

4. Analízis modell kidolgozása 2

10 – itee_team

Konzulens:

Budai Péter

Csapattagok:

Elekes Tamás Csaba	E30C8Z	elekestamas22@gmail.com
Seres Márk Dániel	EUQ8V5	seres.dani@gmail.com
Rédey Bálint Attila	DAVRIZ	botvinnik09@gmail.com
Nagy András	VWBG06	nagyandrasgall@gmail.com
Fuksz Domonkos	GIT0NQ	fukszdomonkos@gmail.com

2014. március 10.

4. Analízis modell kidolgozása 2

4.1. Objektum katalógus

4.1.1. Akadály (Obstacle)

Az akadály (Obstacle) objektum felelőssége egyrészt az, hogy amikor áthalad rajta egy ellenség (Enemy) lelassítsa. Másfelől felelőssége az is, hogy egy-egy ellenség áthaladtával amortizálódjon, valamint ha már teljesen elhasználódott, értesítse azt az út elemet (Path), amelyiken áll.

4.1.2. Ellenség (Enemy)

Egy ellenséget (tünde, hobbit, törp vagy ember) megvalósító objektum. Az ő felelőssége, hogy egy adott lövedék (Bullet) hatására sebződjön, vagy ha már sokat sebződött, akkor haljon meg, valamint az, hogy celláról cellára mozgassa magát, és ha eléri a végzet hegyét értesítse a játék (Game) osztályt, hogy intézkedjen.

4.1.3. Játék (Game)

A játék (Game) objektum felelőssége többek közt a játék ütemezése, az idő múlásának kontrollálása. Ezenkívül az inicializálás, vagyis a játék kezdeti állapotának felvétele, továbbá a modell állapotának folyamatos változása miatti frissítés, valamint ennek a grafikus felületen való megjelenítése. A game osztály hozza létre az ellenségeket és indítja el őket az úton.

4.1.4. Kristály (Gem)

Ha a játékos vesz a toronyra/akadályra valamilyen kristályt, akkor jön létre, megkapja a torony, és beépíti magába. Felelőssége, hogy általa érvényre jussanak a fejlesztések.

4.1.5. Lövedék (Bullet)

A toronyok egy lövedéket tárolnak, amit minden lövésnél átadnak a lövő függvénynek. Ennek a lövedéknek a feladata, hogy az ellenségnek megmondja mennyit sebez rajta.

4.1.6. Mező (Field)

A Field osztály a Cell osztály leszármazottja. A nem út típusú cellákat (mező) reprezentálja. Egy mezőre egy torony helyezhető.

4.1.7. Pálya (Map)

A Map osztály a játéktér elemeit, mint cellák tárolja, egy két dimenziós tömbben. Megadja minden egyes cellához, a szomszédjai referenciáját.

4.1.8. Torony (Tower)

Az egyetlen tervezett toronytípus, le lehet rakni a pályán az úton kívül bárhova. A hatósugarába belépett ellenségekre lőnie kell, lehet fejleszteni lövési sebességét, erejét, újratöltési idejét és egy ellenségtípusra még hatásosabbá tenni a lövedékeit. A játékos varázserőért tud lerakni, illetve eladni toronyokat. Ez a legfontosabb eszköz amivel a játékos meg tudja akadályozni az ellenségek célbajutását.

4.1.9. Út (Path)

A Path a Cell osztály leszármazottja. Az út típusú cellákat reprezentálja. Tartalmazza a rajta lévő ellenségeket és esetleg akadályt. Minden út tudja azt is, hogy hova lehet lépni róla egy lépésben.

4.1.10. Obstaclehez tartozó kristályok: Intensity, Repair

Akadályra helyezhető kristályok, ami növeli a lassítás értékét vagy megjavítja az akadályt.

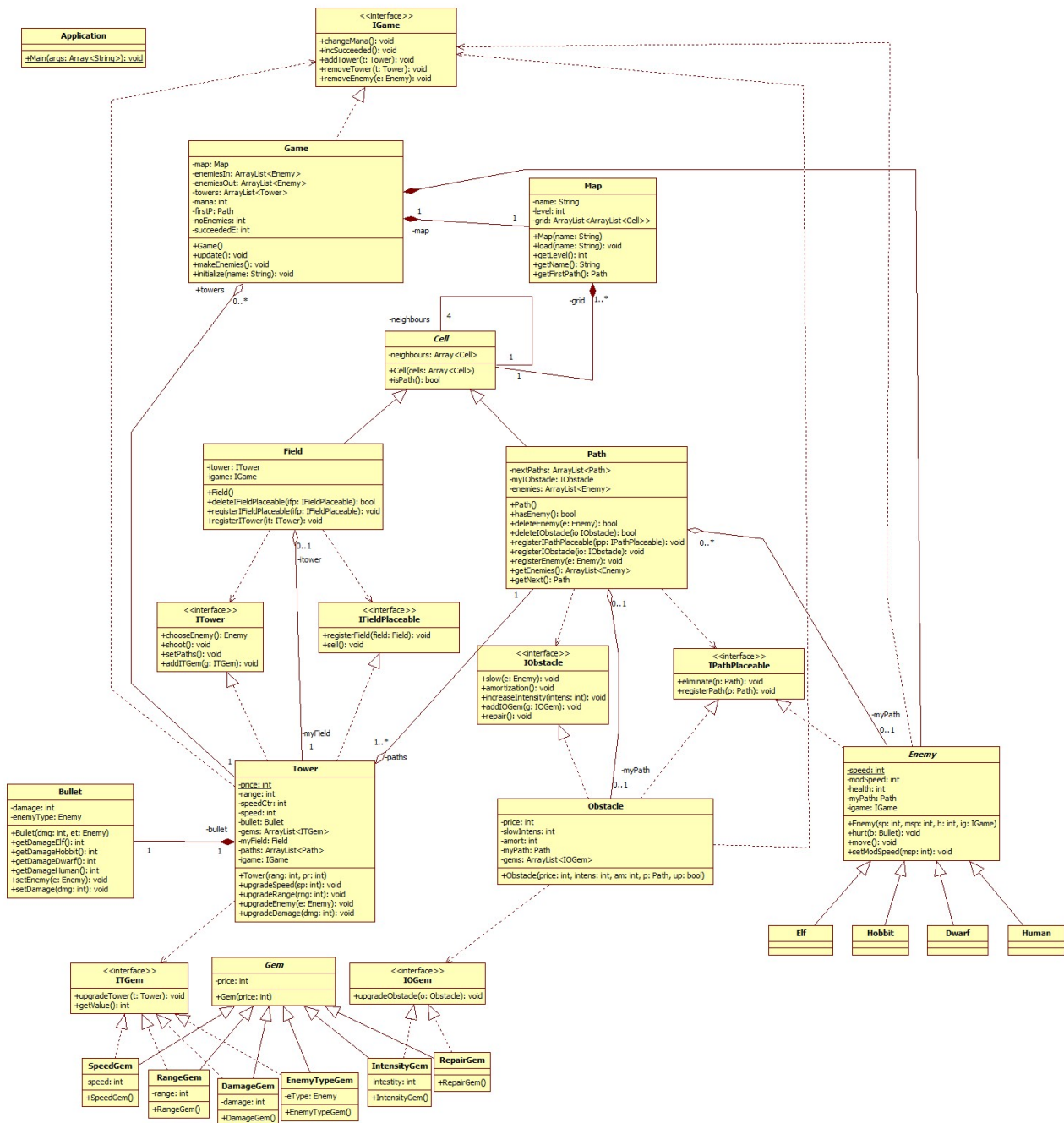
4.1.11. Towerhez tartozó kristályok: Range, Speed, Damage, EnemyType

A torony lövésének hatósugarát, gyorsítását, sebzését, ellenféltípusra sebzés, növelő kristályok osztályai.

4.1.12. ITGem, IOGem

Ezen keresztül lehet kezelni a kristályokat, és a tornyok/akadályok fejlesztését. Eladáshoz ITGem-nek felelősége, hogy az árát a kristálynak visszaadja.

4.2. Statikus struktúra diagramok



4.1. ábra. Osztálydiagram

4.3. Osztályok leírása

4.3.1. Bullet

- Felelősség
Ellenség kapja meg, és ebből tudja meg mennyire sebződik.
- Ősosztályok

2014. március 10.

Object

- Interfészek
Nincs
- Attribútumok

int damage alapsebzés

Enemy enemyType a torony itt tárolja, hogy melyik ellenség típusra erősebb a sebzése

- Metódusok

Bullet(int damage, Enemy enemyType) Konstruktor

int getHobbitDamage() ha hobbitot sebez, ezzel a függvénnyel kérdezi le a sebzés értékét.

int getHumanDamage() ha embert sebez, ezzel a függvénnyel kérdezi le a sebzés értékét.

int getDwarfDamage() ha törpöt sebez, ezzel a függvénnyel kérdezi le a sebzés értékét.

int getElfDamage() ha tündét sebez, ezzel a függvénnyel kérdezi le a sebzés értékét

void setEnemy(Enemy e) beállítja az ellenséget akire specializált

void setDamage(int damage) beállítja a lövedék sebzését

4.3.2. Enemy

- Felelősség
Tudja, hogy mennyi élete van még, milyen sebességgel haladt eredetileg, és milyen sebességgel halad most. Ez egy absztrakt ősszotály, ami összefogja a 4 ellenségtípust (Hobbit, Elf, Dwarf, Human).
- Ősszotályok
Object
- Interfészek
IPathPlaceable
- Attribútumok

int speed A két lépés között eltelt idő.

int modSpeed Az ellenség belső idő mérője. A setModSpeed változtathatja – jellemzően negatív irányba, akadályokon.

int health Életerejét tárolja ebben. Hurt függvényben csökkenti.

Path myPath az a mező, ahol tartózkodik

IGame igrade ezen keresztül tudja módosítani a manát, amikor meghal, illetve ha elér a végzet hegyére módosítani a számlálót (Game.succeededE), hogy nőjön egyel

- Metódusok

hurt(Bullet b) sebződik (abstract method)

move() mozog, a következő path-ra lép, cellát vált

Enemy(int sp, int msp, int h, ig: IGame) konstruktor

void setModSpeed(int msp) modSpeed változót változtatja. Lassítani lehet vele.

4.3.3. Enemy subclasses: Elf, Hobbit, Dwarf, Human

- Felelősség
Sebződés: egy Bullet alapján a saját életét csökkenteni, és ha kell, meghalni. Tehát felüldefiniálja az Enemy őssosztály hurt metódusát.
- Őssosztályok
Object → Enemy
- Interfészek
IPathPlaceable
- Metódusok

hurt(Bullet) sebződik

4.3.4. Game

- Felelősség
Lásd objektum katalógus.
- Őssosztályok
Object
- Interfészek
Nincs
- Attribútumok

Map map játék térképe

List<Enemy> enemiesOut pályára még be nem lépett ellenségek

List<Enemy> enemiesIn pályára már belépett ellenségek

List<Tower> towers a tornyok, amik a pályán vannak

int mana maradék varázserő

Path firstP az út kezdő cellája

int noEnemies kezdeti hullámérték, amely folyamatosan nő, azt mutatja meg, hogy következő körben hány ellenséget kell létrehozni és beküldeni a pályára

int succeeded végzet hegyét elért enemy-k száma

- Metódusok

void update() frissíti a modellt, grafikát

void initialize(String name) kiinduló állapot felvétele

Game() konstruktor

void makeEnemies() létrehoz néhány ellenséget, ezeket beteszi az enemiesIn-be

4.3.5. IGame

- Felelősség
Az IGame interfész szolgáltatást nyújt az akadályoknak, tornyoknak, ellenségeknek, hogy rajta keresztül manát írjanak jóvá/csökkentsenek, illetve ellenségek esetén a végzet hegyét elért ellenségek számát módosítsák. Speciális interfész a Game osztályhoz.

- Metódusok

void changeMana() manát megváltoztató metódus

int incSucceeded() succeededE értékét megváltoztató metódus

void addTower(Tower t) hozzáad egy tornyot a listához

void deleteEnemy(Enemy e) ellenség törlése enemiesIn-ből

void deleteTower(Tower t) torony törlése towers-ből

4.3.6. Gem

- Felelősség
A kristály osztály felel a torony fejlesztéséért.
- Ősosztályok
Object
- Interfészek
Nincs
- Attribútumok

static int price az ár, amennyi varázserőbe kerül.

- Metódusok

Gem(int price) konstruktor

4.3.7. IObstacle

- Felelősség
Olyan metódusok használatát teszi lehetővé, amelyek az Obstacle típusú elemek viselkedését modellezzik
- Ősosztályok
Nincs
- Metódusok

void slow(int intensity, Path p) szól p-nek, hogy lassítsa le az ellenséget intensity-vel

void amortization() amortizál

void increaseIntensity(int intens) megnöveli az intensity-t intens-el

void addIOGem(IOGem: iog) iog kristályt hozzáadja az akadályhoz

repair() megjavítja az akadályt

4.3.8. Obstacle

- Felelősség
Lásd objektum katalógus.
- Ősosztályok
Nincs
- Interfészek
IObsacle, IPathPlaceable
- Attribútumok

int slowIntens lassítás mértéke

Path myPath a mező, amin rajta van

int amort az elhasználódottság mértéke

static final int price az ára

ArrayList<IOGem> gems a megvett kristályok listája

- Metódusok

Obstacle(int intens, Path p, int amort, int price, bool up) konstruktor

4.3.9. ITower

- Felelősség
A torony funkciói vannak benne.
- Metódusok

void setPaths() a saját cellájából kiindulva a hatósugarával lefedett területen felkeresi, és beregisztálja a paths listába a path cellákat.

void shoot() A torony akkor lő, ha letelt az újratöltési idő, ekkor megnézi, hogy lőtávon belül van-e ellenség, és ha van meghívja a sebzés függvényét, átadva paraméterként a lövedékét.

void addITGem(ITGem gem) paraméterként megkapja a kiválasztott kristályt, a gameStat-ot frissíti, és a bullet-et is.

Enemy chooseEnemy() A torony tárolja a hatókörbe eső path cellákat. Minden tick-ben végig megy rajtuk, és kiválaszt egyet, amelyiken van ellenség, és oda fog lőni. Az ellenséggel tér vissza.

4.3.10. Tower

- Felelősség
Lásd objektumkatalógus
- Ősosztályok
Nincs
- Interfészek
ITower, IFieldPlaceable
- Attribútumok

static final int price az ára varázserőben.

int range lőtáv, hatókör.

int speedCtr A torony belső idő mérője. Ezt vizsgálja minden lövés előtt, hogy eltelt e elég idő.

int speed két lövés között eltelt minimális idő.

Bullet bullet A torony tárol egy lövedéket, mindig ezt lövi ki.

ArrayList<ITGem> gems A megvásárolt kristályokat tárolja.

Field myField a mezőt tárolja amin áll.

ArrayList<Path> paths Hatósugárba eső út cellák.

Field myField mező, amin áll.

IGame igrade Egy interfész a játék logikára, amivel a bejutott ellenségek számát, és a varázserőt is lehet állítani.

- Metódusok

Tower(int rang, int pr) konstruktor

void upgradeSpeed(int sp) fejleszti a lövési sebességét.

void upgradeRange(int rng) fejleszti a lőtávot.

void upgradeEnemy(Enemy e) egy ellenségtípusra növeli a sebzést.

void upgradeDamage(int dmg) növeli s sebzést.

4.3.11. Map

- Felelősség
Ld. objektum katalógus
- Ősosztályok
Nincs
- Interfészek
Nincs
- Attribútumok

String name a pálya neve, egyben az azonosítója

int level a pálya szintje

Array<Array<Cell>> grid A cellákat tartalmazó 2 dimenziós tömb

- Metódusok

Map(string name) az osztály konstruktora, a paraméterként megadott névvel rendelkező fájlból betölti a pálya térképét

void load(string name) megnyitja a paraméterként kapott nevű fájlt, és abból betölti a pálya celláinak tulajdonságait, felépíti a pályát.

int getLevel() visszaadja a pálya szintjét.

String getName() visszaadja a pálya nevét

Path getFirstPath() visszaadja a pálya belépési pontjának referenciáját

4.3.12. Cell

- Felelősség
A Cell a pálya egy egységét reprezentáló osztály. Létrehozásakor megkapja a 4 szomszédja referenciáját. Maga a cella nem tudja, hogy hol van a térképen. A cella tárolja a rajta éppen tartózkodó ellenségek referenciáit. A Cell osztály absztrakt.
- Ősosztályok
Nincs
- Interfészek
Nincs
- Attribútumok

Array<Cell> neighbours 4 elemű tömb, tárolja 4 irányban a szomszédjai referenciáját.

- Metódusok

Cell(Array<Cell>) konstruktor, paraméterként kapja a szomszédos mezők referenciáit.

bool isPath() olyan értékkel tér vissza amilyen típusú a cella

4.3.13. Field

- Felelősség
Ld. objektum katalógus
- Ősosztályok
Object → Cell
- Interfészek
Nincs
- Attribútumok

ITower itower a mezőn álló torony interfészű elem tárolása

- Metódusok

bool isPath() hamis értékkel tér vissza

void addFieldPlaceable() egy új tornyot ad hozzá a mezőhöz

void deleteFieldPlaceable(IFieldPlaceable ifield) eltávolítja a tornyot a mezőről

void registerITower(itower ITower) beteszi ifieldbe a kapott tornyot

Field() konstruktor

4.3.14. Path

- Felelősség
Ld objektum katalógus
- Ősosztályok
Object → Cell

- Interfészek
Nincs
- Attribútumok

IObstacle iobstacle az esetleg az úton levő akadályt tárolja

ArrayList<Enemy> enemies az éppen áthaladó ellenségek listája

ArrayList<Path> paths következő path-ok címei

- Metódusok

ArrayList<Enemy> hasEnemy() visszaadja a rajta lévő ellenségek listáját

bool isPath() igaz értékkel tér vissza

void deletePathPlaceable(IPathPlaceable ipath) kitörli a tárolójából a paraméterként kapott referenciával megegyező tárolt referenciát

void registerIPathPlaceable(IPathPlaceable ipath) beregisztrálja a paraméterként kapott objektumot, mint saját magán tartózkodó ellenség

bool hasEnemy() megmutatja, hogy van-e a cellán ellenség

void registerEnemy(Enemy e) a kapott ellenséget beteszi az enemies-be

void registerObstacle(Obstacle o) a o kapott akadály lesz az obstacle

ArrayList<Enemy> getEnemies() visszatér az enemies-el

Path getNext() paths-ből ad vissza egy elemet

4.3.15. Towerhez tartozó krsitályok: Range, Speed, Damage, EnemyType

- Felelősség
Ld objektum katalógus
- Ősosztályok
Object → Gem
- Interfészek
ITGem
- Attribútumok

int range/ speed/ damage/ eType

- Metódusok

Konstruktorok

4.3.16. Obstaclehez tartozó kristályok: Intensity, Repair

- Felelősség
Ld objektum katalógus
- Ősosztályok
Object → Gem

- Interfészek
IOGem
- Attribútumok

int intensity

- Metódusok

Konstruktorok

4.3.17. IOGem

- Felelősség
Akadályra helyezhető kristályok interfésze.
- Metódusok

void upgradeObstacle(Obstacle o) a kapott akadályt fejleszti.

4.3.18. ITGem

- Felelősség
Toronyra illeszthető kristályok interfésze.
- Metódusok

void upgradeTower(Tower t) fejleszti a kapott t tornyot magával

int getValue() visszaad egy, a torony árával képzett értéket, a torony eladásakor jóváírandó mana érték kiszámításához

4.3.19. IFieldPlaceable

- Felelősség
Interfész a mezőre helyezhető osztályok számára. Azonosítja azokat az objektumokat, amelyeket csak a mező típusú pályaelem tartalmazhat.
- Metódusok

void registerField(Field field) a mezőre helyezhető objektumnak megadja paraméterben annak a mezőnek a referenciáját, amelyikre helyezve lesz.

void sell() a mezőre helyezhető objektum eladása, annak megfelelő manát ad a játékosnak amennyit az objektum ér, majd a mező törli magáról az objektumot.

4.3.20. IPathPlaceable

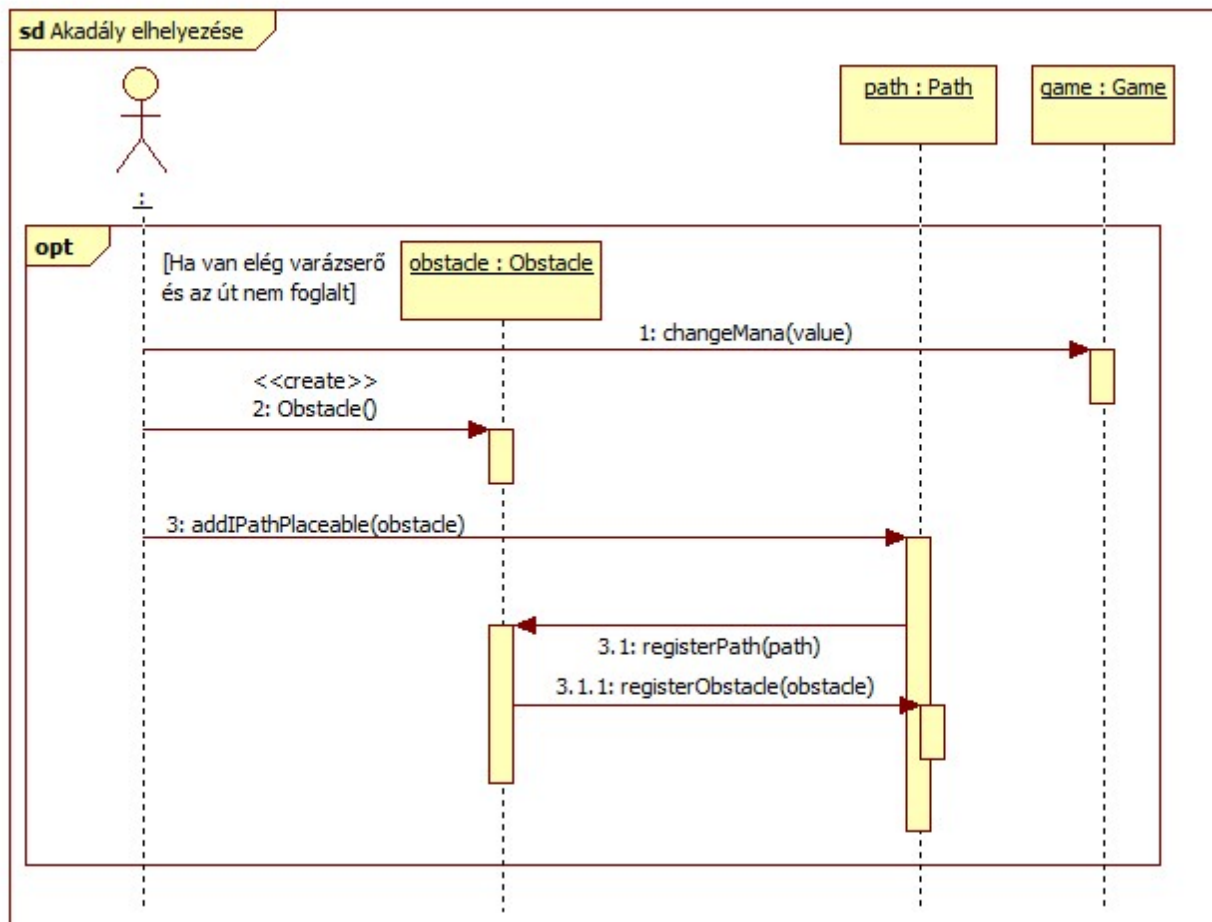
- Felelősség
Interfész az útra helyezhető osztályok számára. Azonosítja azokat az objektumokat, amelyeket csak az út típusú pályaelem tartalmazhat.
- Metódusok

void eliminate(Path p) az útra helyezhető objektum eltávolítása az útról.

void registerPath(Path p) az útra helyezhető objektumnak megadja paraméterben annak az útnak a referenciáját, amelyikre helyezve lesz.

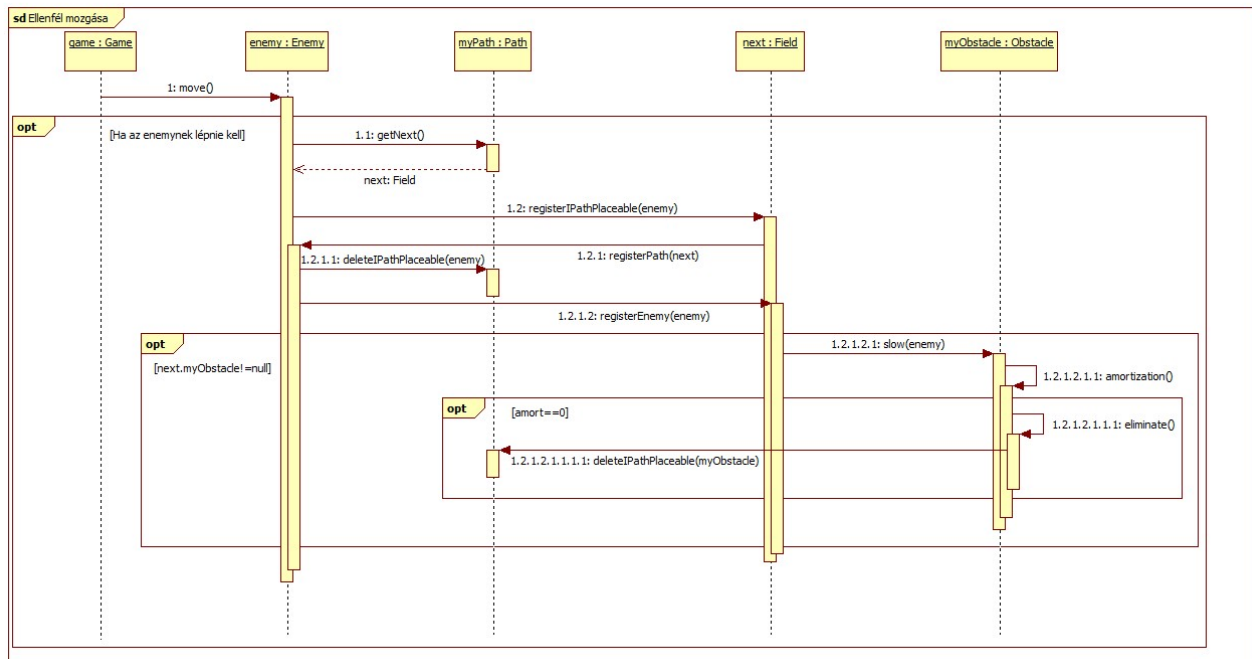
4.4. Szekvencia diagramok

4.4.1. Akadály elhelyezése



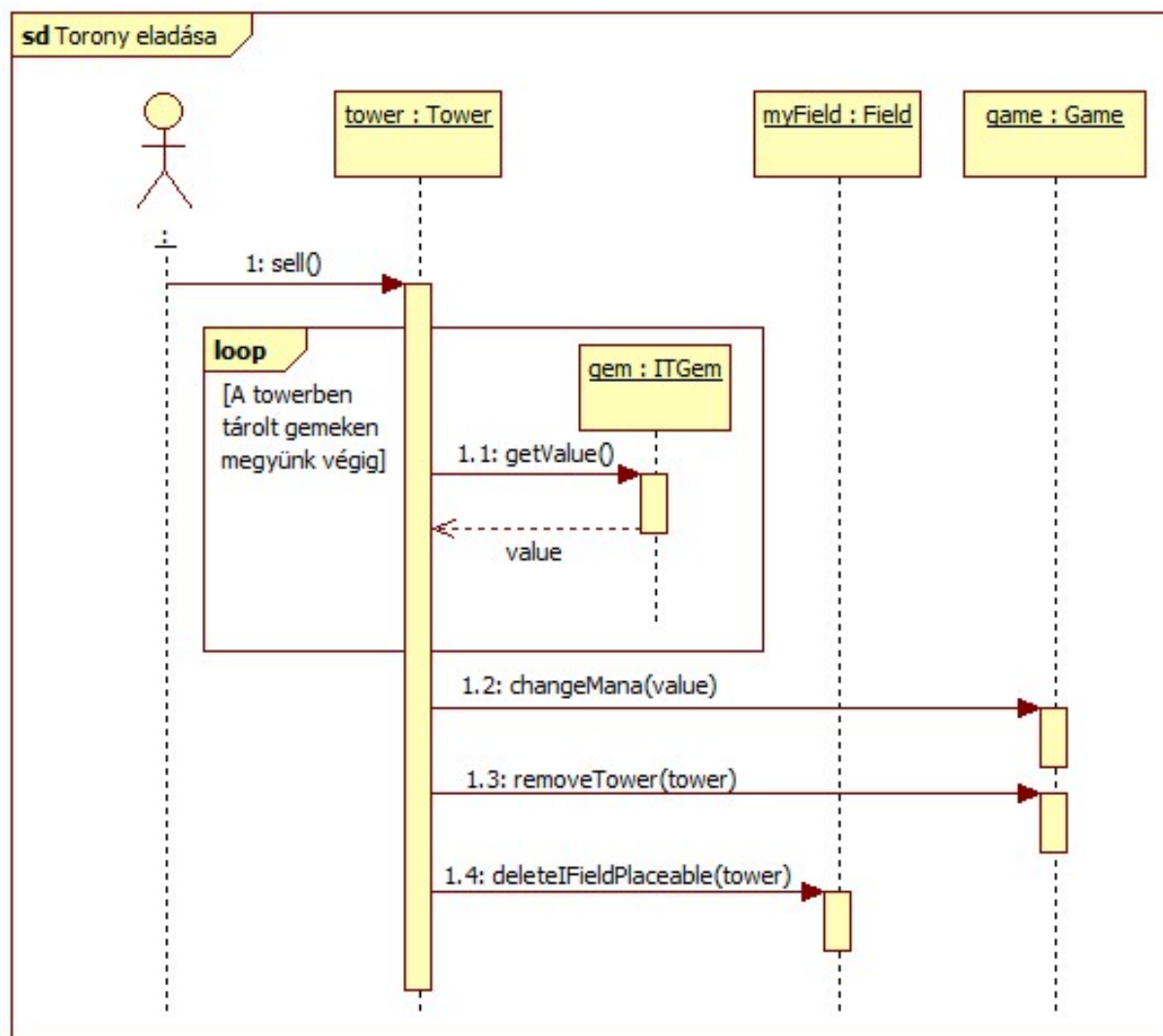
4.2. ábra. Akadály elhelyezése szekvenciadiagram

4.4.2. Ellenfél mozgása



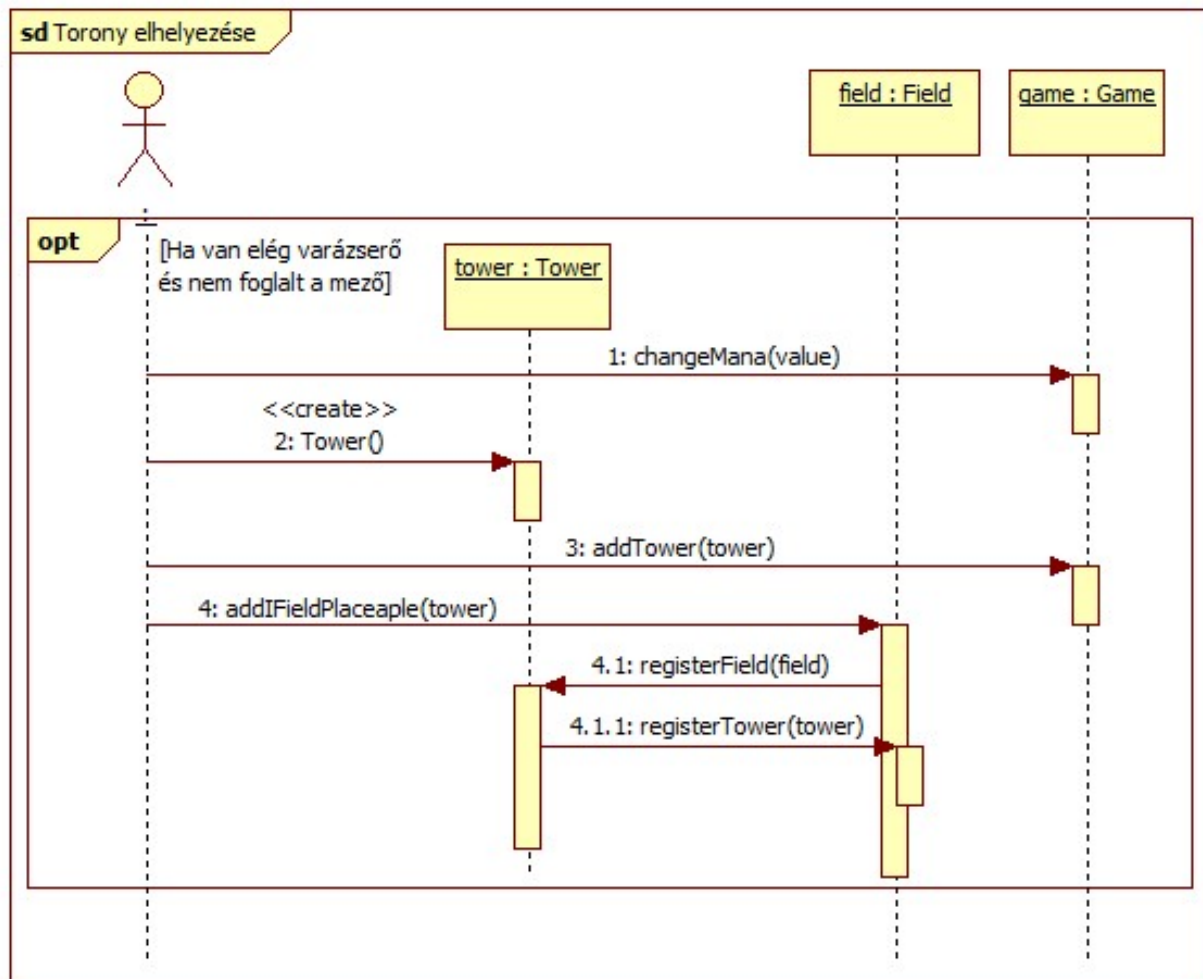
4.3. ábra. Ellenfél mozgása szekvenciadiagram

4.4.3. Torony eladása



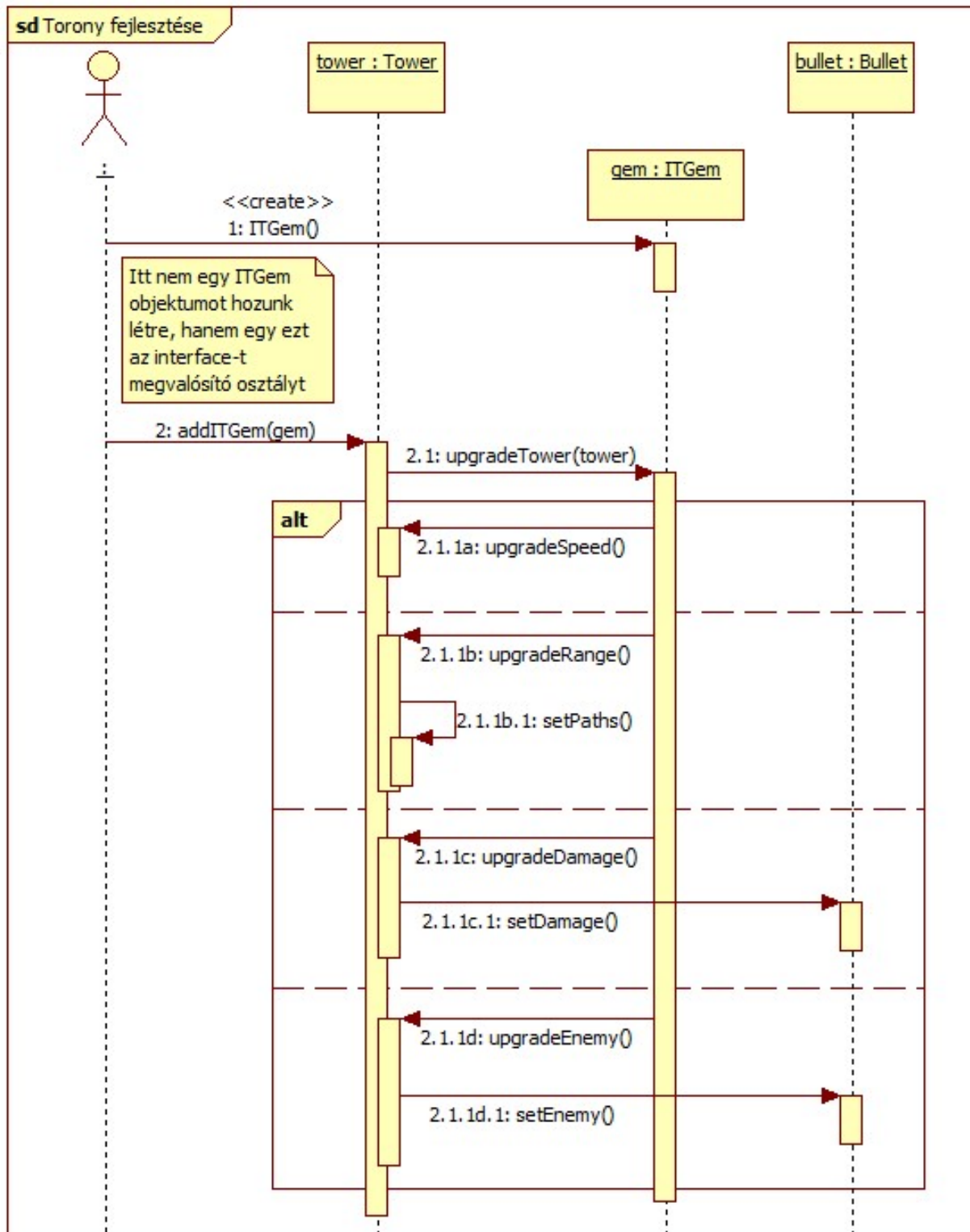
4.4. ábra. Torony eladása szekvenciadiagram

4.4.4. Torony elhelyezése



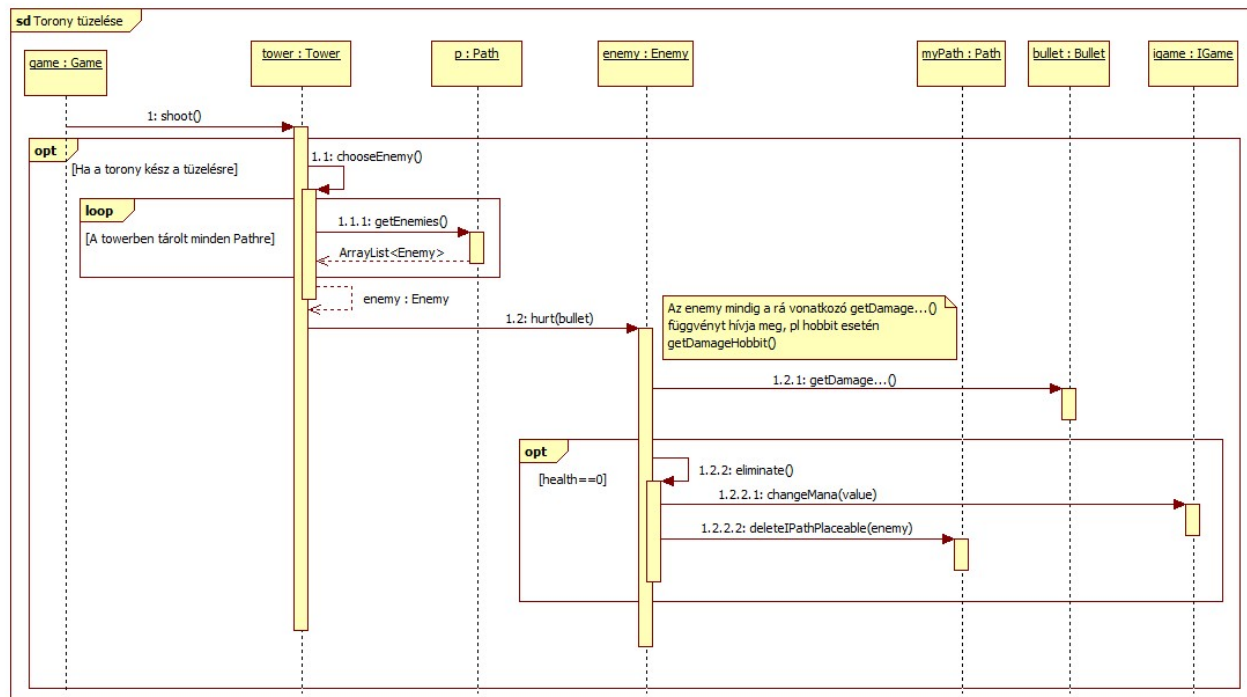
4.5. ábra. Torony elhelyezése szekvenciadiagram

4.4.5. Torony fejlesztése



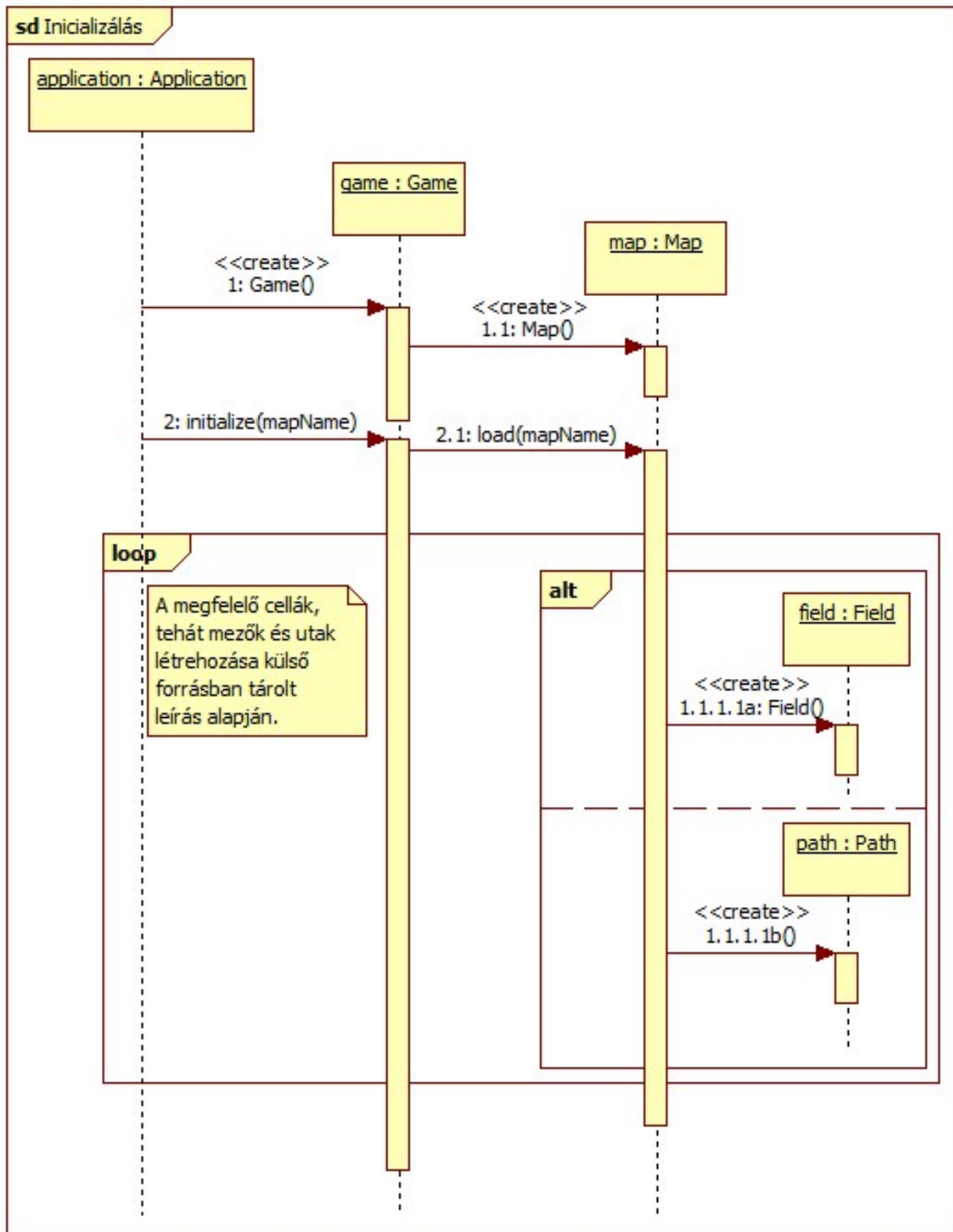
4.6. ábra. Torony fejlesztése szekvenciadiagram

4.4.6. Torony tüzelése



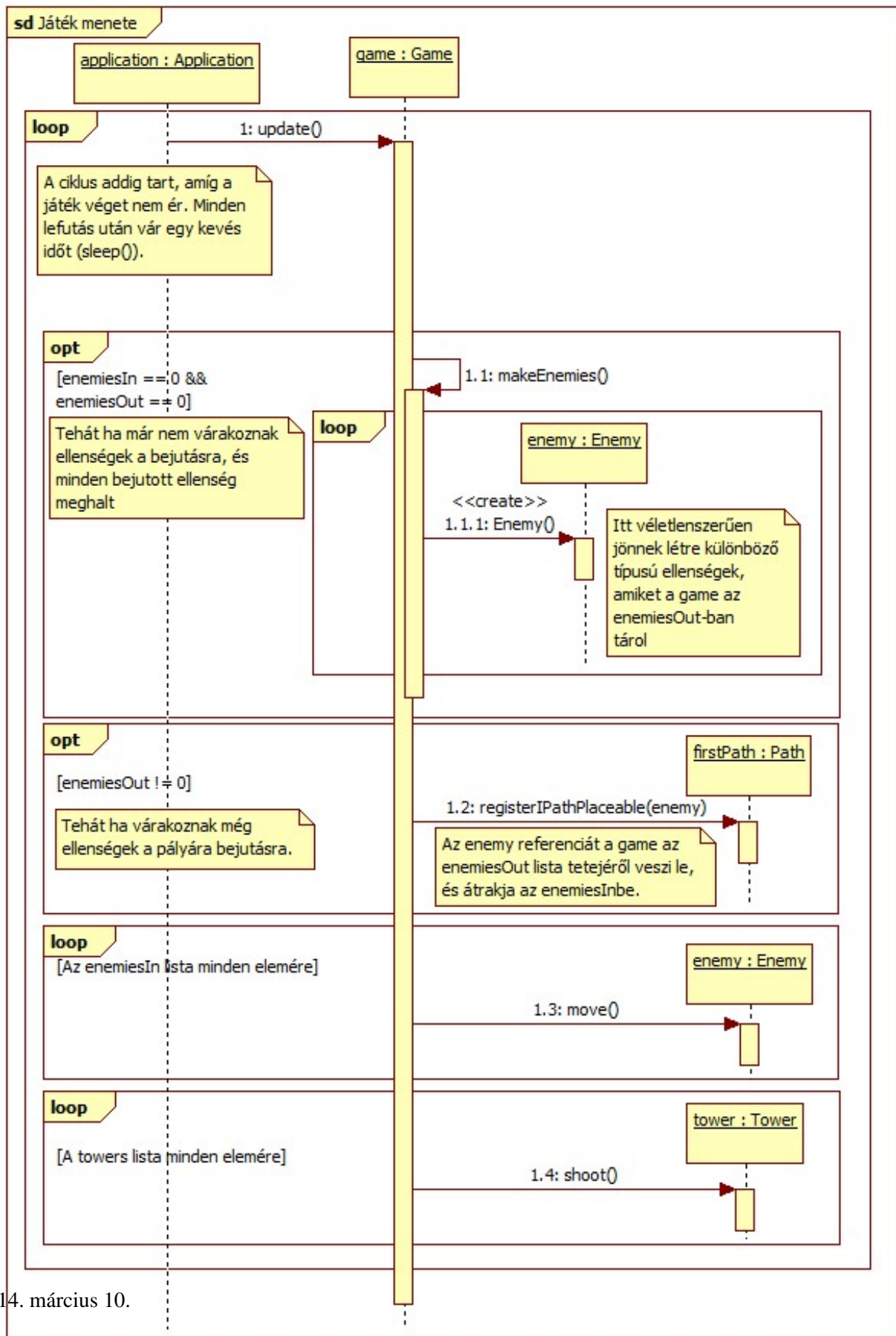
4.7. ábra. Torony tüzelése szekvenciadiagram

4.4.7. Inicializálás



4.8. ábra. Inicializálás szekvenciadiagram

4.4.8. Játék menete



4.5. State-chartok**4.6. Napló**

Kezdet	Időtartam	Résztevők	Leírás
2014.03.05 08:00	1,5 óra	Elekes Seres Ré-dey Nagy Fuksz	Konzultáció
2014.03.05 15:00	30 perc	Rédey	Gem, IOGem, ITGem és a leszármazottai/implementációik bevezetése az osztálydiagrammba, IGame interfész felvétele
2014.03.06. 11:45	45 perc	Elekes	IOGem, ITGem, Speed, Range, Damage, EnemyType, Intensity, Repair osztályleírások
2014.03.07 14:00	3 óra	Elekes Seres Ré-dey	A megírt gem-ek átbeszélése, inicializálás, ellenségek beküldése, ellenség elér a végzet hegyére, egyéb szekvenciák kidolgozása
2014.03.08. 14:00	1 óra	Elekes	Bullet, Tower, Gem-ek. Változtatások átvi-tele az osztályleírásokba.
2014.03.08. 16:00	1 óra	Rédey	Game, IGame, Obstacle, IObstacle, Enemy és leszármazottai osztályok leírások, osztálydiagramm frissítése, észlelt hibák javítása
2014.03.09. 21:00	3 óra	Seres	Szekvencia diagramok elkészítése, módosítása
2014.03.10 01:00	3 óra	Fuksz	Dokumentáció véglegesítése