatása.

- **Elvárt kimenet**

```

kekko.map.txt betoltese sikerult, jatek inditasa.
1000 varazseronk van
jatek létrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
1000 varazseronk van
Ember1 ralepett az (2, 1) cellara
1000 varazseronk van
Ember1 ralepett az (3, 1) cellara
1000 varazseronk van
Ember1 ralepett az (4, 1) cellara
Ember1 ralepett a Végzet Hegyére, vesztettünk

```

8.2.13 Játék megnyerése

- **Leírás**
Úgy alakítjuk a körülményeket, hogy minden ellenséget el tudjanak pusztítani a tornyok.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy amikor már az összes ellenség elfogyott, jelzi-e a program azt, hogy nyertünk.
- **Bemenet**
indit meghal.map.txt
random kettevagass off
tick 1
addTorony 4 2
addTorony 6 2
tick 17
quit
- **Térkép fájl (meghal.map.txt)**
Lásd 8.2.9 Ellenség meghal.
- **Elvárt kimenet**
meghal.map.txt betöltése sikerült, játék indítása.
kettevagass véletlenszerűségének atallítása.
1000 varazseronk van
játék létrehozta Ember1-et
Ember1 ralepett az (1, 1) cellara
Sikerült lerakni a tornyot az (4, 2) koordinataju cellara.
Sikerült lerakni a tornyot az (6, 2) koordinataju cellara.
200 varazseronk van
Ember1 ralepett az (2, 1) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
200 varazseronk van
Ember1 ralepett az (3, 1) cellara
200 varazseronk van
Ember1 ralepett az (3, 2) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
200 varazseronk van
Ember1 ralepett az (3, 3) cellara
200 varazseronk van
Ember1 ralepett az (4, 3) cellara
Torony1 megsebezte Ember1-et 250 sebzessel
Torony2 megsebezte Ember1-et 250 sebzessel
Ember1 meghalt
játék létrehozta Ember2-et
Ember2 ralepett az (1, 1) cellara
300 varazseronk van
Ember2 ralepett az (2, 1) cellara
Torony1 megsebezte Ember2-et 250 sebzessel
játék létrehozta Ember3-et
Ember3 ralepett az (1, 1) cellara
300 varazseronk van

```

Ember2 ralepett az (3, 1) cellara
Ember3 ralepett az (2, 1) cellara
300 varazseronk van
Ember2 ralepett az (3, 2) cellara
Ember3 ralepett az (3, 1) cellara
Torony1 megsebezte Ember2-et 250 sebzessel
300 varazseronk van
Ember2 ralepett az (3, 3) cellara
Ember3 ralepett az (3, 2) cellara
a tornyokra kod szállt
300 varazseronk van
Ember2 ralepett az (4, 3) cellara
Ember3 ralepett az (3, 3) cellara
Torony1 megsebezte Ember2-et 250 sebzessel
Torony1-rol a kod felszállt
300 varazseronk van
Ember2 ralepett az (5, 3) cellara
Ember3 ralepett az (4, 3) cellara
300 varazseronk van
Ember2 ralepett az (5, 2) cellara
Ember3 ralepett az (5, 3) cellara
Torony1 megsebezte Ember2-et 250 sebzessel
Ember2 meghalt
Torony2 megsebezte Ember3-et 250 sebzessel
Torony2-rol a kod felszállt
400 varazseronk van
Ember3 ralepett az (5, 2) cellara
400 varazseronk van
Ember3 ralepett az (5, 1) cellara
Torony1 megsebezte Ember3-et 250 sebzessel
Torony2 megsebezte Ember3-et 250 sebzessel
400 varazseronk van
Ember3 ralepett az (6, 1) cellara
400 varazseronk van
Ember3 ralepett az (7, 1) cellara
Torony2 megsebezte Ember3-et 250 sebzessel
Ember3 meghalt
minden ellenség meghalt, nyertünk

```

8.3 A tesztelést támogató programok tervei

A tesztelést nem támogatjuk általunk írt programmal, a diff parancssori eszközt fogjuk használni.

8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2010.04.03. 19:00	0,5 óra	Lipták Sebők Szepes Varga	Hetedik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása.

2014.04.04. 21:00	1,5 óra	Varga	8.1.4, 8.1.5, 8.1.7, 8.1.9, 8.1.15, 8.1.17 elkészítése
2014.04.04. 22:00	1,5 óra	Lipták	8.1.1 8.1.2 8.1.3 8.1.11 8.1.12 8.1.19 8.1.20
2014.04.04. 21:30	1 óra	Szepes	8.1.6, 8.1.8, 8.1.10, 8.1.13, 8.1.14, 8.1.16, 8.1.18, 8.1.21 elkészítése.
2014.04.05. 20:00	3 óra	Sebők	8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén elkészítése
2014.04.05. 23:00	0,5	Lipták	Javítások, ellenőrzések
2014.04.06. 11:30	0,5 óra	Szepes	8.1.6, 8.1.8, 8.1.10, 8.1.13, 8.1.14, 8.1.16, 8.1.18, 8.1.21 formázása, apróbb hibák javítása
2014.04.07. 10:00	0,5 óra	Sebők	Javítások.

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
src\Akadaly.java	1 286 B	2014.04.21. 19:35	Akadaly osztály
src\Cella.java	1 570 B	2014.04.21. 14:26	Cella osztály
src\Domborzat.java	905 B	2014.04.21. 19:14	Domborzat osztály
src\Ellenseg.java	3 174 B	2014.04.21. 19:29	Ellenseg osztály
src\Ember.java	1 764 B	2014.04.21. 14:05	Ember osztály
src\EmberPirosKo.java	1 104 B	2014.04.21. 19:16	EmberPirosKo osztály
src\Hobbit.java	1 768 B	2014.04.21. 14:05	Hobbit osztály
src\HobbitPirosKo.java	1 115 B	2014.04.21. 19:16	HobbitPirosKo osztály
src\Jatek.java	12 532 B	2014.04.22. 12:36	Jatek osztály
src\KekKo.java	364 B	2014.04.21. 19:16	KekKo osztály
src\Main.java	3 822 B	2014.04.21. 15:29	Main osztály
src\Pont.java	580 B	2014.04.21. 14:05	Pont osztály
src\SargaKo.java	933 B	2014.04.21. 20:30	SargaKo osztály
src\Torony.java	5 308 B	2014.04.21. 19:47	Torony osztály
src\ToronyKo.java	4 711 B	2014.04.21. 19:19	ToronyKo osztály
src\Torp.java	1 737 B	2014.04.21. 14:05	Torp osztály
src\TorpPirosKo.java	1 089 B	2014.04.21. 19:16	TorpPirosKo osztály
src\Tunde.java	1 753 B	2014.04.21. 14:05	Tunde osztály
src\TundePirosKo.java	1 101 B	2014.04.21. 19:16	TundePirosKo osztály
src\Ut.java	2 641 B	2014.04.21. 15:36	Ut osztály
src\VegzetHegye.java	1 152 B	2014.04.21.	VegzetHegye osztály

		14:34	
src\ZoldKo.java	360 B	2014.04.21. 19:20	ZoldKo osztály
test\akadaly.in.txt	75 B	2014.04.18. 23:45	akadaly teszteset bemenet
test\akadaly.test.txt	429 B	2014.04.21. 19:46	akadaly teszteset elvárt kimenet
test\diff.exe	150 528 B	2014.04.19. 00:18	diff parancssori segédprogram
test\elagazas.in.txt	93 B	2014.04.21. 15:44	elagazas teszteset bemenet
test\elagazas.map.txt	94 B	2014.04.18. 23:51	elagazas térkép fájl
test\elagazas.test.txt	345 B	2014.04.21. 15:45	elagazas teszteset elvárt kimenet
test\fordit.bat	267 B	2014.04.22. 12:38	Batch-fájl a fordításhoz
test\futtat.bat	1 387 B	2014.04.21. 19:55	Batch-fájl a futtatáshoz
test\kekko.in.txt	139 B	2014.04.18. 23:48	kekko teszteset bemenet
test\kekko.map.txt	67 B	2014.04.18. 23:48	kekko térkép fájl
test\kekko.test.txt	592 B	2014.04.21. 19:18	kekko teszteset elvárt kimenet
test\kod.in.txt	80 B	2014.04.18. 23:50	kod teszteset bemenet
test\kod.map.txt	192 B	2014.04.18. 23:50	kod térkép fájl
test\kod.test.txt	1 715 B	2014.04.21. 19:49	kod teszteset elvárt kimenet
test\libiconv2.dll	898 048 B	2014.04.19. 00:19	Függvénykönyvtár a diff-hez
test\libintl3.dll	92 672 B	2014.04.19. 00:19	Függvénykönyvtár a diff-hez
test\meghal.in.txt	97 B	2014.04.21. 20:01	meghal teszteset bemenet
test\meghal.map.txt	160 B	2014.04.18. 23:50	meghal térkép fájl
test\meghal.test.txt	837 B	2014.04.21. 20:00	meghal teszteset elvárt kimenet
test\nyeres.in.txt	98 B	2014.04.21. 21:25	nyeres teszteset bemenet
test\nyeres.test.txt	2 332 B	2014.04.21. 21:26	nyeres teszteset elvárt kimenet
test\osztodik.in.txt	112 B	2014.04.18. 23:49	osztodik teszteset bemenet
test\osztodik.test.txt	597 B	2014.04.21. 19:57	osztodik teszteset elvárt kimenet

test\palya.in.txt	27 B	2014.04.18. 23:44	palya teszteset bemenet
test\palya.map.txt	85 B	2014.04.17. 20:31	palya térkép fájl
test\palya.test.txt	51 B	2014.04.18. 23:44	palya teszteset elvárt kimenet
test\pirosko.in.txt	165 B	2014.04.18. 23:47	pirosko teszteset bemenet
test\pirosko.map.txt	148 B	2014.04.18. 23:47	pirosko térkép fájl
test\pirosko.test.txt	1 062 B	2014.04.18. 23:47	pirosko teszteset elvárt kimenet
test\sargako.in.txt	122 B	2014.04.18. 23:46	sargako teszteset bemenet
test\sargako.test.txt	649 B	2014.04.21. 19:47	sargako teszteset elvárt kimenet
test\torony.in.txt	97 B	2014.04.18. 23:45	torony teszteset bemenet
test\torony.map.txt	157 B	2014.04.18. 23:43	torony térkép fájl
test\torony.test.txt	779 B	2014.04.18. 23:46	torony teszteset elvárt kimenet
test\vesztes.in.txt	35 B	2014.04.18. 23:51	vesztes teszteset bemenet
test\vesztes.test.txt	354 B	2014.04.18. 23:52	vesztes teszteset elvárt kimenet
test\zoldko.in.txt	143 B	2014.04.18. 23:48	zoldko teszteset bemenet
test\zoldko.map.txt	124 B	2014.04.18. 23:49	zoldko térkép fájl
test\zoldko.test.txt	853 B	2014.04.21. 19:23	zoldko teszteset elvárt kimenet

10.1.2 Fordítás

A követelményekben leírtaknak megfelelően a program fordításához egy olyan környezetre van szükség, melyben a Java 1.7-es fordítókörnyezet (**javac** program) elérhető.

(Egy parancssori ablakban a **javac** mappáját a PATH környezeti változóhoz a

```
set PATH=%PATH%;C:\Program Files\Java\jdk1.7.0_40\bin
```

paranccsal adhatjuk hozzá, az elérési útvonalat értelemszerűen átírva a saját környezetnek megfelelően.)

Egy ilyen környezetben indítsuk el a **test** könyvtárban a **fordit.bat** batch-fájlt, mely kitörli a meglévő binárisokat és újrafordítja a programot.

10.1.3 Futtatás

A követelményekben leírtaknak megfelelően a program futtatásához egy olyan környezetre van szükség, melyben a Java 1.7-es futtatókörnyezet (java program) elérhető.

(Egy parancssori ablakban a java mappáját a PATH környezeti változóhoz a

```
set PATH=%PATH%;C:\Program Files\Java\jre7\bin
```

paranccsal adhatjuk hozzá, az elérési útvonalat értelemszerűen átírva a saját környezetnek megfelelően.)

Egy ilyen környezetben indítsuk el a `test` könyvtárban a `futtat.bat` batch-fájlt, mely a teszteseteken végigiterálva a következőket végzi el:

- a szabványos bemenetre irányítja a teszteset bemenetét (`<teszteset>.in.txt`)
- a szabványos kimenetet a `<teszteset>.out.txt` fájlba irányítja
- lefuttatja a programot
- a diff parancssori eszközzel összehasonlítja a kapott eredményt (`<teszteset>.out.txt`) és a tesztesetekenél leírt elvárt kimenetet (`<teszteset>.test.txt`), az összehasonlítás eredményét kiírja a `<teszteset>.diff.txt` fájlba
- kijelzi, hogy a teszteset az elvárt kimenetet adta-e

A program a futás végén automatikusan kijelzi a sikeres tesztesetek számát is, részletesebb vizsgálódáshoz pedig a kimeneti fájlok saját szemmel is összehasonlíthatók az elvárttal (illetve a program a parancssorban a `java -cp bin Main` utasítással elindítva kézzel is tesztelhető, ilyenkor figyeljünk a térkép-fájlok megadásánál az elérési útra).

10.2 Tesztek jegyzőkönyvei

10.2.1 Akadály lassítása (akadaly)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 20:00

10.2.2 Elágazások (elagazas)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 20:05
Teszt eredménye	A kimenet nem egyezik az elvárttal.
Lehetséges hibaok	A bemenet és az elvárt kimenet hibásan lettek meghatározva, az elágazás „véletlenszerűségét” egy léptetéssel hamarabb kellett volna átállítani.
Változtatások	A bemenetben és az elvárt kimenetben a feltárt hibát javítottuk, ezek új verziója a <code>test/elagazas.in.txt</code> és <code>test/elagazas.test.txt</code> fájlokban érhető el.

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 20:15

10.2.3 Kékkő hatása (kekko)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 20:20
Teszt eredménye	A kimenet nem egyezik az elvárttal.
Lehetséges hibaok	Az az elvárt kimenet hibásan lett meghatározva, „kekko” helyett „hobbit pirosko” elírások maradtak benne.
Változtatások	Az elvárt kimenetben a feltárt hibát javítottuk, ennek új verziója a <code>test/kekko.test.txt</code> fájlban érhető el.

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 20:30

10.2.4 A toronyra köd ereszkedik (kod)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 20:35
Teszt eredménye	A kimenet nem egyezik az elvárttal.
Lehetséges hibaok	Az az elvárt kimenet hibásan lett meghatározva, a torony koordinátái el lettek írva.
Változtatások	Az elvárt kimenetben a feltárt hibát javítottuk, ennek új verziója a <code>test/kod.test.txt</code> fájlban érhető el.

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 20:45

10.2.5 Ellenség meghal (meghal)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 20:50
Teszt eredménye	A kimenet nem egyezik az elvárttal.
Lehetséges hibaok	A bemenet hibásan lett meghatározva, egyel tovább léptettük a játékot, mint kellett volna.
Változtatások	A bemenetben a feltárt hibát javítottuk, ennek új verziója a <code>test/meghal.in.txt</code> fájlban érhető el.

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:00

10.2.6 Játék megnyerése (nyeres)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:05
Teszt eredménye	A kimenet nem egyezik az elvárttal.
Lehetséges hibaok	A bemenet és az elvárt kimenet hibásan lettek meghatározva, egy torony újratöltési idejét egyszer nem vettük figyelembe. Így összességében elég volt eggyel kevesebbszer léptetni a játékot.
Változtatások	A bemenetben és az elvárt kimenetben a feltárt hibát javítottuk, ezek új verziója a <code>test/nyeres.in.txt</code> és <code>test/nyeres.test.txt</code> fájlokban érhető el.

Tesztelő neve	Sebők Márton
---------------	--------------

Teszt időpontja	2014.04.21. 21:15
------------------------	-------------------

10.2.7 Ellenség osztódik (osztodik)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:20
Teszt eredménye	A kimenet nem egyezik az elvárttal.
Lehetséges hibaok	Az elvárt kimenet hibásan lett meghatározva, az utolsó sor, amiben az újonnan „osztódott” ellenség rálépett a cellára, lemaradt.
Változtatások	Az elvárt kimenetben a feltárt hibát javítottuk, ennek új verziója a <code>test/osztodik.test.txt</code> fájlban érhető el.

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:30

10.2.8 Pálya betöltése (palya)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:35

10.2.9 Piroskő hatása (pirosko)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:40

10.2.10 Sárgakő hatása (sargako)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:45

10.2.11 Torony lövése (torony)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:50

10.2.12 Játék elvesztése (vesztes)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 21:55

10.2.13 Zöldkő hatása (zoldko)

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 22:00
Teszt eredménye	A kimenet nem egyezik az elvárttal.
Lehetséges hibaok	Az elvárt kimenet hibásan lett meghatározva, a zöldkő második sebzését egy léptetéssel később írtuk, mint kellett volna.
Változtatások	Az elvárt kimenetben a feltárt hibát javítottuk, ennek új verziója a <code>test/zoldko.test.txt</code> fájlban érhető el.

Tesztelő neve	Sebők Márton
Teszt időpontja	2014.04.21. 22:10

10.3Értékelés

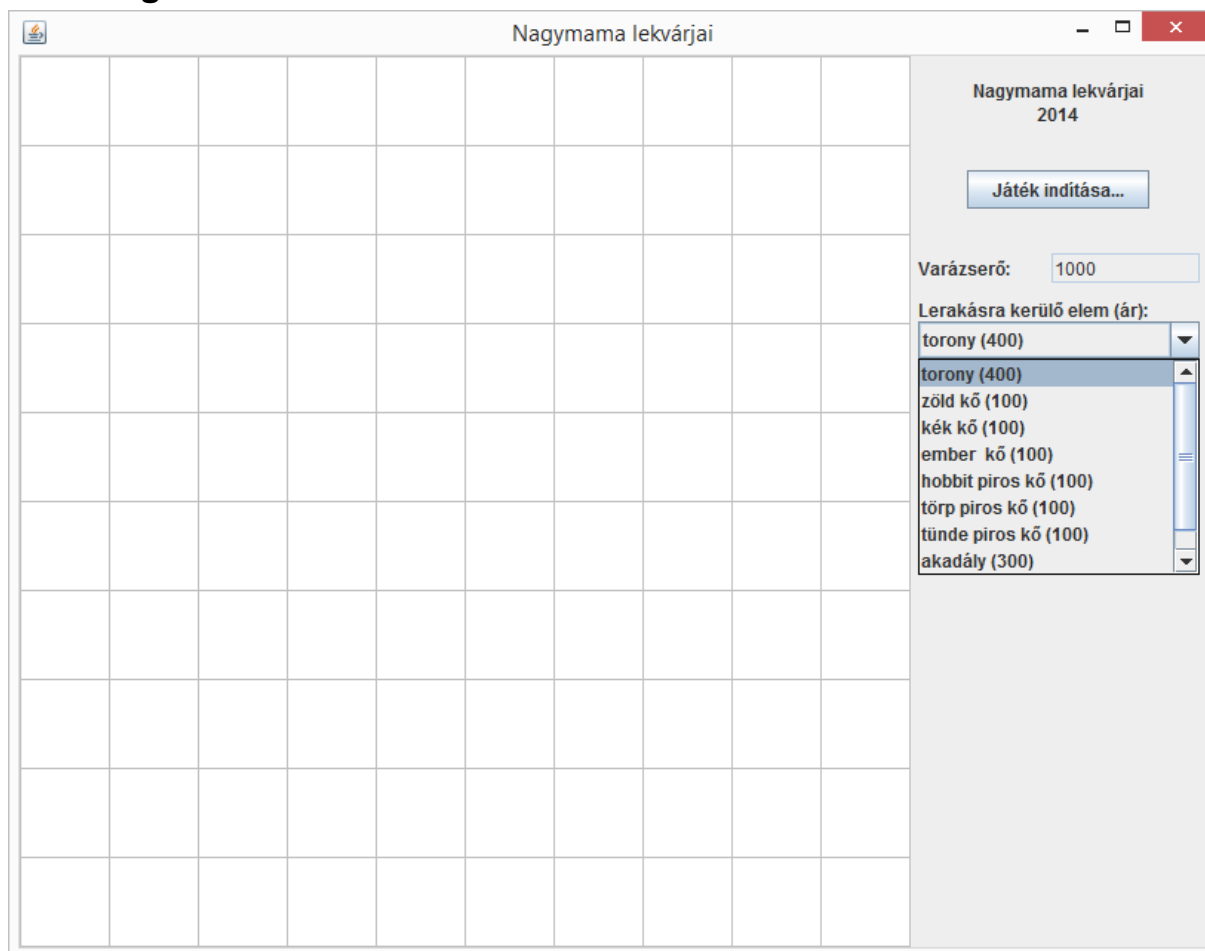
Tag neve	Munka százalékban
Lipták Levente	21%
Szepes Nóra	21%
Varga Gergő	23%
Sebők Márton	35%

10.4Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.04.17. 18:00	0,5 óra	Lipták Sebők Szepes Varga	Nyolcadik értekezlet. Döntések: jelen dokumentum elkészítésének és a kódolásnak a felosztása.
2014.04.18. 20:00	2 óra	Sebők	Main és Jatek osztályok átírása, a szkeleton-specifikus részek eltávolítása, a tesztelési és fordítási keretrendszerhez a batch-fájlok megírása, a tesztesetek átültetése fájlokba.
2014.04.18. 22:00	1 óra	Szepes	Torony és toronykövek implementálása, szkeleton kódjának eltávolítása
2014.04.19. 17:00	0,5 óra	Szepes	Torony és toronykövek javítása, kiírások implementálása
2014.04.20. 22:00	1.5 óra	Lipták	A Pont, Cella, Ut, Vegzethegye, Akadaly, Sargako, Domborzat osztályok implementálása
2014.04.21. 10:30	1,5 óra	Varga	Ellenség és leszármazottainak implementálása
2014.04.21. 14:00	0,5 óra	Szepes	Torony és toronykövek javítása, objektumazonosítók felvétele
2014.04.21. 18:00	2 óra	Sebők	A többi kód és a tesztesetek javítása.
2014.04.21. 20:00	2 óra	Sebők	A dokumentáció megírása.

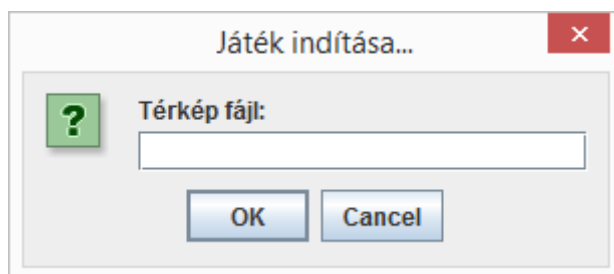
11. Grafikus felület specifikációja

11.1 A grafikus interfész

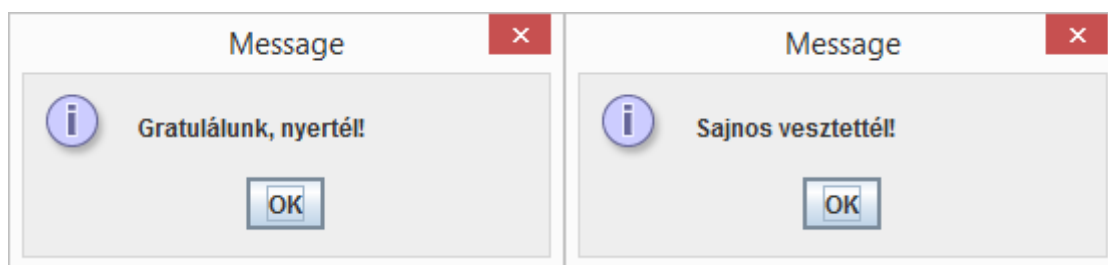


1. ábra A program főablaka

A játék indításakor a program megkérdezi a betöltendő térkép fájl nevét (elérési útját), a játék megnyerésekor és elvesztésekor üzenetablakot jelenít meg.



2. ábra: Játék indítása dialógusablak



3. ábra Üzenetek a játék végén

A bal oldali négyzetrácsos rész a játék pályája, az egyes elemeket a következő ikonok jelölik (egy cellán több ilyen is lehet, értelemszerűen egymásra rétegezve):

	út		ember piros kő (toronyra)
	domborzat		hobbit piros kő (toronyra)
	Végzet Hegye		törp piros kő (toronyra)
	akadály (útra)		tünde piros kő (toronyra)
	torony (domborzatra)		ember (útra)
	torony ködben		hobbit (útra)
	sárga kő (akadályra)		törp (útra)
	zöld kő (toronyra)		tünde (útra)
	kék kő (toronyra)		

Egy cellán a rajta lévő első 4 ellenséget jelöljük, egymás alatt elhelyezve, ezeknél a színes csíkok hossza az életükkel arányos. Az egyes elemek lerakásához a felhasználó először kiválasztja a lenyíló listából, hogy mit szeretne lerakni, majd rákattint a megfelelő cellára.

11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

A feladat megoldásához Modell-View-Controller tervezési mintát valósítunk meg. A grafikai megjelenítés elkülönül a program többi elemétől, amennyire csak lehetséges, így könnyedén át lehet alakítani a grafikus felületet.

11.2.1.1 A model

A modellben csak kisebb változtatások lesznek, melyek célja, hogy a pályán történő összes változás látható legyen a Jatek központi osztályból, mely kommunikál a controller résszel. A változások push alapelvűek, azokat a Jatek üzeni meg a beállító függvények visszatérési értékeként, vagy függvényhívással.

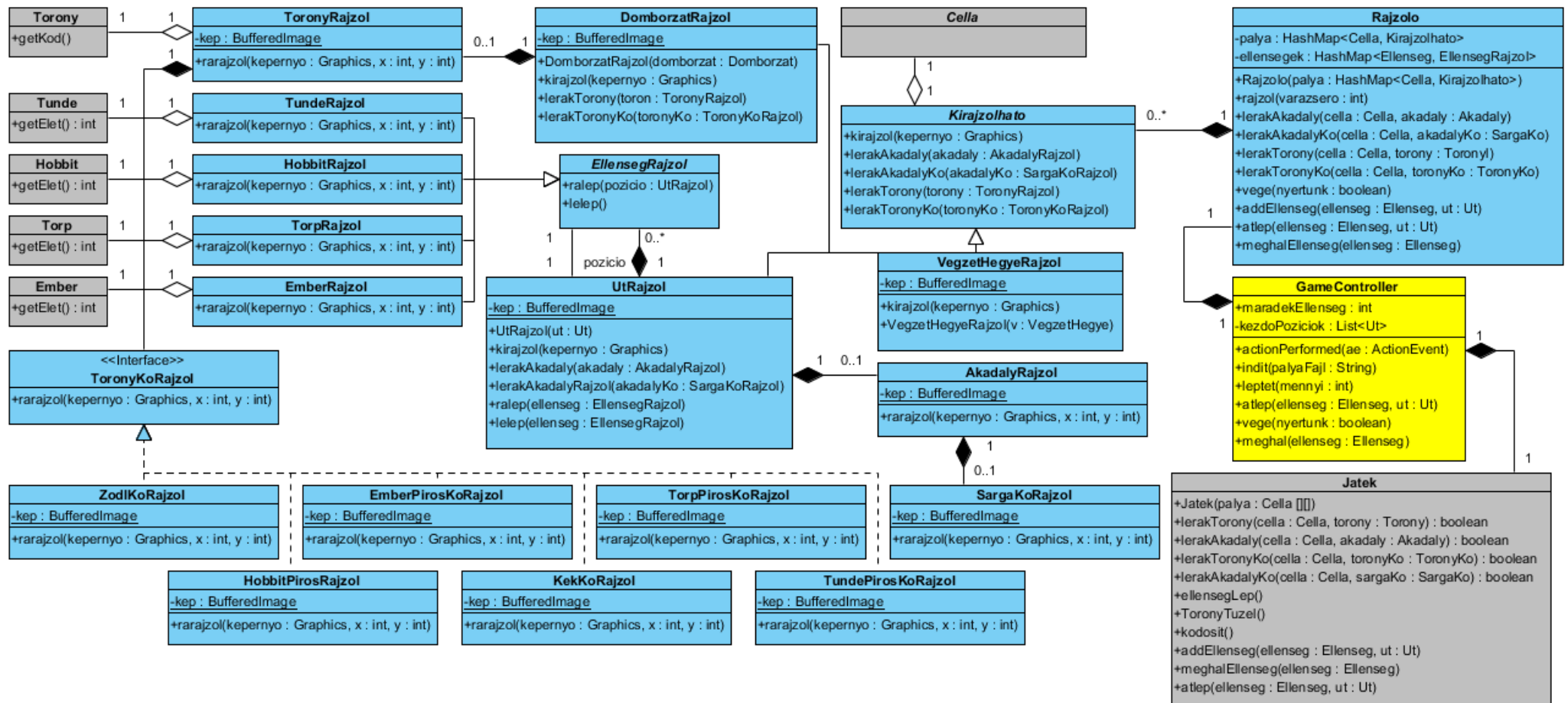
11.2.1.2 A view

A nézetet úgy valósítjuk meg, hogy egy központi osztály, a Rajzolo felépíti a térképet, ahogy a modellben a Jatek, de az eredeti osztályok helyett kirajzoló osztályokat használva. Ahol szükséges, ezek az osztályok referenciával rendelkeznek a modellbeli megfelelőjükre, hogy kirajzoláskor felhasználhassák azok belső állapotát. A kirajzoló osztályok a felület változtatására használható függvényeket valósítják meg (pl. torony vagy toronykő lerakása), illetve a kirajzol/rárajzol függvényt, ami egy képet kap paraméterül, és erre rajzolja ki magát (a modellbeli változatának pozíciója szerinti helyre). Miután minden objektum kirajzolta magát, a Rajzolo kicseréli az eddig mutatott térképet az újra. Saját maga képe minden rajzoló osztályban el van mentve, statikus attribútumként, azt a Rajzolo állítja be még a játék kezdete előtt. A konkrét megvalósításban csak azok a rajzoló osztályok tartalmazzák a modellbeli változatukat, amelyek információhoz is jutnak ezáltal (pl. pozíció kiolvasása).

11.2.1.3 A controller

A controller részt a GameController osztály valósítja meg, mely felhasználói vezérlőkben bekövetkezett események hatására továbbhív a Jatek illetve Rajzolo osztály függvényeibe a megfelelő paraméterekkel (pl. torony lerakása adott cellára.). Továbbá kapcsolatot tart a modell és a nézet között: a modell változásairól értesíti a nézetet, ill. a nézet eseményei alapján módosítja a modellt.

11.2.2 A felület osztály-struktúrája



11.3 A grafikus objektumok felsorolása

11.3.1 Rajzolo

- **Felelősség**

A modell képernyőn való megjelenítéséért felelős. Ő tartja a kapcsolatot a GameController és a View között. Ő rajzolja ki például a térképet és a legördülő listát.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **HasMap<Ellenseg, EllensegRajzol> ellensegek:** Az ellenségek, és a hozzájuk tartozó kirajzolandó elemek.
- - **HashMap<Cella, Kirajzolható> palya:** A cellák, és a hozzájuk tartozó kirajzolandó elemek.

- **Metódusok**

- + **Rajzolo(HashMap<Cella, Kirajzolható>: cellak):** a nézet konstruktora, felépíti az ablakot.
- + **void lerakAkadaly(cella : Cella, Akadaly: akadaly):** Lerak egy akadályt a cellára.
- + **void lerakAkadalyko(cella : Cella, SargaKo: akadalyKo):** Lerak egy akadálykövet a cellára.
- + **void lerakTorony(cella : Cella, Torony: torony):** Lerak egy tornyot a cellára.
- + **void lerakToronyKo(cella : Cella, ToronyKo: toronyKo):** Lerak egy toronykövet a cellára.
- + **void rajzol(int: varazsero):** Kirajzolja a képernyőre az aktuális varázserő mennyiséget, és a térképet.
- + **void vege(boolean: nyertunk):** A játék megnyerése/elvesztése esetén megjelenítő popupot készíti el.
- + **void addEllenseg(Ellenseg: ellenseg, Ut: ut):** Ellenség hozzáadása a térképhez.
- + **void meghalEllenseg(Ellenseg: ellenseg):** Ellenség levétele a térképről.
- + **void ralep(Ellenseg: ellenseg, Ut: ut):** Rálépteti az útra az ellenséget.

11.3.2 GameController

- **Felelősség**

Az MVC-ben ő a Controller osztályunk. Ő tartja a kapcsolatot a modellel és a nézettel is.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- + **int maradekEllenseg**: A maradék ellenségek száma.
- - **List<Ut> kezdPoziok**: Ezeken az utakon kezdhetnek az ellenségek.
- - **Jatek jatek**: A modell, melyet irányít.
- - **Rajzolo rajzolo**: A view, melyet irányít.

- **Metódusok**

- + **void actionPerformed(ActionEvent: ae)**: Az eseménykezelő függvény, ez hívja meg az ablakban történt kattintások után a megfelelő függvényeket.
- + **void ralep(EllensegRajzolo: ellenseg)**: Átlépteti az ellenséget az egyik celláról a másikra.
- + **void indit(String: palyaFajl)**: Elindítja a játékot. Betölti a paraméterként kapott fájlnev alapján a térképájl, felépíti a modell és a nézet térképét.
- + **void leptet(int: mennyi)**: Lépteti a játékot a megfelelő időegységgel.
- + **void meghal(EllensegRajzol: ellenseg)**: Ha meghal az ellenség, szól a View-nak, hogy vegye le a térképről.
- + **void vege(boolean: nyertunk)**: Megjeleníti nyeres illetve vesztes esetén a megfelelő felugró ablakot.

11.3.3 Kirajzolható

- **Felelősség**

A modellben lévő cellához hasonló és annak a kirajzolásáért felelős.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **Cella cella**: A modellbeli cella megfelelője.

- **Metódusok**

- + **void kirajzol(Graphics: kepernyo)**: Kirajzolja a képernyőre a cellát.
- + **void lerakAkadaly(Akadaly: akadaly)**: Lerak egy akadályt a cellára.
- + **void lerakAkadalyko(SargaKo: akadalyKo)**: Lerak egy akadálykövet a cellára.
- + **void lerakTorony(Torony: torony)**: Lerak egy tornyot a cellára.
- + **void lerakToronyKo(ToronyKo: toronyKo)**: Lerak egy toronykövet a cellára.

11.3.4 VegzetHegyeRajzol

- **Felelősség**

A végzet hegyének térképen való megjelenítéséért felelős.

- **Ősosztályok**

Kirajzolható

- **Interfészek**

-

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void kirajzol(Graphics: kepernyo):** Kirajzolja a végzet hegyét a képernyőre.
- + **void VegzetHegyeRajzol(VegzetHegye v):** Konstruktor. Paraméterként megkapja a modellbeli változatát.

11.3.5 DomborzatRajzol

- **Felelősség**

A domborzat térképen való megjelenítéséért felelős.

- **Ősosztályok**

Kirajzolható

- **Interfészek**

-

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.
- - **ToronyRajzol torony:** A domborzaton lévő torony.

- **Metódusok**

- + **void kirajzol(Graphics: kepernyo):** Kirajzolja a domborzatot a képernyőre.
- + **void lerakTorony(ToronyRajzol: torony):** Lerak egy tornyot a domborzatra.
- + **void lerakToronyKo(ToronyKoRajzol: toronyKo):** Lerak egy toronykövet a domborzatra.
- + **DomborzatRajzol(Domborzat: domborzat):** Konstruktor. Paraméterként megkapja a modellbeli változatát.

11.3.6 UtRajzol

- **Felelősség**

Az út térképen való megjelenítéséért felelős.

- **Ósosztályok**

Kirajzolható

- **Interfészek**

-

- **Attribútumok**

- - **AkadalyRajzol akadaly:** Az úton lévő akadály.
- - **EllensegRajzol[] ellensegek:** Az úton lévő ellenségek tömbje.
- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void kirajzol(Graphics: kepernyo):** Kirajzolja az utat a képernyőre.
- + **void lelep(EllensegRajzol: ellenseg):** Lelépteti a paraméterül kapott ellenséget az útról.
- + **void lerakAkadaly(AkadalyRajzol: akadaly):** Lerak egy akadályt az útra.
- + **void lerakAkadalyKo(AkadalyKoRajzol: akadalyKo):** Lerak egy sárgakövet az útra.
- + **void ralep(EllensegRajzol: ellenseg):** Rálépteti a paraméterül kapott ellenséget az útra.
- + **void UtRajzol(Ut: ut):** Konstruktor. Paraméterként megkapja a modellbeli változatát.

11.3.7 EllensegRajzol

- **Felelősség**

A modellbeli ellenség megfelelője. Absztrakt osztály, helyette a leszármazottakkal dolgozunk. A leszármazottai a tündék, törpék, emberek és hobbitok. Az ellenség térképen való megjelenítéséért felelős. A heterogén kollekcióhoz szükséges.

- **Ósosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **UtRajzol pozicio:** Az út, melyen az ellenség tartózkodik.

- **Metódusok**

- + **void ralep(UtRajzol: pozicio):** Átlépteti a paraméterül kapott útra az ellenséget.
- + **void lelep():** Lelépteti az ellenséget az aktuális pozícióról.

11.3.8 AkadalyRajzol

- **Felelősség**

Az akadály térképen való megjelenítéséért felelős.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **SargaKoRajzol sargaKo:** Az akadályon lévő sárgakő.
- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja az akadályt a képernyőre.

11.3.9 ToronyRajzol

- **Felelősség**

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **Torony torony:** A ToronyRajzol modellben lévő megfelelője. A kód lekérése miatt van rá szükség.
- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a tornyot a képernyőre.

11.3.10 TundeRajzol

- **Felelősség**

A tünde térképen való megjelenítéséért felelős.

- **Ősosztályok**

EllensegRajzol

- **Interfészek**

-

- **Attribútumok**

- - **Tunde tunde:** A modellben lévő ellenség. Az élet lekérdezése miatt van rá szükség.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a tündét a képernyőre.

11.3.11 HobbitRajzol

- **Felelősség**

A hobbit térképen való megjelenítéséért felelős.

- **Ősosztályok**

EllensegRajzol

- **Interfészek**

-

- **Attribútumok**

- - **Hobbit hobbit:** A modellben lévő ellenség. Az élet lekérdezése miatt van rá szükség.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a hobbitot a képernyőre.

11.3.12 TorpRajzol

- **Felelősség**

A törp térképen való megjelenítéséért felelős.

- **Ősosztályok**

EllensegRajzol

- **Interfészek**

-

- **Attribútumok**

- - **Torp torp:** A modellben lévő ellenség. Az élet lekérdezése miatt van rá szükség.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a törpét a képernyőre.

11.3.13 EmberRajzol

- **Felelősség**

Az ember térképen való megjelenítéséért felelős.

- **Ősosztályok**

EllensegRajzol

- **Interfészek**

-

- **Attribútumok**

- - **Ember ember:** A modellben lévő ellenség. Az élet lekérdezése miatt van rá szükség.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja az embert a képernyőre.

11.3.14 SargaKoRajzol

- **Felelősség**

A sárgakő térképen való megjelenítéséért felelős.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a sárgakövet a képernyőre.

11.3.15 ToronyKoRajzol

- **Felelősség**

Absztrakt osztály, belőle származik a tünde piros kő, az ember piros kő, a törp piros kő a hobbit piros kő, a zöld kő és a kék kő. A heterogén kollekcióhoz szükséges.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a toronykövet a képernyőre.

11.3.16 ZoldKoRajzol

- **Felelősség**

A zöld kő térképen való megjelenítéséért felelős.

- **Ősosztályok**

-

- **Interfészek**

ToronyKoRajzol

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a toronykövet a képernyőre.

11.3.17 KekKoRajzol

- **Felelősség**

A kék kő térképen való megjelenítéséért felelős.

- **Ősosztályok**

-

- **Interfészek**

ToronyKoRajzol

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a toronykövet a képernyőre.

11.3.18 HobbitPirosKoRajzol

- **Felelősség**

A hobbit piros kő térképen való megjelenítéséért felelős.

- **Ősosztályok**

-

- **Interfészek**

ToronyKoRajzol

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a toronykövet a képernyőre.

11.3.19 EmberPirosKoRajzol

- **Felelősség**

Az ember piros kő térképen való megjelenítéséért felelős.

- **Ősosztályok**

-

- **Interfészek**

ToronyKoRajzol

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a toronykövet a képernyőre.

11.3.20 TundePirosKoRajzol

- **Felelősség**

A tünde piros kő térképen való megjelenítéséért felelős.

- **Ősosztályok**

-

- **Interfészek**

ToronyKoRajzol

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a toronykövet a képernyőre.

11.3.21 TorpPirosKoRajzol

- **Felelősség**

A törp piros kő térképen való megjelenítéséért felelős.

- **Ősosztályok**

-

- **Interfészek**

ToronyKoRajzol

- **Attribútumok**

- - **BufferedImage kep:** A megjelenítendő kép.

- **Metódusok**

- + **void rarajzol(Graphics: kepernyo, int: x, int: y):** Kirajzolja a toronykövet a képernyőre.

11.3.22 Jatek

- **Felelősség**

A modell belüli verzió változott.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- + **boolean lerakAkadaly(Cella: cella, Akadaly: akadaly):** Bővült egy boolean visszatérési értékkel, hogy tudjuk, frissíteni kell-e a térképen a cellát.
 - + **boolean lerakAkadalyKo(Cella: cella, SargaKo: sargaKo):** Bővült egy boolean visszatérési értékkel, hogy tudjuk, frissíteni kell-e a térképen a cellát.
 - + **boolean lerakTorony(Cella: cella, Torony: torony):** Bővült egy boolean visszatérési értékkel, hogy tudjuk, frissíteni kell-e a térképen a cellát.

- + **boolean lerakToronyKo(Cella: cella, ToronyKo: toronyKo):** Bővült egy boolean visszatérési értékkel, hogy tudjuk, frissíteni kell-e a térképen a cellát.
- + **void atlep(Ellenseg ellenseg, Ut ut):** A paraméterként kapott ellenséget átlépteti az aktuális pozíciójáról a paramtérként kapott útra.
- + **void meghal(Ellenseg ellenseg):** Ha egy ellenség meghal, visszajelez a jateknak, aki továbbítja a nézet felé az eseményt, hogy követni tudja a kirajzolás a valós állapotot.
- + **Jatek(Cella[][]: palya):** Konstruktor. Paraméterként megkapja a kész pályát.
- + **ellensegLep():** Ellenségek léptetése.
- + **ToronyTuzel():** Tornyok tüzelgetése.
- + **kodosit():** Tornyok elködösítése.
- + **addEllenseg(Ellenseg: ellenseg, Ut: ut):** Ellenség hozzáadása az adott úthoz.
- + **atlep(Ellenseg ellenseg, Ut: ut):** Jelzi az ellenség, ha átlépett az egyik útról a másikra, hogy átlépett, illetve hova, így a nézet követheti.

11.3.23 Torony

- **Felelősség**

A modell belüli verzió változott.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- + **boolean getKod():** Visszaadja, hogy van-e a tornyon kód.

11.3.24 Ember

- **Felelősség**

A modell belüli verzió változott.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- + **int** **getElet()**: Visszaadja az ellenségfajta aktuális életerejét.

11.3.25 Hobbit

- **Felelősség**

A modell belüli verzió változott.

- **Össztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- + **int** **getElet()**: Visszaadja az ellenségfajta aktuális életerejét.

11.3.26 Tunde

- **Felelősség**

A modell belüli verzió változott.

- **Össztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- + **int** **getElet()**: Visszaadja az ellenségfajta aktuális életerejét.

11.3.27 Torp

- **Felelősség**

A modell belüli verzió változott.

- **Össztályok**

-

- **Interfészek**

-

- **Attribútumok**

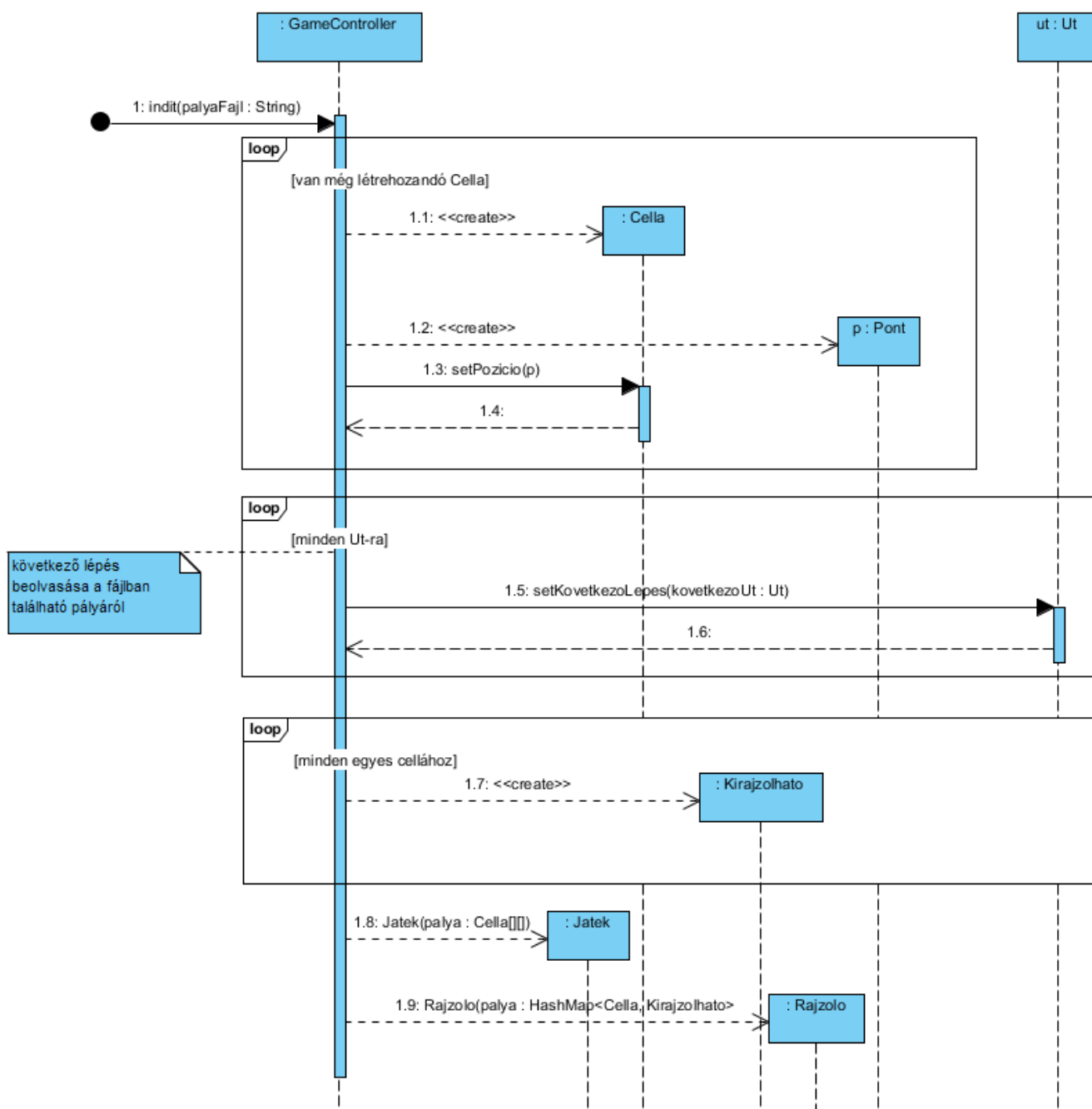
-

- **Metódusok**

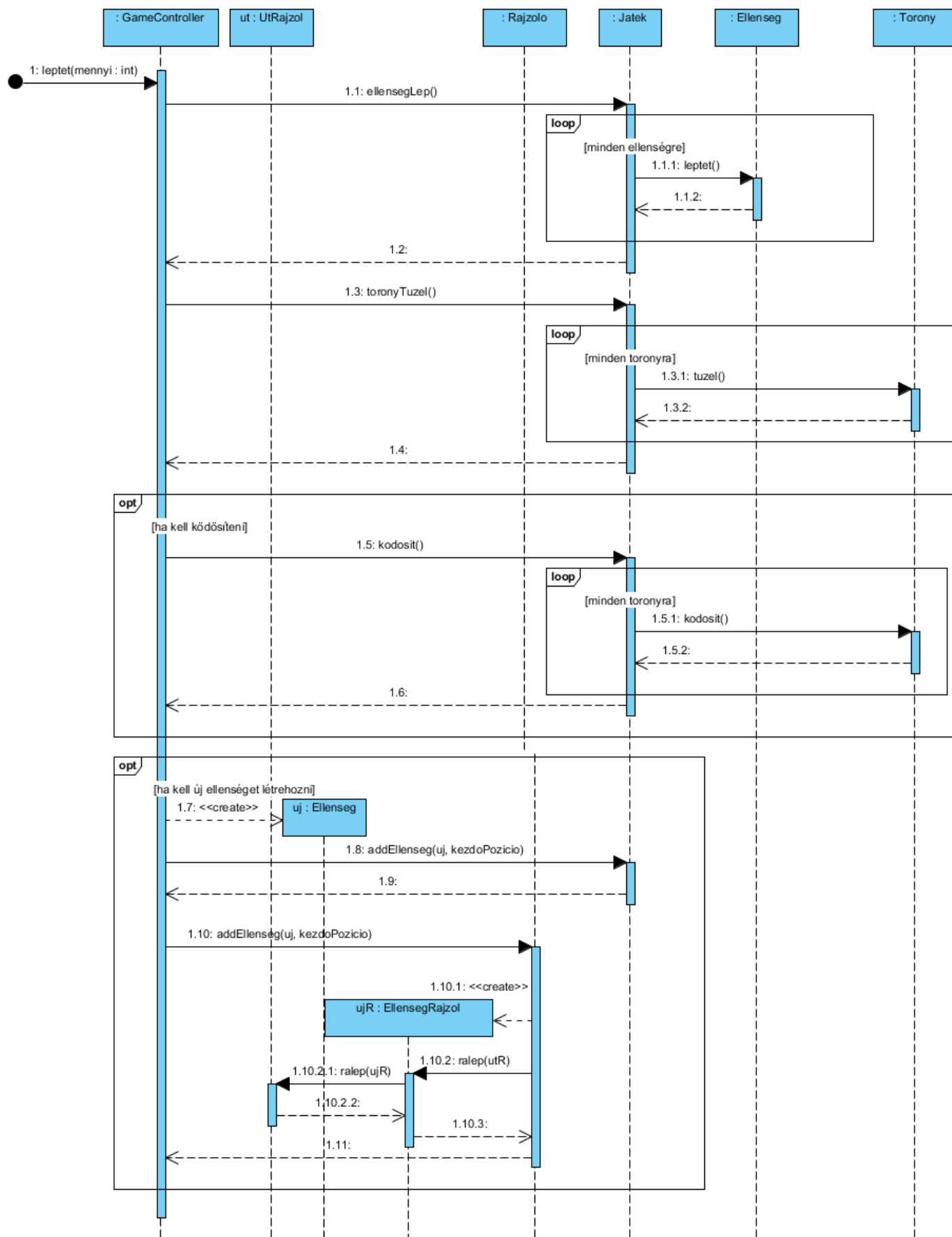
- **+ int getElet():** Visszaadja az ellenségfajta aktuális életerejét.

11.4 Kapcsolat az alkalmazói rendszerrel

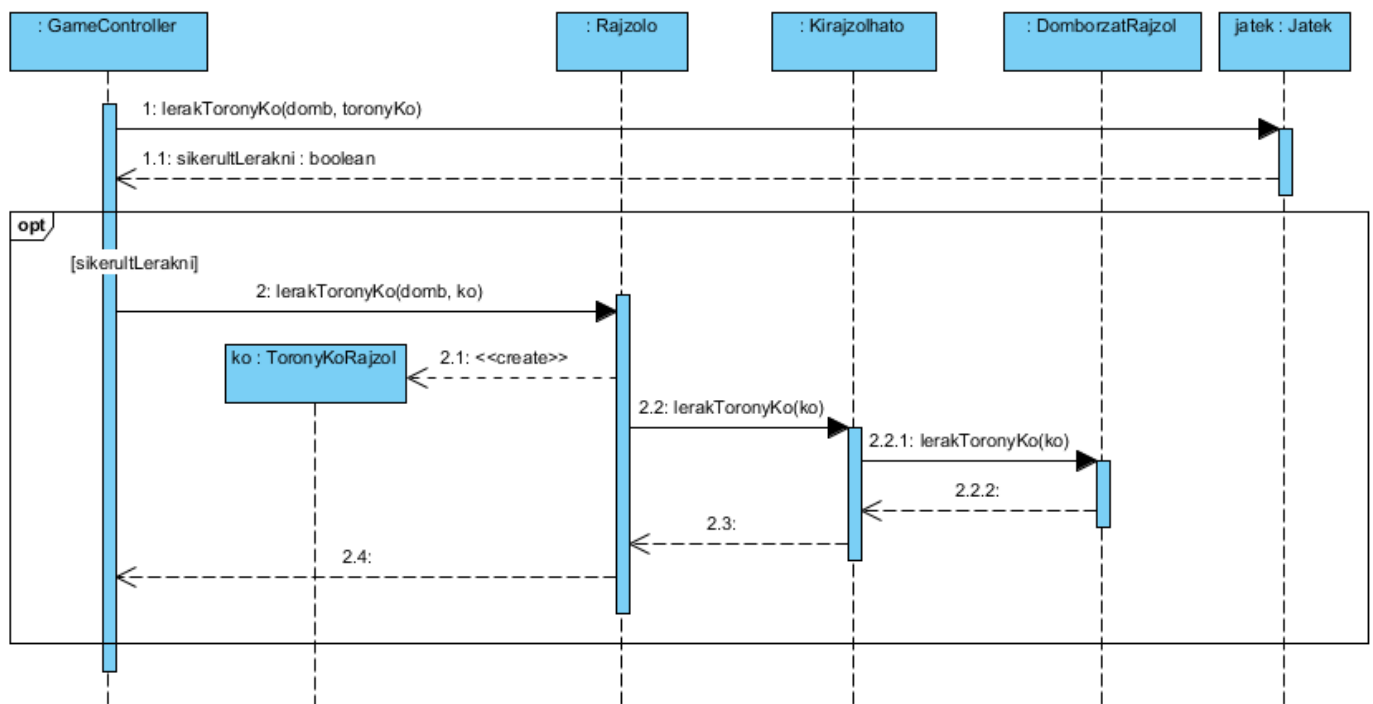
11.4.1 Indítás



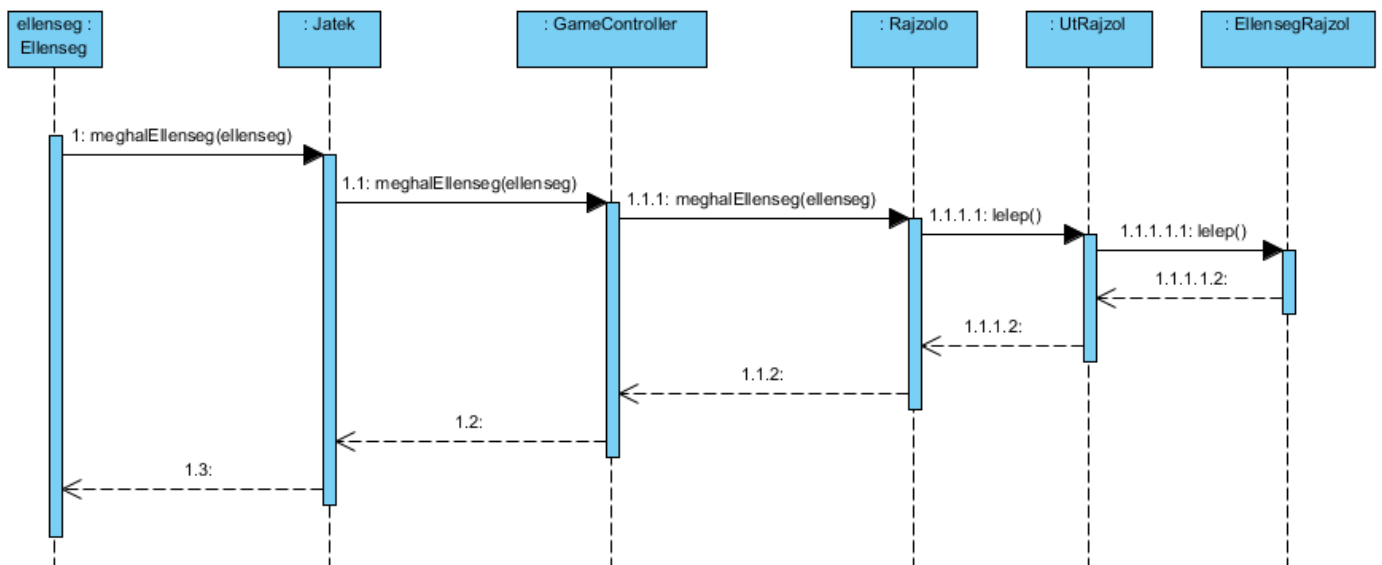
11.4.2 Léptetés



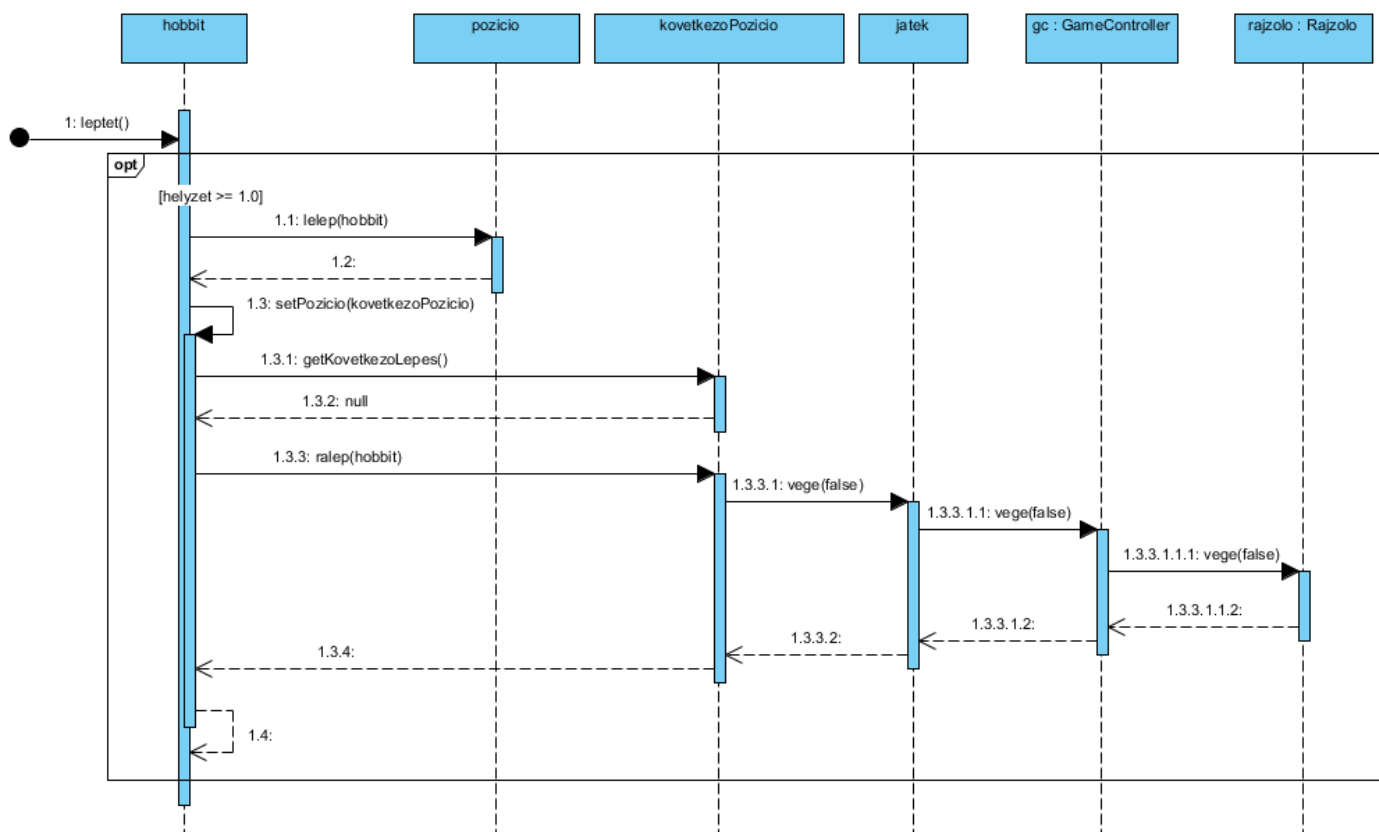
11.4.3 Toronykő lerakása



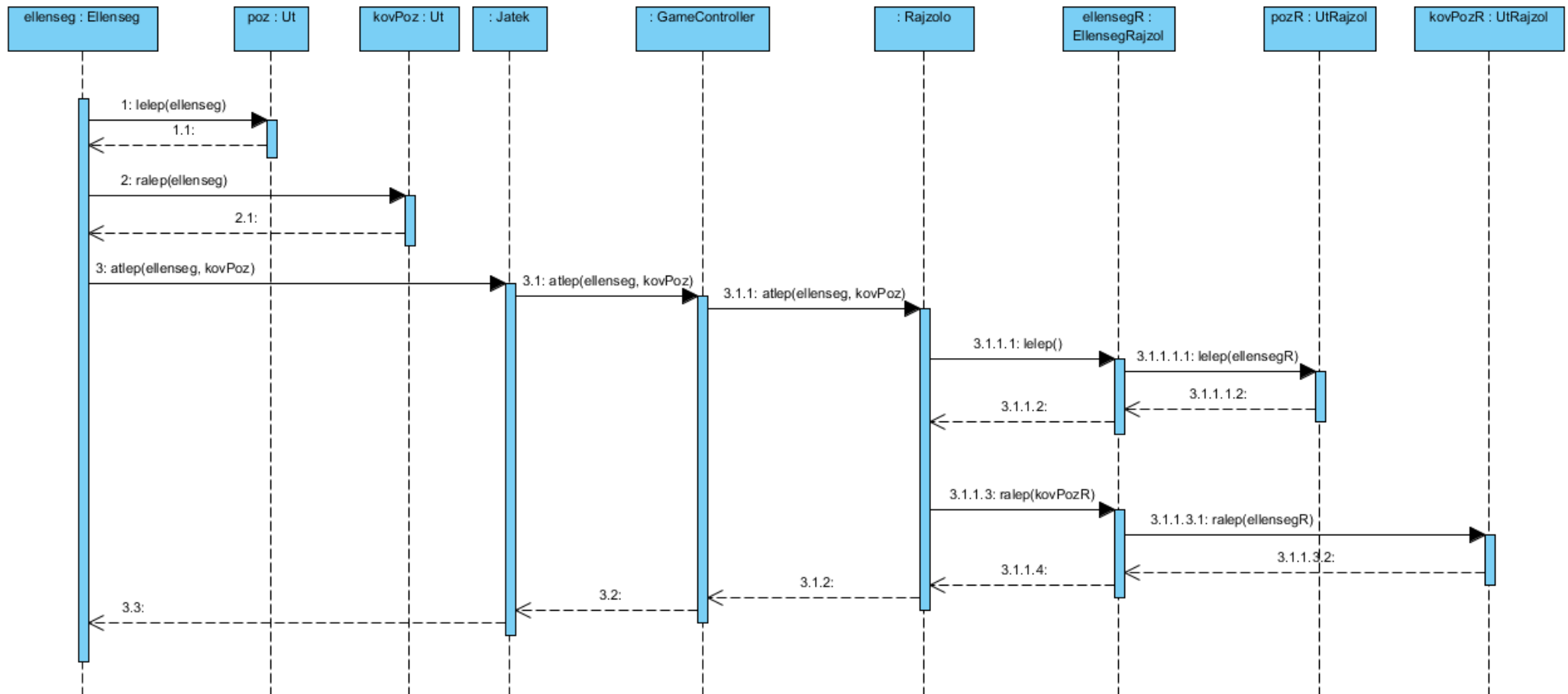
11.4.4 Ellenség meghal



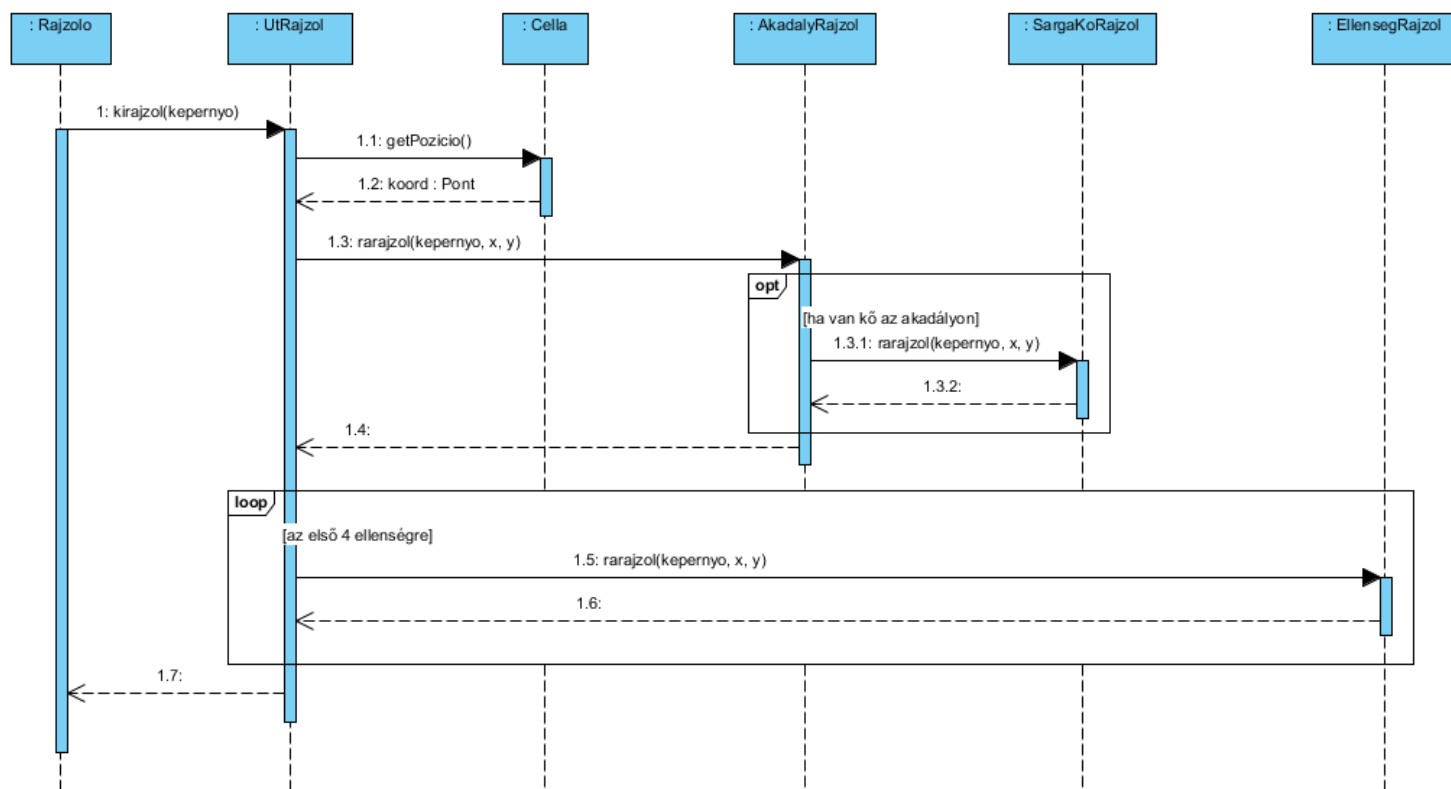
11.4.5 Játék elvesztése



11.4.6 Ellenség átlép



11.4.7 Rajzolás



11.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.04.23. 23:30	0,5 óra	Lipták Sebők Szepes Varga	Kilencedik értekezlet. Döntések: jelen dokumentum elkészítésének a felosztása.
2013.04.25. 16:00	1 óra	Sebők	11.1 Grafikus interfész elkészítése
2013.04.25. 17:00	1,5 óra	Lipták Sebők	Az MVC modell megtervezése
2013.04.26. 19:00	3 óra	Lipták	Az grafikus rendszer architektúrájának leírása, a strukturális diagram elkészítése.
2013.04.26. 20:30	2,5 óra	Szepes	11.3 alapjainak megírása
2013.04.27. 16:20	1,5 óra	Szepes	11.3 javítása a változott strukturális diagram alapján
2013.04.27. 23:00	4 óra	Lipták	Változtatások az architektúrában, javítások a strukturális diagramban, ellenőrzések
2013.04.27. 23:00	1 óra	Szepes	11.3 javítása a változott strukturális diagram alapján
2014.04.28. 01:00	4 óra	Varga	Kapcsolat az alkalmazói rendszerrel

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
img\akadaly.png	17 374 B	2014.04.26. 19:02	Akadály képe
img\domborzat.png	17 132 B	2014.04.26. 19:00	Domborzat képe
img\emberpirosko.png	17 491 B	2014.04.26. 19:04	EmberPirosKő képe
img\hobbitpirosko.png	17 463 B	2014.04.26. 19:09	HobbitPirosKő képe
img\kekko.png	17 404 B	2014.04.26. 19:03	KékKő képe
img\kod.png	18 204 B	2014.04.26. 19:20	Ködös torony képe
img\sargako.png	17 398 B	2014.04.26. 19:02	SárgaKő képe
img\torony.png	18 481 B	2014.04.26. 19:01	Torony képe
img\torppirosko.png	17 568 B	2014.04.26. 19:06	TörpPirosKő képe
img\tundepirosko.png	17 508 B	2014.04.26. 19:06	TündePirosKő képe
img\ut.png	17 129 B	2014.04.26. 19:00	Út képe
img\vegzethegye.png	18 691 B	2014.05.11. 20:52	VégzetHegye képe
img\zoldko.png	17 403 B	2014.04.26. 19:03	ZöldKő képe
src\Akadaly.java	1 174 B	2014.05.09. 00:22	Akadály osztály
src\AkadalyRajzol.java	1 187 B	2014.05.11. 21:13	AkadalyRajzol osztály
src\Cella.java	1 570 B	2014.04.21. 14:26	Cella osztály
src\Domborzat.java	905 B	2014.04.21.	Domborzat osztály

		19:14	
src\DomborzatRajzol.java	1 693 B	2014.05.11. 21:05	DomborzatRajzol osztály
src\Ellenseg.java	3 278 B	2014.05.11. 22:04	Ellenseg osztály
src\EllensegRajzol.java	1 722 B	2014.05.12. 00:50	EllensegRajzol osztály
src\Ember.java	1 787 B	2014.05.11. 22:07	Ember osztály
src\EmberPirosKo.java	428 B	2014.05.09. 00:25	EmberPirosKo osztály
src\EmberPirosKoRajzol.java	897 B	2014.05.11. 21:05	EmberPirosKoRajzol osztály
src\EmberRajzol.java	268 B	2014.05.11. 21:41	EmberRajzol osztály
src\GameController.java	14 478 B	2014.05.12. 00:33	GameController osztály
src\Hobbit.java	1 794 B	2014.05.11. 22:07	Hobbit osztály
src\HobbitPirosKo.java	434 B	2014.05.09. 00:25	HobbitPirosKo osztály
src\HobbitPirosKoRajzol.java	853 B	2014.05.11. 21:06	HobbitPirosKoRajzol osztály
src\HobbitRajzol.java	266 B	2014.05.11. 21:41	HobbitRajzol osztály
src\Jatek.java	8 106 B	2014.05.11. 21:44	Jatek osztály
src\KekKo.java	364 B	2014.04.21. 19:16	KekKo osztály
src\KekKoRajzol.java	837 B	2014.05.11. 21:06	KekKoRajzol osztály
src\Kirajzolható.java	448 B	2014.05.11. 18:40	Kirajzolható osztály
src\Pont.java	557 B	2014.05.11. 21:41	Pont osztály
src\Rajzolo.java	9 825 B	2014.05.11. 23:33	Rajzolo osztály
src\SargaKo.java	822 B	2014.05.09. 00:25	SargaKo osztály
src\SargaKoRajzol.java	811 B	2014.05.11. 21:41	SargaKoRajzol osztály

src\Torony.java	4 867 B	2014.05.11. 21:41	Torony osztály
src\ToronyKo.java	2 007 B	2014.05.09. 00:25	ToronyKo osztály
src\ToronyKoRajzol.java	406 B	2014.05.11. 19:38	ToronyKoRajzol osztály
src\ToronyRajzol.java	1 644 B	2014.05.11. 21:29	ToronyRajzol osztály
src\Torp.java	1 763 B	2014.05.11. 22:07	Torp osztály
src\TorpPirosKo.java	418 B	2014.05.09. 00:25	TorpPirosKo osztály
src\TorpPirosKoRajzol.java	847 B	2014.05.11. 21:06	TorpPirosKoRajzol osztály
src\TorpRajzol.java	263 B	2014.05.11. 21:41	TorpRajzol osztály
src\Tunde.java	1 779 B	2014.05.11. 22:07	Tunde osztály
src\TundePirosKo.java	425 B	2014.05.09. 00:25	TundePirosKo osztály
src\TundePirosKoRajzol.java	851 B	2014.05.11. 21:06	TundePirosKoRajzol osztály
src\TundeRajzol.java	266 B	2014.05.11. 21:40	TundeRajzol osztály
src\Ut.java	2 324 B	2014.05.11. 21:41	Ut osztály
src\UtRajzol.java	2 408 B	2014.05.11. 21:31	UtRajzol osztály
src\VegzetHegye.java	1 152 B	2014.04.21. 14:34	VegzetHegye osztály
src\VegzetHegyeRajzol.java	1 629 B	2014.05.11. 21:31	VegzetHegyeRajzol osztály
src\ZoldKo.java	360 B	2014.04.21. 19:20	ZoldKo osztály
src\ZoldKoRajzol.java	839 B	2014.05.11. 21:07	ZoldKoRajzol osztály
test\elagazas.map.txt	94 B	2014.04.18. 23:51	elagazas tesztelés térkép fájl
test\grafikus.map.txt	503 B	2014.05.12. 00:35	Grafikus felület teszteléséhez térkép fájl
test\kekko.map.txt	67 B	2014.04.18.	kekko tesztelés térkép fájl

		23:48	
test\kod.map.txt	192 B	2014.04.18. 23:50	kod teszteset térkép fájl
test\meghal.map.txt	160 B	2014.04.18. 23:50	meghal teszteset térkép fájl
test\palya.map.txt	85 B	2014.05.11. 22:14	palya teszteset térkép fájl
test\pirosko.map.txt	148 B	2014.04.18. 23:47	pirosko teszteset térkép fájl
test\torony.map.txt	157 B	2014.04.18. 23:43	torony teszteset térkép fájl
test\zoldko.map.txt	124 B	2014.04.18. 23:49	zoldko teszteset térkép fájl

13.1.2 Fordítás és telepítés

A követelményekben leírtaknak megfelelően a program fordításához egy olyan parancssori környezetre van szükség, melyben a Java 1.7-es fordítókörnyezet (**javac** program) elérhető.

(Egy parancssori ablakban a **javac** mappáját a PATH környezeti változóhoz a

```
set PATH=%PATH%;C:\Program Files\Java\jdk1.7.0_40\bin
```

paranccsal adhatjuk hozzá, az elérési útvonalat értelemszerűen átírva a saját környezetnek megfelelően.)

Egy ilyen parancssorban navigáljunk a projekt gyökérkönyvtárába és adjuk ki a

```
del bin\*.class
javac src\*.java -d bin -encoding utf8
```

parancsokat. (A figyelmeztetés a Java 6-os kompatibilitás miatt jelenik meg, attól a fordítás még sikeres.) A programot külön telepíteni nem szükséges, a lefordított fájlok azonnal futtathatók az alább leírtak szerint.

13.1.3 Futtatás

A követelményekben leírtaknak megfelelően a program futtásához egy olyan parancssori környezetre van szükség, melyben a Java 1.7-es futtatókörnyezet (**java** program) elérhető.

(Egy parancssori ablakban a **java** mappáját a PATH környezeti változóhoz a

```
set PATH=%PATH%;C:\Program Files\Java\jre7\bin
```

paranccsal adhatjuk hozzá, az elérési útvonalat értelemszerűen átírva a saját környezetnek megfelelően.)

Egy ilyen parancssorban navigáljunk a projekt gyökérkönyvtárába és adjuk ki a

```
java -cp bin GameController
```

parancsot.

A program használatához a térképállományok a `test` könyvtárban találhatóak, a játék indításkor a térkép elérési útja így `test/<teszteset>.map.txt` alakú lesz.

Az élvezhetőbb teszteléshez a prototípusból megmaradt pályákon kívül elkészítettük a `test/grafikus.map.txt` pályát is, ajánljuk ennek használatát.

13.2Értékelés

Tag neve	Munka százalékban
Lipták Levente	40%
Szepes Nóra	15%
Varga Gergő	15%
Sebők Márton	30%

13.3 Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.04.30. 18:00	0,5 óra	Lipták Sebők Szepes Varga	Tizedik értekezlet. Döntések: jelen dokumentum elkészítésének a felosztása.
2014.05.08. 20:00	3 óra	Sebők	Kontroller kódolása.
2014.05.10. 20:00	4,5 óra	Lipták	View kódolása.
2014.05.12. 20:00	3 óra	Sebők	Kódok javítása, dokumentáció elkészítése.

14. Összefoglalás

14.1A projektre fordított összes munkaidő, személyenként és összesen táblázatban

Név:	Munkaidő (óra):	Százalékosan:
Lipták Levente, HKTMJ1	46	26,59 %
Szepes Nóra, N2WIN2	40	23,12 %
Varga Gergő, KG3LTA	37	21,39 %
Sebők Márton, G78SPI	50	28,90 %
Összesítve:	173	

14.2A feltöltött programok forrásokainak száma fázisonként táblázatban

Fázis	Kód	Komment	Komment/Kód	Összes
Szkeleton	962	478	50%	1440
Prototípus	987	603	61%	1590
Grafikus	1537	1213	79%	2750

14.3Projekt összegzés

14.3.1 Mit tanultak a projektből konkrétan és általában?

A projekt során megtapasztaltuk a csapatban történő tervezett szoftverfejlesztést. Kis, négy fős csapatban dolgoztunk, de így is rengeteg hasznos tapasztalatot szereztünk. Legfontosabb tapasztalatunk a csapaton belüli információáramlás megoldása, valamint a projekt mind a dokumentumai, mind a forráskódjainak közösen való szerkesztése.

Mivel a projekt ütemterve szigorú, így rá voltunk kényszerítve a folyamatos, részletesen dokumentált munkára, ami szintén hasznos lesz a karrierünk során. A részletes tervezés leegyszerűsítette a tervezést és a követelmény megváltozásakor könnyen meg tudtuk változtatni a modellt is.

14.3.2 Mi volt a legnehezebb és a legkönnyebb?

A legnehezebb az analízis modell elkészítése, míg a legkönnyebb pedig a prototípus logikájának megvalósítása volt.

14.3.3 Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal? Ha nem, akkor hol okozott ez nehézséget?

A sok dokumentációt igénylő beadásoknál, főleg a félév második felében, már szűkösebbnek éreztük a határidőket. Ilyenkor nem jutott elég időnk a precíz munkára.

14.3.4 Milyen változási javaslatuk van?

A dokumentációk sablonjai lehetnének konkrétabbak, sok helyen nem érthetőek az elvárások.

14.3.5 Milyen feladatot ajánlanának a projektre?

Mindenképpen valamilyen játékot, hiszen a hallgatók mindig szívesebben foglalkoznak egy játék fejlesztésével.

14.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2014.02.19. 18:00	1 óra	Sebők	Eclipse fejlesztőkörnyezet összeállítása, Git és Visual Paradigm integrálása, kezdő projekt elkészítése
2014.02.20. 18:00	2,5 óra	Lipták Sebők Szepes Varga	Első értekezlet. Döntések: jelen dokumentum elkészítésének felosztása (alább látható), alapelvek megfogalmazása.
2014.02.20. 21:30	2,5 óra	Szepes	2.1 Bevezetés elkészítése
2014.02.20. 22:00	2 óra	Lipták Sebők	2.2 Áttekintés elkészítése
2014.02.21. 16:00	2 óra	Sebők	2.3 Követelmények és 2.4 Lényeges use-case-ek elkészítése
2014.02.21. 17:00	2 óra	Varga	2.6 Projekt terv elkészítése
2014.02.21. 23:00	1 óra	Lipták	2.5 Szótár elkészítése

2014.02.28. 19:30	1 óra	Lipták Sebők Szepes Varga	Második értekezlet. Döntések: jelen dokumentum elkészítésének felosztása
2014.02.28. 20:00	1 óra	Sebők	3.2 Statikus struktúra diagramok felvázolása
2014.02.28. 21:00	2 óra	Lipták Sebők	3.2 Statikus struktúra diagramok elkészítése
2014.03.01. 10:00	2 óra	Szepes	3.3 Osztályok leírásának elkezdése a statikus diagram alapján
2014.03.01. 20:00	2 óra	Szepes	3.3 Osztályok leírásának kiegészítése új attribútumokkal és metódusokkal
2014.03.01. 20:00	2 óra	Varga	3.4 Szekvencia diagramok elkészítése
2014.03.01. 23:00	1 óra	Sebők Szepes	3.2 Statikus struktúra diagramok finomítása, ezek alapján 3.3 Osztályok leírásának kiegészítése
2014.03.02. 00:30	1 óra	Varga Sebők	3.4 Szekvencia diagramok javítása
2014.03.02. 14:00	1 óra	Lipták	3.1 Objektum katalógus elkészítése
2014.03.02. 17:30	1 óra	Varga	3.4 Szekvencia diagramok javítása, pontosítása
2014.03.07. 19:00	2 óra	Lipták Sebők Szepes	Harmadik értekezlet. Döntések: jelen

		Varga	dokumentum elkészítésének felosztása
2014.03.08. 16:00	2 óra	Sebők	4.2 elkészítése
2014.03.09. 00:00	3 óra	Szepes	4.3 elkészítése, 4.2 apró hibáinak észrevétele
2014.03.09. 16:00	2 óra	Sebők	4.2 javítása, 4.4.4 elkészítése, 4.5 elkészítése
2014.03.09. 19:00	1.5 óra	Varga	4.4.1, 4.4.2, 4.4.3 elkészítése
2014.03.09. 23:00	1 óra	Varga	4.4 Javítások
2014.03.09. 23:30	4 óra	Lipták	4.1 javítása, 4.4.5, 4.4.6, 4.4.7 elkészítése
2014.03.10. 10:00	1 óra	Sebők	Javítások, készre formázás
2014.03.14. 19:00	0,5 óra	Lipták Sebők Szepes Varga	Negyedik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása.
2014.03.15. 18:30	2 óra	Szepes	5.3.1, 5.3.2, 5.3.3, 5.3.4 elkészítése
2014.03.16. 00:30	2 óra	Szepes	5.3.1, 5.3.2 javítása, use case leírások, 5.4.1.1, 5.4.1.2, 5.4.1.3 elkészítése
2014.03.16. 07:00	2,5 óra	Lipták	5.1.2.3, 5.1.2.4, 5.3.5, 5.3.6 elkészítése
2014.03.16. 11:30	2,5 óra	Varga	5.3.6, 5.3.7, 5.3.8, 5.3.9 elkészítése
2014.03.16. 15:30	2,5 óra	Varga	5.3.8, 5.3.9 javítása, 5.1.2.5, 5.1.2.6, 5.1.2.7, 5.1.2.8 elkészítése
2014.03.16. 20:30	1 óra	Lipták	Javítások, 5.4.1.4, 5.4.1.5 elkészítése
2014.03.16. 22:00	2 óra	Sebők	Többiek

			munkájának javítása, 5.1.1 Use-case diagram, 5.2 A szkeleton kezelői felületének terve, dialógusok, 5.1.2.8 Hobbit rálép a Végzet Hegyére, 5.3.10 Hobbit rálép a Végzet Hegyére elkészítése, 5.4 Kommunikációs diagramok kiegészítése
2014.03.17. 19:00	0,5 óra	Szepes	Osztály diagramból a program kódjának implementálása.
2014.03.19. 20:00	0,5 óra	Lipták Sebők Szepes Varga	Ötödik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása.
2014.03.20. 20:00	1 óra	Sebők	Szkeleton interfészének implementálása.
2014.03.22. 12:00	3 óra	Szepes	5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.10 use-casek, kommentelése, javítása
2014.03.22. 16:00	3 óra	Lipták	5.3.5, 5.3.6 use-casek implementálása,
2014.03.22. 22:00	4 óra	Varga	5.3.6, 5.3.7, 5.3.8, 5.3.9 use-casek implementálása
2014.03.23. 00:00	0,5 óra	Szepes	Fentebbi use-case diagramok hibáinak javítása
2014.03.23. 20:00	2 óra	Sebők	6.0 Szkeleton tervének javításai,

			6.1 Fordítási és futtatási útmutató elkészítése
2014.03.23. 20:00	0,5 óra	Varga	Javítások
2014.03.23. 23:00	1,5 óra	Lipták	Javítások, az elkészült részek mergelése
2014.03.28. 19:00	0,5 óra	Lipták Sebők Szepes Varga	Hatodik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása.
2014.03.29. 19:45	0,5 óra	Szepes	7.2 Elkészítése.
2014.03.30. 04:00	1,5 óra	Lipták	7.3 Elkészítése
2014.03.30. 09:00	0,5 óra	Szepes	7.2 Hibáinak javítása.
2014.03.30. 20:00	2,5 óra	Sebők	7.0 Specifikáció változás, 7.1.1 Az interfész általános leírása, térképájl formátum, 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása elkészítése
2014.03.30. 21:45	0,5 óra	Szepes	7.2 Hibáinak javítása
2014.03.30. 22:00	2 óra	Varga	7.1.2, 7.1.3 elkészítése.
2014.03.30. 23:00	0,5 óra	Lipták	7.3 javítása, kiegészítése.
2010.04.03. 19:00	0,5 óra	Lipták Sebők Szepes Varga	Hetedik értekezlet. Döntések: jelen dokumentum elkészítésének felosztása.
2014.04.04. 21:00	1,5 óra	Varga	8.1.4, 8.1.5, 8.1.7, 8.1.9, 8.1.15, 8.1.17 elkészítése
2014.04.04. 22:00	1,5 óra	Lipták	8.1.1 8.1.2 8.1.3

			8.1.11 8.1.12 8.1.19 8.1.20
2014.04.04. 21:30	1 óra	Szepes	8.1.6, 8.1.8, 8.1.10, 8.1.13, 8.1.14, 8.1.16, 8.1.18, 8.1.21 elkészítése.
2014.04.05. 20:00	3 óra	Sebők	8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén elkészítése
2014.04.05. 23:00	0,5	Lipták	Javítások, ellenőrzések
2014.04.06. 11:30	0,5 óra	Szepes	8.1.6, 8.1.8, 8.1.10, 8.1.13, 8.1.14, 8.1.16, 8.1.18, 8.1.21 formázása, apróbb hibák javítása
2014.04.07. 10:00	0,5 óra	Sebők	Javítások.
2014.04.17. 18:00	0,5 óra	Lipták Sebők Szepes Varga	Nyolcadik értekezlet. Döntések: jelen dokumentum elkészítésének és a kódolásnak a felosztása.
2014.04.18. 20:00	2 óra	Sebők	Main és Jatek osztályok átírása, a szkeleton-specifikus részek eltávolítása, a tesztelési és fordítási keretrendszerhez a batch-fájlok megírása, a tesztesetek átültetése fájlokba.
2014.04.18. 22:00	1 óra	Szepes	Torony és toronykövek implementálása, szkeleton kódjának

			eltávolítása
2014.04.19. 17:00	0,5 óra	Szepes	Torony és toronykövek javítása, kiírások implementálása
2014.04.20. 22:00	1.5 óra	Lipták	A Pont, Cella, Ut, Vegzethegye, Akadaly, Sargako, Domborzat osztályok implementálása
2014.04.21. 10:30	1,5 óra	Varga	Ellenség és leszármazottainak implementálása
2014.04.21. 14:00	0,5 óra	Szepes	Torony és toronykövek javítása, objektumazonosítók felvétele
2014.04.21. 18:00	2 óra	Sebők	A többi kód és a tesztesetek javítása.
2014.04.21. 20:00	2 óra	Sebők	A dokumentáció megírása.
2014.04.23. 23:30	0,5 óra	Lipták Sebők Szepes Varga	Kilencedik értekezlet. Döntések: jelen dokumentum elkészítésének a felosztása.
2013.04.25. 16:00	1 óra	Sebők	11.1 Grafikus interfész elkészítése
2013.04.25. 17:00	1,5 óra	Lipták Sebők	Az MVC modell megtervezése
2013.04.26. 19:00	3 óra	Lipták	Az grafikus rendszer architektúrájának leírása, a strukturális diagram elkészítése.
2013.04.26. 20:30	2,5 óra	Szepes	11.3 alapjainak megírása
2013.04.27. 16:20	1,5 óra	Szepes	11.3 javítása a

			változott strukturális diagram alapján
2013.04.27. 23:00	4 óra	Lipták	Változtatások az architektúrában, javítások a strukturális diagramban, ellenőrzések
2013.04.27. 23:00	1 óra	Szepes	11.3 javítása a változott strukturális diagram alapján
2014.04.28. 01:00	4 óra	Varga	Kapcsolat az alkalmazói rendszerrel
2014.04.30. 18:00	0,5 óra	Lipták Sebők Szepes Varga	Tizedik értekezlet. Döntések: jelen dokumentum elkészítésének a felosztása.
2014.05.08. 20:00	3 óra	Sebők	Kontroller kódolása.
2014.05.10. 20:00	4,5 óra	Lipták	View kódolása.
2014.05.12. 20:00	3 óra	Sebők	Kódok javítása, dokumentáció elkészítése.
2014.05.12. 20:00	0,5 óra	Lipták Sebők Szepes Varga	Tizenegyedik értekezlet. Döntések: jelen dokumentum elkészítésének a felosztása.
2014.05.12. 22:00	2 óra	Szepes	Dokumentum vázlatának elkészítése. Naplófájl kiegészítése, óraszám és százalékok kiszámítása. 14.4 Féléves napló elkészítése, összegzések

			begépelése
2014.05.12. 23:00	0,5 óra	Varga	Dokumentumok összeolvasztása, 14.2 elkészítése