# Teljes dokumentáció

## 77 – WildPointers

**Konzulens:**

## dr. László Zoltán

Csapattagok:


| | | |
|---|---|---|
| Bana Szabolcs | GPNWUV | banas0021@gmail.com |
| Czirják Balázs | Y1UEOT | balazs.czirjak@gmail.com |
| Csontos Valentin | NTR8BN | csontos.valentin@gmail.com |
| Tóth Dávid | CN3Y8L | adtoth13@gmail.com |
| Török Odett | O1JZM3 | odett.trk@gmail.com |


2014. 05. 16. WildPointers

# Tartalomjegyzék

# 2. Követelmény, projekt, funkcionalitás

## 2.1 Bevezetés

### 2.1.1 Cél

A dokumentum elsődleges célja a követelmények megfogalmazása a szoftverrel szemben. Emellett fontos még a programfejlesztés követhetősége és az elkészített szoftver későbbi karbantarthatóságának támogatása. Megalkotása során fontos szempont egy adott csapattag/fejlesztő helyettesíthetősége, nélkülözhetősége – hasonlóan az éles helyzethez egy munkahelyen: ha a projekt valamelyik tagját elbocsájtják, nélkülözhető legyen az adott személy és nélküle is gördülékenyen folytatódhasson a fejlesztés.
A dokumentációkészítés során törekszünk arra is, hogy minden pontján a megfelelő absztrakciós szinten részletezzük az aktuális pontokat. Ez is a követhetőséget és a továbbfejleszthetőséget szolgálja.

### 2.1.2 Szakterület

Az elkészített szoftverrel a játékok kedvelőit célozzuk, pontosabban a Tower Defense típusú játékok fanatikusainak kívánunk ezzel egy új szórakozási lehetőséget biztosítani. A játék otthoni felhasználásra készül, de kellemes szórakozást nyújt az irodában is.

### 2.1.3 Definíciók, rövidítések

| Definíció | Jelentés |
|---|---|
| JRE | Java Runtime Environment<br>Java nyelven írt programok futtatására szolgál |
| JDK | Java Development Kit - Java fejlesztői környezet |

| | |
|---|---|
| JVM | Java Virtual Machine - Java Virtuális Gép - a lefordított byte kód futtatóeszköze. |
| Java SE | Java Standard Edition |
| MVC | Model-View-Controller szerkezeti minta |
| Követelmény | A szoftverrel szemben támasztott elvárások, igények és korlátozások. |
| Aktor | Use-case-ben szereplő elem, ami egy szerepkört reprezentál. Lehet valós felhasználó vagy egy belső motor. |
| Use-case | Elemi funkcionalitás, ami a szereplők cselekedeteit reprezentálja. |
| Use-case diagram | Statikus képet ad arról, hogy az aktorok és az egyes use-case-ek között milyen kapcsolat van. |

### 2.1.4 Hivatkozások

A tantárggyal és a feladattal kapcsolatos elvárások az alábbi oldalakon tekinthetőek meg:
https://www.iit.bme.hu/~szoftlab4/
https://www.iit.bme.hu/~szoftlab4/feladat.shtml

Felhasznált eszközök:

| **Eszköz neve** | **Elérhetősége** |
|---|---|
| GitHub | https://github.com/ |
| Eclipse | https://www.eclipse.org/ |
| Google Drive | https://drive.google.com/ |
| Microsoft Office | http://office.microsoft.com/ |
| Microsoft Visio | http://visio.microsoft.com/ |
| TortoiseSVN | http://tortoisesvn.net/ |

| | |
|---|---|
| Microsoft Project | http://office.microsoft.com/hu-hu/project/ |
| JDK és JRE | http://docs.oracle.com/javase/7/docs/webnotes/install/windows/windows-system-requirements.html |
| Skype | http://www.skype.com |
| Facebook group | http://facebook.com |

### 2.1.5 Összefoglalás

A következő részek funkciója alapvetően a megkapott specifikáció alapján egy alacsonyabb absztrakciós szinten lévő követelményrendszer megfogalmazása - akárcsak egy életciklus modellt követő szoftverfejlesztéskor.
A dokumentáció további részében felvázoljuk az általunk megvalósítandó programot. Majd ezt követően áttekintjük a célközönség szerepét, tulajdonságait a program működésében illetve szót ejtünk a szoftverre vonatkozó funkciókról, korlátozásokról. A fejezetet a dokumentumban használt anyagok, web-oldalak felsorolása zárja. Ezt követi a különböző követelmények táblázatos formában történő részletezése. Tüzetesen taglalva a funkcionális, erőforrással kapcsolatos, nem funkcionális és az átadással kapcsolatos követelményeket. Folytatásként az elemi funkcionalitásokat vázoljuk használati eset leírások segítségével, először táblázatos formában, majd use-case diagram használatával. A követelményekben használt, további magyarázatra szoruló kifejezéseket a szótár segítségével tesszük egyértelművé. A dokumentum a 2.6-os projekt terv ponttal zárul, amelyben a megvalósításhoz szükséges lépések, határidők, a szerepkörök kijelölése, a tagok beosztása majd a folyamatban felhasznált fejlesztői eszközöket ismertetjük. Emellett, minden dokumentum végén megtalálható lesz az úgynevezett Napló, amiben részletesen napra és feladatra lebontva nyomon követjük a csapattagok munkáját.

## 2.2 Áttekintés

### 2.2.1 Általános áttekintés

A szoftver három nagy alrendszerre különül el: adatmodell, vezérlés és megjelenítés (MVC architektúra). Az egyes alrendszerek kapcsolatban állnak egymással. Az adatmodell elsődleges feladata az adatok tárolása illetve módosítása. A megjelenítés felelős az adatmodell grafikus megjelenítéséért, ezt látja majd a felhasználó. A vezérlés felelős a felhasználói interakciók kezeléséért. Felhasználói beavatkozások hatására a vezérlés manipulálja az adatmodellt és az adatmodell frissíti a megjelenítés modult.
Szoftverünk egy offline Java SE alkalmazás lesz.

### 2.2.2 Funkciók

#### 2.2.2.1. A Játék célja és kimenetelei

A játék során a játékos egyetlen célja a pályára elhelyezhető tornyokkal és akadályokkal megelőzni azt, hogy az ellenségek eljussanak a pályán (Mordor földjén) elhelyezkedő Végzet hegyéhez. Az ellenségek a játék előrehaladtával egyre nagyobb csoportokban, hullámokban érkeznek, ezzel nehezítve a felhasználó esélyeit a játék sikeres teljesítésére.

Az ellenséges csoportok a pályán előre kijelölt utakon haladnak, amiről nem térhetnek le, és aminek a végén helyezkedik el a hegy (Végzet hegye). Egyetlen célkitűzésük ide elérni, és ha ez bekövetkezik, tehát akár egyetlen ellenség is túléli a tornyok támadását és végighalad az egyik hegyhez vezető úton ezzel elérve a Végzet Hegyét az Egy Gyűrű végérvényesen és visszafordíthatatlanul megsemmisül. Ekkor a játék véget ér, a játékos veszít, mivel nem tudta ezt megakadályozni. A játék során meghatározott számú ellenséget kell legyőznünk. Ha az összes ellenséget sikeresen megsemmisítettük, akkor a játéknak vége, a játékos nyer és az Egy Gyűrű megmenekül, így Szarumán diadalmaskodik a Gyűrű Szövetség felett.

#### 2.2.2.2. Védekezés

A játékosnak kétféle védekezés áll rendelkezésére:

- tornyok, melyeket az út mentén helyezhet el
- akadályok, melyeket az útra tehet

A tornyok adott hatótávolságon belül adott erősségű és mennyiségű lőszert lőnek ki a Gyűrű szövetségeseire. Ezeket az értékeket lehetséges növelni varázskövekkel, ezekhez a játékosnak a varázserejét kell felhasználnia. Egy torony egyszerre egy ellenséget támad és

mindig azt, amelyik a legközelebb van a Végzet hegyéhez. Az akadály, amit az útra helyezhetünk a fejlettségétől függően lassítja a rajta áthaladó ellenséges egységet. Fejleszteni ezt is varázskővel tudjuk. Egy akadály a rajta tartózkodó ellenséget lassítja.

#### 2.2.2.3. Varázserő

Kezdőállásban a játékos rendelkezik egy kezdeti varázserővel, amiért cserébe a játék során elhelyezheti, majd fejlesztheti a tornyait és akadályait, ezzel hatékonyabban visszaverve a hullámokban érkező ellenséges egységeket.

A 7 féle varázskő, ami közül választhat:

- hatótávolság
- tüzelési gyakoriság
- speciális képességek:
  - tündés
  - törpés
  - hobbitos
  - emberes
- lassítás

A varázskő vásárlásához az elpusztított ellenfelekért kapott varázserő szükséges. Az egyes varázskövekhez szükséges varázserő azonos lesz.
A varázskő nevéből egyértelműen következtethetni lehet a toronyra gyakorolt hatására.
Fontos megjegyezni, hogy egy varázskő egyszer használható fel, valamint a torony tulajdonságai közül csak egyre van pozitív hatással. Egy toronyra viszont több kő is elhelyezhető.

#### 2.2.2.4. Ellenségek

Négy fajta ellenséggel találkozhatunk a játékban, akik a Gyűrű Szövetségét alkotják és a Végzet Hegyének megsemmisítésére esküdtek fel. Ezek a következők:

- tündék
- törpék
- hobbitok
- emberek

Egy-egy fajjal szemben növelhető a torony képessége ezekkel a bizonyos varázskövekkel.

Ezen kívül az ellenségeknek van bizonyos értékű életenergiája, amit a tornyok minden - az adott ellenségre irányuló - lövése csökkent, és ha eléri a nullát, akkor elpusztulnak. Minden elpusztított ellenség után növekszik a játékos által birtokolt varázserő.

### 2.2.3 Felhasználók

A játékszoftvert átlag felhasználóknak szánjuk. Feltételezzük, hogy szórakozás és kikapcsolódás céljából használják a szoftvert. Bonyolultságát tekintve a telepítéséhez és kezeléséhez semmilyen komolyabb szoftveres előképzettség nem szüksége szükséges.

### 2.2.4 Korlátozások

Előírások és korlátozások:

- A játékszoftver futtatásához szükséges a megfelelő verziójú JRE telepítése.
- A játék irányításához szükséges egér (mouse) kontroller, mivel a játékban elérhető összes funkció egérrel érhető el.
- JVM: A futtató hardveren beépített Java Virtual Machine megléte szükséges.

### 2.2.5 Feltételezések, kapcsolatok

A dokumentumban lexikális információkat különböző hiteles forrásokból szedjük össze. Ezek az alábbiak:

- Tárgyhonlap: www.iit.bme.hu/~szoftlab4
- Wikipedia: www.wikipedia.org
- Szofvertechnológiák tárgyweboldal: http://directory.iit.bme.hu/belso/st/stbelso.html
- Java API dokumentáció: http://docs.oracle.com/javase/7/docs/api/
- Adatszerkezethez XML: http://www.w3.org/

## 2.3 Követelmények

### 2.3.1 Funkcionális követelmények

| Azonosító | Leírás | Ellenőrzés | Prioritás | Forrás | Use-case | Komment |
|---|---|---|---|---|---|---|
| FK_01 | Az ellenségek csak az előre megadott úton haladhatnak. | Bemutatás | Alapvető | Megrendelő | Ellenség | Mordor földje |
| FK_02 | Az ellenségek folyamatosan jönnek, de számuk véges. | Bemutatás | Alapvető | Megrendelő | Ellenség | |
| FK_03 | Az ellenségek csak előrefelé haladhatnak. | Bemutatás | Fontos | Csapat | Ellenség | Ellenség előrefelé haladása szótárban van értelmezve. |
| FK_04 | Egy ellenség bizonyos mennyiségű sebzés hatására elpusztul. | Bemutatás | Alapvető | Megrendelő | Ellenség Védekezés | |
| FK_05 | Egy ellenség akadályra lépéskor lelassul. | Bemutatás | Alapvető | Megrendelő | Ellenség, Akadály | Lassítás értelmezve van a szótárban. |
| FK_06 | Ellenség életereje sebzés hatására csökken. | Bemutatás | Alapvető | Megrendelő | Ellenség Védekezés | |
| FK_07 | Ellenség elpusztul, ha életereje a sebzések hatására 0 lesz. | Bemutatás | Fontos | Csapat | Ellenség | |
| FK_10 | Nem beépíthető | Bemutatás | Fontos | Csapat | Torony és | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | csempére nem lehet tornyot illetve akadályt elhelyezni. | | | | akadály | |
| FK_11 | Nem szabad út-csempére is lehet akadályt tenni. | Bemutatás | Opcionális | Csapat | Védekezés | |
| FK_12 | Az egyazon hullámon belüli ellenségek szinkronban mozognak. | Bemutatás | Fontos | Csapat | Ellenség | |
| FK_13 | Egy ellenség elpusztulása során bizonyos mértékben nő a játékos varázsereje. | Bemutatás | Alapvető | Csapat | Varázserő Torony | |
| FK_20 | Az út mellett tornyok helyezhetőek el. | Bemutatás | Alapvető | Megrendelő | Torony | |
| FK_21 | Torony csak a hatótávon belüli ellenséget képes megsebezni. | Bemutatás | Alapvető | Megrendelő | Torony | |
| FK_22 | Torony a hatótávolságán belül a Végzet hegyéhez legközelebb eső ellenséget sebzi meg. | Bemutatás | Fontos | Csapat | Torony | |
| FK_23 | A tornyok és akadályok építéséhez | Bemutatás | Alapvető | Megrendelő | Torony, Akadály, Varázserő | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | meghatározott számú varázserő kell. | | | | | |
| FK_24 | Toronyra elhelyezett varázskő növeli a torony hatótávolságát, tüzelés gyakoriságát vagy a lövedékek sebzési erejét. | Bemutatás | Alapvető | Megrendelő | Torony, Varázskő | |
| FK_25 | Akadályra elhelyezett varázskő tovább lassítja az akadályon keresztülhaladó ellenséget. | Bemutatás | Fontos | Csapat | Akadály Varázskő | |
| FK_26 | Varázserő felhasználásával varázskövet is tudunk létrehozni. | Bemutatás | Alapvető | Megrendelő | Varázserő Varázskő | |
| FK_30 | A játék akkor ér véget, ha egy ellenség eljut a Végzet hegyéhez vagy sikerült az összes ellenséget elpusztítani. | Bemutatás | Alapvető | Megrendelő | Játék indítása | |
| FK_31 | Játék indítása során a program kezdő állásba kerül. | Bemutatás | Alapvető | Megrendelő | Játék indítása | Kezdő állás szótárban megtalálható. |

### 2.3.2 Erőforrásokkal kapcsolatos követelmények

| Azonosító | Leírás | Ellenőrzés | Prioritás | Forrás | Komment |
|---|---|---|---|---|---|
| EK_01 | Pentium 2266 MHz processzor | Kiértékelés | Alapvető | Csapat | JRE és JDK minimális követelménye |
| EK_02 | ~400 MB lemezterület | Kiértékelés | Alapvető | Csapat | JRE és JDK együttes minimális követelménye |
| EK_03 | ~100 MB | Kiértékelés | Alapvető | Csapat | Szoftver fejlesztése során létrejövő fájlok mérete |
| EK_04 | 128 MB memória | Kiértékelés | Alapvető | Csapat | JRE és JDK minimális követelménye |
| EK_05 | Egér | Kiértékelés | Alapvető | Csapat | Fejlesztéshez, Játék funkcióihoz |
| EK_06 | Billentyűzet | Kiértékelés | Alapvető | Csapat | Fejlesztéshez |
| EK_07 | Videókártya | Kiértékelés | Alapvető | Csapat | Futtatáshoz |
| EK_08 | Monitor | Kiértékelés | Alapvető | Csapat | Futtatáshoz, fejlesztéshez |
| EK_09 | Internet | Kiértékelés | Fontos | Csapat | Fejlesztéshez |
| EK_10 | Operációs rendszer | Kiértékelés | Alapvető | Csapat | Futtatáshoz, Fejlesztéshez |
| EK_11 | Java Runtime Environment | Kiértékelés | Alapvető | Csapat | Futtatáshoz, Fejlesztéshez |
| EK_12 | Java Development Kit | Kiértékelés | Alapvető | Csapat | Fejlesztéshez |
| EK_13 | Eclipse | Kiértékelés | Opcionális | Csapat | Fejlesztéshez |
| EK_14 | Microsoft Office | Kiértékelés | Opcionális | Csapat | Dokumentáláshoz |

### 2.3.3 Átadással kapcsolatos követelmények

| Azonosító | Leírás | Ellenőrzés | Prioritás | Forrás | Komment |
|---|---|---|---|---|---|
| ÁK_01 | HSZK gépen fut | Bemutatás | Alapvető | Megrendelő | |
| ÁK_02 | Felhasználói útmutató | Bemutatás | Fontos | Megrendelő | |

### 2.3.4 Egyéb nem funkcionális követelmények

| Azonosító | Leírás | Ellenőrzés | Prioritás | Forrás | Komment |
|---|---|---|---|---|---|
| EN_01 | Program hordozhatósága | Bemutatás | Alapvető | Csapat | Minden, a program által használt fájl rendszerfüggetlenül elérhető |
| EN_02 | Operációsrendszer függetlenség | Bemutatás | Alapvető | Csapat | Java |
| EN_03 | A felhasználó rendelkezzen alapvető számítógép használati képességgel | Bemutatás | Alapvető | Csapat | |
| EN_04 | Funkcionalitás (Functionality) | Bemutatás | Alapvető | Csapat | A program teljesíti a megrendelő által támasztott igényeket, funkciókat. |

## 2.4 Lényeges use-case-ek

### 2.4.1 Use-case leírások

| | |
|---|---|
| **Use-case neve** | Játék indítása |
| **Rövid leírás** | Felhasználó elindítja a játékot. |
| **Aktorok** | Felhasználó |
| **Forgatókönyv** | A játék kezdő állásba kerül. |

| | |
|---|---|
| **Use-case neve** | Védekezés |
| **Rövid leírás** | A játékos számára elérhető védelmi eszközök használata. |
| **Aktorok** | Felhasználó |
| **Forgatókönyv** | Meghatározott számú varázserő segítségével védelmi eszközök elhelyezése a pálya bizonyos részein. |

| | |
|---|---|
| **Use-case neve** | Varázserő használat |
| **Rövid leírás** | A játékos számára elérhető számérték, amit védekezésre használhat fel. |
| **Aktorok** | Felhasználó |
| **Forgatókönyv** | Felhasználásával tornyokat és akadályokat lehet építeni. |

| **Use-case neve** | Torony építés |
|---|---|
| **Rövid leírás** | Védelmi tornyok elhelyezése a pályán. |
| **Aktorok** | Felhasználó |
| **Forgatókönyv** | A torony hatósugarába eső ellenségeket tüzeli bizonyos gyakorisággal. |

| **Use-case neve** | Akadály építés |
|---|---|
| **Rövid leírás** | Akadályok elhelyezése a pályán. |
| **Aktorok** | Felhasználó |
| **Forgatókönyv** | Az akadály területén belül mozgó ellenséget lelassítja. |

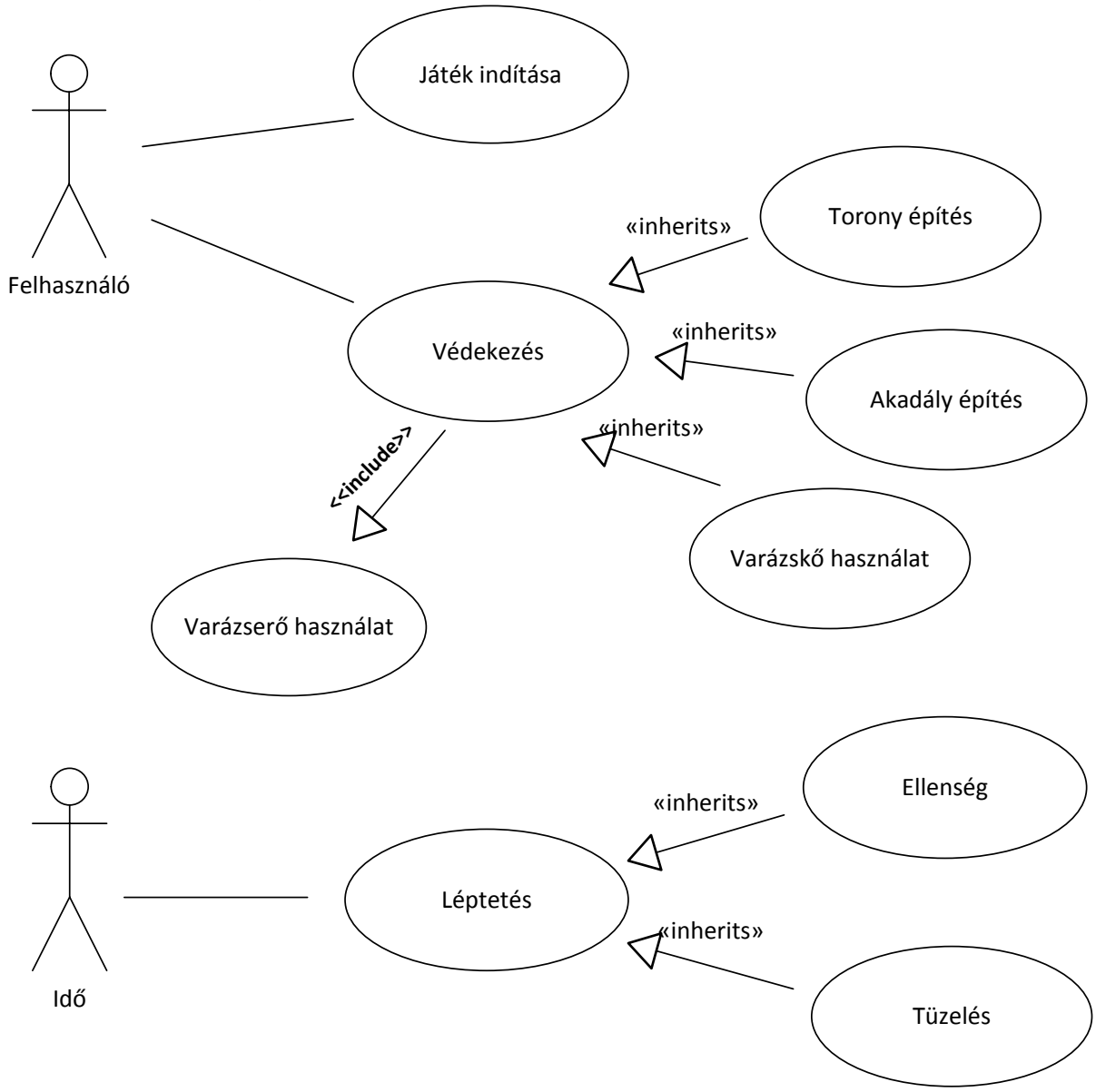
| **Use-case neve** | Varázskő használat |
|---|---|
| **Rövid leírás** | A védelmi tornyok és az akadályok megerősítésére szolgál. |
| **Aktorok** | Felhasználó |
| **Forgatókönyv** | Már meglévő tornyokat és akadályokat ruházza fel speciális képességgel. Egy varázskő csak egyszer használható fel és varázserő fejében lehet beszerezni. |

| **Use-case neve** | Léptetés |
|---|---|
| **Rövid leírás** | Idő kezelése |
| **Aktorok** | Idő |
| **Forgatókönyv** | Az idő a játék elindulásától számolódik és ennek függvényében megy végbe az animáció. |

| **Use-case neve** | Ellenség |
|---|---|
| **Rövid leírás** | Az ellenség léptetése az idő függvényében történik. Egységnyi idő alatt a léptetések száma csak nőhet az idő múlásával. |
| **Aktorok** | Idő |
| **Forgatókönyv** | Az ellenségek mindig csak előrefelé, az úton haladva, egyazon hullámon belül egymással szinkronban mozognak a Végzet hegye felé. |

| **Use-case neve** | Tüzelés |
|---|---|
| **Rövid leírás** | Minden torony tud tüzelni bizonyos tüzelési gyakorisággal és hatótávval. |
| **Aktorok** | Idő |
| **Forgatókönyv** | Egy torony hatósugarába eső területen megsebzi a célhoz legközelebb lévő ellenséget. |

### 2.4.2 Use-case diagram



## 2.5 Szótár

**Akadály**: Játékos út-csempére építhető védelmi eszköze, ami lassítja a csempére rálépő ellenség haladási sebességét.

**Akadály építés**: Játékos által egy új akadály elhelyezése az út olyan csempéjére, amely beépíthető.

**Beépíthető csempe**: Olyan csempe, amire lehet építeni.

- út esetében, amin még nincs akadály, forrás és hegy, de út
- torony esetében, amin még nincs semmi, csak egy üres csempe

**Csempe/Tile**: A játéktér csempékből épül fel, ez egy egységnyi méretű pálya elem, amin elhelyezhető például egy torony vagy egy akadály.

**Egy Gyűrű**: A megsemmisítendő tárgyat jelöli, ami a Végzet Hegyénél található.

**Életerő**: Az ellenségre jellemző számérték, ami csak csökkenhet, és nullára csökkenés esetén az ellenség elpusztul.

**Ellenség**: Ellenséges lény, melynek célja, hogy eljusson a toronyhoz és elpusztítsa a gyűrűt (emberek, tündék, hobbitok, törpök). Véges szám van belőle. Egyre nagyobb számban érkeznek a játék előrehaladtával, amíg el nem fogynak. Csak út-csempén haladhatnak.

**Ellenség előrefelé haladása:** Egy ellenség léptetésének sorozata.

**Ellenségek forrása**: Olyan csempe, ahol ellenség létrejön, és automatikusan elindul a Végzet Hegyének irányába vezető úton.

**Ellenség léptetése:** Egy adott út-csempén tartózkodó ellenség egy szomszédos út-csempére történő lépése. Ha több szomszédos út-csempére is léphet, akkor véletlenszerűen választ.

**Elpusztul**: Az ellenség életereje nullára csökken és elpusztul. Az ellenség eltűnik a pályáról, további interakciója nincs. A felhasználó varázsereje növekszik.

**Építés**: (elhelyezés szinonimaként): Torony vagy akadály létrehozásának folyamata, amit a játékos (Szarumán) kezdeményez.

**Fejlesztés** (felruházással szinonimák): A torony felruházása egy bizonyos ellenséggel szemben speciális képességgel, magasabb tüzelési gyakorisággal vagy magasabb hatótávolsággal.

**Fejlettség**: Egy torony fejlesztés által elért képességeinek száma.

**Győzelem**: A játékos elpusztította az összes ellenséget mielőtt a Végzet Hegyéhez ért volna, így megvédte az Egy Gyűrűt.

**Haladás:** Egy ellenség Végzet Hegyének irányába tett pozícióváltoztatás.

**Hatótávolság**: Egy toronynak azon sugarú környezete, amelyen belül az ellenséget sebezni tudja.

**Hullám**: Meghatározott időközönként érkező, meghatározott számú, logikailag összefüggő ellenségek csoportja.

**Játékos**: A felhasználó szinonimája.

**Játéktér**: A felhasználó által látható terület, ami csempékből épül fel.

**Kezdőállás**: A játék legelső pillanata, amikor még a játékosnak (Szarumán) nincsenek tornyai és akadályai a pályán, és varázsereje egy kezdőértéket vesz fel.

**Képesség:** A tüzelési hatékonyság, a hatótávolság és a speciális képesség gyűjtőneve.

**Lassítás:** Egy adott ellenség haladásának visszafogása oly módon, hogy a léptetések során kimarad bizonyos számú lépésből.

**Lövedék**: A torony által a legközelebbi ellenségre periodikusan kilőtt lőszer, ami az ellenséget sebzi a torony tüzelési hatékonyságával arányos módon.

**Megsemmisülés/vereség**: Egy ellenfél elért a Végzet hegyéhez.

**Mordor földje**: Az ellenfél mozgásterülete.

**Nehézségi szint**: Az idő előrehaladtával folyamatosan növekvő érték, ami az aktuálisan induló hullámmal érkező ellenségek számát jelöli. Hullámok egyre gyakrabban indulnak.

**Sebzés**: Az ellenség életerejének lövedék által okozott csökkenése.

**Speciális képesség**: A torony azon képessége, hogy egy adott ellenségtípust magasabb mértékben sebez.

**Szarumán**: A játékost reprezentáló karakter, képes tornyot építeni az utak mellé, akadályokat elhelyezni az utakon, és mind a tornyokat, mind az akadályokat felruházni varázskövekkel.

**Torony**: A játékos által elhelyezhető építmény, ami védekezésre szolgál.

**Toronyépítés**: Játékos által egy új torony elhelyezése beépíthető csempére.

**Torony fejlesztései**: A torony tüzelési gyakoriságának, hatótávolságának és speciális képességének gyűjtőneve.

**Tüzelés**: A lövedék egy adott, hatótávolságon belül lévő ellenségre való elindítása.

**Tüzelési hatékonysága**: A torony tüzelési gyakoriságának, speciális képességének és varázskő által felruházott többletsebzésnek az összege.

**Tüzelési gyakoriság**: Torony által egységnyi idő alatt kilőtt lövedékek száma.

**Út**: Az ellenségek forrása és a Végzet hegye között csempékből álló egyértelmű útvonal, ahol az ellenség haladhat. Lehetnek több forrás és elágazás is előfordulhat.

**Varázserő**: A játékos (Szarumán) egy számmal kifejezhető tulajdonsága, amit építésre vagy varázskő használatára fordíthat. Értéke egy-egy ellenség elpusztulásakor növekszik.

**Varázskő**: A játékos által birtokolt tárgy, amit felhasználhat tornyai vagy akadályai fejlesztésére. Torony felruházása varázskővel megfelelő mennyiségű varázserő birtokában lehetséges.

**Végzet Hegye**: Egy olyan mező, amit ha egy ellenfél elér, akkor a játék vereséggel véget ér. A csempe, amin el van helyezve nem beépíthető (más nem kerülhet rá).

# 2.6 Projekt terv

## 2.6.1 Végrehajtás lépései

### 2.6.1.1 Fejlesztés fő mérföldkövei

A feladatot megvalósítása három nagy részből fog állni:

- szkeleton
- prototípus
- grafikus felület

#### 2.6.1.1.1 Szkeleton

A változat funkciója annak a szemléltetése, hogy az objektum és a dinamikus modellek a kiadott feladat hiteles modelljét alkotják. Ebben a változatban már az összes, a végső programban is szereplő objektum megtalálható. Az objektumoknak (osztályoknak) egyelőre csak az interfészüket készítjük el.
A szkeletonnak képesnek kell lennie a különböző forgatókönyvek és szekvencia diagramok ellenőrizhetőségére.
Az egyes metódusok mindössze annyit csinálnak majd, hogy kiírják a saját nevüket, és meghívják azokat a további metódusokat, ami az adott funkció ellátásához szükséges.

#### 2.6.1.1.2 Prototípus

A változat az elkészült program, kivéve a GUI. A program célja, hogy bemutassa, hogy a program működik, azaz teljesíti az összes, vele szemben támasztott követelményt, ellátja az összes elvárt funkciót. Ezen a ponton az objektumok függvényei már a végleges algoritmusokat tartalmazzák.
Fontos, hogy a megjelenítés egy fájlban naplózható, ezzel támogatva a rendszer tesztelhetőségét. A működtetés illetve megjelenítés alfanumerikus képernyőn látható.

#### 2.6.1.1.3 Grafikus felület (GUI)

A szoftver végső, elkészült verziója, ami egy grafikus felülettel felruházott változat lesz.

### 2.6.2 A fejlesztés részletesebb lépései, határidők

| 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 |

### 2.6.3 A fejlesztőcsapat tagjai, felosztott feladatkörök

A táblázatban enumeráljuk, hogy az egyes csapattagok a projektmunka mely feladatköreire, fázisaira fordítják leginkább az erőforrásaikat. Egy nagyobb projekt eredményes abszolválása nagyban múlik a csapattagok hatékony együttműködésén, ezért a csapatunkon belül mindenki tisztában van azzal, hogy egymás folyamatos segítése, kiegészítése, helyenként felülvizsgálása/korrigálása a cél. Az itt feltüntetett feladatkörök iránymutatóak, és inkább egyfajta felelősségkört jelölnek ki az adott csapattagunkra

vonatkozóan.

| Név | Feladatkör |
|---|---|
| Bana Szabolcs | Tervezés, UML, implementálás |
| Czirják Balázs | munka összehangolása, UML |
| Csontos Valentin | Csapatvezetés, tervezés, implementálás, tesztelés |
| Tóth Dávid | Implementálás, tesztelés, UML |
| Török Odett | Dokumentáció, tesztelés, UML |

### 2.6.4 Kommunikáció a csapaton belül

A csapatunk tagjai mind különböző helyen élnek, így a projekt során folyamatos találkozókra lesz szükségünk. Ez fontos momentum a hatékony feladatfelosztáshoz, a számonkéréshez és további finomításhoz. Komoly összehangolást igényel, amit a csapat minden tagja tudomásul vett.
Egy találkozó az aktuálisan leadandó dokumentumtól függően kétféle lesz: személyes és virtuális.

Személyes találkozóra olyankor lesz szükség, amikor hosszútávú döntéseket hozunk, illetve előfordulhat, hogy a vizuális kapcsolat is szükséges. Ilyen például a szótár megalkotása, a modellezés, vagy a prototípus koncepciójának elkészítése.

Virtuális találkozó elegendő azokban a helyzetekben, amikor a felmerülő kérdések és a meghozandó döntések a WBS-ből (Work Breakdown Structure) kifolyólag az egyes csapattagok szintjén merülnek fel. Ekkor integráljuk a rendszerbe a csapattagra kiosztott feladatrészt és segítünk egymásnak felmerülő kérdésekben. Egymást ellenőrizve integrálunk.

### 2.6.5 Felhasznált fejlesztőeszközök

- Eclipse és JDK:
  Fejlesztőkörnyezetnek a korábbi évek – labor, illetve otthoni hobbiprojekt - tapasztalataira alapozva közösen az Eclipse-t választottuk. Fejlesztői körökben egyébként is népszerű ez a környezet.

- Microsoft Visio:
  UML diagramok elkészítéséhez a Microsoft Visio nevű alkalmazást fogjuk használni. Előnyei közé soroljuk a vektorgrafikus ábrázolást, egyszerű szerkeszthetőséget és a könnyű hordozhatóságot. A diagramok Microsoft Word-be könnyen átmásolhatóak. A Microsoft ezt trialware szoftverként adta ki, ami számunkra elegendő funkciót biztosít a projekt UML modellezésére.

- GIT - verziókezelés:
  Ez a legnépszerűbb verziókezelő eszköz, ajánlásból kipróbáltuk és közösen megegyeztünk a használatában. A csapat nagy része egyébként is használt már élesben GIT-et, így kézenfekvőnek tűnt a döntés.

- Microsoft Word:
  Elegendő funkcióval rendelkező, minden csapattagunk által jól ismert szövegszerkesztő szoftver. Offline dokumentumkészítésre használjuk majd főként. Számlájára írandó még, hogy kellemesen másolható a közösen szerkeszthető GoogleDocs-ba.

- GoogleDocs:
  Az egyes csapattagok elkészített feladatrészeit egy közös Google Dokumentumba gyűjtjük.
  Ez követhetővé teszi a kiosztott feladatok elkészültségi szintjét és a projekt állapotát.
  A fájlok a Google Drive tárhelyén találhatóak, amely minden felhasználónak 15 GB ingyenes tárhelyet biztosít.

### 2.6.6 Kockázatelemzés

A tervezés során figyelembe kell venni számos kockázat lehetőségét. Ennek célja, hogy az adott kockázat bekövetkezésekor valamilyen kockázatkezelési stratégiával tudjunk előállni a probléma kezelésére.
A kockázattervezésre három stratégiát ismerünk:

- elkerülés stratégiája
- minimalizálás - bekövetkezett kockázat hatásának csökkentése
- folytatás - ha már bekövetkezett a kockázat, akkor hogyan tudjuk folytatni

A kockázatokat előfordulásuk valószínűsége alapján a következő csoportokba osztjuk: nagyon alacsony, alacsony, közepes, nagy, nagyon nagy.
A kockázat hatásait tekintve négy csoportba fogjuk osztani: katasztrofális, komoly, elviselhető, semleges.

| **Kockázat neve** | **Valószínűsége** | **Hatása** | **Lépések** |
|---|---|---|---|
| kilép egy csapattárs | nagyon alacsony | komoly | jelentés a konzulensnek és teendők/felelősségkörök újraelosztása |
| hardver meghibásodása | Közepes | elviselhető | biztonsági másolatok folyamatos készítése |
| követelmények módosítása időközben | Közepes | elviselhető | dinamikus alkalmazkodás |
| szoftver meghibásodása | Közepes | elviselhető | biztonsági másolat készítése |
| csapattárs túlterhelődése | Közepes | elviselhető | csapattárs feladatainak áttekintése, erőforrások újraelosztása |
| leadási határidő túllépése | nagyon alacsony | katasztrofális | Lehetőleg minél hamarabb leadni a beadandót. Ezenfelül csapattagok közötti feladat- és időbeosztás újragondolása, hogy a jövőben ne fordulhasson ilyen |

| | | | |
|---|---|---|---|
| | | | elő. |

## 2.7 Napló

| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.01.27. 18:00 | 2 óra | Török<br>Csontos<br>Bana<br>Tóth<br>Czirják | Értekezlet – Skype -<br>Csapatkapitány választás és követelmények tanulmányozása. |
| 2014.02.05. 18:00 | 3 óra | Csontos<br>Tóth<br>Bana<br>Czirják | Értekezlet - Követelmény, projekt, funkcionalitás feladatok áttekintése. Fejlesztés során felmerülő programok konkretizálása. |
| 2014.02.19. 13:30 | 2,5 óra | Török<br>Csontos<br>Bana<br>Tóth<br>Czirják | Értekezlet - Czirják és Török által külön-külön elkészített szótárakat áttekintettük, majd közösen megalkottuk a közös elfogadott verziót. Czirják vezetésével a feladatok kiosztásra kerültek illetve pontosítottuk a következő találkozó időpontját. |
| 2014.02.19. 16:00 | 4 óra | Bana | Szótár finomítás, funkciók megfogalmazása, kis kiegészítés a többi ponthoz |
| 2014.02.21. 18:00 | 1,5 óra | Csontos | Use-case diagram<br>Use-case leírások |
| 2014.02.22. 9:00 | 2 óra | Török | Szótár finomítás, formázás, dokumentáció, napló, funkcionális követelmények |

| | | | |
|---|---|---|---|
| 2014.02.22.<br>13:00 | 3 óra | Czirják | Projekt terv, cél, felhasználók, korlátozások |
| 2014.02.22.<br>17:00 | 0,5 óra | Tóth | Szakterület, nem funkcionális követelmények |
| 2014.02.22.<br>18:00 | 2 óra | Török<br>Csontos<br>Bana<br>Tóth<br>Czirják | Értekezlet - Skype - A beadandó feladatok leellenőrzése és részletes áttekintése. |
| 2014.02.23.<br>8:30 | 3 óra | Török | Spreadsheet - naplózás, funkciók, összefoglalás |
| 2014.02.23.<br>14:00 | 2,5 óra | Csontos | Definíciók, rövidítések<br>Hivatkozások<br>Általános áttekintés<br>Erőforrásokkal kapcsolatos követelmények<br>Átadással kapcsolatos követelmények |
| 2014.02.23.<br>16:00 | 2 óra | Török<br>Tóth<br>Czirják | Értekezlet - Skype - Funkciók rész áttekintése és szótárral való párhuzamba állítása, teljes dokumentáció áttekintése |
| 2014.02.23<br>18:00 | 2,5 óra | Csontos | Szótár kiegészítése<br>Funkcionális követelmények kiegészítése |
| 2014.02.23.<br>20:00 | 0,25 óra | Török | Utólagos dokumentáció |
| 2014.02.23<br>23:00 | 1 óra | Czirják | Dokumentáció végleges formázása |
| 2014.02.23.<br>23:00 | 2,5 óra | Bana | Funkciók véglegesítése, végső kiegészítések és észrevételek megvitatása a többi |

| | | | |
|---|---|---|---|
| | | | ponthoz, dokumentáció végleges formázása - utolsó simítások |

# 4. Analízis modell kidolgozása

## 4.1 Objektum katalógus

### 4.1.1 Enemy

Ellenséget reprezentáló objektum.
Az Enemy felelőssége az ellenség mozgatása az úton. Ez jelenti konkrétan a pozícióváltoztatást. Végül az ellenség dolga saját életerejének számontartása.

### 4.1.2 Engine

Az egész játék mozgatórugója, az időkezelésért felelős. Ezen belül felelős az ellenségek mozgatásért, továbbá a torony védekező függvényének meghívásáért. Kezeli a játék végkimenetelével kapcsolatos folyamatokat és tárolja a játékban nyilvántartott összes ellenséget.

### 4.1.3 Fellowship

A szövetséget reprezentáló objektum, ami tárolja az aktív és passzív ellenségeket illetve a teljes létszámot. A passzív ellenségek még nincsenek a pályán, de ők is a szövetség tagjai, míg az aktívak már a pályán láthatjuk. A játék előrehaladtával egyre nagyobb számú ellenség lép a pályára.

### 4.1.4 Main

A main példányosítja a mindent vezérlő Engine-t. Ennek az osztálynak a feladata a program elindítása, azaz a Main-ben található a program belépési pontja.

### 4.1.5 Obstacle

Akadályt reprezentáló objektum. Ha áll rajta ellenség, akkor lelassítja azt, a rajta elhelyezett varázskövek számától függően. A rá vonatkozó lassítási tényezőt és az eddig ráhelyezett varázskövek számát tárolja.

### 4.1.6 Player

A játékost reprezentáló objektum. Tárolja a játékos varázserejét, ami szükséges a tornyok és az akadályok építéséhez.
Felelős a már meglévő tornyok és akadályok fejlesztéséért varázskövek felhasználásával.
Felelős a kiválasztott építési terület (Tile) beépíthetőségének, valamint az építéshez szükséges varázserő megállapításáért, és Player dolga továbbá a saját varázserejének folyamatos menedzselése.
A Player objektum indítja a játékot, látja az egész pályát és iniciailizálja.

### 4.1.7 PlayingArea

A játékteret reprezentáló objektum.
A pályát felépítő elemtípusok listáit tartalmazza.
Feladata a pálya inicializálása. Felelős azért, hogy információt szolgáltasson a térkép egyes mezőinek tulajdonságairól, ilyen információszolgáltatás például a játékos informálása a mező beépíthetőségről vagy az ellenség informálása a Player objektumon keresztül az út következő mezőjéről. Azért is felelős, hogy ha a hegyen ellensgé áll, akkor vereséggel véget érjen a játék.

### 4.1.8 Position

Egy adott csempe két koordinátáját kompakt módon tároló objektum.
Felelős a koordináták tárolásáért, módosításáért.

### 4.1. 9 Road

Az ellenség által járható út-csempét reprezentáló objektum.
Számon tartja a saját pozícióját és a következő út-csempét is tárolja, ami az ellenségek útvonalához szükséges. Továbbá tárolja a típusát, ami a beépíhetőség miatt fontos.

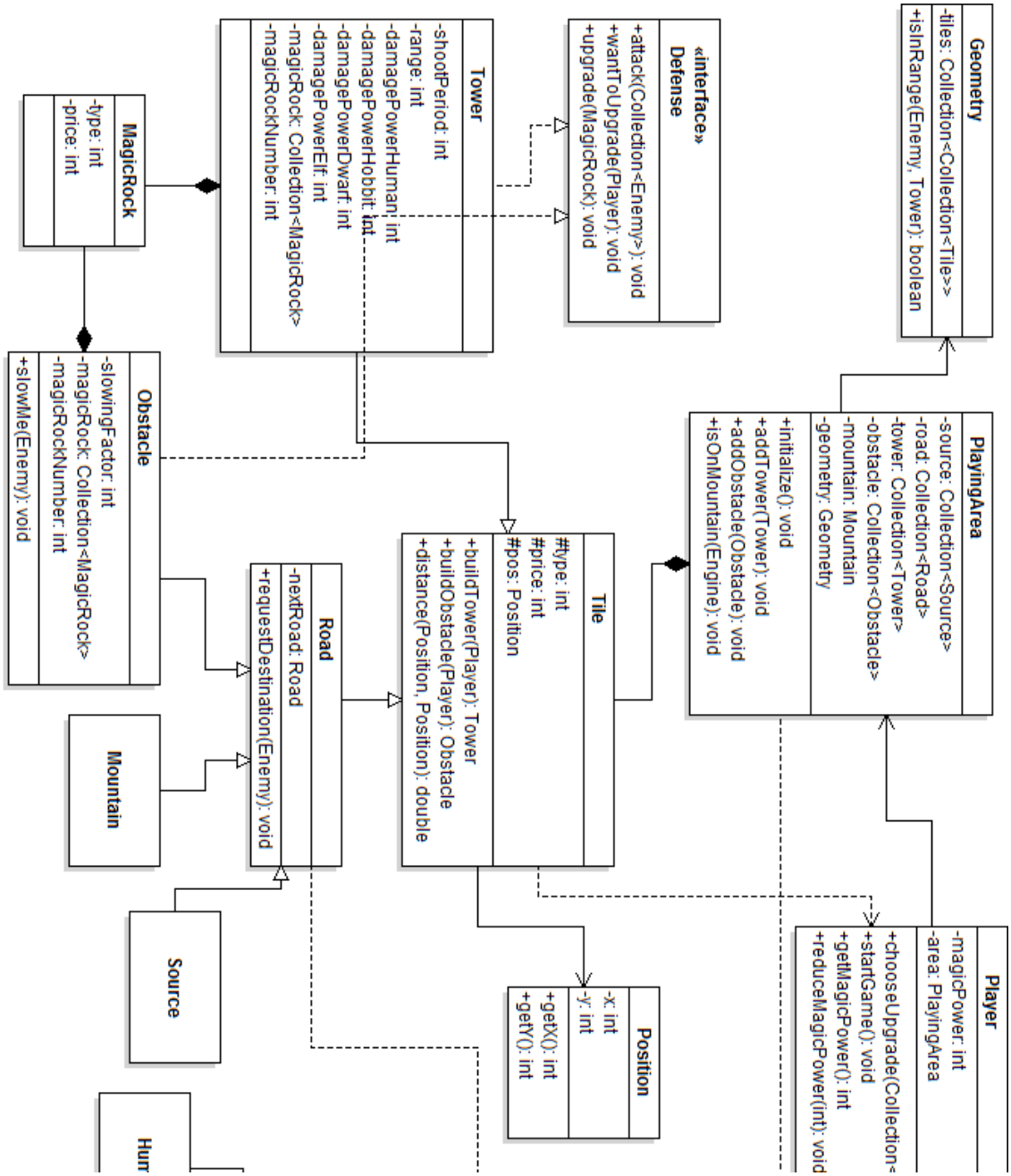
### 4.1.10 Tile

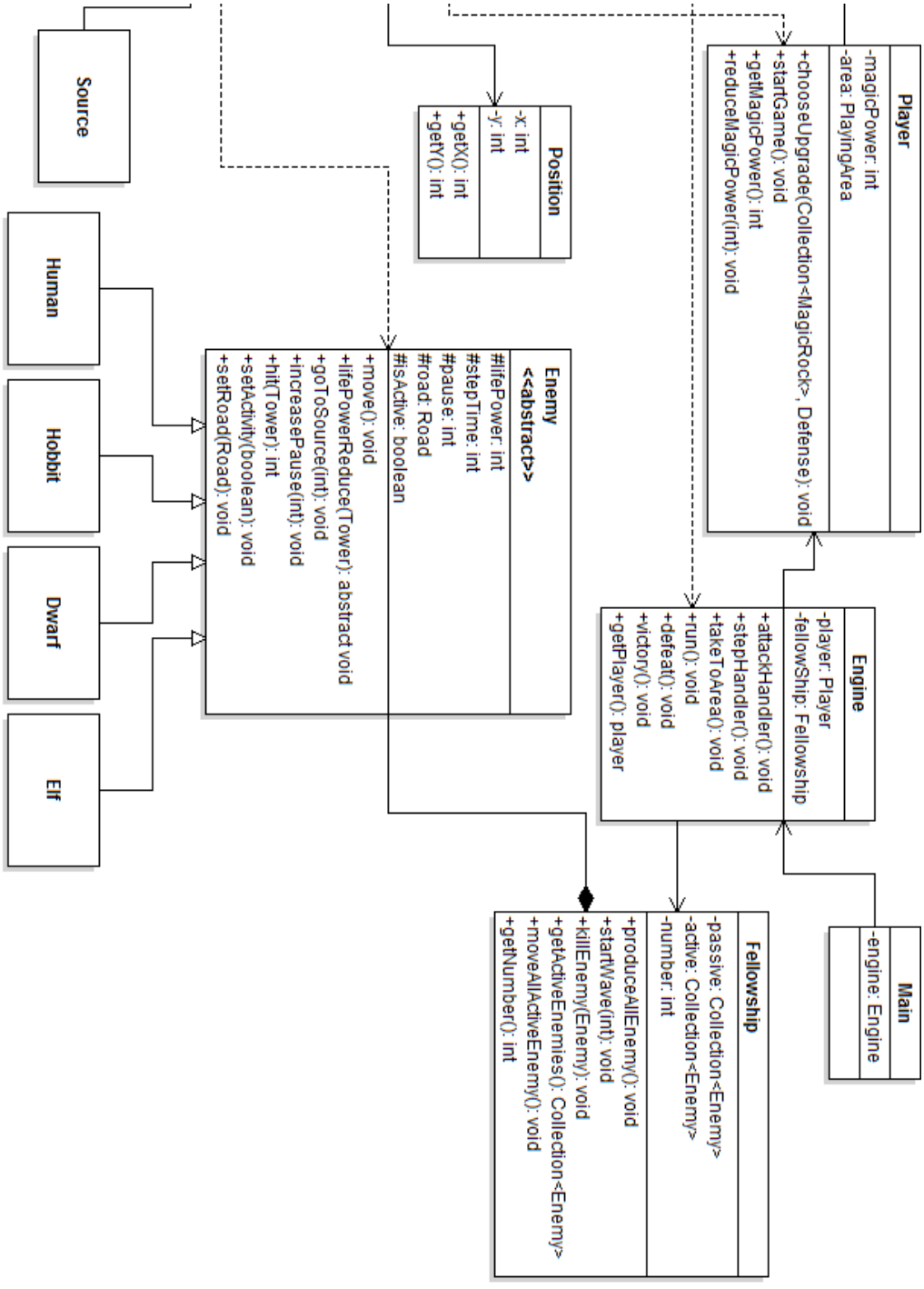
A kétdimenziós játéktér egy-egy elemét reprezentáló objektum. Tudja a saját beépíthetőségének típusát és ismeri a saját pozícióját illetve a csempére építés költségét is tárolja.
Felelős a magára vonatkozó tulajdonságok közléséért a külvilág felé. Továbbá meg tudja határozni bármely két csempe között mekkora a távolság illetve azonos-e két csempe.
Továbbá ő felel az akadályok és tornyok építéséért.

### 4.1.11 Tower

A játékos által elhelyezhető tornyot reprezentáló objektum, aki a tüzelésért felelős.
Tárolja a lövésre vonatkozó paramétereket (tüzelési hatékonyság, hatótávolság, sebzés ereje), és varázskő használata esetén növeli is a hatékonyságukat. Tárolja még a ráhelyezhető varázskövek aktuális számát, korlátozva annak maximális értékét.

## 4.2 Statikus struktúra diagramok


Geometry
-tiles: Collection<Collection<Tile>>
+isInRange(Enemy, Tower): boolean
«interface»
Defense
+attack(Collection<Enemy>): void
+wantToUpgrade(Player): void
+upgrade(MagicRock): void
Tower
-shootPeriod: int
-range: int
-damagePowerHuman: int
-damagePowerHobbit: int
-damagePowerDwarf: int
-damagePowerElf: int
-magicRock: Collection<MagicRock>
-magicRockNumber: int
MagicRock
-type: int
-price: int
PlayingArea
-source: Collection<Source>
-road: Collection<Road>
-tower: Collection<Tower>
-obstacle: Collection<Obstacle>
-mountain: Mountain
-geometry: Geometry
+initialize(): void
+addTower(Tower): void
+addObstacle(Obstacle): void
+isOnMountain(Engine): void
Obstacle
-slowingFactor: int
-magicRock: Collection<MagicRock>
-magicRockNumber: int
+slowMe(Enemy): void
Tile
#type: int
#price: int
#pos: Position
+buildTower(Player): Tower
+buildObstacle(Player): Obstacle
+distance(Position, Position): double
Road
-nextRoad: Road
+requestDestination(Enemy): void
Mountain
Source
Position
-x: int
-y: int
+getX(): int
+getY(): int
Player
-magicPower: int
-area: PlayingArea
+chooseUpgrade(Collection<
+startGame(): void
+getMagicPower(): int
+reduceMagicPower(int): void


Player
-magicPower: int
-area: PlayingArea
+chooseUpgrade(Collection<MagicRock>, Defense): void
+startGame(): void
+getMagicPower(): int
+reduceMagicPower(int): void
Position
-x: int
-y: int
+getX(): int
+getY(): int
Source
Human
Hobbit
Dwarf
Elf
Enemy
<<abstract>>
#lifePower: int
#stepTime: int
#pause: int
#road: Road
#isActive: boolean
+move(): void
+lifePowerReduce(Tower): abstract void
+goToSource(int): void
+increasePause(int): void
+hit(Tower): int
+setActivity(boolean): void
+setRoad(Road): void
Engine
-player: Player
-fellowShip: Fellowship
+attackHandler(): void
+stepHandler(): void
+takeToArea(): void
+run(): void
+defeat(): void
+victory(): void
+getPlayer(): player
Fellowship
-passive: Collection<Enemy>
-active: Collection<Enemy>
-number: int
+produceAllEnemy(): void
+startWave(int): void
+killEnemy(Enemy): void
+getActiveEnemies(): Collection<Enemy>
+moveAllActiveEnemy(): void
+getNumber(): int
Main
-engine: Engine

# 4.3 Osztályok leírása

## 4.3.1 Dwarf

**Felelősség**
Ez egy törpe objektum, ami az életerő módosításáért felelős.

**Ősosztályok**
Enemy

**Interfészek**
Nincs

**Attribútumok**
Nincs

**Metódusok**

- **void lifePowerReduce(Tower)**: életerőt csökkenti a Tower által megadott mértékben.

## 4.3.2 Elf

**Felelősség**
Ez egy tünde objektum, ami az életerő módosításáért felelős.

**Ősosztályok**
Enemy

**Interfészek**
Nincs

**Attribútumok**
Nincs

**Metódusok**

- **void lifePowerReduce(Tower)**: életerőt csökkenti a Tower által megadott mértékben.

### 4.3.3 Enemy

**Felelősség**

Ez egy absztrakt osztály, ami egy ellenséget reprezentál. Az ellenség pozíciójának, életerejének tárolásáért, az ellenség léptetéséért és az ellenségek közötti szünetek beiktatásáért felelős.

**Ősosztályok**

Nincs

**Interfészek**

Nincs

**Attribútumok**

- **int lifePower**: életerőt tárolja
- **int stepTime:** léptetés sebességét tárolja
- **int pause**: a kihagyott lépések számát tárolja
- **Road road**: aktuális út-csempét tárolja, amin az ellenség áll
- **boolean isActive**: ellenség állapotának igazságértékét adja meg (passzív vagy aktív)

**Metódusok**

- **void move():** az ellenség léptetését végzi
- **abstact void lifePowerReduce(Tower)**: A leszármazott osztályoknak kell megvalósítani.
- **void goToSource(int pause)**: ellenség forrásra helyezése
- **void increasePause(int slowingFactor)**: pause attribútum növelése a lassító faktorral, ezáltal az ellenség pause mennyiségű lépésből fog kimaradni (akadály lassítás)
- **int hit(Tower)**: ellenség sebzése egy adott torony által
- **void setActivity(boolean)**: ellenség isActive attribútumának igazságértéket változtatja meg
- **void setRoad(Road)**: ellenség a következő csempére lép, road attribútumának értéke módosul az új road-ra

### 4.3.4 Engine

**Felelősség**

Engine felelős az idő kezeléséért. Peroiodikus időközönként meghívja az összes ellenség move() metódusát, és az összes torony attack(Collection<Enemy>) metódusát. Továbbá az ellenségek pályára helyezésének időzítéséért és a vereség/győzelem figyeléséért is felelős.

**Ősosztályok**
Thread

**Interfészek**
Nincs

**Attribútumok**
- **Player player**: referencia a játékosra
- **Fellowship fellowship:** referencia a szövetségre

**Metódusok**
- **void attackHandler()**: Minden toronyra meghívja az attack(Collection<Enemy>) metódust.
- **void stepHandler()**: Az ellenségek move() metódusát hívja meg
- **void takeToArea():** az ellenség pályára helyezésekor hívódik meg, itt mondjuk meg, hogy hány hullám lesz, egy hullámon belül hány ellenség és azok milyen időközönként jönnek
- **void run()**: Időzítő, ami bizonyos időközönként meghívja az attackHandler(), stepHandler() és takeToArea() metódusokat.
- **void defeat()**: kiírja, hogy vereség és felszabadítja az összes objektumot
- **void victory()**: kiírja, hogy győzelem és felszabadítja az összes objektumot
- **Player getPlayer()**: játékost adja vissza

### 4.3.5 Fellowship

**Felelősség**
Egy szövetséget reprezentál, ami az összes aktív és passzív ellenséget tárolja.

**Ősosztályok**
Nincs

**Interfészek**
Nincs

**Attribútumok**
- **Collection<Enemy> passive**: szövetségen belüli passzív ellenségeket tárolja
- **Collection<Enemy> active**: szövetségen belüli aktív ellenségeket tárolja
- **int number**: szövetségen belüli ellenségek számát adja meg

**Metódusok**

- **void produceAllEnemy():** összes ellenséget létrehozza és beleteszi a passive listába, de még nincsenek a pályán az ellenségek
- **void startWave(int number):** létrehoz egy number számú ellenségből álló hullámot és kezdeményezi a pályára helyezésüket
- **void killEnemy(Enemy):** ellenség elpusztulásakor törli az ellenséget a kollekcióból
- **Collection<Enemy> getActiveEnemies():** aktív ellenségek kollekcióját adja vissza
- **void moveAllActiveEnemy():** összes aktív ellenség léptetése
- **int getNumber():** number attribútum értékét adja vissza

### 4.3.6 Geometry

**Felelősség**

Felelős az összes csempe tárolásáért. Továbbá meg tudja határozni, hogy egy tetszőleges ellenség egy tetszőleges torony hatótávolságán belül van-e.

**Ősosztályok**

Nincs

**Interfészek**

Nincs

**Attribútumok**

- **Collection<Collection<Tile>> tiles**: összes csempét tárolja

**Metódusok**

- **boolean isInRange(Enemy, Tower):** megadja, hogy egy ellenség egy torony hatótávolságán belül van-e

### 4.3.7 Hobbit

**Felelősség**

Ez egy hobbit objektum, ami az életerő módosításáért felelős.

**Ősosztályok**

Enemy

**Interfészek**
Nincs

**Attribútumok**
Nincs

**Metódusok**
- **void lifePowerReduce(Tower)**: életerőt csökkenti a Tower által megadott mértékben.

### 4.3.8 Human

**Felelősség**
Ez egy ember objektum, ami az életerő módosításáért felelős.

**Ősosztályok**
Enemy

**Interfészek**
Nincs

**Attribútumok**
Nincs

**Metódusok**
- **void lifePowerReduce(Tower)**: életerőt csökkenti a Tower által megadott mértékben.

### 4.3.9 MagicRock

**Felelősség**
Egy varázskövet reprezentál. Egy ilyet használ fel, ha tornyot vagy akadályt akar építeni a játékos.

**Ősosztályok**
Nincs

**Interfészek**
Nincs

**Attribútumok**

- **int type:** varázskő típusa
- **int price:** varázskő ára

**Metódusok**
Nincs

### 4.3.10 Main

**Felelősség**
A program innen indul ki. Létrehozza az Engine objektumot.

**Ősosztályok**
Nincs

**Interfészek**
Nincs

**Attribútumok**

- **Engine engine**: referencia az Engine-re.

**Metódusok**
Nincs

### 4.3.11 Mountain

**Felelősség**
A Végzet-hegyét reprezentálja.

**Ősosztályok**
Road

**Interfészek**
Nincs

**Attribútumok**
Nincs

**Metódusok**
Nincs

### 4.3.12 Obstacle

**Felelősség**
Ez egy akadály objektum, ami lassítja az ellenség haladását az akadály területén belül.

**Ősosztályok**
Road

**Interfészek**
Nincs

**Attribútumok**
- **int slowingFactor**: akadály lassításának mértéke
- **Collection<MagicRock> magicRock**: akadályra helyezhető varázskövek
- **int magicRockNumber:** akadályon lévő varázskövek száma

**Metódusok**
- **void attack(Collection<Enemy>):** figyeli, hogy áll-e az adott akadályon ellenség
- **void slowMe(Enemy)**: az akadályon álló ellenséget lelassítja, úgy hogy növeli az ellenségnek a pause attribútumát
- **void wantToUpgrade(Player)**: akadály fejlesztésének kezdeményezésére hívódik meg
- **void upgrade(MagicRock)**: akadály fejlesztése a megadott varázskővel

### 4.3.13 Player

**Felelősség**
Játékost reprezentáló objektum. A tornyok, akadályok fejlesztéséért és a játék elindításáért felelős.

**Ősosztályok**
Nincs

**Interfészek**
Nincs

**Attribútumok**
- **int magicPower**: varázserőt tárolja
- **PlayingArea area:** referencia a PlayingArea-ra

**Metódusok**
- **void startGame()**: játék indítása
- **void chooseUpgrade(Collection<MagicRock>, Defense)**: játékos megkapja, hogy egy adott toronyra vagy akadályra milyen varázskövek tehetőek és ebből egyet kiválasztva fejleszthet, ha van elég varázsereje
- **int getMagicPower()**: játékos aktuális varázserejét adja vissza.
- **void reduceMagicPower(int price)**: játékos varázserejét csökkenti

### 4.3.14 PlayingArea

**Felelősség**
A pálya különböző csempéinek a listában tárolásáért és a pálya inicializálásáért felelős. Továbbá itt lehet tornyokat és akadályokat hozzáadni a torony és akadály listához.

**Ősosztályok**
Nincs

**Interfészek**
Nincs

**Attribútumok**
- **Collection<Source> source**: forrásokat tárolja
- **Collection<Road> road**: az utakat tárolja, amin az ellenségek haladnak
- **Collection<Tower> tower**: tornyokat tárolja
- **Collection<Obstacle> obstacle**: akadályokat tárolja
- **Mountain mountain**: referencia a Mountain-ra
- **Geometry geometry**: referencia a Geometry-re

**Metódusok**

- **void initialize():** a pályát inicializálja
- **void addTower(Tower):** egy tornyot ad hozzá a tower listához
- **void addObstacle(Obstacle):** egy akadályt ad hozzá az obstacle listához
- **void isOnMountain(Engine):** megadja, hogy áll-e ellenség a hegyen, ha igen, akkor meghívja az Engine defeat() metódusát

### 4.3.15 Position

**Felelősség**

(x, y) koordinátapárok tárolásáért felel, ami a 2D-s pályához szükséges.

**Ősosztályok**

Nincs

**Interfészek**

Nincs

**Attribútumok**

- **int x**: x koordinátát tárolja
- **int y**: y koordinátát tárolja

**Metódusok**

- **int getX()**: x koordinátát adja vissza
- **int getY()**: y koordinátát adja vissza

### 4.3.16 Road

**Felelősség**

Egy út-csempét reprezentál.

**Ősosztályok**

Tile

**Interfészek**

Nincs

**Attribútumok**

- **Road nextRoad:** tárolja a következő út-csempét, ami az ellenségek mozgatásához szükséges

**Metódusok**

- **void requestDestination(Enemy)**: egy adott ellenséget lépteti a következő út-csempére

### 4.3.17 Source

#### Felelősség

Egy forrást reprezentál.

**Ősosztályok**

Road

**Interfészek**

Nincs

**Attribútumok**

Nincs

**Metódusok**

Nincs

### 4.3.18 Tile

#### Felelősség

Egy csempe tárolásáért felel, ami a pálya atomi része. Továbbá tornyok és akadályok építéséért felelős.

**Ősosztályok**

Nincs

**Interfészek**

Nincs

**Attribútumok**

- **int type**: A csempe típusát adja meg. Ez alapján lehet eldönteni, hogy lehet-e egy csempére tornyot illeteve akadályt építeni. Tile esetén az értéke 0 lesz.
- **int price**: Az építés költségét tárolja.
- **Position pos**: A csempe pozícióját tárolja.

**Metódusok**

- **Tower buildTower(Player)**: egy tornyot épít az adott csempére
- **Obstacle buildObstacle(Player)**: egy akadályt épít az adott csempére
- **double distance(pos1, pos2)**: két pozíció közötti távolságot adja meg

### 4.3.19 Tower

**Felelősség**

Torony, ami a hatótávolságán belüli ellenségek sebzéséért felelős.

**Ősosztályok**

Tile

**Interfészek**

Nincs

**Attribútumok**

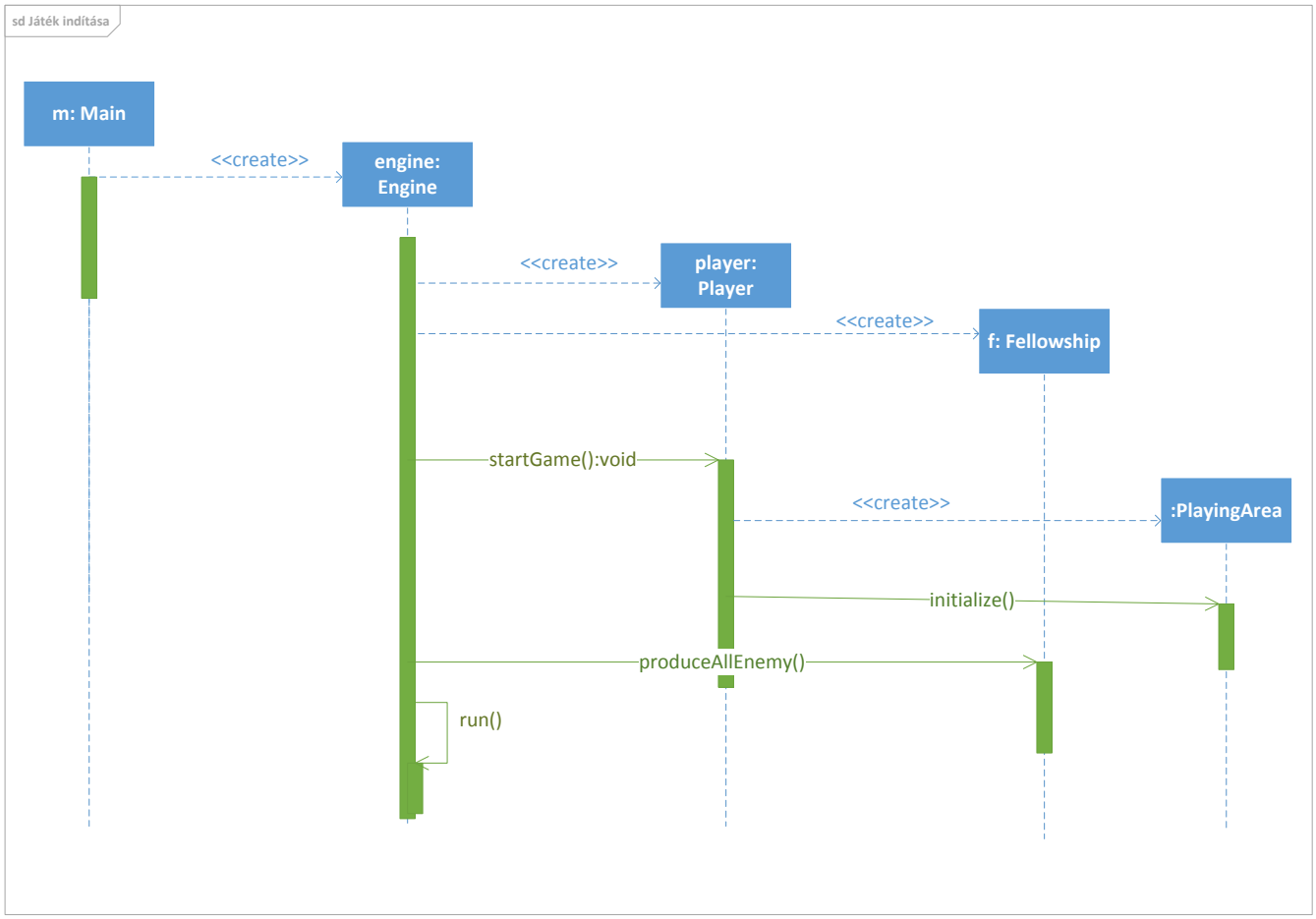
- **int shootPeriod**: A torony tüzelési gyakorisága.
- **int range**: A torony hatótávolsága.
- **int damagePowerHuman**: Torony sebzési ereje egy Human objektumra.
- **int damagePowerHobbit**: Torony sebzési ereje egy Hobbit objektumra.
- **int damagePowerDwarf**: Torony sebzési ereje egy Dwarf objektumra.
- **int damagePowerElf**: Torony sebzési ereje egy Elf objektumra.
- **int magicRockNumber**: Tornyon lévő varázskövek száma.

**Metódusok**

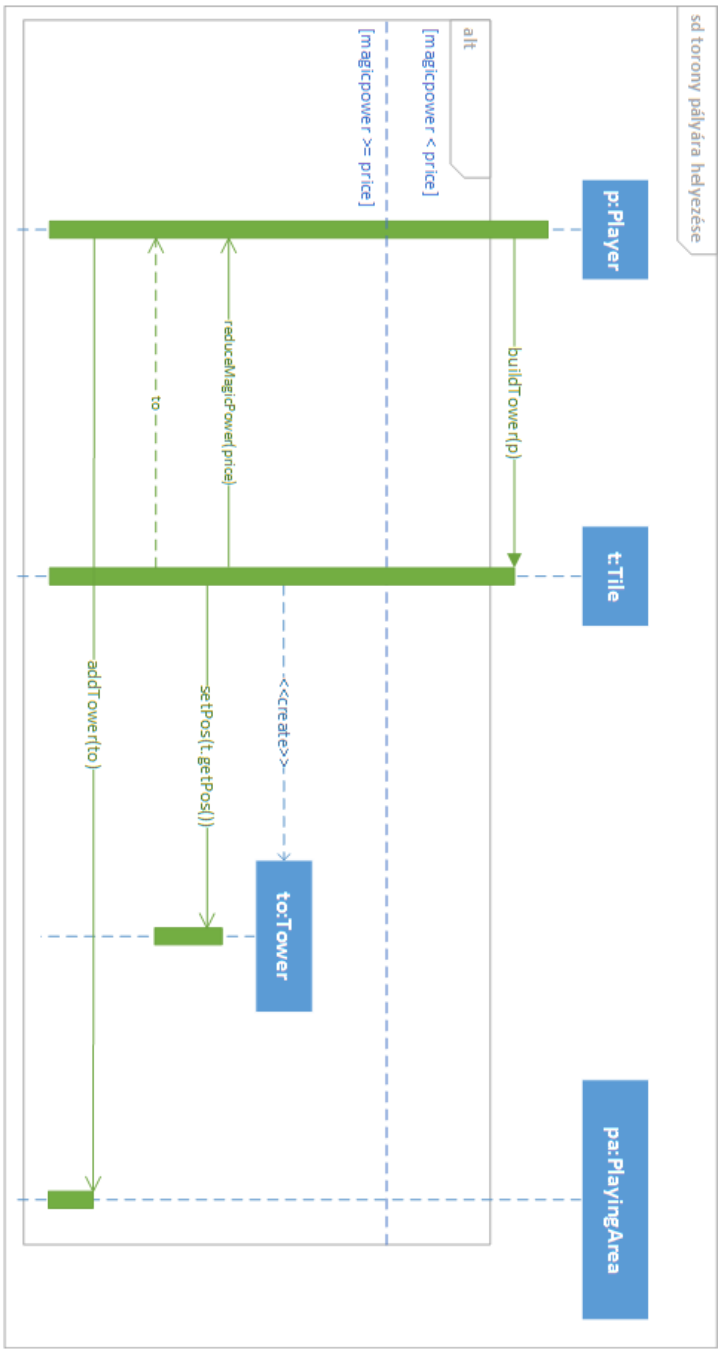
- **void attack(Collection<Enemy>):** végigiterál az összes aktív ellenségen és a hatótávon belüliek közül meglő egy ellenséget
- **void wantToUpgrade(Player)**: torony fejlesztésének kezdeményezésére hívódik meg
- **void upgrade(MagicRock)**: torony fejlesztése a megadott varázskővel

# 4.4 Szekvencia diagramok

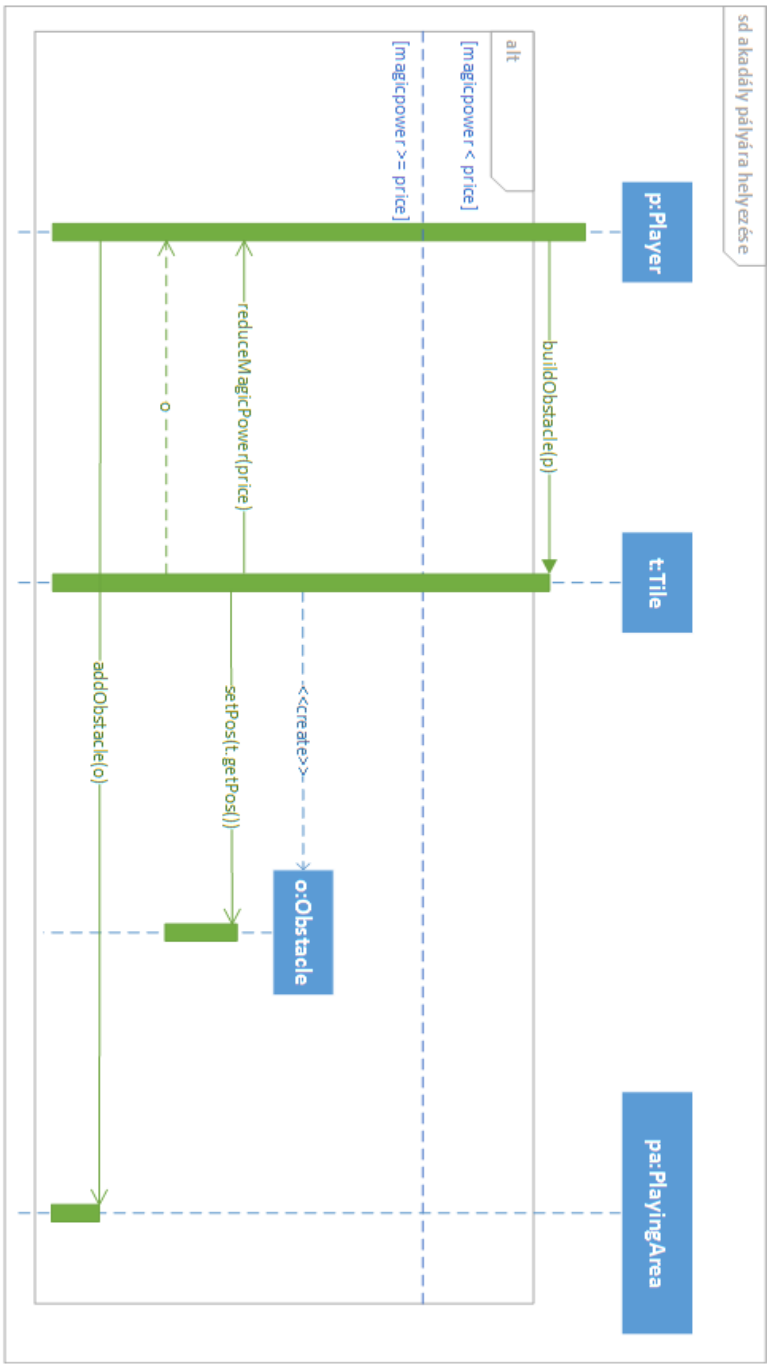
## 4.4.1 INDÍTÁS



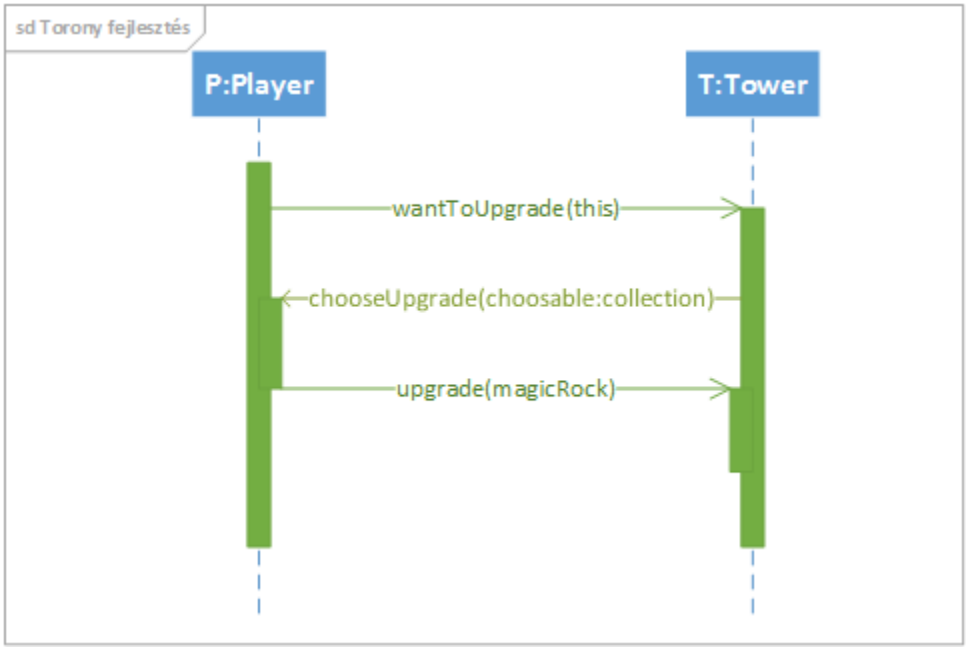
### 4.4.2 TORONY ÉPÍTÉS



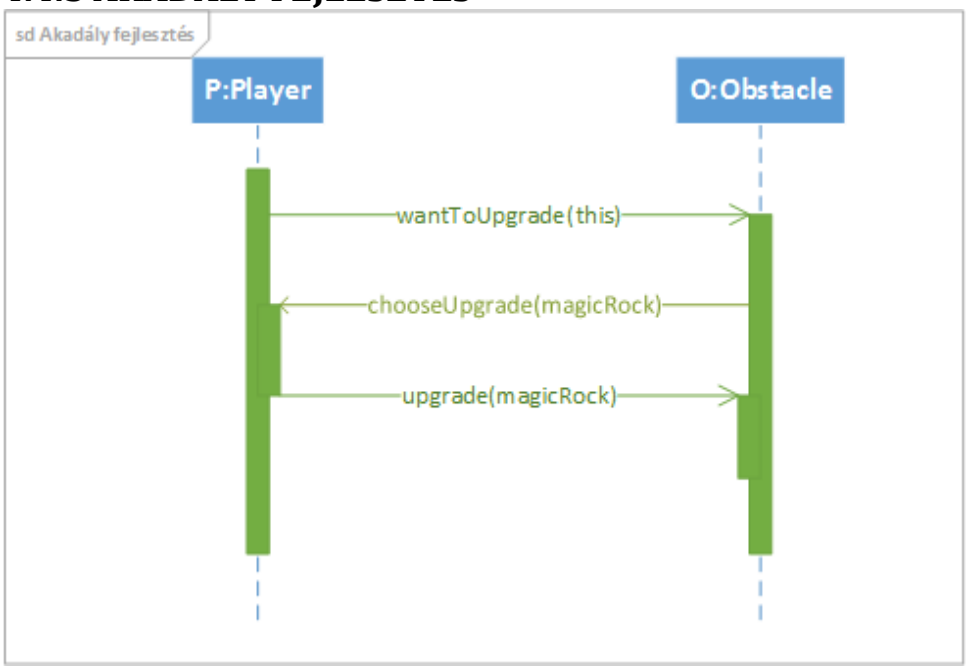
### 4.4.3 AKADÁLY ÉPÍTÉS



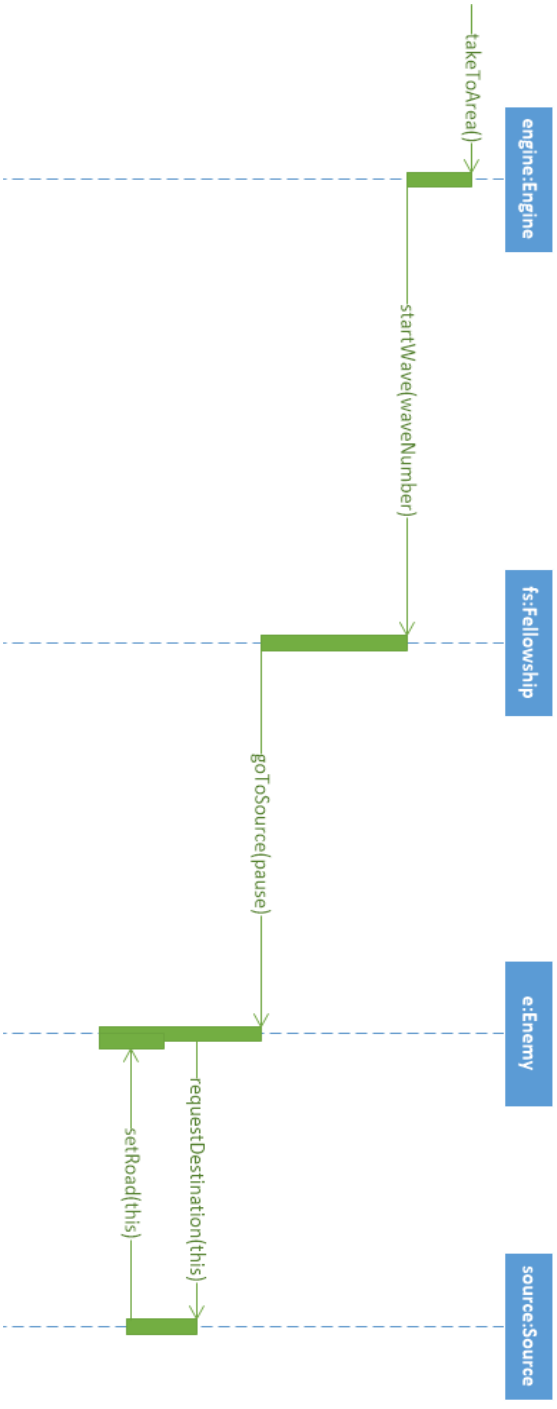
### 4.4.4 TORONY FEJLESZTÉS



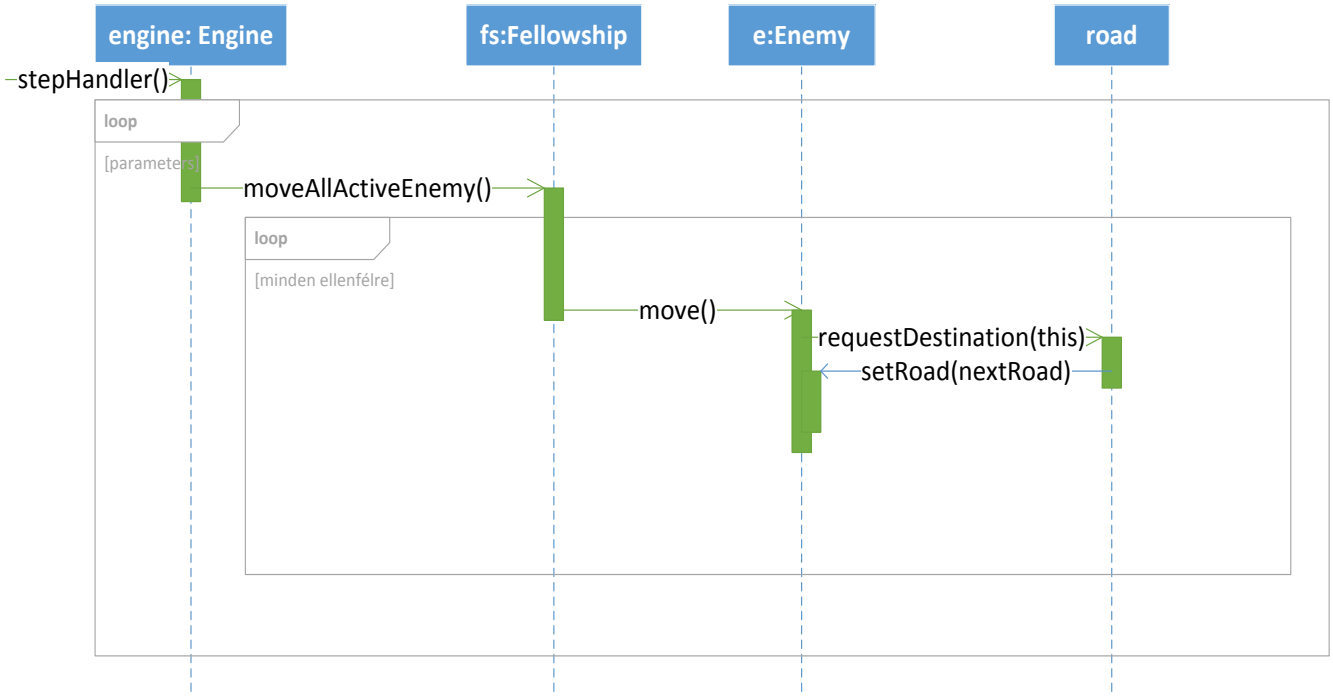

### 4.4.5 AKADÁLY FEJLESZTÉS



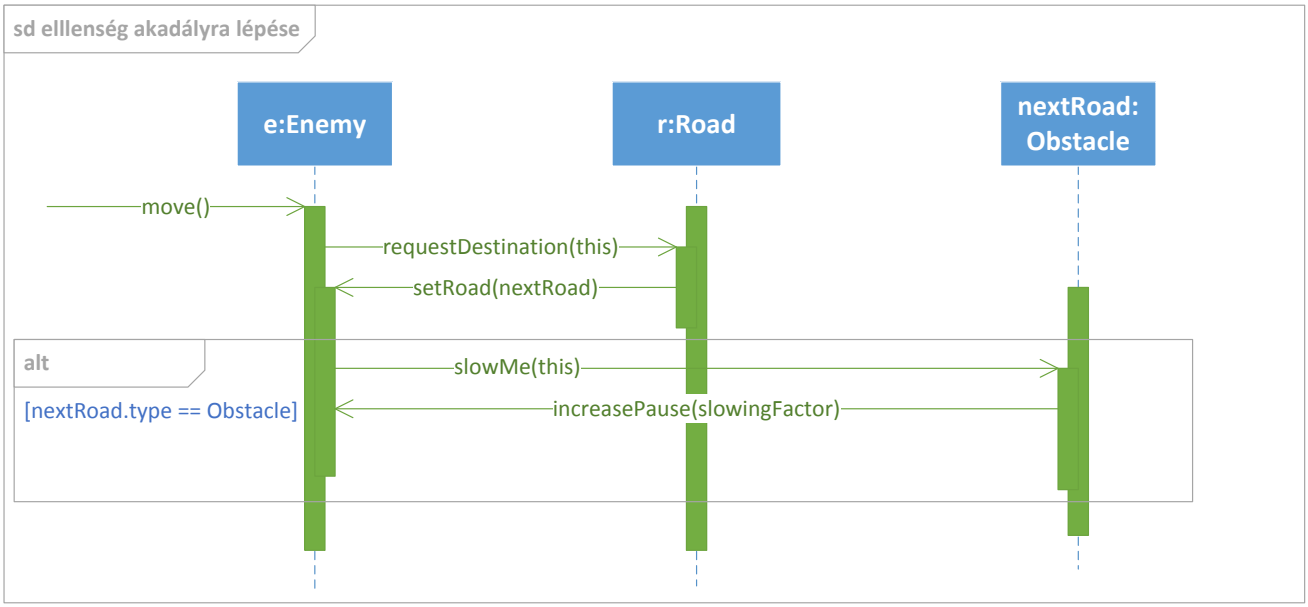
### 4.4.6 ELLENSÉG PÁLYÁRA HELYEZÉSE



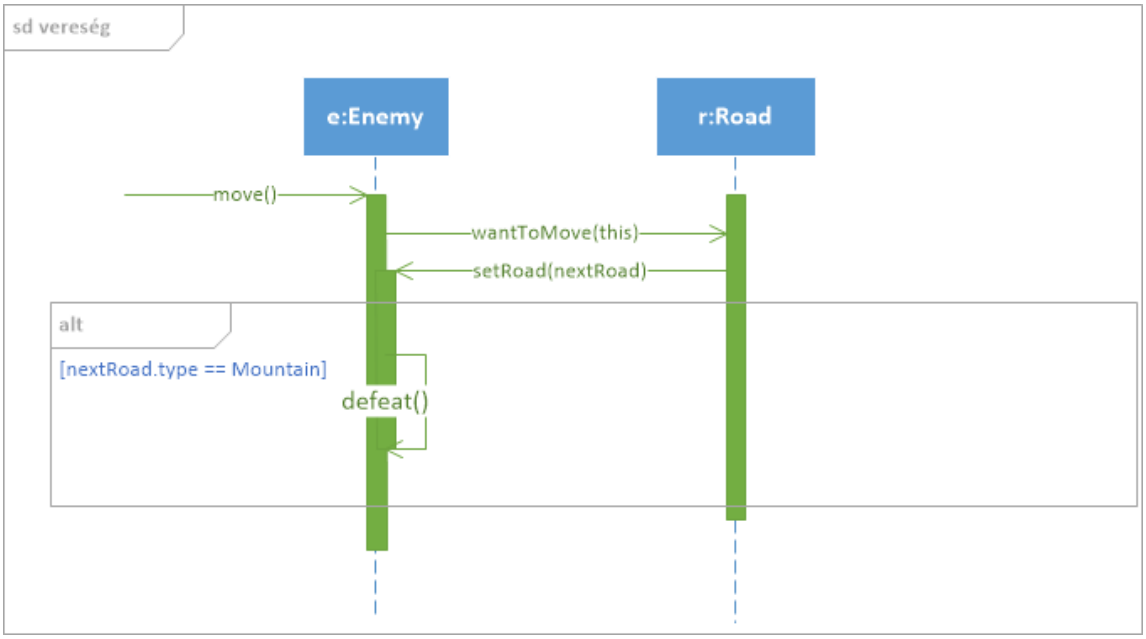
### 4.4.7 ELLENSÉG LÉPTETÉSE



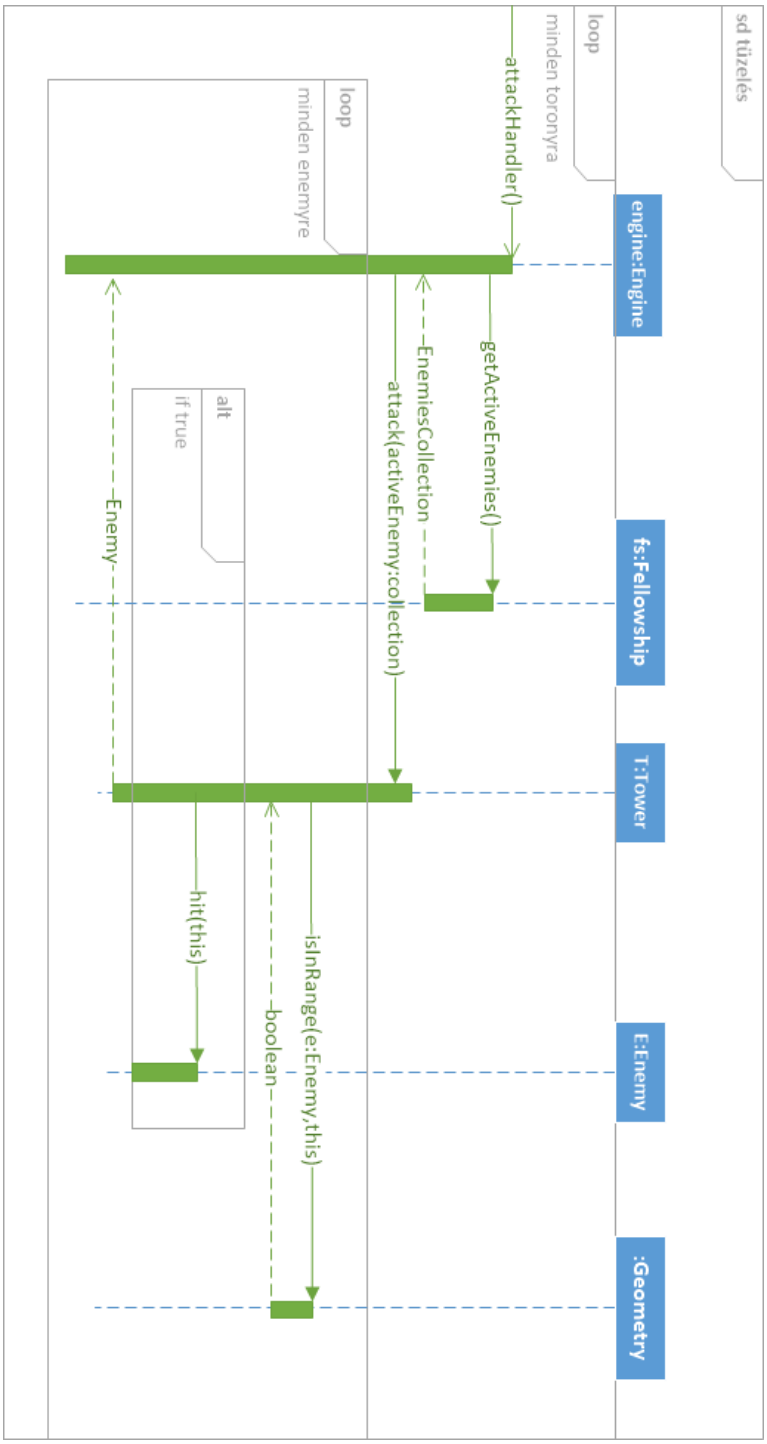
### 4.4.8 ELLENSÉG LASSÍTÁSA



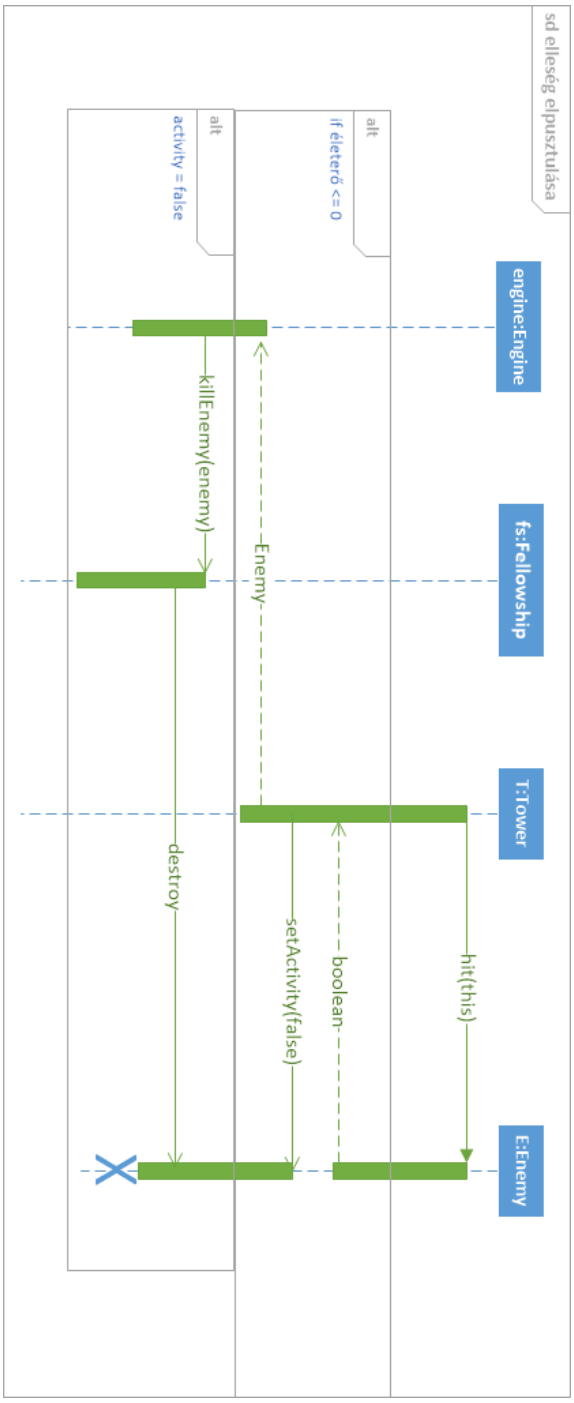

### 4.4.9 VERESÉG



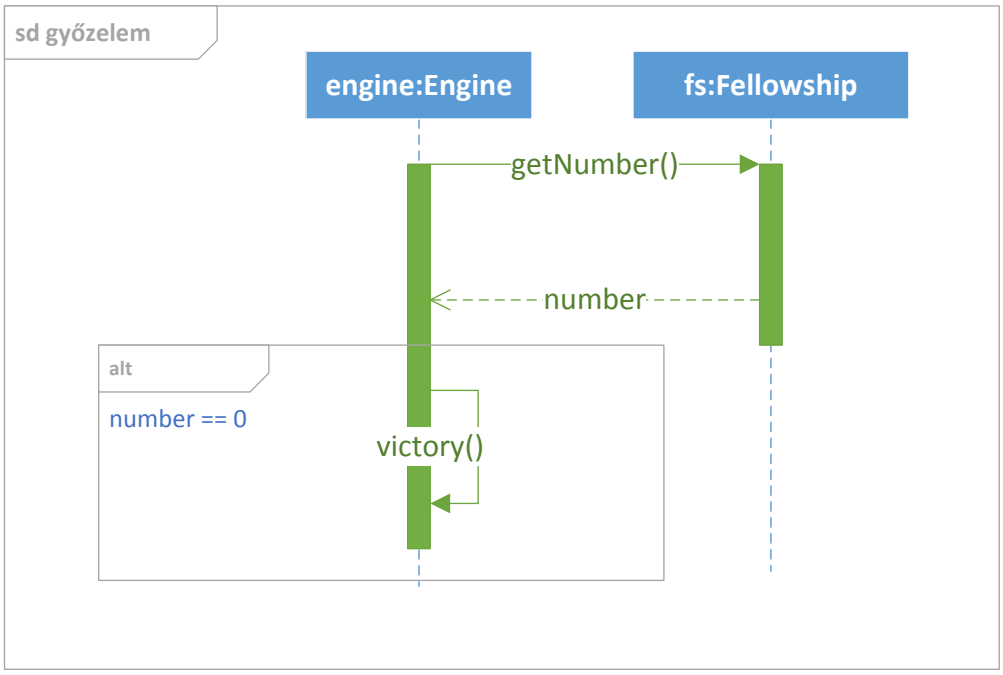
## 4.4.10 TÜZELÉS



### 4.4.11 ELLENSÉG ELPUSZTULÁSA

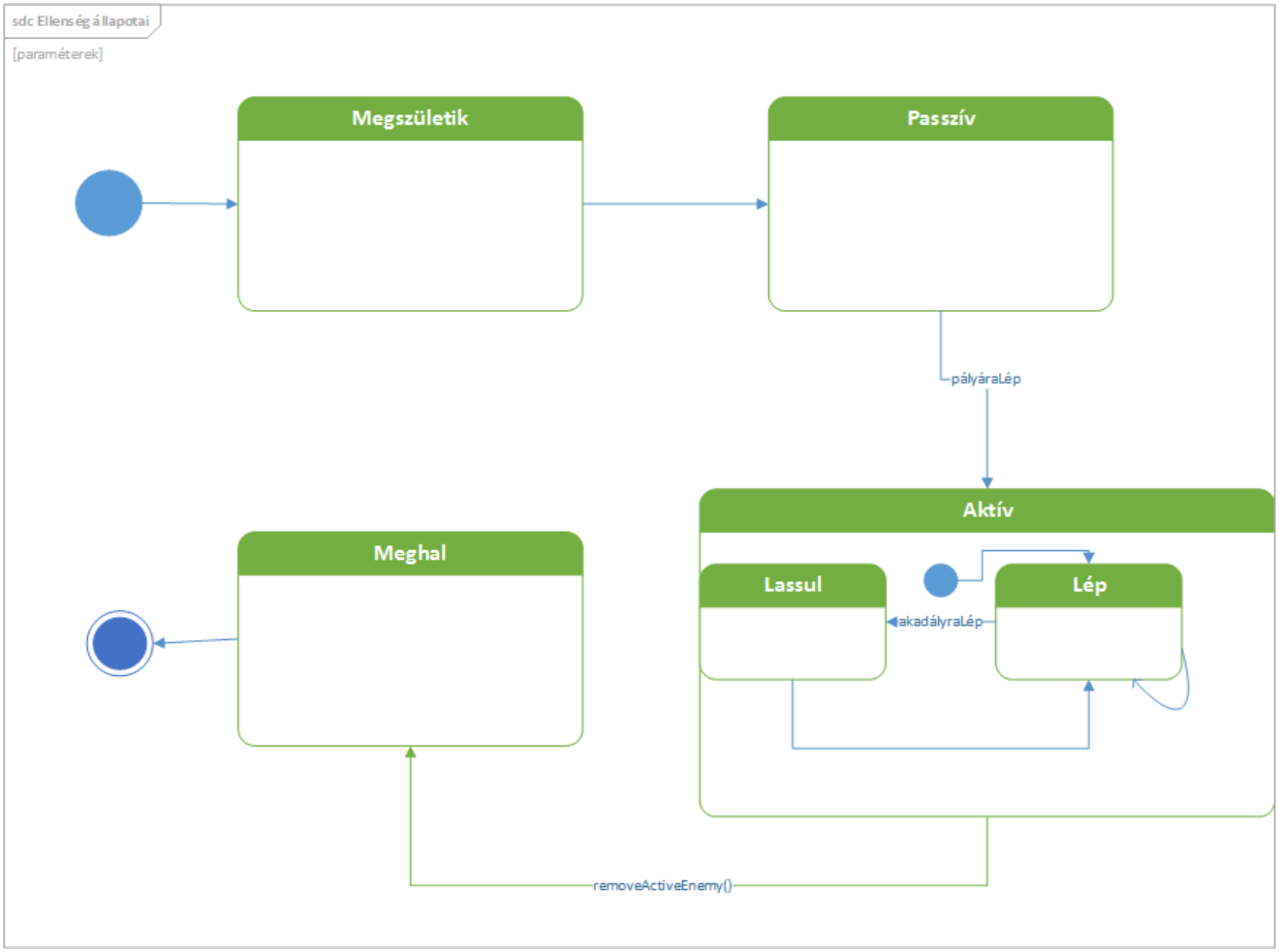


### 4.4.12 GYŐZELEM

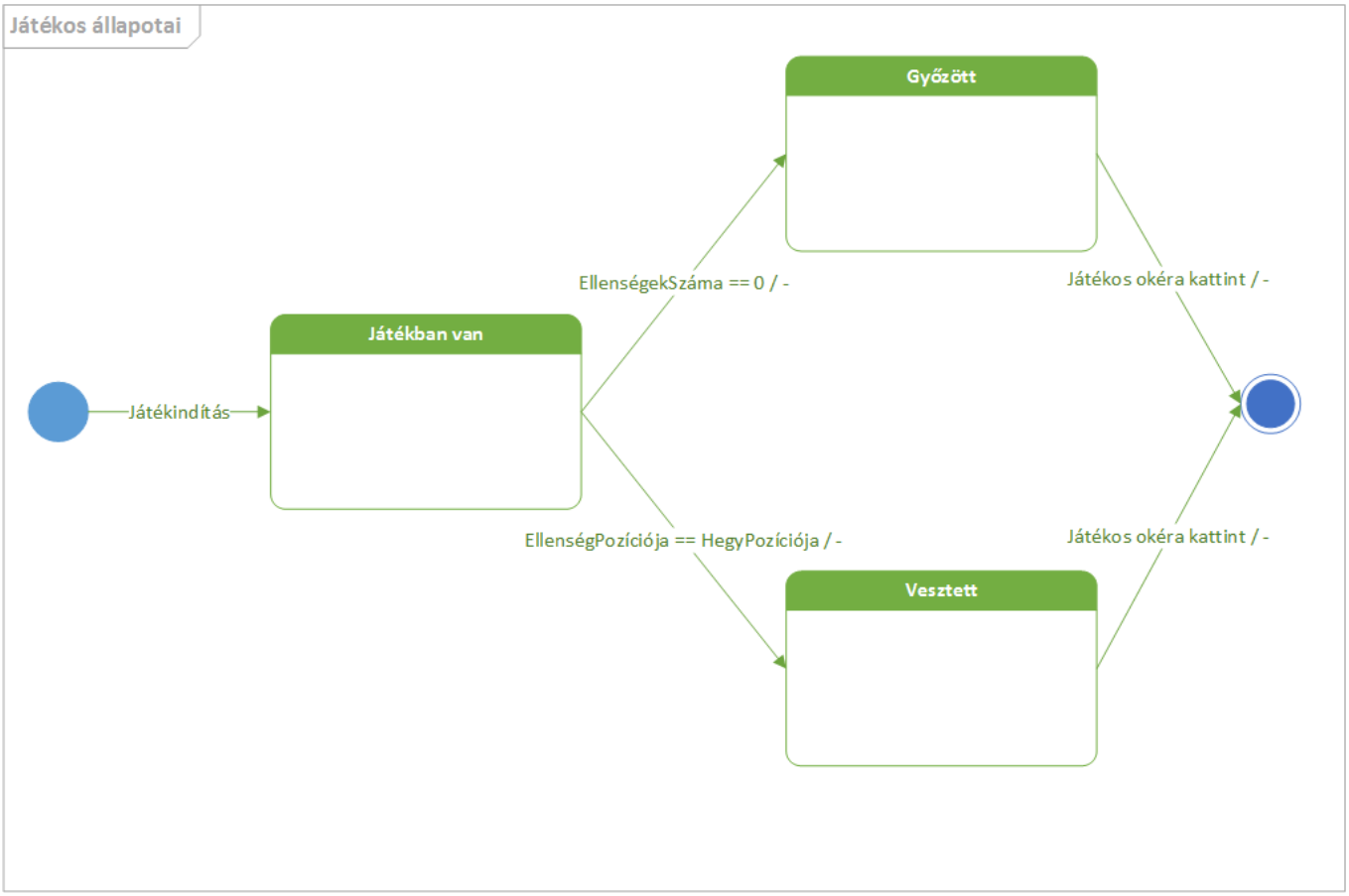


# 4.5 State-chartok

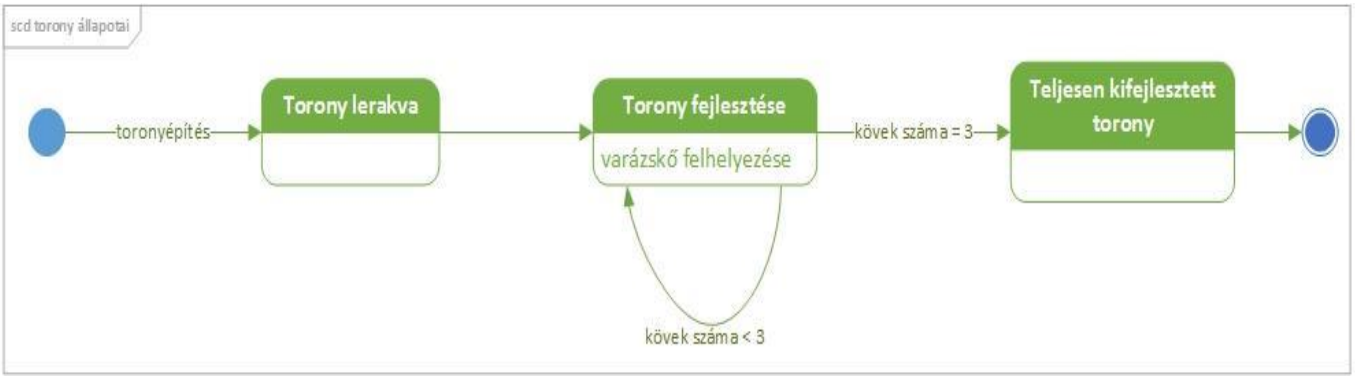
## 4.5.1 Ellenség állapotai



### 4.5.2 Játékos állapotai




### 4.5.3 Torony állapotai



## 4.6 Napló

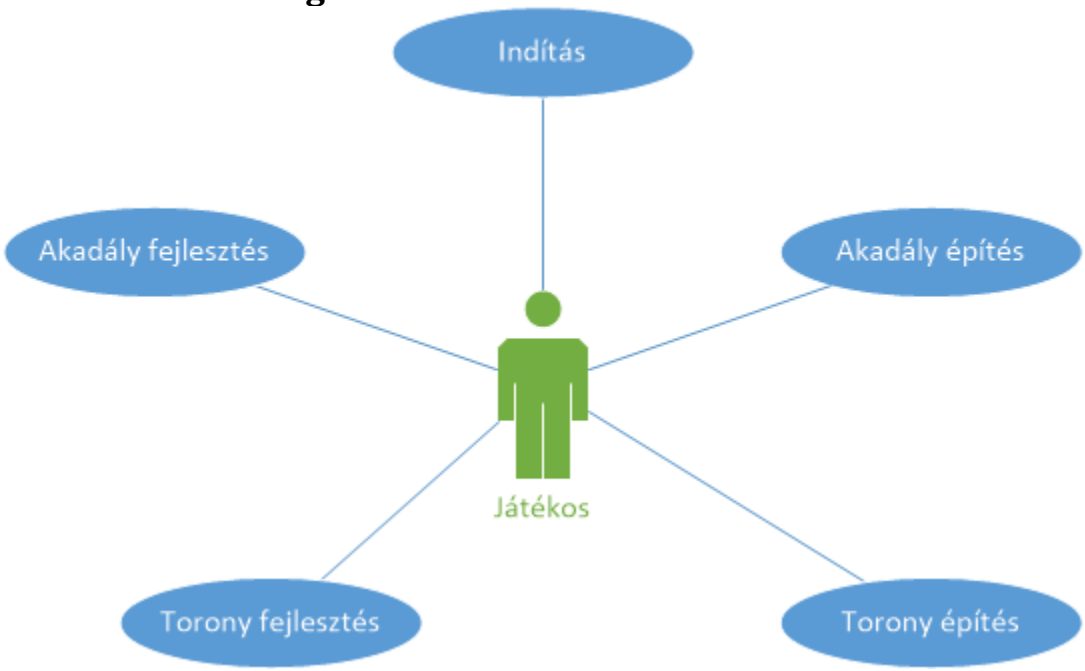
| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.03.07 14:00 | 3 óra | Csontos | Osztálydiagram módosítása<br>objektumok felelősségének újragondolása |
| 2014.03.08 | 4 óra | Tóth | Tüzelés, Torony fejlesztés, Akadály fejlesztés diagram javítása |
| 2014.03.07 16:00 | 1,5 óra | Tóth<br>Bana<br>Czirják<br>Török<br>Csontos | Módosítások megbeszélése |
| 2014.03.08 16:00 | 1,5 óra | Tóth<br>Csontos | 4.4.1, 4.4.2, 4.4.3 és 4.4.10 szekvencia diagramok megtárgyalása skype-on |
| 2014.03.08 20:00 | 1,5 óra | Czirják<br>Csontos | 4.4.4 szekvencia diagram megtárgyalása skype-on |
| 2014.03.08 22:00 | 1,5 óra | Bana<br>Csontos | 4.4.5, 4.4.6 és 4.4.7 szekvencia diagramok megtárgyalása skype-on |
| 2014.03.08 23:30 | 2.5 óra | Bana | 4.4.7 szekvencia diagram elkészítése a korábban tárgyaltak alapján |
| 2014.03.09 09:30 | 2 óra | Csontos | Osztálydiagramon újabb módosítások<br>Konzisztencia vizsgálat osztály- és szekvencia diagramok között |
| 2014.03.09 13:30 | 1 óra | Czirják<br>Csontos | 4.4.8 szekvencia diagram megtárgyalása skype-on |
| 2014.03.09 16:00 | 2 óra | Bana | 4.4.7 szekvencia diagramon észlelt hibák javítása<br>4.4.5, 4.4.6 szekvencia diagram átalakítása a megbeszéltek szerint |
| 2014.03.09. 16:30 | 1,5 óra | Török<br>Csontos | 4.4.11 és 4.4.12 szekvencia diagramok elkészítése |
| 2014.03.09 | 1,5 óra | Csontos | Osztályleírás módosítása |

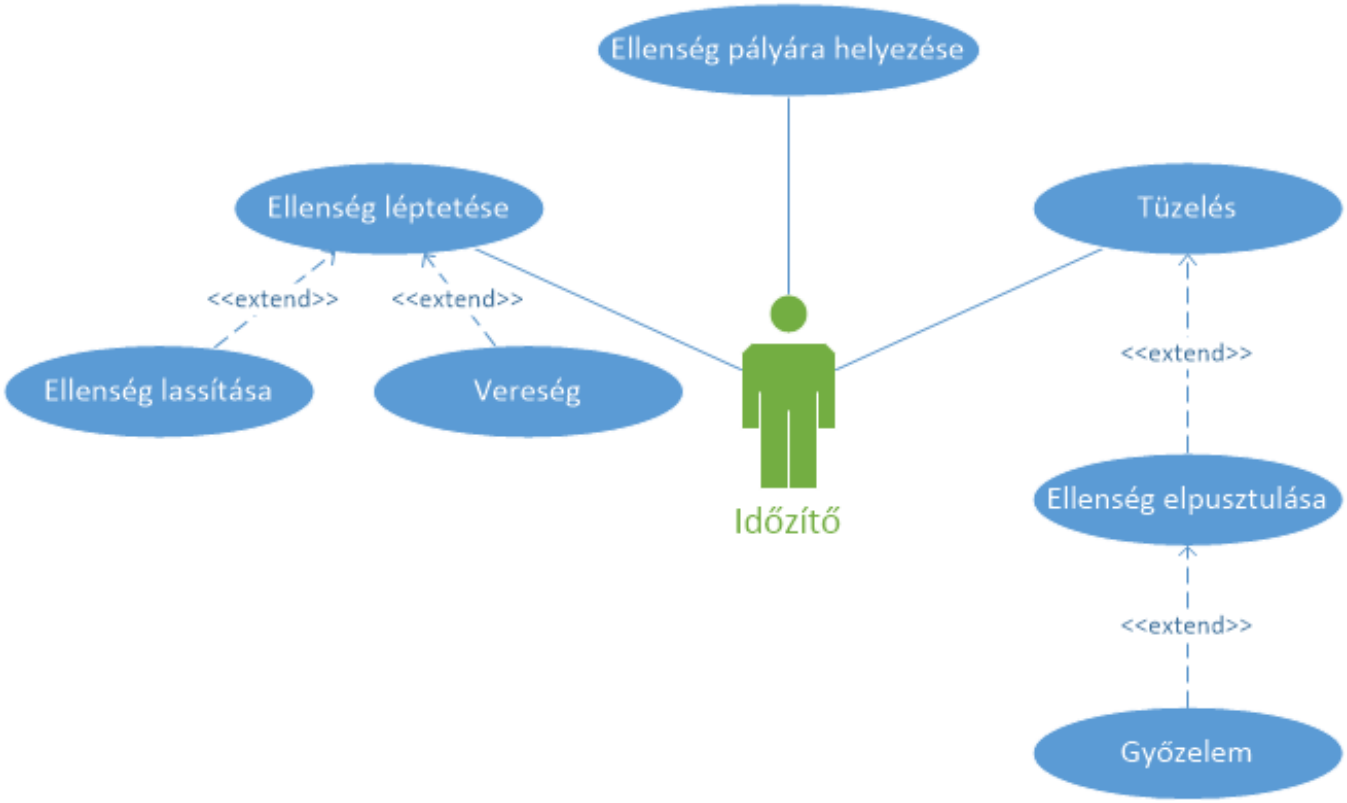
| | | | |
|---|---|---|---|
| 18:00 | | | |
| 2014.03.09. 20:15 | 2 óra | Török | 4.4.9 szekvencia diagram elkészítése |
| 2014.03.09. 22:15 | 1 óra | Csontos Török | Objektum katalógus módosítása |
| 2014.03.09. 23:15 | 1,5 óra | Török | Dokumentáció, szekvenciák javítása |
| 2014.03.10 00:30 | 1 óra | Bana | Dokumentáció átnézése, észrevételek lejegyzése. Szekvencia diagram 4.4.7 átformázása |

# 5. Szkeleton tervezése

## 5.1 A szkeleton modell valóságos use-case-ei

### 5.1.1 Use-case diagram




Ellenség pályára helyezése
Ellenség léptetése
Tüzelés
<<extend>>
<<extend>>
<<extend>>
Ellenség lassítása
Vereség
Időzítő
Ellenség elpusztulása
<<extend>>
Győzelem

### 5.1.2 Use-case leírások

| **Use-case neve** | Indítás |
|---|---|
| **Rövid leírás** | A játékos új játékot kezdeményez. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A játékos a játékindítást választja a program indítását követően. Betöltődik a pálya, első pillanatban még nincs ellenség, torony vagy akadály a pályán. |

| **Use-case neve** | Torony építés |
|---|---|
| **Rövid leírás** | A játékos egy új torony pályára építését kezdeményzi. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Ha a játékos egy olyan csempére kattint, ahova építhető torony, akkor toronyépítést kezdeményezhet. Ha a játékosnak van elegendő varázsereje a toronyépítéshez, akkor a torony megépül. Ha nincs elég varázsereje, akkor a torony nem épül meg, és értesíti a játékost a varázserőhiányról. |

| **Use-case neve** | Akadály építés |
|---|---|
| **Rövid leírás** | A játékos egy új akadály építését kezdeményezi a pályára. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Ha a játékos egy út-csempére kattint, akkor akadályépítést kezdeményezhet. Ha a játékosnak van elegendő varázsereje az akadályépítéshez, akkor az akadály megépül. Ha nincs elég varázsereje, akkor az akadály nem épül meg, és értesíti a játékost a varázserőhiányról. |

| **Use-case neve** | Torony fejlesztés |
|---|---|
| **Rövid leírás** | A játékos a kiválasztott torony fejlesztését kezdeményezi varázserőért cserébe. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Ha a játékos kiválaszt egy már meglévő toronyt, fejlesztési lehetőségeket kap. Meglévő varázskővel a játékos a toronyt a varázskő típusának megfelelő képességgel ruházza fel. |

| **Use-case neve** | Akadály fejlesztés |
|---|---|
| **Rövid leírás** | A játékos a kiválasztott akadály fejlesztését kezdeményezi varázserőért cserébe. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Ha a játékos kiválaszt egy már meglévő akadályt, fejlesztési lehetőséget kap. Egy fajta fejlesztés van, a lassítás faktorának növelése. |

| **Use-case neve** | Ellenség pályára helyezése |
|---|---|
| **Rövid leírás** | Az ellenség út-csempére helyezése. |
| **Aktorok** | Időzítő |
| **Forgatókönyv** | Időzítő a passzív ellenségek egy csoportját aktivizálja és ráteszi őket a forrás csempére. Ezen ellenségek már készen állnak a léptetésre. |

| **Use-case neve** | Ellenség léptetése |
|---|---|
| **Rövid leírás** | Az ellenségek út-csempén történő előrefelé haladása. |
| **Aktorok** | Időzítő |
| **Forgatókönyv** | A pályára helyezett ellenségeket léptetjük az út-csempéken. Egy léptetés megfelel egy aktuális út-csempéről a szomszédos út-csempére való lépéssel. A léptetés periodikus időközönként történik, az ellenségek szinkronban mozognak. |

| **Use-case neve** | Ellenség lassítása |
|---|---|
| **Rövid leírás** | Az ellenség akadályra lépése. |
| **Aktorok** | Időzítő |
| **Forgatókönyv** | Ha egy ellenség léptetése során akadályra (speciális út-csempe) lép, akkor lassítva lesz. A lassítás bizonyos számú léptetés kihagyását jelenti, amely az akadály fejlettségétől függ. |

| **Use-case neve** | Vereség |
|---|---|
| **Rövid leírás** | Az ellenség hegyre lépése. |
| **Aktorok** | Időzítő |
| **Forgatókönyv** | Ha egy ellenség léptetése során hegyre (speciális út-csempe) lép, akkor a játék vereséggel véget ér. |

| **Use-case neve** | Tüzelés |
|---|---|
| **Rövid leírás** | Ellenség torony általi sebzése. |
| **Aktorok** | Időzítő |
| **Forgatókönyv** | A játékos által elhelyezett tornyok periodikus időközönként sebzik a hatótávolságukon belül tartózkódó ellenségek valamelyikét. Az ellenség sebzésének hatékonysága függ a torony fejlettségétől. |

| **Use-case neve** | Ellenség elpusztulása |
|---|---|
| **Rövid leírás** | Egy ellenség torony általi végzetes sebzése. |
| **Aktorok** | Időzítő |
| **Forgatókönyv** | Ha egy ellenség sebzése során életereje 0 vagy az alá csökken, akkor elpusztul és eltűnik a pályáról. |

| **Use-case neve** | Győzelem |
|---|---|
| **Rövid leírás** | Összes ellenség elpusztul. |
| **Aktorok** | Időzítő |
| **Forgatókönyv** | Ha az összes ellenség elpusztul, akkor a játék győzelemmel ér véget. |

# 5.2 A szkeleton kezelői felületének terve, dialógusok

A szkeleton kezelői felülete karakteres lesz. Indításkor egy menüben fel lesznek sorolva az egyes use-case-ek és egy kilépési lehetőség. A felhasználó a menüponthoz megadott azonosító megadásával tudja lefuttatni a kiválasztott use-case-t ill. kilépni.

## Menü felépítése

1. Indítás
2. Torony építés
3. Akadály építés
4. Torony fejlesztés
5. Akadály fejlesztés
6. Ellenség pályára helyezése
7. Ellenség léptetése
8. Ellenség lassítása
9. Vereség
10. Tüzelés
11. Ellenség elpusztulása
12. Győzelem
0. Kilépés

A kiválasztott use-case létrehozza a szükséges objektumokat és a közöttük lévő kapcsolatokat.

**A kiválasztott use-case az alábbiakat fogja kiírni a felületre:**

1. meghívott függvény: jobbra mutató nyíl fogja jelölni
2. visszatért függvény: balra mutató nyíl fogja jelölni
3. egymásba ágyazott hívások tabulátorokkal jelezve
4. döntési helyzet idézőjelek között

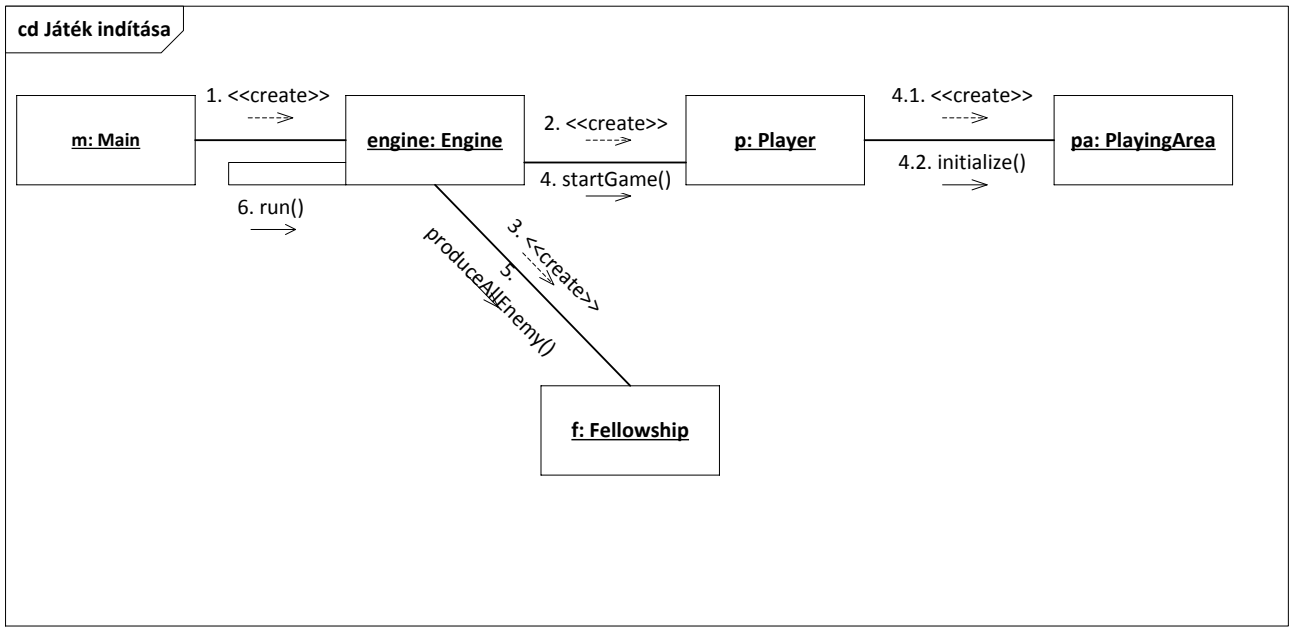
Döntési helyzetnél a felületre egy idézőjelek közötti kérdés íródik ki a lehetséges opciókkal együtt. Ezek közül kell a felhasználónak pontosan egyet kiválasztania. Ha a felhasználót több opció eredménye is érdekli, akkor újra kell futtatni az aktuális use-case-t.

# 5.3 Szekvencia diagramok a belső működésre

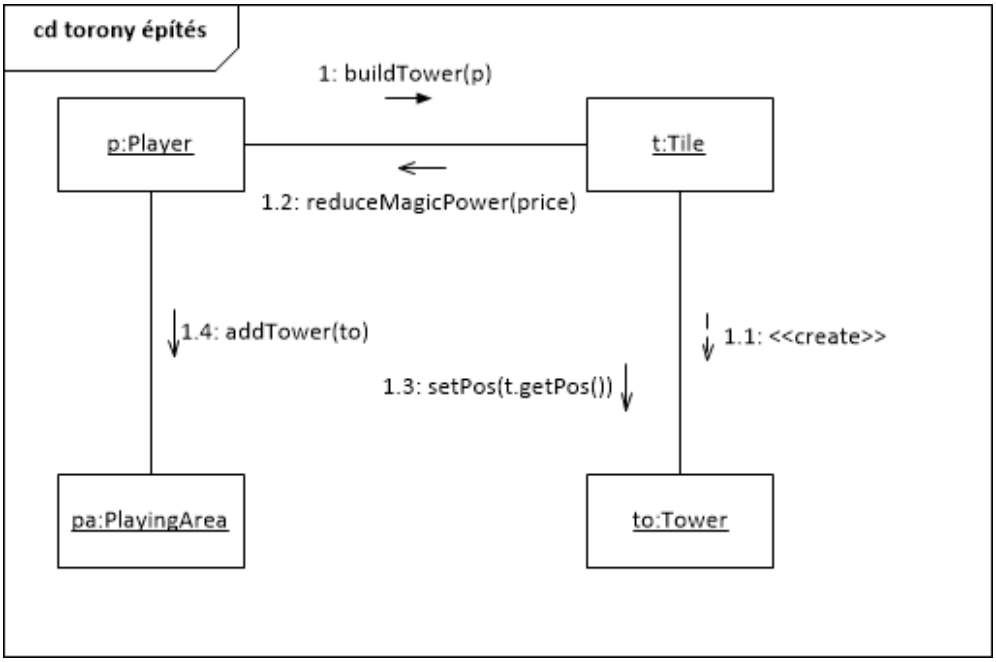
A szekvencia diagramok részletes leírása az Analízis 2. dokumentum 4.4. pontjában találhatóak meg.

# 5.4 Kommunikációs diagramok

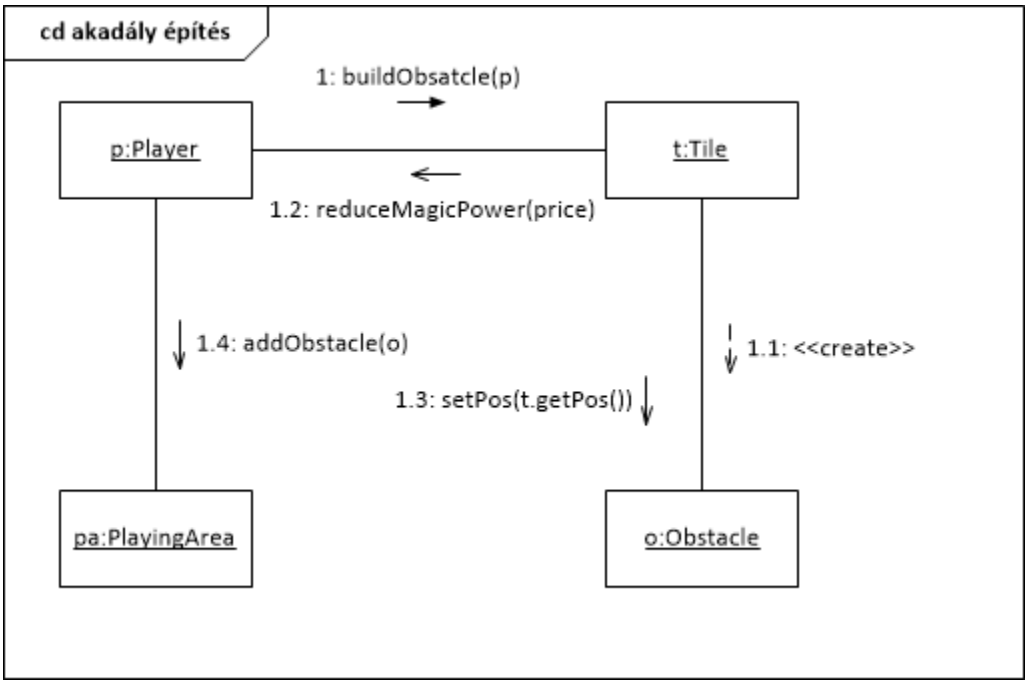
## 5.4.1 JÁTÉK INDÍTÁSA



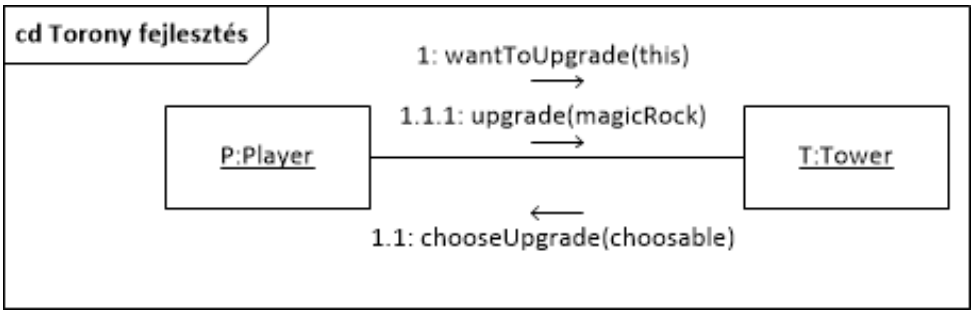
## 5.4.2 TORONY ÉPÍTÉS



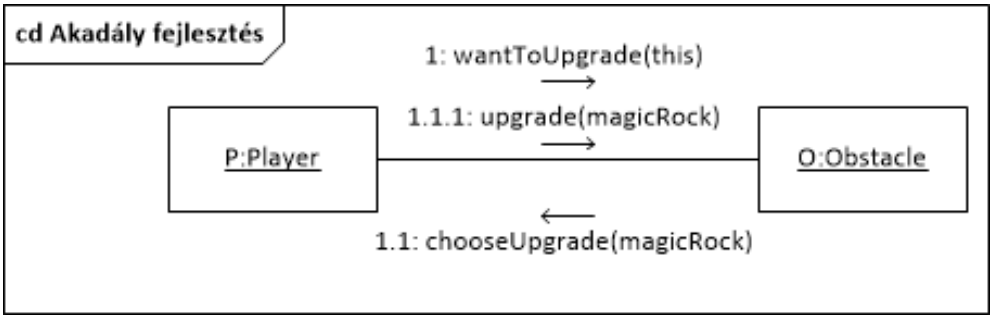

## 5.4.3 AKADÁLY ÉPÍTÉS



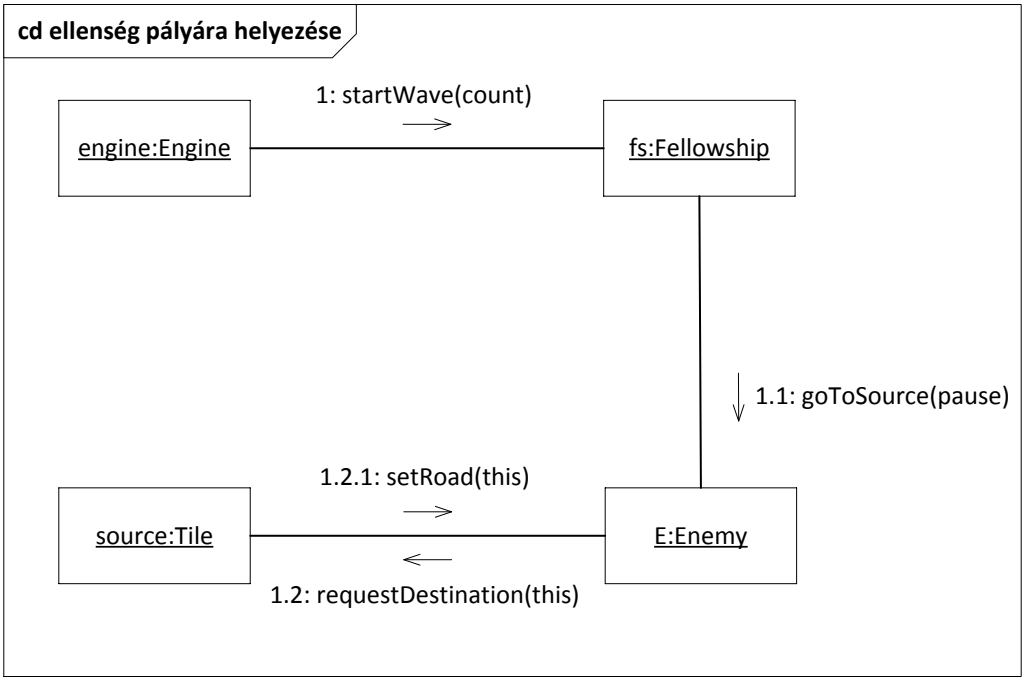
### 5.4.4 TORONYFEJLESZTÉS



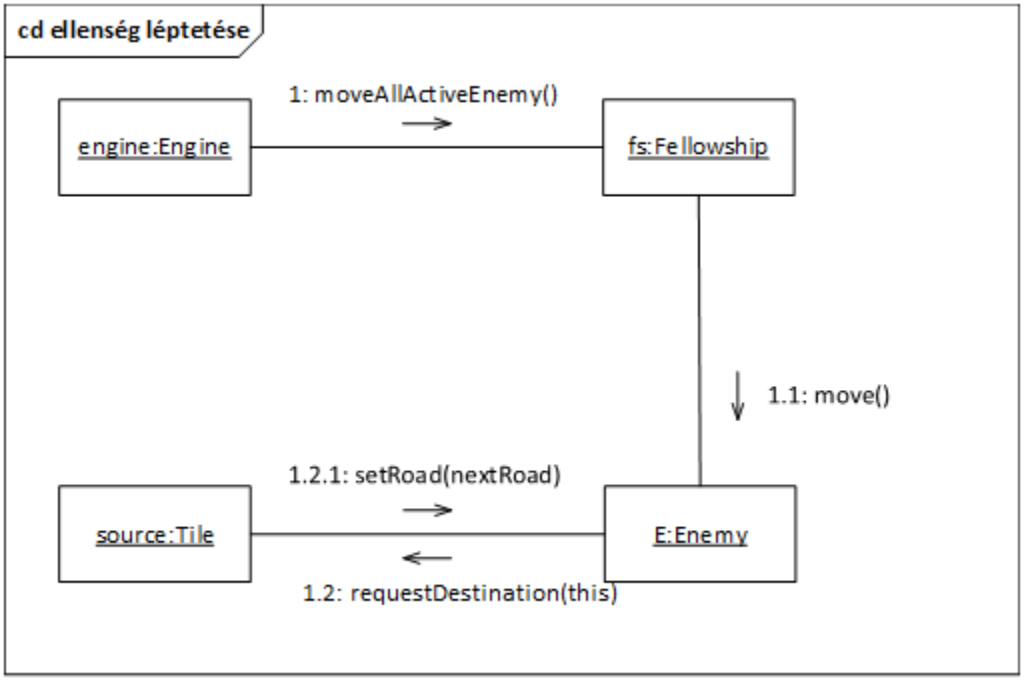

### 5.4.5 AKADÁLY FEJLESZTÉS



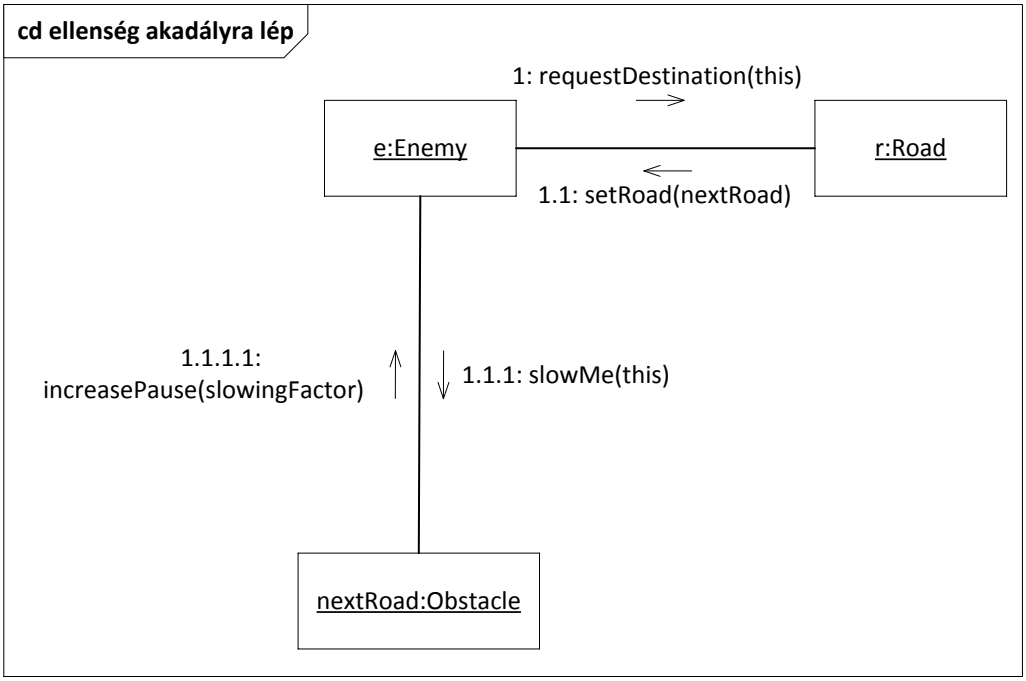
### 5.4.6 ELLENSÉG PÁLYÁRA HELYEZÉSE



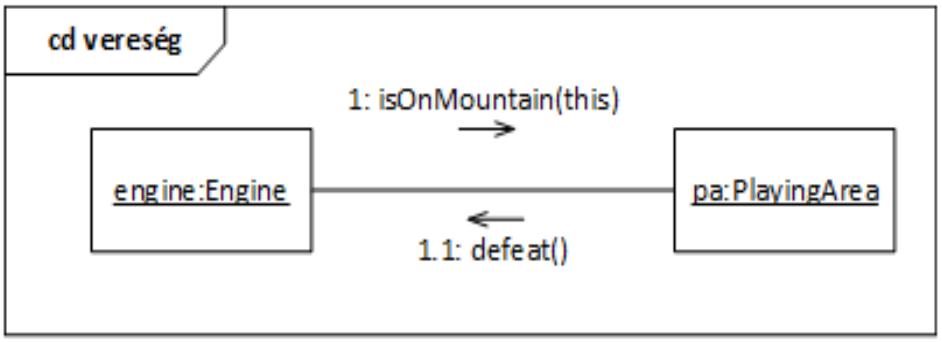

### 5.4.7 ELLENSÉG LÉPTETÉSE



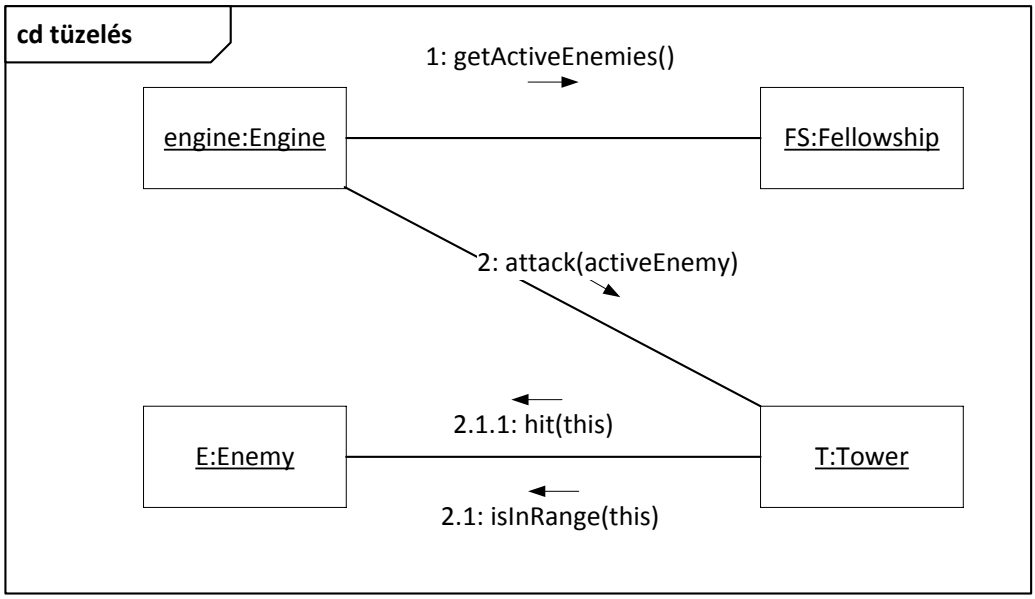
### 5.4.8 ELLENSÉG LASSÍTÁSA



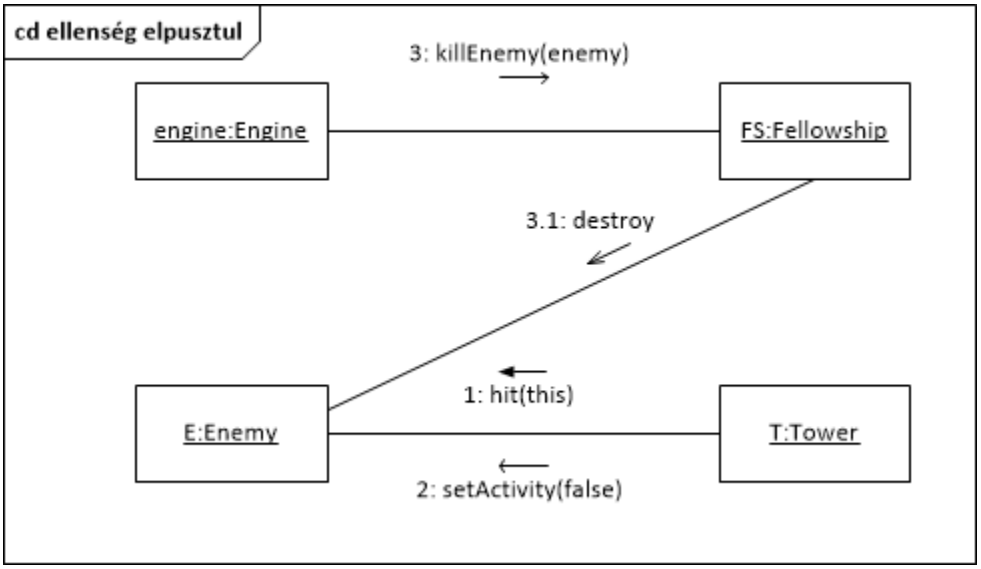

### 5.4.9 VERESÉG



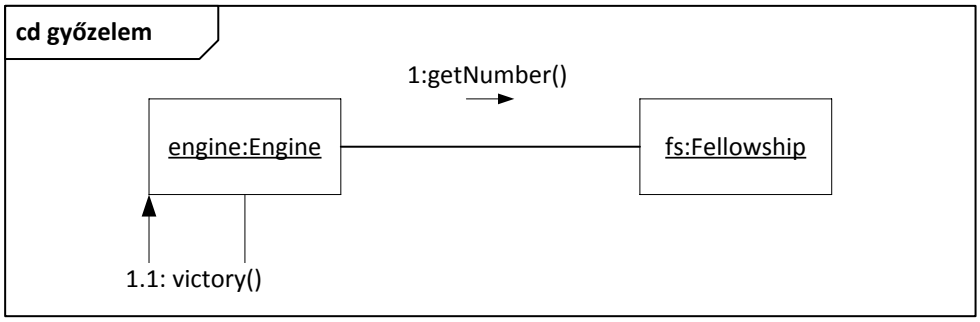
### 5.4.10 TÜZELÉS




### 5.4.11 ELLENSÉG ELPUSZTULÁSA



### 5.4.12 GYŐZELEM



## 5.5 Napló

| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.03.12. 18:00 | 2 óra | Török<br>Bana<br>Tóth<br>Csontos<br>Czirják | Értekezlet - Skype megbeszélés: konzultáció kiértékelése |
| 2014.03.13. 18:00 | 2 óra | Bana<br>Tóth<br>Csontos<br>Czirják | Szekvencia és osztálydiagram módosítások megbeszélése |
| 2014.03.14 10:00 | 3 óra | Bana<br>Tóth<br>Csontos<br>Czirják<br>Török | Szekvencia diagramok elkészítése |
| 2014.03.14 13:00 | 3 óra | Bana<br>Tóth<br>Csontos<br>Czirják | Szekvencia diagramok elkészítése |
| 2014.03.14. 16:00 | 1 óra | Czirják<br>Csontos | Use-case diagram tervezése |
| 2014.03.14 22:30 | 0.5 óra | Bana | 5.4.7, 5.4.8 diagramok formázása |
| 2014.03.15 16:00 | 2 óra | Török<br>Bana<br>Tóth<br>Csontos<br>Czirják | Értekezlet - Skype:<br>- szekvencia és kommunikációs diagramokkal kapcsolatos feladatok elosztása |
| 2014.03.15 18:00 | 30 perc | Czirják | Ellenség léptetése és ellenség pályára helyezése kommunikációs diagramok elkészítése |

| | | | |
|---|---|---|---|
| 2014.03.15<br>20:00 | 30 perc | Tóth | Kommunikációs diagramok (Torony, akadály fejlesztés, Torony tüzelés) |
| 2014.03.16<br>16:00 | 2.5 óra | Bana | Kommunikációs diagramok:<br>5.4.2, 5.4.3, 5.4.8<br>Szekvencia diagramok:<br>5.3.2, 5.3.3, 5.3.8 |
| 2014.03.16<br>18:30 | 1 óra | Török<br>Bana<br>Tóth<br>Csontos<br>Czirják | Értekezlet - Skype:<br>- szkeleton letisztázása<br>- szekvencia diagramok átbeszélése<br>- kommunikációs diagramok átbeszélése |
| 2014.03.16<br>02:00 | 1 óra | Czirják | A Játékos táblázatos use-case-einek elkészítése |
| 2014.03.16<br>12:00 | 4.5 óra | Török | Dokumentáció, 5.3.1., 5.3.9, 5.4.1., 5.4.9. elkészítése |
| 2014.03.16<br>18:00 | 2 óra | Csontos | Időzítő aktorhoz tartozó use-case-ek leírása<br>Szkeleton kezelői felületének terve |
| 2014.03.16<br>21:00 | 1 óra | Csontos | 5.4.1 kommunikációs diagram<br>szekvencia- és kommunikációs diagramok közötti konzisztencia vizsgálat és javítás |
| 2014.03.16<br>23:00 | 1 óra | Csontos | Dokumentum formázása |
| 2014.03.17<br>01:00 | 1 óra | Tóth<br>Csontos | 5.3.11, 5.3.12 szekvenciák és<br>5.4.11, 5.4.12 kommunikációs diagramok elkészítése |

# 6. Szkeleton beadás

## 6.1 Fordítási és futtatási útmutató

### 6.1.1 Fájllista

| Fájl neve | Méret [Byte] | Keletkezés ideje | Tartalom |
|---|---|---|---|
| Defense.java | 406 | 2014.03.22 | |
| Dwarf.java | 146 | 2014.03.22 | |
| Elf.java | 144 | 2014.03.22 | |
| Enemy.java | 1970 | 2014.03.22 | |
| Engine.java | 1064 | 2014.03.22 | |
| Fellowship.java | 2117 | 2014.03.22 | |
| Geometry.java | 492 | 2014.03.22 | |
| Hobbit.java | 147 | 2014.03.22 | |
| Human.java | 146 | 2014.03.22 | |
| MagicRock.java | 158 | 2014.03.22 | |
| Main.java | 5997 | 2014.03.22 | a szkeleton állapotgépe |
| Mountain.java | 315 | 2014.03.22 | |
| Obstacle.java | 867 | 2014.03.22 | |
| Player.java | 1146 | 2014.03.22 | |
| PlayingArea.java | 1737 | 2014.03.22 | |
| Position.java | 97 | 2014.03.22 | |

| | | | |
|---|---|---|---|
| Road.java | 454 | 2014.03.22 | |
| Source.java | 319 | 2014.03.22 | |
| Tile.java | 1115 | 2014.03.22 | |
| Tower.java | 1447 | 2014.03.22 | |
| Writer.java | 2564 | 2014.03.22 | hívott függvények kiírása konzolra |
| compile.bat | 348 | 2014.03.23 | .java fájlok fordítása |
| run.bat | 9 | 2014.03.23 | a fordított program futtatása |

### 6.1.2 Fordítás

A .java forrásfájlok mellett található egy compile.bat fájl, ami egyenként lefordítja az összes olyan java osztályt, ami nem absztrakt. Ezt a javac parancsszóval érhetjük el, ami a Command Prompt-ból elérhető, ha megadjuk a JDK helyét. Ezt az alábbi paranccsal tehetjük meg: “set path=%path%; JDK elérési útvonala”.
Ha ezt sikeresen beállítottuk, akkor a compile.bat fájlt futtatva létejönnek a .class fájlok, ugyanazon mappában, ahol a .java fájlok vannak.

### 6.1.3 Futtatás

Ha megtörtént a fordítás, akkor a run.bat fájlt futtatva elindul a program és egy use-case menü jelenik meg, ami a felhasználától egy számot kér. A run.bat fájlon belül azt a java osztályt kell megadni, amelyikben a public static void main(String[] args) metódus található.

## 6.2 Értékelés

| Munka százalékban | Tag neve |
|---|---|
| 26 | Bana Szabolcs |
| 15 | Czirják Balázs |
| 24 | Csontos Valentin |
| 17 | Tóth Dávid |
| 18 | Török Odett |

## 6.3 Napló

| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.03.20. 14:00 | 1 óra | Török<br>Tóth<br>Csontos | Értekezlet. Heti feladat részletes áttekintése, előző heti feladat hibáinak kijavítása. |
| 2014.03.20. 15:00 | 1 óra | Török<br>Tóth<br>Csontos<br>Bana | Értekezlet. Heti feladat részletes áttekintése, előző heti feladat hibáinak kijavítása és kiiírás megoldásáról való ötletelés. |
| 2014.03.20. 16:00 | 2 óra | Csontos<br>Bana | Értekezlet. Heti feladat részletes áttekintése, előző heti feladat hibáinak kijavítása és kiiírás megoldásáról való ötletelés. |
| 2014.03.22. 17:30 | 5 óra | Csontos<br>Bana | Értekezlet.<br>- Szkeleton működésének kitalálása,<br>- hívott függvények kiiíratásának megoldása.<br>- 1-es use-case kiiratásának elkészítése |
| 2014.03.22. 23:30 | 4 óra | Bana | Fellowship és PlayingArea osztály implementálása és javadoc formázása a szkeletonhoz. |
| 2014.03.22. 23:30 | 3 óra | Csontos | Engine, Tile, Obstacle, Main osztályok skeletonjának elkészítése<br>Menü megírása |
| 2014.03.23. 00:00 | 2 óra | Tóth | Tower, Geometry, Position osztályok megvalósítása |
| 2014.03.23. 8:00 | 2 óra | Török | Source, Mountain, Defense, MagicRock, Spreadsheet, Dokumentáció |
| 2014.03.23. 14:00 | 3 óra | Csontos | Indítás, Ellenség léptetése, Ellenség pályára |

| | | | |
|---|---|---|---|
| | | | helyezése, Ellenség elpusztulása |
| 2014.03.23. 15:30 | 11,5 óra | Bana | - Fellowship, PlayingArea, Enemy, Road osztályok javadoc kommentezése<br>- Ellenség lassítása szekvencia elkészítése a szkeletonban<br>- Vereség szekvencia elkészítése a szkeletonban<br>- Vereség szekvencia diagram javítása<br>- Vereség kommunikációs diagram javítása<br>- Ellenség akadályra lép szekvencia diagram javítása<br>- Ellenség akadályra lép kommunikációs diagram javítása |
| 2014.03.23 15:30 | 0,5 óra | Czirják | Enemy, Elf, Dwarf, Hobbit, Human osztályok implementálása |
| 2014.03.23 16:00 | 3 óra | Tóth | Szekvenciagramoknak megfelelő implementálás |
| 2014.03.23. 16:00 | 4 óra | Török | Toronyépítés és akadályépítés use-case-ek megvalósítása |
| 2014.03.23. 17:00 | 2 óra | Csontos | Torony, akadály építés ill. fejlesztés use-case-ek elkészítésében való aktív részvétel |
| 2014.03.23. 22:00 | 1 óra | Csontos | szkeleton kód kommentezése<br>compile.bat, run.bat elkészítése |
| 2014.03.24. 00:00 | 1,5 óra | Csontos<br>Bana | Szekvencia- és kommunikációs diagramok javítása a szkeleton tervezéshez képest eszközölt módosítások elkészítése a diagramokon |
| 2014.03.24 04:00 | 0,5 óra | Bana | Dokumentum kiegészítése, a projekt kezdete óta végzett munka százalékos lebontásának kiszámítása |

# 7.Prototípus koncepciója

## 7.1 Prototípus interface-definíciója

A prototípus a program egy olyan verziója, mely elkészült és a várakozásoknak megfelelő, helyes működés demonstrálására szolgál. Az alkalmazás funkcióinak teszelésése előredefiniált parancsokon keresztül történik, melyet a parancssorból futtathatunk, mint a szkeletont. A prototípusban a függvényhívásokon kívül már a program belső felépítése, az objektumok metódusai és azok algoritmusai kapnak nagy szerepet.
A tesztelés során, a program kimenetelei előre determináltak lesznek, mivel minden eseményt előre meghatározott parancs formájában adunk meg.
A rendszer véletlen elemeket tartlamaz, így a véletlenszerűség ki-bekapcsolhatóságával tudjuk kezelni a program determinisztikus tesztelhetőségét.

### 7.1.1 Az interfész általános leírása

A felhasználó és a prototípus program a szabványos be- és kimeneten kommunikál egymással, így a parancsokat konzolból vagy parancsfájlból kaphatja, míg a kimenet képernyőre vagy fájlba történik. Egy kommunikációra alkalmas parancssori felületet alakítunk ki, mellyel könnyen ellenőrizhetővé válik a program működése. Használatakor kiadjuk a megfelelő parancsokat és az ahhoz tartozó paramétereket, majd a kijelölt műveletek elvégzése után visszatér az eredménnyel, így ellenőrizve a kívánt működést.

### 7.1.2 Bemeneti nyelv

***buildTower***
**Leírás:** Egy torony építést reprezentál a megadott pozíciójú csempére.
Opciók: buildTower pos

***buildObstacle***
**Leírás:** Egy akadályt építést reprezentál a megadott pozíciójú út-csempére.
**Opciók**: buildObstacle pos

***upgradeTower***
Leírás: A megadott id-jú tornyot fejleszti. A fejlsztés típusát a felhasználó adja meg a konzolon keresztül.
Opciók: upgradeTower id

***upgradeObstacle***
Leírás: A megadott id-jú akadályt fejleszti. A fejlsztés típusát a felhasználó adja meg a konzolon keresztül.
Opciók: upgradeObstacle id

***step***
Leírás: Az idő léptetése a number-nek megfelelő értékkel. (Példa: step 10 - (10 lépés történik)
Opciók: step number

**RANDOM_ON**
Leírás: Randomizálás bekapcsolása. Elágazásoknál és ellenség pályára helyezésénél van jelentősége.
Opciók: -

**RANDOM_OFF**
**Leírás:** A program véletlenszerűségének kikapcsolása.
**Opciók:-**

***fogOn***
**Leírás:** Az összes toronyra köd ereszkedik le.
**Opciók:-**

***fogOff***
**Leírás:** Az összes toronyról felszáll a köd.
**Opciók:-**

**SPLIT_ON**
Leírás: Torony tüzeléskor nemcsak megsebzi az ellenfelet, hanem ketté is lövi.
Opciók: -

***loadMap***
Leírás: Betölt a hivatkozott fájlból egy pályát. Az aktuális pálya felülírásra kerül. A pályaleíró fájlban az alábbi parancsok használhatóak:

-Tower pos: Egy tornyot rak le a meghatározott csempére.
-Obstacle pos: Egy akadályt rak le a meghatározott csempére.
-Source pos: Egy forrást rak le a meghatározott csempére.
-Road pos: Egy út-csempét rak le a meghatározott csempére.
-Mountain pos: Egy hegyet rak le a meghatározott csempére.
-Road ref pos1 pos2: Road pos1 csempét összeköti a pos2-vel.
-Obstacle ref pos1 pos2: Obstacle pos1 csempét összeköti a pos2-vel.
-Source ref pos1 pos2: Source pos1 csempét összeköti a pos2-vel.
-Mountain ref pos1 pos2: Mountain pos1 csempét összeköti a pos2-vel.
-Enemy pos: Egy ellenséget rak le a meghatározott csempére.

Minden parancsot külön sorba kell írni.

Opciók: loadMap filename

***getTowerInfo***
**Leírás:** A megadott id-jú torony aktuális állapotát kilistázza.
**Opciók:** *getTowerInfo id*

***getObstacleInfo***
**Leírás:** A megadott id-jú akadály aktuális állapotát kilistázza.
**Opciók:** *getObstacleInfo id*

***getEnemyInfo***
**Leírás:** A megadott id-jú ellenség aktuális állapotát kilistázza.
**Opciók:** getEnemyInfo id

***getPlayerInfo***
**Leírás:** A játékos aktuális állapotát kilistázza.
**Opciók: -**

***exit***
**Leírás:** A programból való kilépésre szolgál**.**
**Opciók:-**

### 7.1.3 Kimeneti nyelv

<[ID:Objektum típusa]> has <Esemény leírása és a paraméterei>.
Példa:
[1:Tower] has shot an enemy on the Road(Position (3,6)).

Rossz paraméter vagy hibásan megadott parancs esetén: <Error: hiba leírása>
Példa:
Error: Invalid command

A pályán található objektum paramétereit szövegesen jelenítjük meg, az alábbi nyelvtan szerint:
<[ID:Objektum típusa]>
<Paraméter neve:Paraméter értéke>
Példa:
[2:Human]
position:(10,20)
pause:0
lifePower:50

## 7.2 Összes részletes use-case

| | |
|---|---|
| **Use-case neve** | Torony építése |
| **Rövid leírás** | A felhasználó a megadott pozícióra épít egy új tornyot. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Ez a ***buildTower*** *paranccsal* végezhető el, paraméterként a torony helyét kell megadni. |

| | |
|---|---|
| **Use-case neve** | Akadály építése |
| **Rövid leírás** | A felhasználó a megadott útpozícióra épít egy új akadályt. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Ez a ***buildObstacle*** paranccsal végezhető el, paraméterként az akadály helyét kell megadni. |

| **Use-case neve** | Torony fejlesztés |
|---|---|
| **Rövid leírás** | A kiválasztott torony valamely képességének fejlesztése. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Az ***upgradeTower*** paranccsal végezhető el, a kiválasztott torony a megadott képessége felfejlődik. |

| **Use-case neve** | Akadály fejlesztés |
|---|---|
| **Rövid leírás** | A kiválasztott akadály fejlesztése. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Az ***upgradeObstacle*** paranccsal végezhető el, a kiválasztott akadály lassítási képessége fejlődik. |

| **Use-case neve** | Léptetés |
|---|---|
| **Rövid leírás** | A program léptetése kézzel. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A ***step*** paranccsal hajtható végre, annyi lépést fog végrehajtani a program, amennyit paraméterül kap a függvény. |

| **Use-case neve** | Torony információ |
|---|---|
| **Rövid leírás** | Adott torony tulajdonságait listázza ki. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A ***getTowerInfo*** paranccsal kilistázódik az adott toronyra vonatkozó információk listája. |

| **Use-case neve** | Akadály információ |
|---|---|
| **Rövid leírás** | Adott akadály tulajdonságait listázza ki. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A ***getObstacleInfo*** paranccsal kilistázódik az adott akadályra vonatkozó információk listája. |

| **Use-case neve** | Ellenség információ |
|---|---|
| **Rövid leírás** | Adott ellenség tulajdonságait listázza ki. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A ***getEnemyInfo*** paranccsal kilistázódik az adott akadályra vonatkozó információk listája. |

| **Use-case neve** | Játékos információ |
|---|---|
| **Rövid leírás** | Játékos tulajdonságait listázza ki. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A ***getPlayerInfo*** paranccsal kilistázódik a játékosra vonatkozó információk listája. |

| **Use-case neve** | Randomizálás bekapcsolása |
|---|---|
| **Rövid leírás** | A program determinisztikus futását kapcsolja ki. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Ha a tesztelő meghívja a **RANDOM_ON** *parancsot*, a program futásában véletlenszerűen lefolyó folyamatok valóban véletlenszerűen futnak majd le. |

| **Use-case neve** | Randomizálás kikapcsolása |
|---|---|
| **Rövid leírás** | A program determinisztikus futását kapcsolja be. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Ha a tesztelő meghívja a **RANDOM_OFF** *parancsot*, a program futásában véletlenszerűen lefolyó folyamatok determinisztikusan futnak majd le. |

| **Use-case neve** | Toronyra köd leszáll |
|---|---|
| **Rövid leírás** | A toronyra leereszkedik a köd. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A ***fogOn*** *paranccsal* hatására a tornyokra köd ereszkedik, a lőtávolságuk lecsökken. |

| **Use-case neve** | Toronyról a köd felszáll |
|---|---|
| **Rövid leírás** | A tornyokról felszáll a köd |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A ***fogOff*** *paranccsal* hatására a tornyokról a köd felszáll, a lőtávolságuk visszaáll az eredeti értékre. |

| **Use-case neve** | Pálya betöltése |
|---|---|
| **Rövid leírás** | A tesztelő fájlból betölti a pályát. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A ***loadMap*** *parancs* hatására a megadott nevű pálya betöltődik. |

| **Use-case neve** | Ellenség kettészakítása |
|---|---|
| **Rövid leírás** | A torony tüzelése során kettévágja a megsebzett ellenséget, azaz klónozza és az újannan keletkezett ellenség is a pályára kerül, méghozzá ugyanarra a csempére, amelyiken a kettélőtt áll. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A **SPLIT_ON** parancs megadásával a torony tüzeléskor mindig kettévágja az ellenséget. |

| **Use-case neve** | Kilépés |
|---|---|
| **Rövid leírás** | A programból való kilépésre szolgál. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | **exit** parancs hatására hajtódik végre |

## 7.3 Tesztelési terv

| | |
|---|---|
| **Teszt-eset neve** | A pálya beolvasása |
| **Rövid leírás** | A program fájlból beolvas egy pályát. |
| **Teszt célja** | Leelenőrizni, hogy a pálya sikeresen beolvasható. |

| | |
|---|---|
| **Teszt-eset neve** | Az ellenség a következő csempére lép |
| **Rövid leírás** | Az ellenséget az ütemező hatására léptetjük, melynek hatására a következő út csemére lép a pályán. |
| **Teszt célja** | A lépés algoritmusának ellenőrzése és a megvalósulás leellenőrzése. |

| | |
|---|---|
| **Teszt-eset neve** | Az ellenség akadályra lép |
| **Rövid leírás** | Az ellenség következő út-csempéjére lép, mely út csempén egy akadály van. |
| **Teszt célja** | Teszteljük, hogy ellenség akadályra lépésekor valóban késleltetve lép-e tovább az adott ellenség. |

| | |
|---|---|
| **Teszt-eset neve** | Toronyépítés |
| **Rövid leírás** | A játékos toronyépítést kezdeményez érvényes területre, elegendő varázserővel a prototípus ‘buildTower pos parancsának segítségével. |
| **Teszt célja** | Teszteljük, hogy a játékos a toronyépítés kezdeményezésére megfelelő jogosultság birtokában (elegendő varázserő és érvényes terület) felépíti-e a tornyot. |

| | |
|---|---|
| **Teszt-eset neve** | Torony tüzel |
| **Rövid leírás** | Egy torony kiválaszt egy, a hatótávolságában lévő ellenséget, melyre tüzel. |
| **Teszt célja** | Célja, hogy megnézzük, hogy a kiválasztott ellenségre tüzeléskor valóban |

| | |
|---|---|
| | csökken-e az ellenség életereje. |

| | |
|---|---|
| **Teszt-eset neve** | Torony fejlesztése |
| **Rövid leírás** | Egy adott toronyra megfelelő varázserő birtokában egy fejlesztést hajtunk végre a proto ‘upgradeTower’ parancsának segítségével, majd a fejlesztett toronnyal újra tüzelünk. |
| **Teszt célja** | A fejlesztés funkció működésének ellenőrzése, azaz a speciális képesség megvalósulásának ellenőrzése. (például eddigi hatótávolságnál távolabb lévő ellenségekre való tüzelés) |

| | |
|---|---|
| **Teszt-eset neve** | Ellenség elpusztul |
| **Rövid leírás** | Egy alacsony életerejű ellenségre úgy tüzelünk, hogy annak életereje biztosan 0 alá csökkenjen. |
| **Teszt célja** | Annak ellenőrzése, hogy egy ellenség életerejének 0-ra csökkenésekor valóban megszűnik-e az ellenség. |

| | |
|---|---|
| **Teszt-eset neve** | A tornyokra köd ereszkedik |
| **Rövid leírás** | Az engine véletlenszerű köd ereszkedést sorsolt, melynek során a tornyok látóviszonya csökken, így a hatátóvolságuk csökken. |
| **Teszt célja** | A köd okozta hatótávolságcsökkenés ellenőrzése. |

| **Teszt-eset neve** | Az ellenség megkettőződik |
|---|---|
| **Rövid leírás** | Az ellenségre egy torony tüzel, melynek hatására éppen az ellenség kettészakad, és a játék így folytatódik tovább. |
| **Teszt célja** | A teszt célja e viszonylag komplex funkció működésének ellenőrzése. Egyik várt eredmény, hogy a lövés hatására az adott út csempén két ellenség lesz. A másik várt funkció, hogy a megkettőződés során az egyik ellenség később indul el a következő csempére, mint a másik. |

| **Teszt-eset neve** | Akadályépítés |
|---|---|
| **Rövid leírás** | A játékos akadályt épít egy adott út csempére megfelelő varázserő birtokában, a proto 'buildObstacle pos parancsával. |
| **Teszt célja** | Az akadály megépülésének ellenőrzése a pályán. |

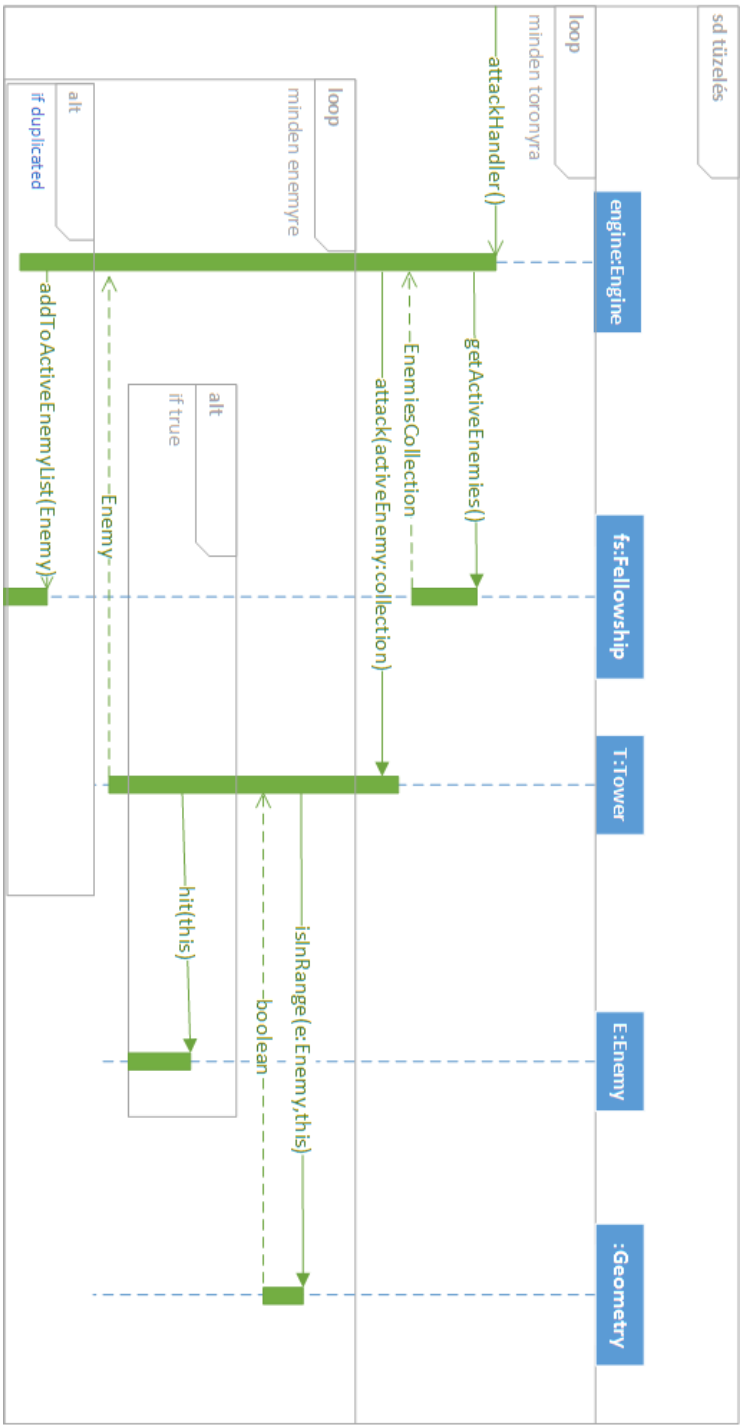
| **Teszt-eset neve** | A játékos győzelmet arat |
|---|---|
| **Rövid leírás** | Az utolsó ellenség elpusztul egy lövés során, ami a Játékos győzelmét eredményezi. |
| **Teszt célja** | Annak ellenőrzése, hogy a játék befejeződik, és a játékos erről értesítést kap. |

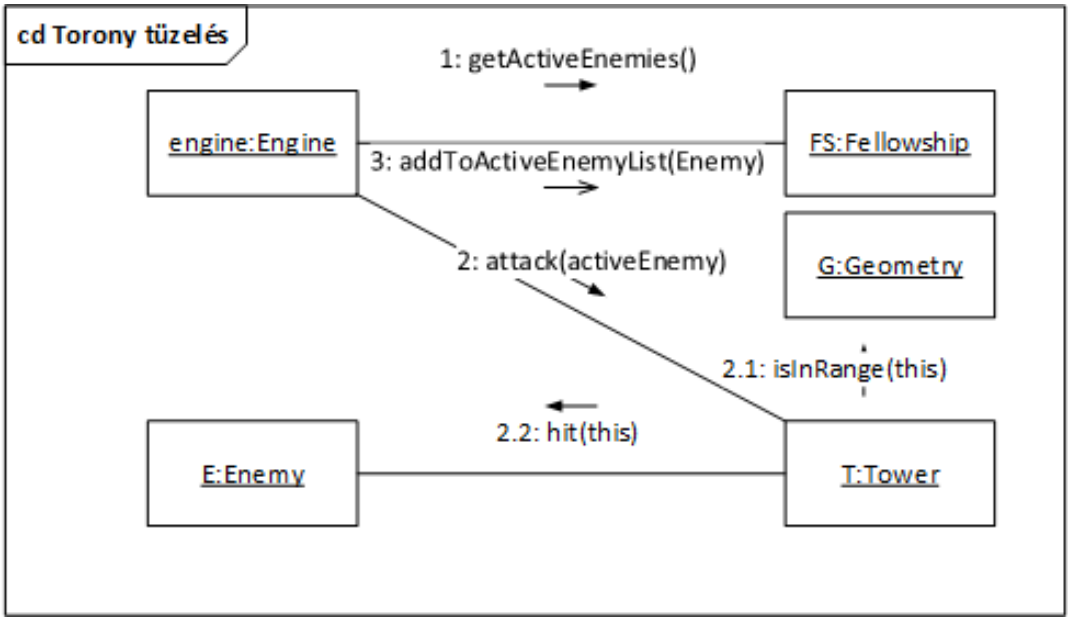
## 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A teszt program lefuttatja a bemeneti fájlban megadott parancsokat, majd létrehoz egy kimeneti fájlt, amiben a legutóbb lefutatott teszteset eredménye lesz benne. Az összehasonlító program megkapja a kimenetként kapott fáljt és a várt eredményt tartalmazó fájlt, a kimenetén pedig eredményt szolgáltat a teszt sikerességéről/sikertelenségéről.

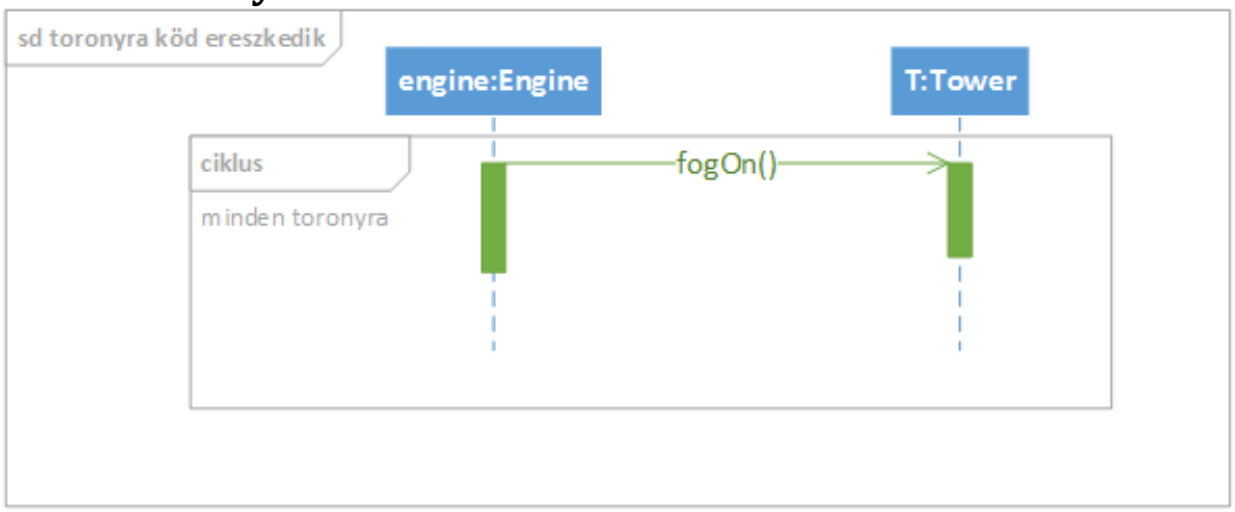
## 7.5 Specifikációmódosítás miatti változtatások

## 7.5.1: Ellenség kettészakad

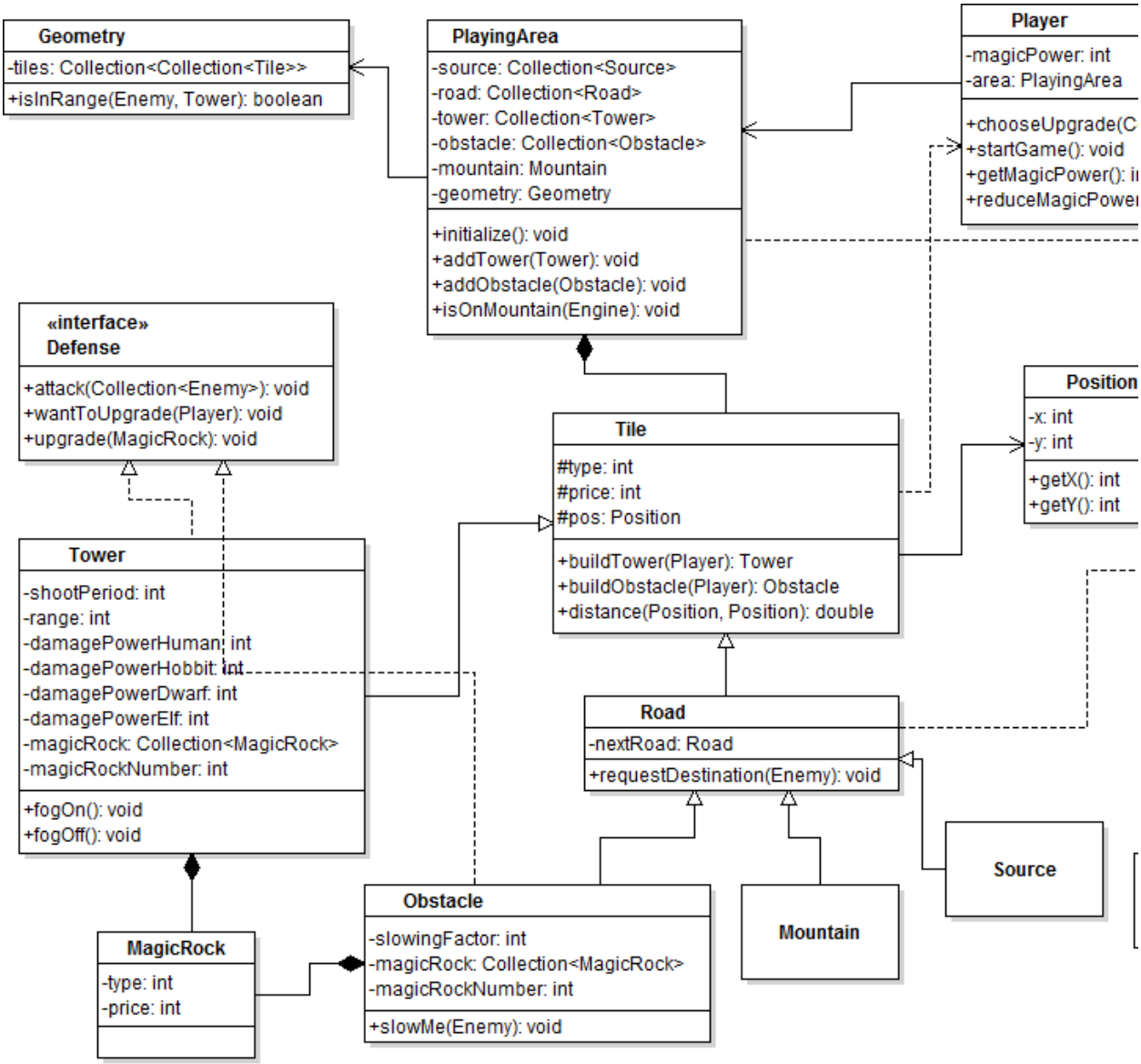


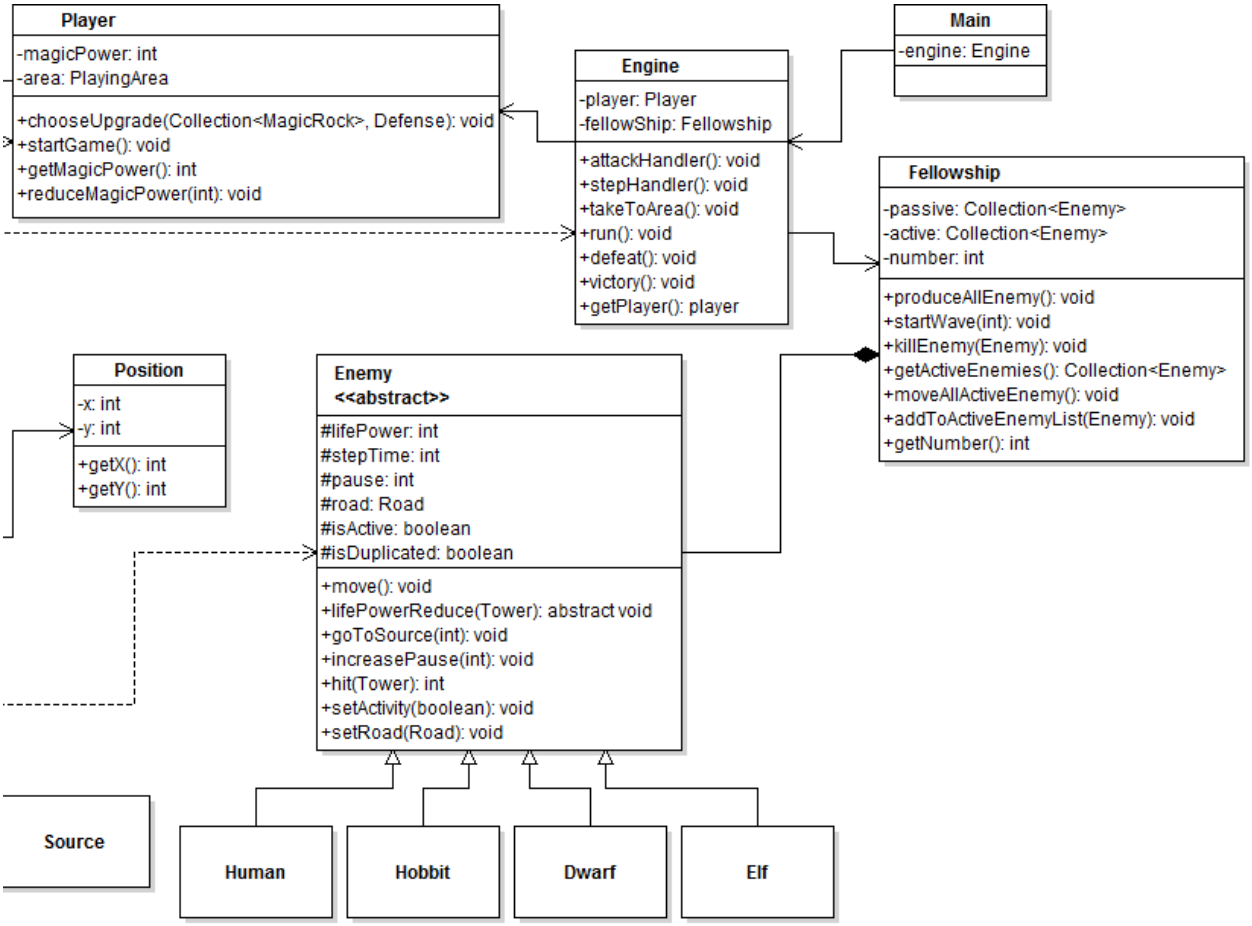



## 7.5.2: A tornyokra köd ereszkedik:




## 7.5.3: Módosult osztálydiagram




Player
-magicPower: int
-area: PlayingArea
+chooseUpgrade(Collection<MagicRock>, Defense): void
+startGame(): void
+getMagicPower(): int
+reduceMagicPower(int): void
Engine
-player: Player
-fellowShip: Fellowship
+attackHandler(): void
+stepHandler(): void
+takeToArea(): void
+run(): void
+defeat(): void
+victory(): void
+getPlayer(): player
Main
-engine: Engine
Fellowship
-passive: Collection<Enemy>
-active: Collection<Enemy>
-number: int
+produceAllEnemy(): void
+startWave(int): void
+killEnemy(Enemy): void
+getActiveEnemies(): Collection<Enemy>
+moveAllActiveEnemy(): void
+addToActiveEnemyList(Enemy): void
+getNumber(): int
Position
-x: int
-y: int
+getX(): int
+getY(): int
Enemy
<<abstract>>
#lifePower: int
#stepTime: int
#pause: int
#road: Road
#isActive: boolean
#isDuplicated: boolean
+move(): void
+lifePowerReduce(Tower): abstract void
+goToSource(int): void
+increasePause(int): void
+hit(Tower): int
+setActivity(boolean): void
+setRoad(Road): void
Source
Human
Hobbit
Dwarf
Elf

## 7.6 Napló

| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.03.25. 17:30 | 3 óra | Bana<br>Csontos<br>Czirják<br>Tóth<br>Török | Értekezlet:<br>A specifikáció módosításainak átbeszélése. Megterveztük mit kell változtatnuk a szoftver eddigi felépítésén a változtatások eszközöléséhez. |
| 2014.03.26. 12:00 | 2 óra | Tóth<br>Csontos<br>Czirják | Szkeleton tesztelés |
| 2014.03.27. 14:30 | 1,5 óra | Török<br>Tóth<br>Csontos | Értekezlet |
| 2014.03.27. 20:00 | 1 óra | Török | 7.1, 7.1.1. |
| 2014.03.29. | 1 óra | Tóth | 7.2 |
| 2014.03.30. | 30 perc | Tóth | 7.2 |
| 2014.03.30. 20:00 | 1 óra | Bana<br>Csontos<br>Czirják<br>Tóth<br>Török | Értekezlet - Skype:<br>A heti feladatok átnézése, az eddigi előrehaladás összeegyeztetése, a felmerülő problémák átbeszélése |
| 2014.03.30.<br>21:00 | 1 óra | Bana | 7.1.3 - Kimeneti nyelv, véglegesítés<br>Dokumentáció |
| 2014.03.30 21:00 | 1.5 óra | Csontos | 7.1.2 Bemeneti nyelv<br>7.4 |

| | | | |
|---|---|---|---|
| 2014.03.30. 22:50 | 0.25 óra | Bana Csontos | Egyeztetés a kimeneti és bemeneti nyelvekről, implementációs megfontolásokról |
| 2014.03.30. 21:00 | 1 óra | Czirják | 7.1.3 - Teszt-esetek elkészítse |
| 2014.03.31. 4:00 | 1 óra | Török | Dokumentáció |

# 8. Részletes tervek

## 8.1 Osztályok és metódusok tervei.

### 8.1.1 Defense

· **Felelősség**
- Ez egy interfész, aminek szerepe, hogy a védelmi funkciókat rögzítse. Az interfészt megvalósított osztályoknak kötelességük ezen előírt funkciókat megvalósítani, ezáltal eleget téve a szerződésnek.

· **Ősosztályok**
- -

· **Interfészek:**
- -

· **Attribútumok**
- -

· **Metódusok**

**Ezen metódusoknak nincs törzse. Szerepük, hogy egyfajta szerződést nyújtsanak ezen interfészt megvalósító osztályok felé.**

- **Enemy attack(Collection<Enemy> Geometry): támadás kezdeményezése az ellenségekre**
- **void wantToUpgrade(Player): fejlesztés kezdeményezése a játékos által**
- **void upgrade(MagicRock): fejlesztés végrehajtása a játékos által**

### 8.1.2 Dwarf

· **Felelősség**
- Ez egy törpe objektum, ami egy ellenség típust reprezentál. Az ellenség összes tulajdonságával rendelkezik, ezenfelül képes egy torony által kezdeményezett életerő csökkentésére.

· **Ősosztályok**
- Enemy

· **Interfészek:**
  - -

· **Attribútumok**
  - -

· **Metódusok**
  - **void lifePowerReduce(Tower)**: életerőt csökkenti a Tower által megadott mértékben.

### 8.1.3 Elf

· **Felelősség**
  - Ez egy tünde objektum, ami egy ellenség típust reprezentál. Az ellenség összes tulajdonságával rendelkezik, ezenfelül képes egy torony által kezdeményezett életerő csökkentésére.

· **Ősosztályok**
  - Enemy

· **Interfészek:**
  - -

· **Attribútumok**
  - -

· **Metódusok**
  - **void lifePowerReduce(Tower)**: életerőt csökkenti a Tower által megadott mértékben.

### 8.1.4 Enemy

· **Felelősség**
  - Ez egy absztrakt osztály, ami egy ellenséget reprezentál. Az ellenség pozíciójának, életerejének tárolásáért, az ellenség léptetéséért és az ellenségek közötti szünetek beiktatásáért felelős.

· **Ősosztályok**
  - -

· **Interfészek:**
  - -

· **Attribútumok**
  - **# static int id:** az enemy azonosítóját generáló változó
  - **# int myId:** adott enemy azonosítója
  - **# int lifePower:** életerőt tárolja
  - **# int stepTime:** léptetés sebességét tárolja
  - **# int pause**: a kihagyott lépések számát tárolja
  - **# Road road:** aktuális út-csempét tárolja, amin az ellenség áll
  - **# boolean isActive:** ellenség állapotának igazságértékét adja meg
  - **#boolean isDuplicated**: ketté lőtték-e az ellenséget
  - **#int counter**: 1-tol stepTime-ig novelodik az erteke, ha eleri a stepTime erteket, akkor lep az ellenseg
  - **#boolean random**: random generator ki/be kapcsolasahoz szukseges

· **Metódusok**
  - **+ void goToSource(Collection<Source>):** ellenség forrás-csempére helyezésének kezdeményezése
  - **+ void move():** az ellenség léptetését végzi
  - **+ void setRoad(Road):** az ellenség pozícióját állítja
  - **+ abstract void lifePowerReduce(Tower):** A leszármazott osztályoknak kell megvalósítani.
  - **+void increasePause(slowingFactor):** az ellenség pause attribútumát növeli a paraméterben átadott értékkel
  - **+ int hit(Tower):** paraméterben megadott torony meglövi az ellenséget, melynek hatására az ellenség életereje a sebzés mértékében lecsökken és bizonyos lövések során az ellenség megduplázódik
  - **+ void setActivity(boolean):** isActive attribútum értékét módosítja

### 8.1.5 Engine

· **Felelősség**
  - Az egész játék mozgatórugója, az időkezelésért felelős. Ezen belül felelős az ellenségek mozgatásért felelős függvény meghívásáért, továbbá a torony és az akadály védekező függvényeinek meghívásáért felelős. Kezeli a játék

végkimenetelével kapcsolatos folyamatokat és tárolja a játékban nyilvántartott összes ellenséget.

· **Ősosztályok**
  - Thread

· **Interfészek:**
  - -

· **Attribútumok**
  - **- Player player:** referencia a játékosra
  - **- Fellowship fellowship:** referencia a szövetségre
  - **-int counter:** lovesvizsgalat gyakorisagahoz szukseges

· **Metódusok**
  - **+ void attackHandler():** a tornyok tüzelésének ütemezéséért felelős
  - **+ void stepHandler() :** ellenségek léptetésért felelős
  - **+ void takeToArea(): :** az ellenségek pályára helyezését időzíti
  - **+ void run()**: Időzítő, ami bizonyos időközönként meghívja az attackHandler(), stepHandler() és takeToArea() metódusokat.
  - **+ void defeat()**: kiírja, hogy vereség és felszabadítja az összes objektumot
  - **+ void victory()**: kiírja, hogy győzelem és felszabadítja az összes objektumot
  - **+ Player getPlayer()**: játékost adja vissza

### 8.1.6 Fellowship

**Felelősség**

- Egy szövetséget reprezentál, ami az összes aktív és passzív ellenséget tárolja. Az ő felelőssége még az engine időzítésének hatására az ellenségek léptetése, valamint ha valakinek elfogy az életereje, akkor ő szünteti meg azt a példányt. Ezen kívül ő adja meg az aktív ellenségek léptetésének sorrendjét is. Továbbá tárolja az összes ellenség számát.

**Ősosztályok**

- -

**Interfészek**

- -

**Attribútumok**

- **- Collection<Enemy> passive**: szövetségen belüli passzív ellenségeket tárolja
- **- Collection<Enemy> active**: szövetségen belüli aktív ellenségeket tárolja
- **- int number**: szövetségen belüli ellenségek számát tárolja

**Metódusok**

- **+ void produceAllEnemy():** összes ellenséget létrehozza és beleteszi a passive listába, de még nincsenek a pályán az ellenségek
- **int getNumber():** number attribútum értékét adja vissza.
- **+ void startWave(waveNumber):** A paraméterül kapott szám alapján, ennyi ellenségből álló hullámot indít el.
- **+ void killEnemy(Enemy):** A paraméterül kapott ellenséget megsemmisíti, törli az active listából.
- **+ Collection<Enemy> getActiveEnemies():** active listát adja vissza
- **+ void moveAllActiveEnemy():** Az összes aktív ellenséget lépteti eggyel.
- **+ void addToActiveEnemyList(Enemy):** A paraméterül kapott ellenséget hozzáadja az aktív kollekcióhoz.

### 8.1.7 Geometry

· **Felelősség**

- A Geometry osztály feladata annak meghatározása, hogy egy adott enemy adott torony hatósugarában van-e, illetve a pálya szerkezetének tárolása.

· **Ősosztályok**

- -

· **Interfészek:**

- -

· **Attribútumok**

- **- Collection<Collection<Tile>> tiles:** a pálya összes csempéjét tárolja

· **Metódusok**

- **+ boolean isInRange(Enemy, Tower)**: Igaz értékkel tér vissza, ha az adott Enemy az adott Tower hatósugarán belül van.
- **+ void createAllTile(int, int):** A játék indításakor a megadott paraméterekkel létrehozunk egy csak Tile-okbol álló pályát,

### 8.1.8 Hobbit

- **Felelősség**
  - Ez egy hobbit objektum, ami egy ellenség típust reprezentál. Az ellenség összes tulajdonságával rendelkezik, ezenfelül képes egy torony által kezdeményezett életerő csökkentésére.
- **Ősosztályok**
  - Enemy
- **Interfészek:**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **+ void lifePowerReduce(Tower)**: életerőt csökkenti a Tower által megadott mértékben.

### 8.1.9 Human

- **Felelősség**
  - Ez egy ember objektum, ami egy ellenség típust reprezentál. Az ellenség összes tulajdonságával rendelkezik, ezenfelül képes egy torony által kezdeményezett életerő csökkentésére.
- **Ősosztályok**
  - Enemy
- **Interfészek:**
  - -
- **Attribútumok**
  - -
- **Metódusok**
  - **+ void lifePowerReduce(Tower)**: életerőt csökkenti a Tower által megadott mértékben.

### 8.1.10 MagicRock

· **Felelősség**
- Egy varázskövet reprezentál, aminek külöböző típusai vannak, eltérő megvételi árral.

· **Ősosztályok**
- -

· **Interfészek:**
- -

· **Attribútumok**
- **int type:** varázskő típusát tárolja
- **int price:** varázskő árát tárolja
- **int static Collection<String> name:** varázskövek nevei

· **Metódusok**
- **int getType():** type attribútum értékét adja vissza
- **int getPrice():** price attribútum értékét adja vissza
- **Collection<String> getName():** egy kő nevét adja vissza

### 8.1.11 Main

**Felelősség**
- A program innen indul ki. Létrehozza az Engine objektumot.

**Ősosztályok**
- -

**Interfészek**
- Serializable

**Attribútumok**
- **Engine engine**: referencia az Engine-re.

**Metódusok**
- **+ void loadInputLanguage():** Betölti a bemeneti fájlokat

### 8.1.12 Mountain

· **Felelősség**
- A Végzet Hegyét reprezentálja, ami csak egy út-csempén állhat.

· **Ősosztályok**
- Road

· **Interfészek**
- **-**

· **Attribútumok**
- **-**

· **Metódusok**
- **-**

### 8.1.13 Obstacle

**Felelősség**
- Ez egy akadály objektum, ami lassítja az ellenség haladását az akadály területén belül. Továbbá tárolja a ráhelyezhető varázsköveket és az akadályon lévő varázskövek számát.

**Ősosztályok**
- Road

**Interfészek**
- Defense

**Attribútumok**
- **- int slowingFactor**: Akadály lassításának mértéke.
- **- int magicRockNumber:** Akadályon lévő varázskövek száma.
- **- Collection<MagicRock> magicRock:** a toronyra helyezhető varázskövek kollekciója

**Metódusok**

- **+ void wantToUpgrade(Player):** A játékos ennek a függvénynek a segítségével tudja jelezni fejlesztési szándékát az adott akadály felé. Ellenőrzi, hogy fejleszthető-e még az akadály illetve a játékosnak elegendő varázsereje van-e. Ezen két feltétel teljesülése szükséges, ahhoz hogy fejleszteni lehessen az adott akadályt.
- **+ void upgrade(MagicRock):** Miután a Player kiválasztotta a varázskövet, amit fel szeretne rakni az akadályra, ezt átadva az akadálynak, az felteszi magára. Ez a függvény növeli meg a slowingFactor és a varázskövek számának értékét és a varázskövek kollekciójához hozzáadja a követ.
- **+ void slowMe(Enemy):** Az akadályon álló ellenséget lelassítja úgy, hogy növeli az ellenségnek a pause attribútumát, a varázskövek számával arányos mértékben.

### 8.1.14 Player

· **Felelősség**

- A játékost reprezentáló objektum. Tárolja a játékos varázserejét, ami szükséges a tornyok és az akadályok építéséhez. Felelős a tornyok és akadályok építéséért illetve kezdeményezheti ezek elhelyezését a játéktér megfelelő részein. Továbbá felelős a már meglévő tornyok és akadályok fejlesztéséért varázskövek felhasználásával. Felelős a kiválasztott építési terület (Tile) beépíthetőségének, valamint az építéshez szükséges varázserő megállapításáért, és Player dolga továbbá a saját varázserejének folyamatos menedzselése. A Player objektum indítja a játékot.

· **Ősosztályok**

- **-**

· **Interfészek**

- **-**

· **Attribútumok**

- **-int magicPower**: a játékos varázserejét tárolja
- **-PlayingArea area**: a játékteret tárolja

· **Metódusok**

- + **void chooseUpgrade(Collection <MagicRock> magicRock)**: Fejlesztés során ezen a függvényen keresztül választja ki a játékos, hogy milyen képességet akar fejleszteni. A paraméterként kapott varázskövekből választhat. Ha a kiválasztott varázskőre van elegendő magicPower, akkor a kiválasztás után csökkentjük az aktuális magicPower értékét a varázskő árával.

- **+ void startGame()**: elindítja a játékot
- **+ void reduceMagicPower(int price):** Egy építés illetve egy fejlesztés után a játékos varázsereje ezen a metóduson keresztül csökken, a paraméterként kapott *price* értékkel.

### 8.1.15 PlayingArea

· **Felelősség**
- A pályán lévő pályaelemek tárolása (út csempe, beépíthető csempe, torony és akadály). A játék betöltését és a pálya kiolvasását követően inicializálja a pályát, valamint a Játékos kérésére tornyot vagy akadályt ad hozzá a megfelelő listához. A PlayingArea felelőssége a Vereség (ellenség hegyre lép) figyelése és kezelése is. Hozzáférést nyújt a megfelelő osztályok számára a pályán lévő entitásokhoz.

· **Ősosztályok**
- **-**

· **Interfészek**
- **-**

· **Attribútumok**
- **- source: Collection<Source>**: A pályán lévő forráscsempéket tároló kollekció.
- **- tower: Collection<Tower>:** A pályá lévő tornyokat tároló kollekció.
- **- obstacle: Collection<Obstacle>**: A pályán lévő akadályokat tároló kollekció.
- **- mountain: Mountain**: A pályán lévő Végzet Hegyét tároló objektum.
- - **road: Collection<Road>:** Az utat tároló változó
- **- geometry: Geometry:** A geometriát tároló változó

· **Metódusok**
- **+ void initialize():** Egy fájlból beolvasott map inicializálásáért felelős függvény.
- **+ void: addTower(Tower):** A paraméterlistán átadott torony hozzáadása a pályán lévő *tower* kollekcióhoz.
- **+ void:addObstacle(Obstacle):** A paraméterlistán átadott akadály hozzáadása a pályén lévő *obstacle* kollekcióhoz.
- **+ void: isOnMountain(Engine)**: A Hegy csempét figyelő függvény, aminek felelőssége, hogy minden lépésben ellenőrizze, van-e ellenség a Hegyen.
- **+ void isBuildable(Tile)**: Beépíthető-e az adott csempe.
- **+ void addReference(pos1, pos2)**: pos1 csempe összekötése a pos2-vel, ami az ellenség léptetéséhez kell, úgy hogy pos1 nextRoad-ja lesz a pos2, ha pos1 út-csempe

### 8.1.16 Road

· **Felelősség**
  - Egy adott út-csempét reprezentál. Felelős az őt követő út-csempe nyilvántartásáért valamint a foglaltság jelzéséért.

· **Ősosztályok**
  - **Tile**

· **Interfészek**
  - **-**

· **Attribútumok**
  - **#Road nextRoad:** A következő út-csempét tároló objektum, ami az ellenségek megadott útvonalon való haladásához szükséges.
  - **#boolean random:** véletlen ki-bekapcsolása miatt szükséges

· **Metódusok**
  - **+void requestDestination(Enemy):** Paraméterben megadott ellenség kérést kezdeményez azon út-csempéhez, amelyen éppen áll, hogy elkérhesse a következő út-csempét.

### 8.1.17 Source

· **Felelősség**
  - Olyan csempét reprezentál, ahova az ellenségek inicializáláskor kerülnek. Felelős saját pozíciója nyilvántartásáért és az őt követő út-csempe nyilvántartásáért.

· **Ősosztályok**
  - Road

· **Interfészek**
  - **-**

· **Attribútumok**
  - **-**

· **Metódusok**
  - -

### 8.1.18 Tile

· **Felelősség**
  - Egy csempe tárolásáért felel. Ki tudja számolni tetszőleges két csempe közötti távolságot. Továbbá felelős tornyok és akadályok építéséért.
· **Ősosztályok**
  - -
· **Interfészek**:
  - -
· **Attribútumok**
  - **# Position pos**: A csempe pozícióját tárolja.
  - **# int type**: adott csempe típusa
  - **# int price:** adott csempe építésének árát tárolja

· **Metódusok**
  - **+ Position getPos()**: pos attribútum értékét adja vissza
  - **+ void setPos(Position)**: pos attribútum értékét módosítja
  - **+ Tower buildTower(Player):** Az adott Tile-ra megpróbál tornyot építeni a játékos.. Ha van elegendő varázserő és nincs a Tile-on se út, se torony, akkor felépíti és visszaadja a Player-nek a létrehozott Tower referenciáját.
  - **+ Obstacle buildObstacle(Player):** Az adott Tile-ra megpróbál akadályt építeni. Ha van elegendő varázserő és az adott Tile út, és nincs még rajta akadály, akkor felépíti és visszaadja a Player-nek a létrehozott Obstacle referenciáját.

### 8.1.19 Tower

· **Felelősség**
  - A játékos által elhelyezhető tornyot reprezentáló objektum. Tárolja a lövésre vonatkozó paramétereket (tüzelési hatékonyság, hatótávolság, sebzés ereje különböző ellenség típusokra), és varázskő használata esetén növeli is a hatékonyságukat. Tárolja még a ráhelyezhető varázskövek aktuális számát. Felelős továbbá a köd le- és felszállásáért.
· **Ősosztályok**
  - **Tile**

- **Interfészek**
  - **- Defense**

- **Attribútumok**
  - **- int shootPeriod**: a torony tüzelési hatékonysága
  - + static int id: azonosító
  - **- int range**: a torony tüzelési távolsága
  - - **int fogRange**: ebbe a változóba menti ki a torony a tüzelési távolságát a köd leszállása előtt, hogy a köd felszállása után visszaállítható legyen az eredeti érték
  - **- int damagePowerHuman:** tárolja azt az értéket, amivel egy Human típusú ellenséget megsebez
  - **- int damagePowerHobbit:** tárolja azt az értéket, amivel egy Hobbit típusú ellenséget megsebez
  - **- int damagePowerElf:** tárolja azt az értéket, amivel egy Elf típusú ellenséget megsebez
  - **- int damagePowerDwarf:** tárolja azt az értéket, amivel egy Dwarf típusú ellenséget megsebez
  - **- Collection <MagicRock> magicRock:** a toronyra helyezhető varázskövek kollekciója
  - **- int magicRockNumber:** a toronyra helyezett varázskövek száma
  - **# boolean random:** véletlen ki-bekapcsolása
  - **# boolean split:** kettélövés igen/nem

- **Metódusok**
  - + **Enemy attack(Collection <Enemy>)**: ezen a függvényen keresztül támadja a torony az ellenségeket, a paraméterként kapott listából egy ellenséget meglő (olyat, aki hatótávon belül van), majd visszatér a meglőtt ellenséggel.
  - **+ void wantToUpgrade(Player):** A játékos ennek a függvénynek a segítségével tudja jelezni fejlesztési szándékát az adott torony felé.
  - + **void upgrade(MagicRock):** a torony fejlesztése, a paraméterként kapott varázskőnek megfelelő képességgel.
  - **+ void fogOn():** toronyra a köd beállítása
  - **+ void fogOff():** toronyról a köd eltávolítása

## 8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

### 8.2.1 Pálya beolvasása

**Bemenet (map01.txt és input01.txt tartalma)**

**map01.txt tartalma:**

**Road 1 1**
**Obstacle 1 2**

**input01.txt tartalma:**

**loadMap map01.txt**
**exit**

**Elvárt kimenet**

**Load map is unsuccessful.**
**Error: (1,2) tile is unbuildable for an Obstacle.**

### 8.2.2 Az ellenség a következő csempére lép

**Bemenet (map02.txt és input02.txt tartalma)**

**map02.txt tartalma:**

**Road 1 1**
**Road 1 2**
**Road ref 1 1 1 2**
**Hobbit 1 1**

**input02.txt tartalma:**

**loadMap map02.txt**
**getEnemyInfo 0**
**step 1**
**getEnemyInfo 0**
**exit**

**Elvárt kimenet**

**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,1)**

**isActive: true**
**isDuplicated: false**
**[Step:1]**
**[0:Hobbit] has moved to Road(1,2)**
**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,2)**
**isActive: true**
**isDuplicated: false**

### 8.2.3 Az ellenség akadályra lép

**Bemenet (map03.txt és input03.txt tartalma)**

**map03.txt tartalma:**

**Road 1 1**
**Road 1 2**
**Obstacle 1 2**
**Road 1 3**
**Road 1 4**
**Road ref 1 1 1 2**
**Obstacle ref 1 2 1 3**
**Road ref 1 3 1 4**
**Hobbit 1 1**

**input03.txt tartalma:**

**loadMap map03.txt**
**getEnemyInfo 0**
**getObstacleInfo 0**
**step 1**
**getEnemyInfo 0**
**getObstacleInfo 0**
**step 1**
**getEnemyInfo 0**
**getObstacleInfo 0**
**step 1**
**getEnemyInfo 0**
**getObstacleInfo 0**

**exit**

**Elvárt kimenet**

**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,1)**
**isActive: true**
**isDuplicated: false**
**[0:Obstacle]**
**slowingFactor: 1**
**magicRockNumber: 0**
**[Step:1]**
**[0:Hobbit] has moved to Obstacle(1,2)**
**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 1**
**position: (1,2)**
**isActive: true**
**isDuplicated: false**
**[0:Obstacle]**
**slowingFactor: 1**
**magicRockNumber: 0**
**[Step:1]**
**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,2)**
**isActive: true**
**isDuplicated: false**
**[0:Obstacle]**
**slowingFactor: 1**
**magicRockNumber: 0**
**[Step:1]**
**[0:Hobbit] has moved to Road(1,3)**

**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,3)**
**isActive: true**
**isDuplicated: false**
**[0:Obstacle]**
**slowingFactor: 1**
**magicRockNumber: 0**

### 8.2.4 Toronyépítés

**Bemenet (map04.txt és input04.txt tartalma)**

**map04.txt tartalma:**

**input04.txt tartalma:**

**[loadMap map04.txt**
**getPlayerInfo 0**
**buildTower 1 1**
**getTowerInfo 0**
**getPlayerInfo 0**
**exit**

**Elvárt kimenet**

**[Player]**
**magicPower: 100**
**[0:Tower] has been built on Tile(1,1)**
**[0:Tower]**
**range: 3**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**
**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**
**[Player]**
**magicPower: 90**

### 8.2.5 Tüzelés

**Bemenet (map.txt és input.txt tartalma)**

**map05.txt tartalma:**

**Tower 2 2**
**Road 1 2**
**Road 1 3**
**Road ref 1 2 1 3**
**Hobbit 1 2**

**input05.txt tartalma:**

**loadMap map05.txt**
**getEnemyInfo 0**
**getTowerInfo 0**
**step 1**
**getEnemyInfo 0**
**getTowerInfo 0**
**exit**

**Elvárt kimenet**

**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,2)**
**isActive: true**
**isDuplicated: false**
**[0:Tower]**
**range: 3**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**
**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**
**[Step:1]**
**[0:Hobbit] has moved to Road(1,3)**

**[0:Tower] has shot [0:Hobbit]**
**[0:Hobbit]**
**lifePower: 90**
**stepTime: 10**
**pause: 0**
**position: (1,3)**
**isActive: true**
**isDuplicated: false**
**[0:Tower]**
**range: 3**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**
**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**

### 8.2.6 Toronyfejlesztés (range)

**Bemenet (map06.txt és input06.txt tartalma) Felhasználó a 3-as varázskövet teszi a toronyra.**

**map06.txt tartalma:**

**Tower 0 1**
**Road 1 1**
**Road 1 2**
**Road 1 3**
**Road ref 1 1 1 2**
**Road ref 1 2 1 3**
**Hobbit 1 1**

**input06.txt tartalma:**

**loadMap map06.txt**
**getEnemyInfo 0**
**getTowerInfo 0**
**step 1**
**getEnemyInfo 0**
**getTowerInfo 0**

**getPlayerInfo 0**
**upgradeTower 0**
**getTowerInfo 0**
**getPlayerInfo 0**
**step 1**
**getEnemyInfo 0**
**exit**

**Elvárt kimenet (3-as varázskő esetén)**

**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,1)**
**isActive: true**
**isDuplicated: false**
**[0:Tower]**
**range: 3**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**
**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**
**[Step:1]**
**[0:Hobbit] has moved to Road(1,2)**
**[0:Tower] has shot [0:Hobbit]**
**[0:Hobbit]**
**lifePower: 90**
**stepTime: 10**
**pause: 0**
**position: (1,2)**
**isActive: true**
**isDuplicated: false**
**[0:Tower]**
**range: 3**

**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**
**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**
**[Player]**
**magicPower: 100**
**[Player] has upgraded [0:Tower]**
**[0:Tower]**
**range: 3**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**
**damagePowerHobbit: 20**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**
**[Player]**
**magicPower: 90**
**[Step:1]**
**[0:Hobbit] has moved to Road(1,3)**
**[0:Tower] has shot [0:Hobbit]**
**[0:Hobbit]**
**lifePower: 70**
**stepTime: 10**
**pause: 0**
**position: (1,3)**
**isActive: true**
**isDuplicated: false**

### 8.2.7 Ellenség elpusztul

**Bemenet (map07.txt és input07.txt tartalma)**

**map07.txt tartalma:**

**Tower 2 2**

**Road 1 2**
**Road 1 3**
**Road 1 4**
**Road ref 1 2 1 3**
**Road ref 1 3 1 4**
**Hobbit 1 2 10**
**Hobbit 1 3**

**input07.txt tartalma:**

**loadMap map07.txt**
**getPlayerInfo 0**
**getEnemyInfo 0**
**getEnemyInfo 1**
**getTowerInfo 0**
**step 1**
**getPlayerInfo 0**
**exit**

**Elvárt kimenet**

**[Player]**
**magicPower: 100**
**[0:Hobbit]**
**lifePower: 10**
**stepTime: 10**
**pause: 0**
**position: (1,2)**
**isActive: true**
**isDuplicated: false**
**[1:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,3)**
**isActive: true**
**isDuplicated: false**
**[0:Tower]**
**range: 3**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**

**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**
**[Step:1]**
**[0:Hobbit] has moved to Road(1,3)**
**[1:Hobbit] has moved to Road(1,4)**
**[0:Tower] has shot [0:Hobbit]**
**[0:Hobbit] has been deleted**
**[Player]**
**magicPower: 110**

### 8.2.8 Toronyra köd leereszkedik

**Bemenet (map08.txt és input08.txt tartalma)**

**map08.txt tartalma:**

**Tower 1 1**

**input08.txt tartalma:**

**loadMap map08.txt**
**getTowerInfo 0**
**fogOn**
**getTowerInfo 0**
**exit**

**Elvárt kimenet**

**[0:Tower]**
**range: 3**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**
**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**
**[Fog has been set on]**
**[0:Tower]**
**range: 2**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**

**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**

### 8.2.9 Ellenség megkettőződik

**Bemenet (map09.txt és input09.txt tartalma)**

**map09.txt tartalma:**

**Tower 1 1**
**Road 1 2**
**Road 1 3**
**Road 1 4**
**Road ref 1 2 1 3**
**Road ref 1 3 1 4**
**Hobbit 1 2**

**input09.txt tartalma:**

**loadMap map09.txt**
**SPLIT_ON**
**getEnemyInfo 0**
**getTowerInfo 0**
**step 1**
**getEnemyInfo 0**
**getEnemyInfo 1**
**exit**

**Elvárt kimenet**

**[0:Hobbit]**
**lifePower: 100**
**stepTime: 10**
**pause: 0**
**position: (1,2)**
**isActive: true**
**isDuplicated: false**
**[0:Tower]**
**range: 3**
**fogRange: 3**
**shootPeriod: 10**
**damagePowerHuman: 10**

**damagePowerHobbit: 10**
**damagePowerElf: 10**
**damagePowerDwarf: 10**
**magicRockNumber: 0**
**[Step:1]**
**[0:Hobbit] has moved to Road(1,3)**
**[0:Tower] has shot [0:Hobbit]**
**[0:Hobbit]**
**lifePower: 45**
**stepTime: 10**
**pause: 0**
**position: (1,3)**
**isActive: true**
**isDuplicated: false**
**[1:Hobbit]**
**lifePower: 45**
**stepTime: 10**
**pause: 0**
**position: (1,3)**
**isActive: true**
**isDuplicated: false**

### 8.2.10 Akadályépítés

**Bemenet (map10.txt és input10.txt tartalma)**

**map10.txt tartalma:**

**Road 0 0**

**input10.txt tartalma:**

**loadMap map10.txt**
**getPlayerInfo 0**
**buildObstacle 0 0**
**getObstacleInfo 0**
**getPlayerInfo 0**
**exit**

**Elvárt kimenet**

**[Player]**
**magicPower: 100**
**[0:Obstacle] has been built on Road(0,0)**
**[0:Obstacle]**
**slowingFactor: 1**
**magicRockNumber: 0**
**[Player]**
**magicPower: 95**

### 8.2.11 A játékos győzelmet arat

**Bemenet (map.txt és input.txt tartalma)**

**map11.txt tartalma:**

**Road 1 2**
**Road 1 3**
**Road ref 1 2 1 3**
**Tower 2 3**
**Hobbit 1 2 2**

**input11.txt tartalma:**

**loadMap map11.txt**
**getEnemyInfo 0**
**step 1**
**exit**

**Elvárt kimenet**

**[0:Hobbit]**
**lifePower: 2**
**stepTime: 10**
**pause: 0**
**position: (1,2)**
**isActive: true**
**isDuplicated: false**
**[Step:1]**
**[0:Hobbit] has moved to Road(1,3)**
**[0:Tower] has shot [0:Hobbit]**
**[0:Hobbit] has been deleted**
**Victory! :)**

## 8.3 A tesztelést támogató programok tervei

A tesztelést egy külön program végzi el, ami végrehajtja a beadott bemeneti tesztet és előállítja a .txt kiterjesztésű kimeneti fájlt. Az összehasonlítást végző program a bemenetén várja a kimeneti fájlt és a várt eredményeket tartalmazó fájlt. Ezeket beolvassa és soronként összehasonlítja. Az összehasonlításnak kétféle kimenetele lehet: sikeres vagy sikertelen volt a teszt.

## 8.4 Napló

| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.04.03 18:00 | 1.5 óra | Bana, Csontos, Czirják, Tóth, Török | Értekezlet - a heti feladatok elosztása, a megoldás menetének átbeszélése |
| 2014.04.04 | 2.5 óra | Czirják, Csontos, Tóth | Konzultáció |
| 2014.04.05 | 1.5 óra | Tóth | 8.2.7, 8.2.8, 8.2.9 tesztesetek elkészítése |
| 2014.04.06 | 1.5 óra | Tóth | Geometry, Player, Tower, Position osztályok leírása |
| 2014.04.06. 19:00 | 3 óra | Török | Engine, Enemy, Hobbit, Human osztályok leírása. 8.2.4, 8.2.5, 8.2.6 tesztesetek elkészítése |
| 2014.04.05 15:30 | 1.5 óra | Czirják | 8.2.10, 8.2.11, 8.2.12 tesztesetek elkészítése |

| | | | |
|---|---|---|---|
| 2014.04.06<br>20:00 | 1,5 óra | Csontos | 8.1.1, 8.1.2, 8.1.3, 8.1.10<br>8.2.1, 8.2.2<br>Osztálydiagram módosítása |
| 2014.04.06<br>20:00 | 6 óra | Bana | - Tile, Obstacle, Fellowship, Main osztályok terveinek elkészítése<br>- 8.13, 8.14, 8.15 Tesztesetek átgondolása, és elkészítése |
| 2014.04.07<br>20:00 | 4 óra | Csontos | Osztálydiagram javítása<br>Osztályleírás javítása<br>Tesztesetek javítása illetve ellenőrzése<br>8.3 |
| 2014.04.08.<br>10:30 | 0,75 óra | Török | Dokumentáció |

# 10. Prototípus

## 10.1 Fordítási és futtatási útmutató

### 10.1.1 Fájllista

| Tartalom | Keletkezés ideje | Méret [Byte] | Fájl neve |
|---|---|---|---|
| Defense interfészt valósítja meg. | 2014.04.17 | 810 | Defense.java |
| Dwarf osztályt valósítja meg. | 2014.04.17 | 802 | Dwarf.java |
| Elf osztályt valósítja meg. | 2014.04.17 | 794 | Elf.java |
| Enemy osztályt valósítja meg. | 2014.04.18 | 5320 | Enemy.java |
| Engine osztályt valósítja meg. | 2014.04.18 | 4448 | Engine.java |
| Fellowship osztályt valósítja meg. | 2014.04.18 | 5865 | Fellowship.java |
| Geometry valósítja meg. | 2014.04.17 | 1582 | Geometry.java |
| Hobbit osztályt valósítja meg. | 2014.04.17 | 814 | Hobbit.java |
| Human osztályt valósítja meg. | 2014.04.17 | 807 | Human.java |
| MagicRock osztályt valósítja meg. | 2014.04.18 | 1445 | MagicRock.java |
| Main osztályt valósítja meg. | 2014.04.18 | 19163 | Main.java |
| Mountain osztályt valósítja meg. | 2014.04.17 | 241 | Mountain.java |
| Obstacle osztályt valósítja meg. | 2014.04.18 | 3230 | Obstacle.java |
| Player osztályt valósítja meg. | 2014.04.18 | 3655 | Player.java |
| PlayingArea osztályt valósítja meg. | 2014.04.18 | 9847 | PlayingArea.java |
| Position osztályt valósítja meg. | 2014.04.17 | 560 | Position.java |
| Road osztályt valósítja meg. | 2014.04.18 | 1360 | Road.java |
| Source osztályt valósítja meg. | 2014.04.17 | 276 | Source.java |
| Tile osztályt valósítja meg. | 2014.04.18 | 1577 | Tile.java |
| Tower osztályt valósítja meg. | 2014.04.18 | 5923 | Tower.java |
| Writer osztályt valósítja meg. | 2014.04.22 | 263 | Writer.java |

### 10.1.2 Fordítás

A .java forrásfájlok mellett található egy compile.bat fájl, ami egyenként lefordítja az összes olyan java osztályt, ami nem absztrakt. Ezt a javac parancsszóval érhetjük el, ami a Command Prompt-ból elérhető, ha megadjuk a JDK helyét. Ezt az alábbi paranccsal tehetjük meg: “set path=%path%; JDK elérési útvonala”.
Ha ezt sikeresen beállítottuk, akkor a compile.bat fájlt futtatva létrejönnek a .class fájlok, ugyanazon mappában, ahol a .java fájlok vannak.

### 10.1.3 Futtatás

Ha megtörtént a fordítás, akkor a run.bat fájlt futtatva elindul a program és a felhasználótól egy szöveges fájl nevét kéri (inputxy.txt), ezek az egyes tesztesetek lesznek. Minden teszteset lefutása során keletkezik egy outputxy.txt, amibe a teszteset által kapott eredmény található. Ezt a szöveges fájlt kell összevetni az elvárt eredménnyel, amit a doksiban adtunk meg.
A run.bat fájlon belül azt a java osztályt kell megadni, amelyikben a public static void main(String[] args) metódus található.

## 10.2 Tesztek jegyzőkönyvei

### 10.2.1 Pálya beolvasása

| | |
|---|---|
| **Tesztelő neve** | Csontos Valentin |
| **Teszt időpontja** | 2014.04.23 |

### 10.2.2 Az ellenség a következő csempére lép

| | |
|---|---|
| **Tesztelő neve** | Tóth Dávid |
| **Teszt időpontja** | 2014.04.23 |
| **Tesztelő neve** | Tóth Dávid |

| | |
|---|---|
| **Teszt időpontja** | 2014.04.22 |
| **Teszt eredménye** | Az ellenség nem lépett a következő csempére |
| **Lehetséges hibaok** | Nincs megfelelően összekötve a két egymást követő csempe |
| **Változtatások** | A két egymást követő csempe referenciája megfelelően be lett állítva |

### 10.2.3 Az ellenség akadályra lép

| | |
|---|---|
| **Tesztelő neve** | Tóth Dávid |
| **Teszt időpontja** | 2014.04.23. |

| | |
|---|---|
| **Tesztelő neve** | Tóth Dávid |
| **Teszt időpontja** | 2014.04.23. |
| **Teszt eredménye** | Az ellenséget nem lassítja az akadály |
| **Lehetséges hibaok** | Nem érzékelte akadálynak a csempét, ahol akadály van |
| **Változtatások** | Megfelelő listában a megfelelő csempét akadályra cseréltük |

### 10.2.4 Toronyépítés

| **Tesztelő neve** | Tóth Dávid |
|---|---|
| **Teszt időpontja** | 2014.04.22 |

| **Tesztelő neve** | Tóth Dávid |
|---|---|
| **Teszt időpontja** | 2014.04.21 |
| **Teszt eredménye** | A játékos varázsereje nem csökkent az építés után |
| **Lehetséges hibaok** | A Tile osztály buildTower() metódusában nem módosítjuk a játékos varázserejét. |
| **Változtatások** | A tile osztály buildTower() metódusában megfelelően csökkentjük a játékos varázserejét. |

### 10.2.5 Tüzelés

| **Tesztelő neve** | Tóth Dávid |
|---|---|
| **Teszt időpontja** | 2014.04.23 |

| **Tesztelő neve** | Tóth Dávid |
|---|---|
| **Teszt időpontja** | 2014.04.22 |

| | |
|---|---|
| **Teszt eredménye** | A torony nem lövi meg az ellenséget |
| **Lehetséges hibaok** | Rossz az inRange függvény |
| **Változtatások** | inRange metódus számítás javítva |

### 10.2.6 Toronyfejlesztés (rang)

| | |
|---|---|
| **Tesztelő neve** | Bana Szabolcs |
| **Teszt időpontja** | 2014.04.22 |

| | |
|---|---|
| **Tesztelő neve** | Bana Szabolcs |
| **Teszt időpontja** | 2014.04.22 |
| **Teszt eredménye** | Nem csökkent a játékos varázsereje |
| **Lehetséges hibaok** | Rossz paraméterezéssel hívtuk az upgrade függvényt |
| **Változtatások** | A Player átadásával már jól működött a callback, és levonta a varázserőt |

### 10.2.7 Ellenség elpusztul

| **Tesztelő neve** | Tóth Dávid |
|---|---|
| **Teszt időpontja** | 2014.04.22 |

| **Tesztelő neve** | Tóth Dávid |
|---|---|
| **Teszt időpontja** | 2014.04.22 |
| **Teszt eredménye** | Az ellenség elpusztulása után nem növekedik a játékos varázsereje |
| **Lehetséges hibaok** | Az engine attackHandler() metódusában nem változtatjuk a játékos varázserejét |
| **Változtatások** | Az engine attackHandler() megfelelő részén növeljük a varázserőt |

### 10.2.8 Toronyra köd leereszkedik

| **Tesztelő neve** | Török Odett |
|---|---|
| **Teszt időpontja** | 2014.04.22. |
| **Tesztelő neve** | Török Odett |
| **Teszt időpontja** | 2014.04.21. |
| **Teszt eredménye** | Nem ereszkedett köd a toronyra. |
| **Lehetséges hibaok** | A Torony osztályában nem változtatjuk a torony range-t |
| **Változtatások** | A Tower osztály megfelelő részén csökkentjük a range-t |

### 10.2.9 Ellenség megkettőződik

| | |
|---|---|
| **Tesztelő neve** | Bana Szabolcs |
| **Teszt időpontja** | 2014.04.22 |
| **Tesztelő neve** | Bana Szabolcs |
| **Teszt időpontja** | 2014.04.22 |
| **Teszt eredménye** | Nem jött létre újabb ellenség példány |
| **Lehetséges hibaok** | A random faktort nem kalkuláltuk bele |
| **Változtatások** | A bemeneti nyelvben a random faktor kikapcsolható, így determinisztikusan mindig fixen létrejön az ellenség klónja |

### 10.2.10 Akadályépítés

| | |
|---|---|
| **Tesztelő neve** | Tóth Dávid |
| **Teszt időpontja** | 2014.4.22 |

### 10.2.11 Játékos győzelmez arat

| | |
|---|---|
| **Tesztelő neve** | Tóth Dávid |
| **Teszt időpontja** | 2014.04.22 |

## 10.3 Értékelés

| Tag neve | Munka százalékban |
|---|---|
| Bana Szabolcs | 22,06% |
| Csontos Valentin | 26,59% |
| Czirják Balázs | 16,40% |
| Tóth Dávid | 18,40% |
| Török Odett | 16,55% |

## 10.4 Napló

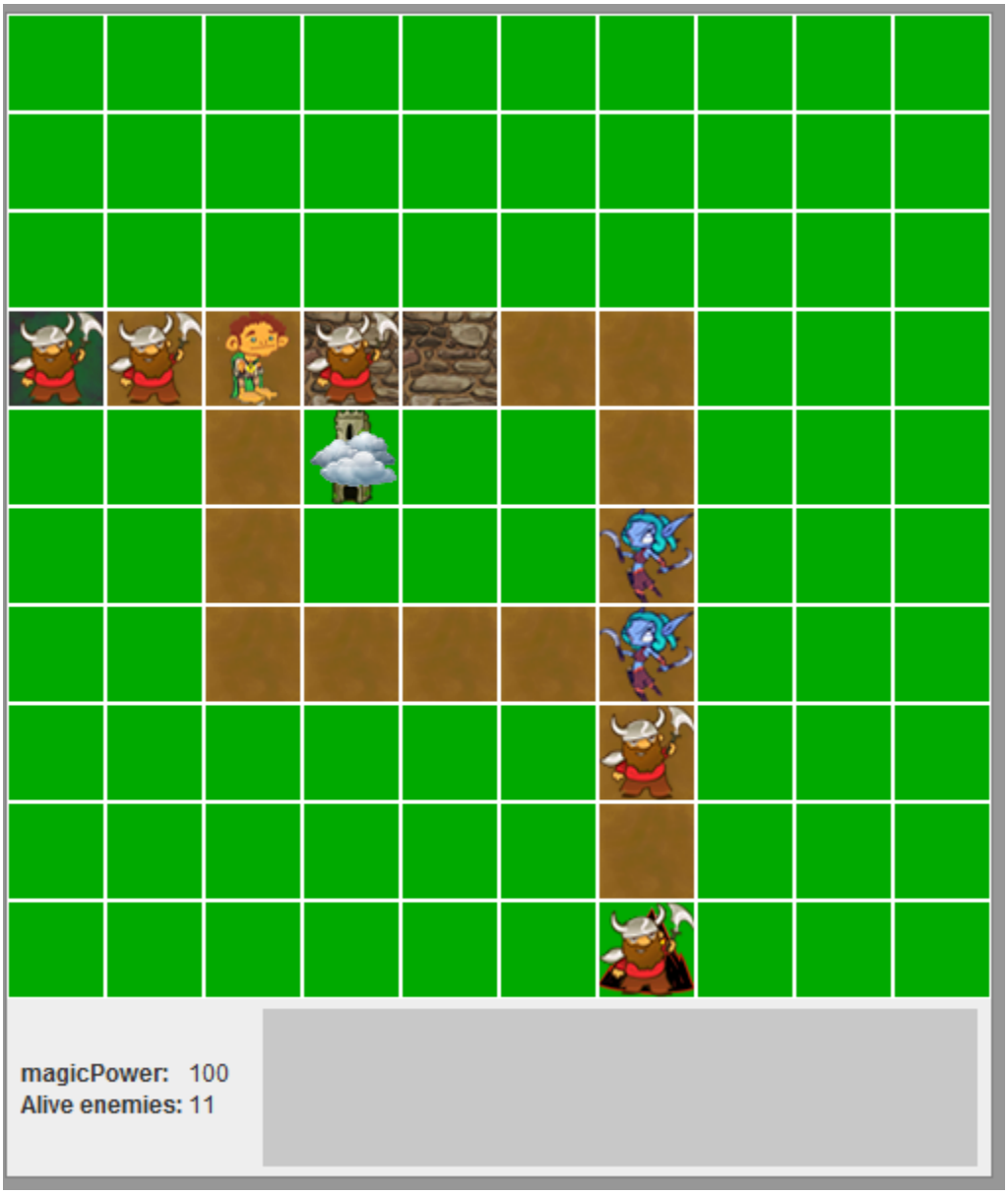
| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.04.09. 12:15 | 2 óra | Csontos<br>Czirják | Konzultáció |
| 2014.04.16. 17:30 | 2,5 óra | Bana<br>Csontos<br>Czirják<br>Tóth | Értekezlet. |
| 2014.04.17. 14:00 | 3 óra | Csontos | Implementáció szekvencia diagramok alapján |

| | | | |
|---|---|---|---|
| 2014.04.17. 17:00 | 4 óra | Bana<br>Csontos<br>Czirják | Értekezlet |
| 2014.04.18 00:00 | 1,5 óra | Csontos | Main osztály implementációja |
| 2014.04.18 16:00 | 1,5 óra | Csontos | Main implementációjának folytatása |
| 2014.04.18 18:00 | 1,5 óra | Csontos | Main implementációjának folytatása |
| 2014.04.18 11:30 | 2,5 óra | Csontos | implementáció + kommentezés |
| 2014.04.19. 14:00 | 5 óra | Bana<br>Csontos<br>Czirják<br>Tóth<br>Török | Értekezlet |
| 2014.04.20 15:30 | 3 óra | Bana | Ellenség megkettőződik, Torony fejlesztés tesztesetek implementálása |
| 2014.04.20. 17:00 | 4 óra | Czirják | Akadályfejlesztés, Ellenség akadályra lép |
| 2014.04.20 18:00 | 2,5 óra | Csontos | Ellenség léptetésének a megoldása<br>Pálya betöltés<br>Ellenség következő csempére lép |

| | | | |
|---|---|---|---|
| 2014.04.20. 19:00 | 4 óra | Tóth | Toronyépítés, ellenség elpusztul, akadály építés |
| 2014.04.21 10:00 | 1 óra | Csontos | Input file name bekérése konzolról<br>eredmény generálása egy szövegfájlba, implementáció |
| 2014.04.21 20:00 | 4 óra | Tóth | Toronyépítés, ellenség elpusztul, akadály építés |
| 2014.04.22 16:30 | 5,5 óra | Csontos<br>Czirják<br>Bana<br>Tóth | Tesztesetek közös áttekintése, hibák javítása |
| 2014.04.22 11:00 | 4 óra | Csontos | bemeneti nyelv változások<br>10.1.1, 10.1.2, 10.1.3<br>Összes teszteset ellenőrzése command prompt-ban |
| 2014.04.22. | 5 óra | Török | 10.0.3, 10.0.8 tesztesetek, dokumentáció, spreadsheet |
| 2014.04.23 00:10 | 2 óra | Bana | Kommentezés, kommentek % számítása, végső tesztelés, dokumentáció |
| 2014.04.23 00:00 | 2 óra | Tóth | Kommentek, jegyzőkönyv, dokumentáció |
| 2014.04.23 6:00 | 2 óra | Török | Jegyzőkönyv, dokumentáció, spreadsheet |

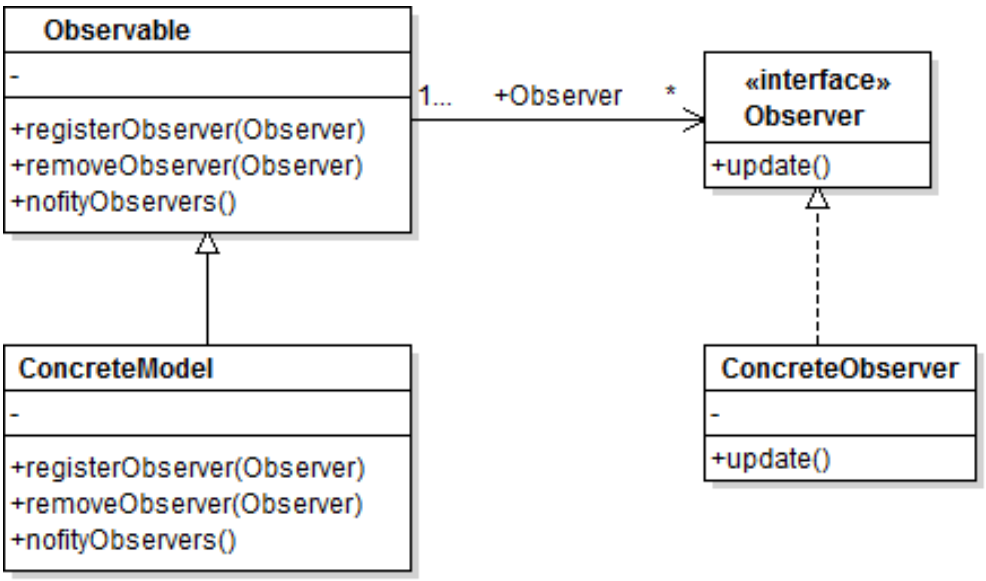
# 11. Grafikus felület specifikációja

## 11.1 A grafikus interfész



Középen a játéktér helyezkedik el, amely csempékből áll. A különböző típusú csempék, és az egyes ellenségtípusok különböző megjelenítést kapnak. Alul két adatot írunk ki: az egyik a játékos aktuális varázsereje illetve a még életben lévő ellenségek száma.

# 11.2 A grafikus rendszer architektúrája

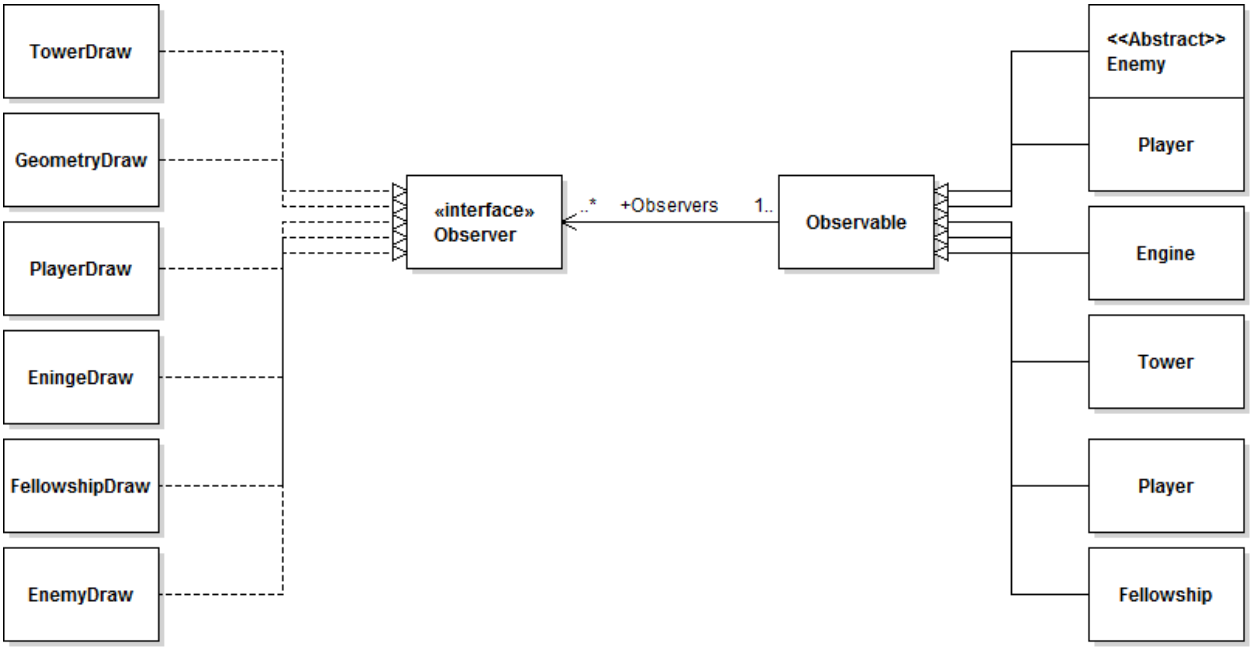



A ConcreteModel osztály reprezentálja a már implementált modell osztályainkat, amelyek leszármaznak az Observable osztályból. Minden modell osztály kollekcióban tárolja az állapot változásáról értesíteni kívánt ConcreteObserver osztályokat. A ConcreteObserver osztályok megvalósítják az Observer interfészt. A modell osztály notifyObservers() metódus lefutására meghívódik a regisztrált ConcreteObserver osztályok update metódusa, amelyben a grafikus felület módosítását hajtjuk végre.
Tehát minden olyan modell osztálynak, amiben egy állapotváltozást fel szeretnénk tüntetni a grafikus felületen, ahhoz kell egy hozzátartozó Draw osztály, ami a felület módosításért felelős.

### 11.2.1 A felület működési elve

A program megalkotása során az MVC modellt követtük. A modell rész már implementálva van. A modell és a grafikus felület push alapú Observer mintájú kommunikációra fog épülni, tehát a modell értesíti a felületet, ha változott az állapota. Ezt úgy érjük el, hogy minden modell osztály mellé létrehozunk egy Draw osztályt, ami a változásokat észleli, és ennek megfelelően módosítja a felületet. Kell egy osztály, ami magát a grafikus felületet tartalmazza. Az újonnan létrehozott Draw osztályoknak kell majd elérni a felület azon komponenseit, amit változtatni szeretnének. A felület komponenseinek elérésére különböző lehetőségek vannak, például a felület elérni kívánt komponenseit globális változóként kezeljük, így biztosítva a láthatóságot.
A kontroller rész megvalósítása úgy fog történni, hogy minden csempére egy MouseListener lesz állítva, így könnyen lekezelhető, hogy a csempére katintás során mi történjen. Ha egy toronyra kattintunk, akkor felugrik majd egy új ablak, ami megkérdezi a felhasználótól, hogy szeretné-e fejleszteni a tornyot, majd egyúttal ki is választhatja, hogy melyik varázskövet szeretné rátenni, ha van elég varázserő, akkor a torony fejlesztése sikeres lesz, különben kapunk egy „Nincs elég varázserőd” szövegű felugró ablakot. Akadályra kattintás ugyanez. Ha egy olyan csempére kattintunk, amire sem építeni sem fejleszteni nem lehet, akkor nem történik semmi. Ha egy olyan sima útra kattintunk, akkor felugrik egy új ablak, ami megkérdezi a felhasználótól, hogy szeretne-e ide akadályt építeni, igen válasszal és elegendő varázserővel egy akadály jelenik meg az adott csempén. Hansonló elven a torony is.

## 11.2.2 A felület osztály-struktúrája



# 11.3 A grafikus objektumok felsorolása

## 11.3.1 Enemy

- ***Felelősség***

  Ez egy absztrakt osztály, ami egy ellenséget reprezentál. Az ellenség pozíciójának, életerejének tárolásáért, az ellenség léptetéséért és az ellenségek közötti szünetek beiktatásáért felelős.

- ***Ősosztályok***

  Object -> Observable

- ***Interfészek***

  Nincs

- ***Attribútumok***
  - **# ArrayList<EnemyDraw> observers:** azon observereket tárolja, amelyeket egy Enemy osztály értesíteni akar, ha változott az állapota
  - **# boolean isSource:** ha a nextRoad egy forrás, akkor az egy pályára helyezés, különben pedig egy sima léptetés

- ***Metódusok***
  - **+ void registerObserver(EnemyDraw):** egy EnemyDraw objektumot ad hozzá az observers listához, ezzel az adott objektum felkerül az értesítési listára
  - **+ void removeObserver(EnemyDraw):** egy EnemyDraw objektumot töröl az observers listából, ezzel az adott objektum értesítése megszűnik
  - **+ void notifyObservers(Observable):** observerek értesitése az állapotváltozásról, paraméterként átadjuk a modell osztályt, ahol a változás történt
  - **+ArrayList<EnemyDraw> getObservers():** observers listát adja vissza

### 11.3.2 EnemyDraw

- ***Felelősség***

  Enemy osztály leszármazottjának egy példányának állapot változásáról értesül és ennek alapján módosítja a grafikus felületet.

- ***Ősosztályok***

  Object

- ***Interfészek***

  Observer

- ***Attribútumok***

  Nincs

- ***Metódusok***

  - **+ void update(Observable, Object)**: Módosítja a grafikus felületet a paraméterként átadott modell osztály állapotváltozása alapján. Ha meghívódik az Enemy setRoad(Road) metódusa, akkor fog meghívódni egy update, ami kirajzolja az új csempére az ellenséget.

### 11.3.3 GeometryDraw

- ***Felelősség***

  A geometria kirajzolásáért felelős osztály.

- ***Ősosztályok***

  Object

- ***Interfészek***

  Observer

- ***Attribútumok***

  Nincs

- ***Metódusok***
  - **+ void update(Observable, Object):** Módosítja a grafikus felületet a paraméterként átadott modell osztály állapotváltozása alapján.

## 11.3.4 TowerDraw

- ***Felelősség***

  A Tower osztály egy példányán az attribútumok függvényében elvégzi a szükséges grafikai változtatásokat.

- ***Ősosztályok***

  Object

- ***Interfészek***

  Observer

- ***Attribútumok***

  Nincs

- ***Metódusok***
  - **+ void update(Observable, Object):** Módosítja a grafikus felületet a paraméterként átadott modell osztály állapotváltozása alapján.

## 11.3.5 EngineDraw

- ***Felelősség***

  A győzelem és vereség felugró ablakok miatt szükséges, így hozzuk összhangba a grafikus felületet az adott állapottal

- ***Ősosztályok***

  Object

- ***Interfészek***

  Observer

· ***Attribútumok***

Nincs

· ***Metódusok***

- **+ void update(Observable, Object):** Módosítja a grafikus felületet a paraméterként átadott modell osztály állapotváltozása alapján.
- **+ void updateDefeat(Observable, Object):** A kijelzendő eredményhez szükséges flaget állítja vereség esetén.
- **+ void updateVictory(Observable, Object):** A kijelzendő eredményhez szükséges flaget állítja győzelem esetén.
- **+ void updateStatus():** A tényleges rajzoló függvényt hívja meg.

## 11.3.7 Player

- ***Felelősség***

  A játékost reprezentáló objektum. Tárolja a játékos varázserejét, ami szükséges a tornyok és az akadályok építéséhez. Akadályok és tornyok elhelyezését kezdeményezheti a játéktér megfelelő részein. Továbbá felelős a már meglévő tornyok és akadályok fejlesztéséért varázskövek felhasználásával. Felelős a kiválasztott építési terület (Tile) beépíthetőségének, valamint az építéshez szükséges varázserő megállapításáért, és Player dolga továbbá a saját varázserejének folyamatos menedzselése. A Player objektum indítja a játékot.

- ***Ősosztályok***

  Object -> Observable

- ***Interfészek***

  Nincs

- ***Attribútumok***

  - **-ArrayList<PlayerDraw> observers:** azon observereket tárolja, amelyeket egy Player osztály értesíteni akar, ha változott az állapota

- ***Metódusok***

  - **+registerObserver(PlayerDraw):** egy PlayerDraw objektumot ad hozzá az observers listához, ezzel az adott objektum felkerül az értesítési listára
  - **+removeObserver(PlayerDraw):** egy PlayerDraw objektumot töröl az observers listából, ezzel az adott objektum értesítése megszűnik
  - **+notifyObservers(Observable):** observerek értesitése az állapotváltozásról, paraméterként átadjuk a modell osztályt, ahol a változás történt
  - **+ArrayList<PlayerDraw> getObservers():** observers listát adja vissza

## 11.3.8 PlayerDraw

- ***Felelősség***

  Player osztály egy példányának állapot változásáról értesül és ennek alapján módosítja a grafikus felületet.

· ***Ősosztályok***

Object

· ***Interfészek***

Observer

· ***Attribútumok***

Nincs

· ***Metódusok***

- **+ void update(Observable, Object)**: Módosítja a grafikus felületet a paraméterként átadott modell osztály állapotváltozása alapján.
- **+ void updateMagicPower(int):** Player osztály reduceMagicPower(int) metódusának lefutására frissíti a grafikus felületen a játékos varázserejének értékét.

## 11.3.9 Fellowship

· ***Felelősség***

Egy szövetséget reprezentál, ami az összes aktív és passzív ellenséget tárolja. Az ő felelőssége még az engine időzítésének hatására az ellenségek léptetése, valamint ha valakinek elfogy az életereje, akkor ő szünteti meg azt a példányt. Ezen kívül ő adja meg az aktív ellenségek léptetésének sorrendjét is. Továbbá tárolja az összes ellenség számát.

· ***Ősosztályok***

Object -> Observable

· ***Interfészek***

Nincs

· ***Attribútumok***

- **-ArrayList<FellowshipDraw> observers:** azon observereket tárolja, amelyeket egy Player osztály értesíteni akar, ha változott az állapota

- ***Metódusok***
  - **+registerObserver(FellowshipDraw):** egy FellowshipDraw objektumot ad hozzá az observers listához, ezzel az adott objektum felkerül az értesítési listára
  - **+removeObserver(FellowshipDraw):** egy FellowshipDraw objektumot töröl az observers listából, ezzel az adott objektum értesítése megszűnik
  - **+notifyObservers(Observable):** observerek értesitése az állapotváltozásról, paraméterként átadjuk a modell osztályt, ahol a változás történt
  - **+ArrayList<PlayerDraw> getObservers():** observers listát adja vissza

## 11.3.10 FellowshipDraw

- ***Felelősség***

  Fellowship osztály egy példányának állapot változásáról értesül és ennek alapján módosítja a grafikus felületet.

- ***Ősosztályok***

  Object

- ***Interfészek***

  Observer

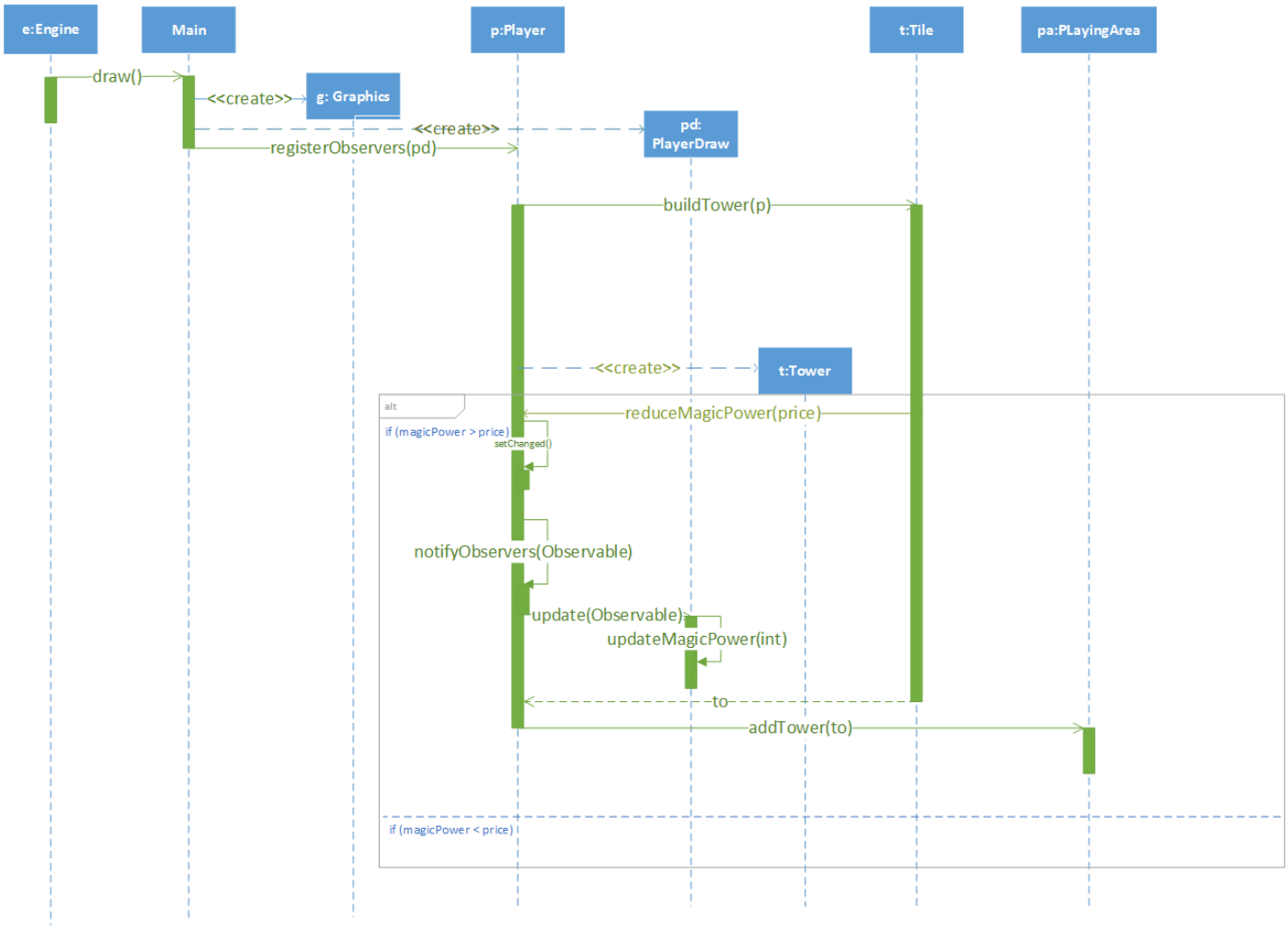
- ***Attribútumok***

  Nincs

- ***Metódusok***
  - **+ void update(Observable, Object)**: Módosítja a grafikus felületet a paraméterként átadott modell osztály állapotváltozása alapján.

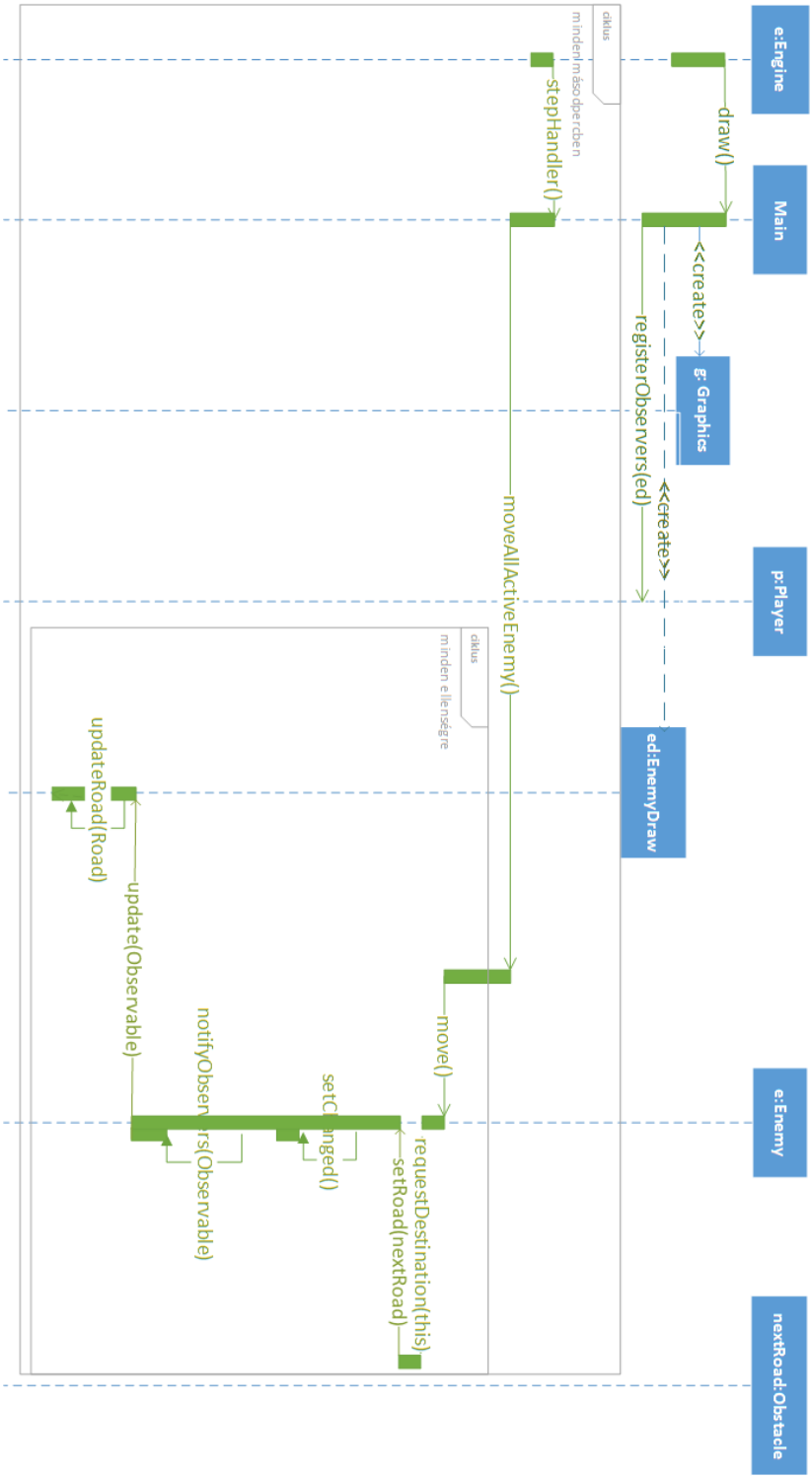
# 11.4 Kapcsolat az alkalmazói rendszerrel

## 11.4.1 Játékos varázserejének frissítése a felületen torony építésre

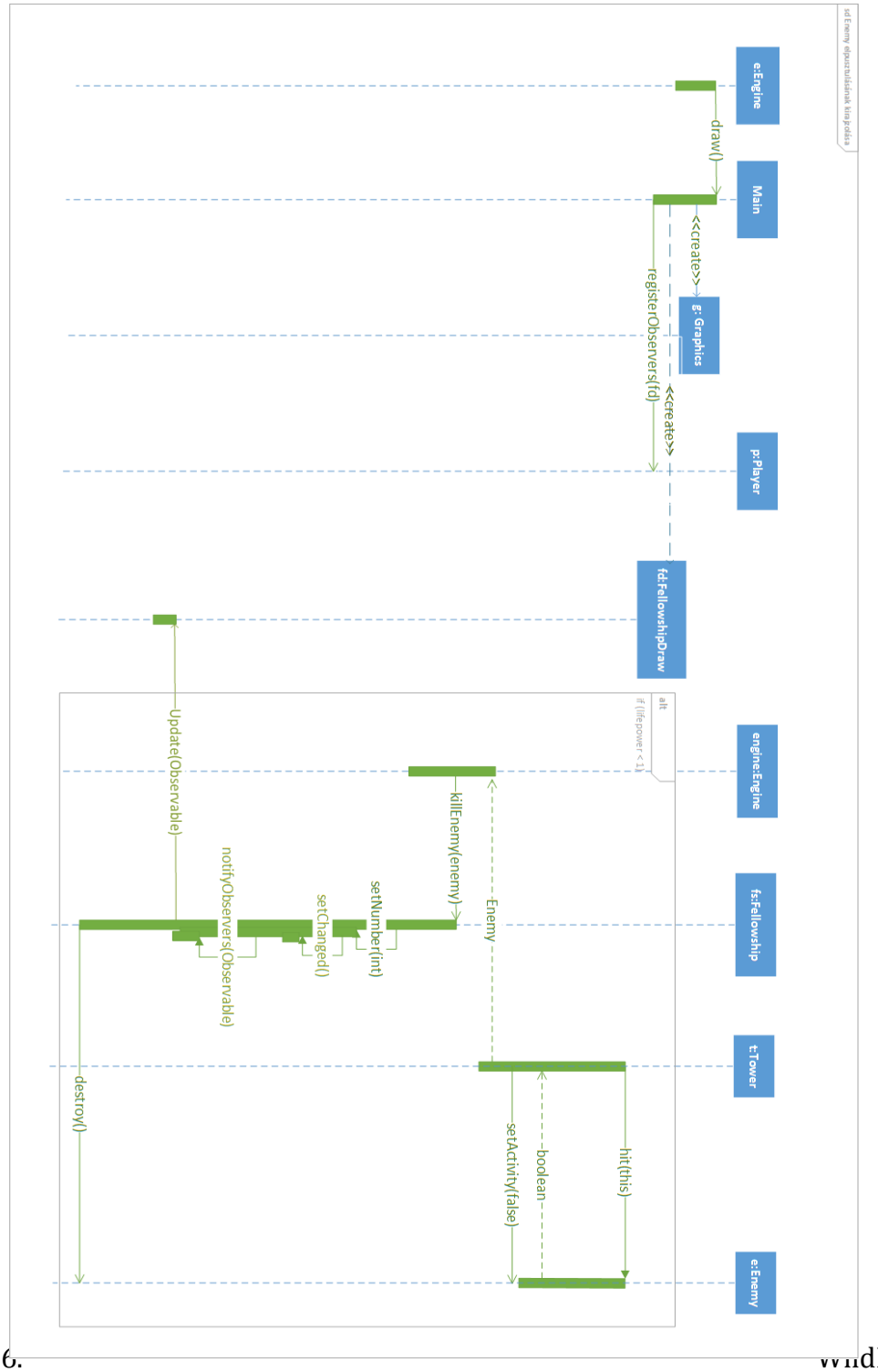



Játékos varázsereje még változik torony fejlesztésre is, de az Observer-es része ugyanaz, mint a torony építésnek, ezért nem készítettünk neki külön szekvencia diagramot.

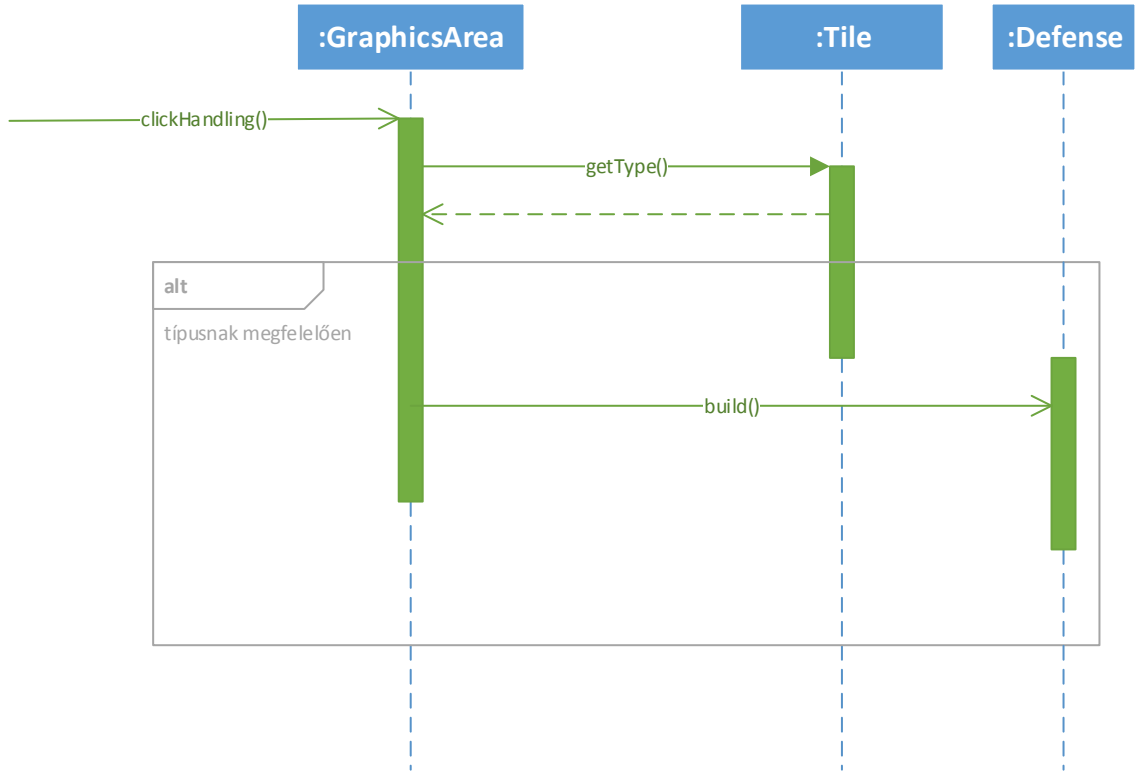
## 11.4.2 Ellenség léptetésének frissítése grafikus felületen



### 11.4.3 Ellenség elpusztul frissítése a grafikus felületen



### 11.4.4 Kattintás kezelés




## 11.5 Napló

| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.04.23 20:00 | 1.5 óra | Tóth<br>Bana<br>Török<br>Czirják | Konzultáció |
| 2014.04.23 21:00 | 0.5 | Csontos | Konzultáció |
| 2014.04.23 23:00 | 3 óra | Csontos | Observer minta |

| | | | |
|---|---|---|---|
| 2014.04.24. 8:00 | 7 óra | Török | Grafikus interfész |
| 2014.04.24 00:00 | 1.5 óra | Csontos | Observer minta<br>11.2.1 |
| 2014.04.26 18:20 | 1 óra | Bana<br>Czirják<br>Tóth<br>Török | Értekezlet |
| 2014.04.26 10:30 | 1 óra | Czirják<br>Csontos<br>Török<br>Tóth<br>Bana | Értekezlet<br>Feladatok kiosztása |
| 2014.04.26 11:30 | 1óra | Tóth | Osztálydiagram és Struktúradiagram megrajzolása |
| 2014.04.26 00:00 | 2 óra | Csontos | 11.3.1, 11.3.2, 11.3.7, 11.3.8<br>11.4.1 tervezése |
| 2014.04.26. 13:30 | 3.5 óra | Török | 11.3.9, 11.3.10<br>11.4.1 elkészítése<br>11.1 ábra javítása |
| 2014.04.27. 14:00 | 1,5 óra | Czirják | 11.3.3-11.3.6 |
| 2014.04.27 11:00 | 0.5 óra | Csontos | 11.1 és 11.2 leírás elkészítése |
| 2014.04.27 20:00 | 3.5 óra | Czirják | 11.4.2, 11.4.3 |
| 2014.04.28 00:00 | 1 óra | Csontos | Dokumentum formázása |

# 13. Grafikus változat

## 13.1 Fordítási és futtatási útmutató

### 13.1.1 Fájllista

| Fájl neve | Méret [bájt] | Keletkezés ideje | Tartalom |
|---|---|---|---|
| Defense.java | 810 | 2014.04.17 | Defense interfészt valósítja meg. |
| Dwarf.java | 1725 | 2014.04.17 | Dwarf osztályt valósítja meg. |
| Elf.java | 1698 | 2014.04.17 | Elf osztályt valósítja meg. |
| Enemy.java | 6684 | 2014.04.18 | Enemy osztályt valósítja meg. |
| EnemyDraw.java | 9489 | 2014.05.11 | EnemyDraw osztályt valósítja meg. |
| Engine.java | 6197 | 2014.04.18 | Engine osztályt valósítja meg. |
| EngineDraw.java | 1570 | 2014.05.11 | EngineDraw osztályt valósítja meg. |
| Fellowship.java | 8194 | 2014.04.18 | Fellowship osztályt valósítja meg. |
| FellowshipDraw.java | 3755 | 2014.05.11 | FellowshipDraw osztályt valósítja meg. |
| Geometry.java | 2448 | 2014.04.17 | Geometry osztályt valósítja meg. |
| GeometryDraw.java | 5773 | 2014.05.11 | GeometryDraw osztályt valósítja meg. |
| GraphicsArea.java | 11660 | 2014.05.11 | GraphicsArea osztályt valósítja meg. |
| Hobbit.java | 1723 | 2014.04.17 | Hobbit osztályt valósítja meg. |
| Human.java | 1713 | 2014.04.17 | Human osztályt valósítja meg. |
| MagicRock.java | 1259 | 2014.04.18 | MagicRock osztályt valósítja meg. |
| Main.java | 6167 | 2014.04.18 | Main osztályt valósítja meg. |
| Mountain.java | 241 | 2014.04.17 | Mountain osztályt valósítja meg. |

| | | | |
|---|---|---|---|
| Obstacle.java | 3128 | 2014.04.18 | Obstacle osztályt valósítja meg. |
| Player.java | 4473 | 2014.04.18 | Player osztályt valósítja meg. |
| PlayerDraw.java | 541 | 2014.05.11 | PlayerDraw osztályt valósítja meg. |
| PlayingArea.java | 14096 | 2014.04.18 | PlayingArea osztályt valósítja meg. |
| Position.java | 560 | 2014.04.17 | Position osztályt valósítja meg. |
| Road.java | 1866 | 2014.04.18 | Road osztályt valósítja meg. |
| Source.java | 276 | 2014.04.17 | Source osztályt valósítja meg. |
| Tile.java | 1711 | 2014.04.18 | Tile osztályt valósítja meg. |
| Tower.java | 7380 | 2014.04.18 | Tower osztályt valósítja meg. |
| TowerDraw.java | 1578 | 2014.05.11 | TowerDraw osztályt valósítja meg. |
| Writer.java | 263 | 2014.04.22 | Writer osztályt valósítja meg. |

### 13.1.2 Fordítás és telepítés

A .java forrásfájlok mellett található egy compile.bat fájl, ami egyenként lefordítja az összes olyan
java osztályt, ami nem absztrakt. Ezt a javac parancsszóval érhetjük el, ami a Command Prompt-ból elérhető, ha megadjuk a JDK helyét. Ezt az alábbi paranccsal tehetjük meg: “set path=%path%; JDK elérési útvonala”.
Ha ezt sikeresen beállítottuk, akkor a compile.bat fájlt futtatva létrejönnek a .class fájlok, ugyanazon mappában, ahol a .java fájlok vannak.

### 13.1.3 Futtatás

Ha megtörtént a fordítás, akkor a run.bat fájlt futtatva elindul a program és megjelenik a grafikus felület, ami egy felugró ablakban egy map nevet kér, jelen esetben a “map.txt” nevet kell beírni, melyre statikusan betöltődik a pálya és egyből el is kezdődik a játék.
A játék során csempére bal kattintással lehet tornyot illetve akadályt építeni illetve fejleszteni. Ha olyan csempére akarunk építeni, amelyre nem lehet, akkor kattintáskor nem történik semmi.

## 13.2 Értékelés

| Tag neve | Munka százalékban |
|---|---|
| Bana Szabolcs | 21,31% |
| Czirják Balázs | 16,18% |
| Csontos Valentin | 26,47% |
| Tóth Dávid | 17,83% |
| Török Odett | 18,20% |

## 13.3 Napló

| Kezdet | Időtartam | Résztvevők | Leírás |
|---|---|---|---|
| 2014.04.29. 22:00 | 1,5 óra | Török | Spreadsheet |
| 2014.04.30 13:30 | 2,5 óra | Török<br>Csontos | Grafikus felület implementációja |
| V2014.04.30 | 1,5 óra | Csontos | Csempe típusok kirajzoltatása |
| 2014.05.01 | 3 óra | TöröCk<br>Csontos | Torony építés<br>Torony fejlesztés |
| 2014.05.03 | 3 óra | Tóth | Kép beolvasás, megjelenítés |
| 2014.05.03 | 2 óra | Tóth | Objektumok rajzainak optimalizálása a grafikus felülethez |
| 2014.05.10 13:00 | 1 óra | Bana | Akadály rárakása a csempére, mikor az ellenség rajta áll |
| 2014.05.11 | 2 óra | Tóth | Torony és akadály fejlesztés |
| 2014.05.11. 8:00 | 2 óra | Török | EngineDraw, Engine, popup ablakok |

| | | | |
|---|---|---|---|
| 2014.05.11. 22:00 | 3 óra | Czirják | Köd leereszkedés grafikus megjelenítése |
| 2014.05.11. 21:00 | 1 óra | Bana<br>Czirják<br>Csontos<br>Tóth<br>Török | Skype konferencia |
| 2014.05.11 22:00 | 2 óra | Tóth | Ellenség letörlése elpusztulás esetén |
| 2014.05.11 15:00 | 2 óra | Bana | dinamikusan elhelyezett akadály is lassítson |
| 2014.05.12 09:00 | 2 óra | Csontos | Köd kirajzolása<br>Akadály kirajzolása + lassítás javítása<br>Ellenségek számának kiíratás a felületre |
| 2014.05.12 14:00 | 2 óra | Csontos | Ellenség pályára helyezése |
| 2014.05.12. 17:00 | 3 óra | Bana<br>Czirják<br>Csontos<br>Török | Utolsó simítások |
| 2014.05.13 13:30 | 1 óra | Bana | Kód commentezése |
| 2014.05.13 11:00 | 2 óra | Csontos | Ellenség kettészakítás kirajzolása<br>Kommentezés |
| 2014.05.13. 13:45 | 1.25 óra | Török | Kommentezés: Hobbit, Elf, Human, Dwarf, EngineDraw, PlayerDraw, EnemyDraw, spreadsheet |
| 2014.05.13 14:15 | 0.5 óra | Tóth | GraphicsArea, EnemyDraw kommentezés |
| 2014.04.13 13:30 | 0.5 óra | Bana | Végső simítások, commentek ellenörzése |

# 14. Összegzés

## 14.1 Projekt összegzés

### 14.1.1 Mit tanultak a projektből konkrétan és általában?

Fontosabb észrevételünk, hogy a csapatmunka során az egyik legnehezebb feladat a megfelelő kommunikáció, pontosabban a kommunikáció módjának megválasztása, az összhang, és a megfelelő időpont megválasztása.
Érdemes pontosan tudni egymás kapacitásairól: ha egy adott csapattag váratlanul elérhetetlenné válik, vagy túlterhelődik, akkor értesítse a többi csapattagot, hogy a társai időben kompenzálni tudják a kiesést, és át tudják venni a rá hárult teendőket.

### 14.1.2 Mi volt a legnehezebb és a legkönnyebb?

A legnehezebbnek a csapatmunka összehangolása bizonyult. Miután ez megtörtént, már gördülékenyen haladtak a fázisok kidolgozásai. A legjelentősebb feladatnak megvalósítás szintjén az adott rész követelményeinek becslése volt tapasztalat hiányában. Adódott olyan eset, amikor az általunk implementált feladatot a követelményeknek megfelelőnek tartottuk, azonban az értékelés során derült ki, hogy bizonyos dolgokban tévúton jártunk.
Ami meglepően összehangoltan alakult az első szakaszban lévő modellezési problémák feloldása, valamint a többi feladathoz mérten a program implementációja (kódja) már a tervezési fázist követően könnyen ment.

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

A Szkeleton számíthatna nagyobb súlyozással a végső jegyszámításnál, de ettől eltekintve az egyes mérföldköveken belüli a pontok eloszlásai érthetőek és teljes mértékben egyetértünk vele.

### 14.1.4 Milyen változtatási javaslatuk van?

Hiányoltuk a folyamatos kérdezési/konzultálási lehetőséget.
Emellett a csapat három tagjának hétfői időponttól eltérő leadási határidő lett volna ideális. A hétvége nagyon sokat segített, azonban ideális lenne, ha több konzultációs időpont közül lehetne választani.

# 14.2 Táblázat

| Név | Összesített munkaóra | %-ban |
|---|---|---|
| Bana Szabolcs | 146 | 21.4% |
| Czirják Balázs | 110 | 16.1% |
| Csontos Valentin | 181 | 26.5% |
| Tóth Dávid | 122 | 17.9% |
| Török Odett | 124 | 18.1% |
| Összesen: | 683 | 100% |

| Név | Analízis/óra | Szkeleton/óra | Prototípus/óra | Grafikus/óra | Összesen: |
|---|---|---|---|---|---|
| Bana Szabolcs | 58 | 42 | 34 | 12 | 146 |
| Czirják Balázs | 43 | 16 | 36 | 15 | 110 |
| Csontos Valentin | 57 | 39 | 58 | 27 | 181 |
| Tóth Dávid | 46 | 21 | 39 | 16 | 122 |
| Török Odett | 50 | 21 | 25 | 28 | 124 |
| Összesen: | 254 | 139 | 182 | 98 | 683 |

### Forrássorok száma

| | |
|---|---|
| Szkeleton | 1019 |
| Prototípus | 2161 |
| Grafikus | 3425 |