

# 11. Grafikus felület specifikációja

10 – itee\_team

Konzulens:

**Budai Péter**

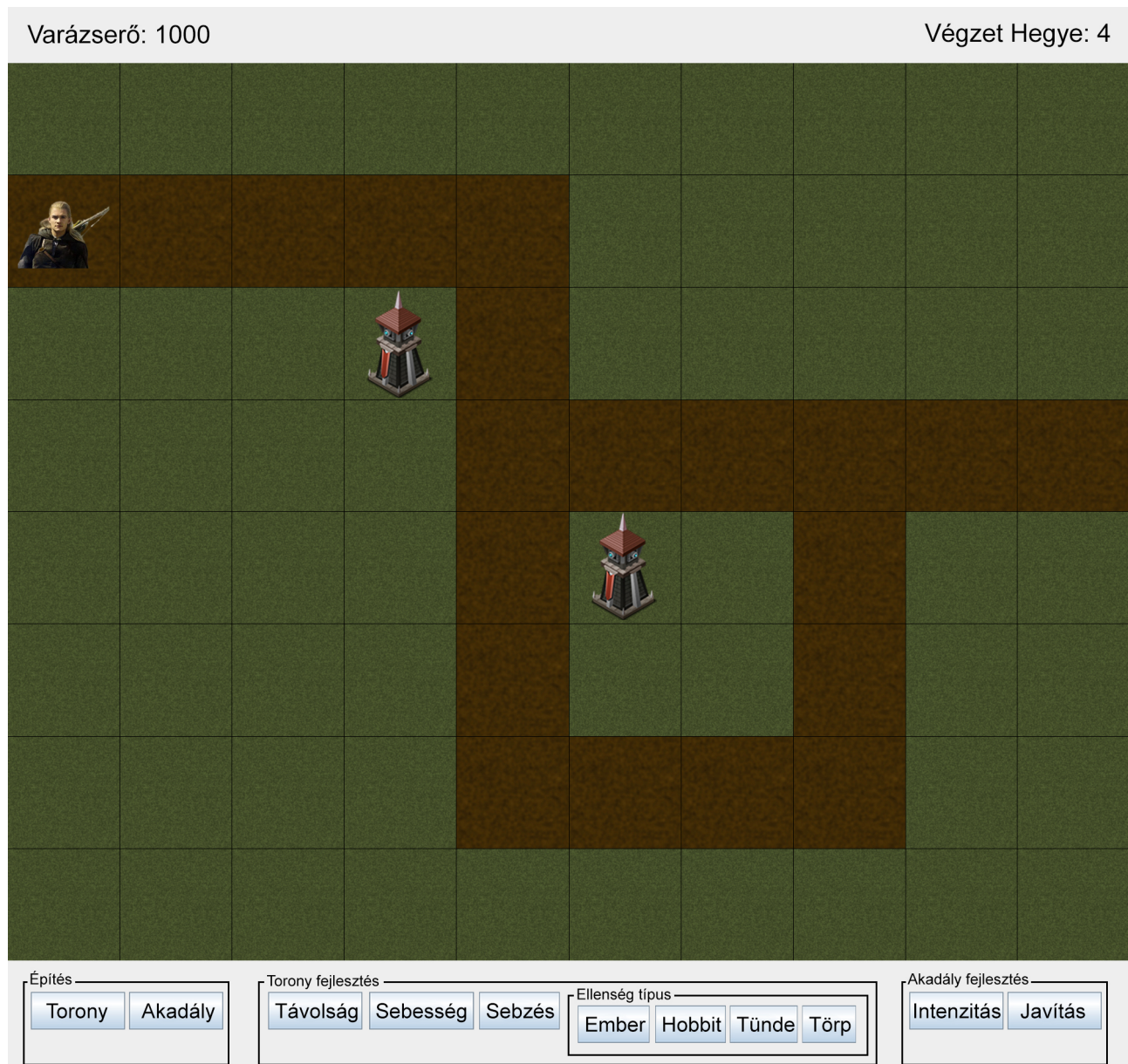
## Csapattagok:

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. április 28.

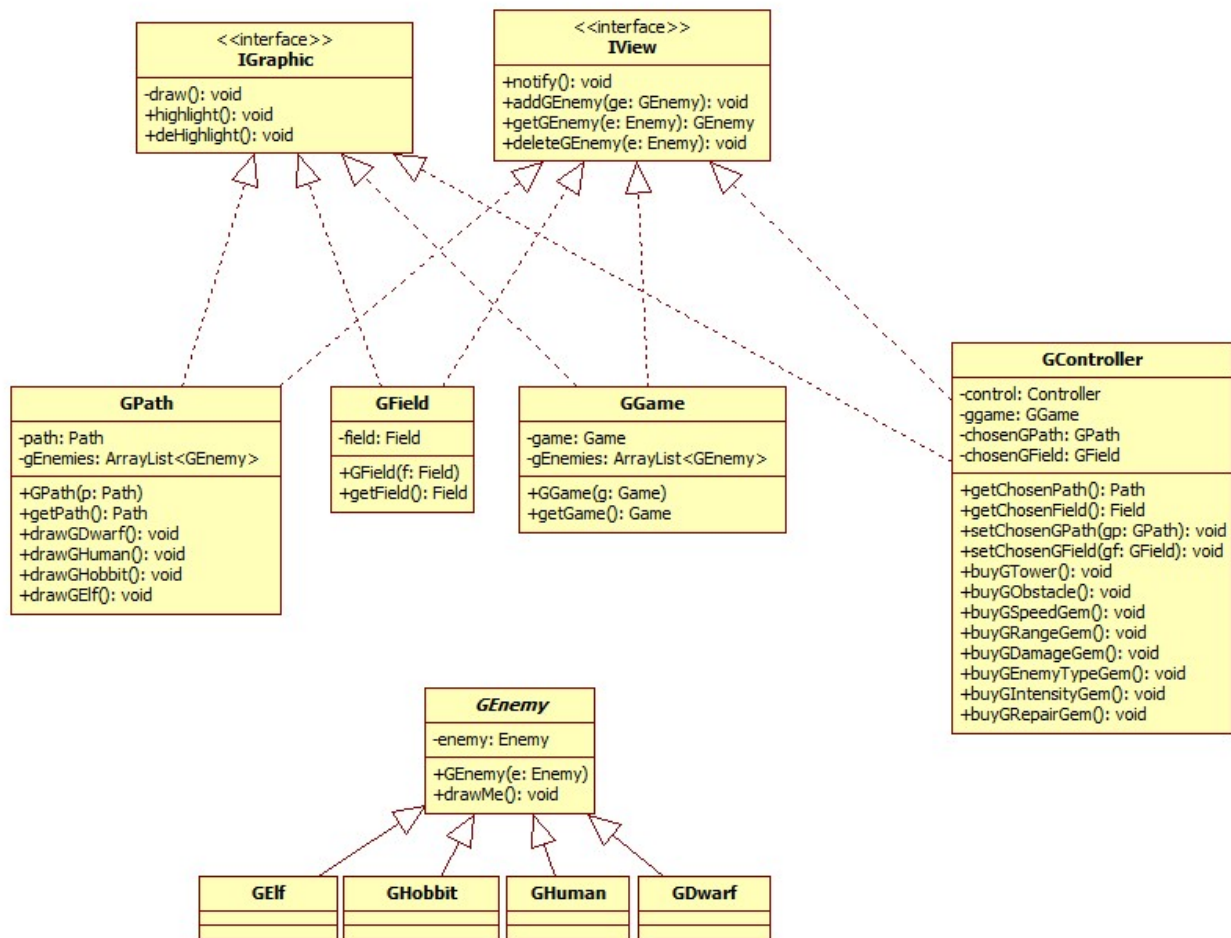
## 11. Grafikus felület specifikációja

### 11.1. A grafikus interfész



11.1. ábra. Grafikus interfész

## 11.2. A grafikus rendszer architektúrája



11.2. ábra. Grafikus rendszer struktúra diagram

### 11.2.1. A felület működési elve

A grafikus felület kialakításánál elsődleges célunk az volt, hogy a program modelljébe ne épüljön bele a megjelenítés mivolta, könnyen cserélhető legyen.

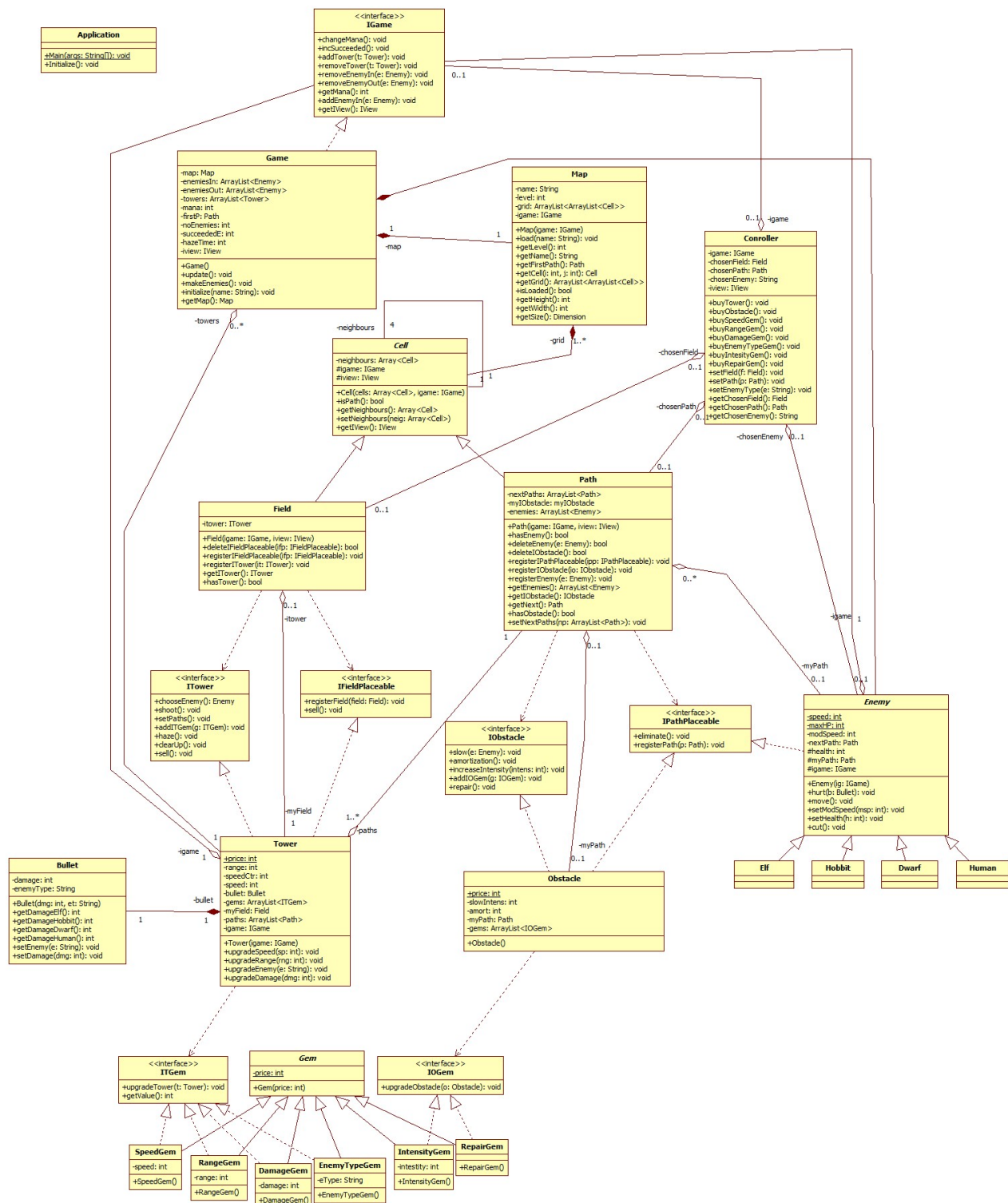
Ezért létrehoztunk a megjelenítendő elemeknek egy-egy testvér osztályt amik az objektum kirajzolásáért felelnek. Két interfésszel általánosítottuk a grafikus objektumokat.

Az IGraphic interfészt megvalósító osztályok a draw() metódussal tudják kirajzolni magukat.

Az IView interfészen keresztül tudnak a testvérosztályaikra hivatkozni, és a notify() metóduson keresztül értesítheti, hogy változott, aminek hatására kirajzolja magát.

Ez a megoldás, miszerint modell belüli objektum minden változáskor értesíti a felületet push alapú megoldás.

## 11.2.2. A felület osztály-struktúrája



11.3. ábra. Osztály diagram

### 11.3. A grafikus objektumok felsorolása

#### 11.3.1. GPath

- Felelősség  
A GPath felel a Path osztály megjelenítéséért, kirajzolásáért. Minden Path-hoz tartozik egy, és mind-egyikhez egy Path tartozik.
- Ősosztályok  
Object
- Interfészek  
IGraphic, IView
- Attribútumok  
  
**-path: Path** Referencia arra az útra, amelyikhez tartozik. Ezen keresztül kérheti le a szükséges adatokat a modellből.  
**-gEnemies: ArrayList<GEnemy>** Az path enemies-hez tartozó GEnemy-k listája.
- Metódusok  
  
**+drawGHuman/GHobbit/GElf/GDwarf: void** A különböző GEnemy-khez tartozó rajzoló függvény.  
**+getPath(): Path** Getter path-hoz  
**+GPath(p: Path)** Konstruktor.

#### 11.3.2. GField

- Felelősség  
A GField felel a Field osztály megjelenítéséért, kirajzolásáért. Minden Field-hez tartozik egy, és mind-egyikhez egy Field tartozik.
- Ősosztályok  
Object
- Interfészek  
IGraphic, IView
- Attribútumok  
  
**-field: Field** Referencia arra mezőre, amelyikhez tartozik. Ezen keresztül kérheti le a szükséges adatokat a modellből.
- Metódusok  
  
**+getField(): void** Getter fieldhez.  
**+GField(f: Field)** Konstruktor.

## 11.3.3. GGame

- Felelősség  
A GGame felel a Game osztály megjelenítéséért. Elsősorban a felső sávért, vagyis a Game azon attribútumainak megjelenéséért, amiket a játékosnak célszerű látnia: mana, bejutott ellenségek száma, pálya szintje stb.
- Ősosztályok  
Object
- Interfészek  
IGraphic, IView
- Attribútumok

**-game: Game** A Game referencia, ami ahhoz kell, hogy meg tudja jeleníteni az adatokat.

**-gEnemies** ArrayList<GEnemy>: A Game enemiesIn és enemiesOut-ban lévő Enemy-khez tartozó megjelenítők listája.

- Metódusok

**+getGame()** Game: Getter game-hez.

**+GGame(g: Game)** Konstruktor.

## 11.3.4. GEnemy és leszármazottai: GHobbit, GHuman, GElf, GDwarf

- Felelősség  
A GEnemy felel azért, hogy a GPath tudja, hogy milyen típusú Enemy-ket kell rajzolni. Ezt az enemy attribútumból tudja meg, illetve drawMe metódus különféle implementációinak segítségével üzen GEnemy a GPath-nak. Minden Enemy-hez létrehozáskor társítunk egy példányt, amit elhelyezünk a GGame-ben.
- Ősosztályok  
Object
- Interfészek  
Nincs
- Attribútumok

**-enemy: Enemy** Ez a hivatkozási pont a modellbeli Enemy-re.

- Metódusok

**+drawMe(): void** Ez a metódus gondoskodik arról, hogy a megfelelő típusú Enemy kirajzolása történjen meg a Path-on.

**+GEnemy(e: Enemy)** Konstruktor.

## 11.3.5. Controller

- Attribútumok

**-iview: IView** Interfész az értesítéshez.

- Metódusok

**+getView()** IView: Getter az iview-hoz.

#### 11.3.6. GController

- Felelősség  
A GController irányítja a grafikus elemeket, végzi a felhasználói interakciók érvényre jutásának a felületre vonatkozó részeit, illetve kezdeményezi a modell megváltoztatását.
- Ősosztályok  
Object
- Interfészek  
IGraphic, IView
- Attribútumok

**-control: Controller** Ez a hivatkozási pont a modellhez.

**-chosenGPath: Path** A kijelölt út, pl akadály vétele.

**-chosenGField: Field** A kijelölt mező, pl torony vétele.

**-ggame: GGame** Ez a hivatkozási pont a grafika másik részéhez.

- Metódusok

**+set chosenGField/GPath(f/p Field/Path)** void: GField/GPath setter.

**+get chosenGField/GPath(): void** GField/GPath getter.

**+buyGSpeed/Range/Damage/EnemyType/Intesity/RepairGem(): void** Gem-ek vétele.

**+buyTower/Obstacle()** Tower/Obstacle vétele.

#### 11.3.7. IView

- Felelősség  
Az IView interfész teremt kapcsolatot a modellből a grafikus felület felé. Segítségével lehet értesíteni a felületet a modellbeli változásokról. Ehhez különböző értesítő metódusokat tartalmaz.
- Ősosztályok  
Nincs
- Interfészek  
Nincs
- Metódusok

**+notify()** void: Alap értesítő metódus.

**+notify(ge: GEnemy)** void: Értesítő metódus Enemy és GEnemyk kezeléséhez. Pl Enemy létrehozásakor vele azonos típusú GEnemy-t hozunk létre, amit odaadunk a megfelelő megjelenítőnek.

## 11.3.8. IGraphic

- Felelősség  
Az IGraphic a rajzoló interfész. Különböző rajzolásokat tesz lehetővé.
- Ősosztályok  
Nincs
- Interfészek  
Nincs
- Metódusok

**-draw()** void: Alap rajzoló metódus.

**+highlight()** void: Kijelölő metódus, pl mezőre kattintáshoz.

**+deHighlight()** void: Highlight inverze.

## 11.3.9. Cell

- Attribútumok

**#iview: IView** Interfész a grafika értesítéséhez.

**#igame: IGame** A Path és Field igrace-je felkerült ide, protected lett.

- Metódusok

**+getIView()** IView: Getter az iview-hoz.

## 11.3.10. Game és IGame

- Attribútumok

**-iview: IView** Game-ben interfész az értesítéshez.

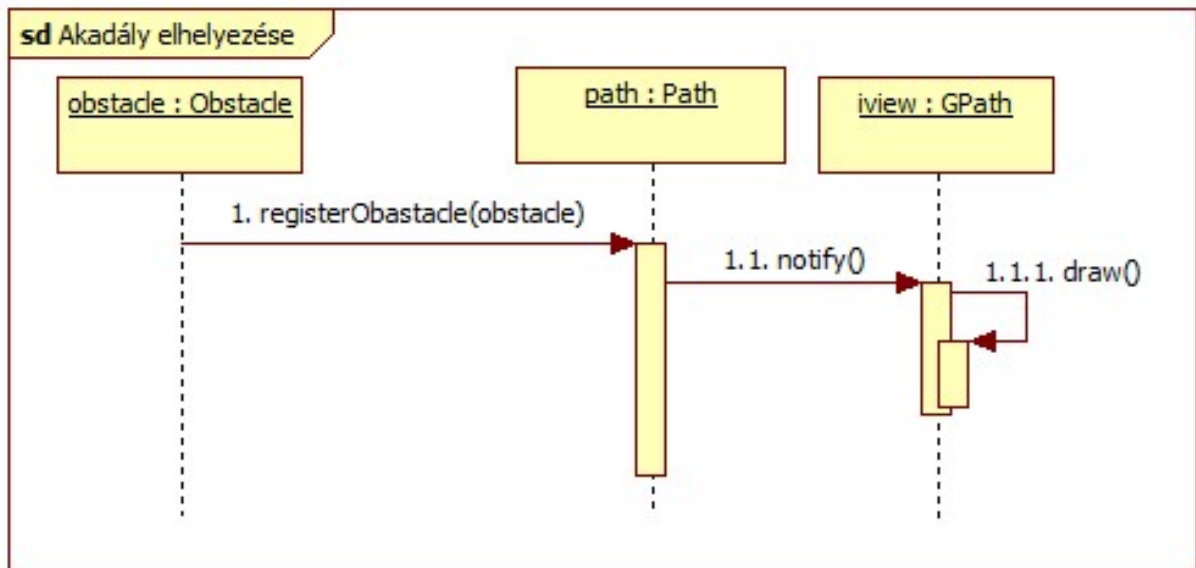
- Metódusok

**+getIView(): IView** IGame-be getter, hogy mások is elérjék, akik a Gamet elérik az interfészén keresztül.



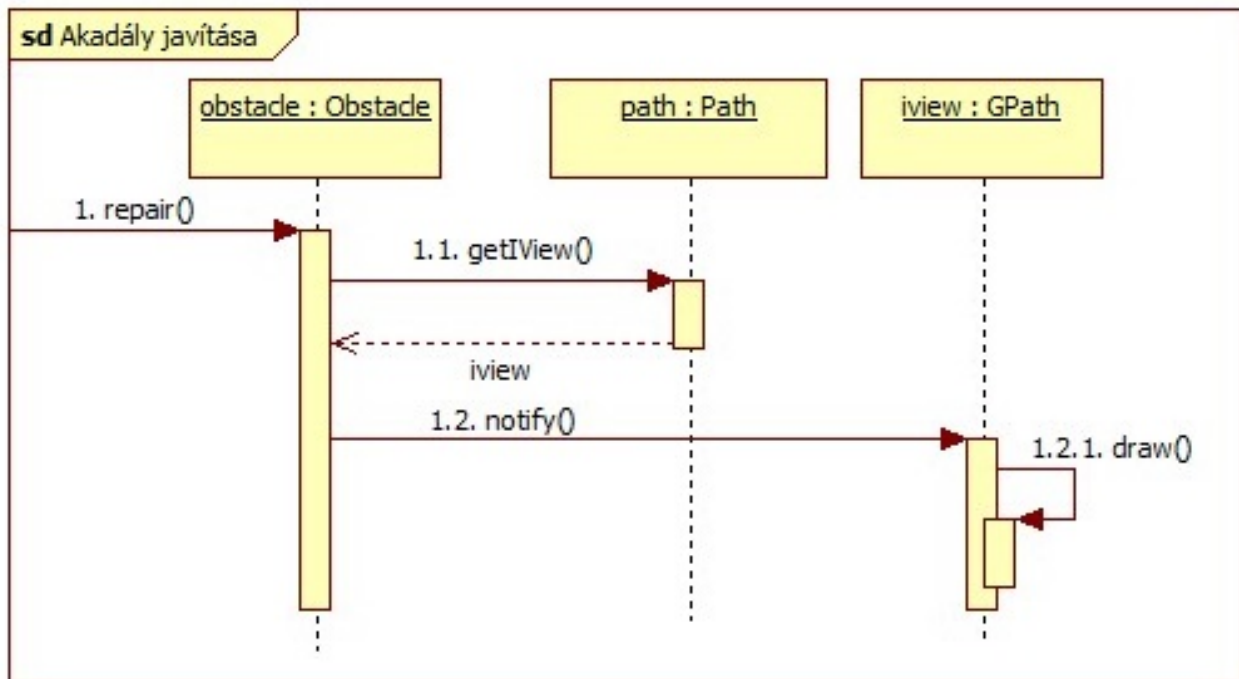
## 11.4. Kapcsolat az alkalmazói rendszerrel

### 11.4.1. Akadály elhelyezése



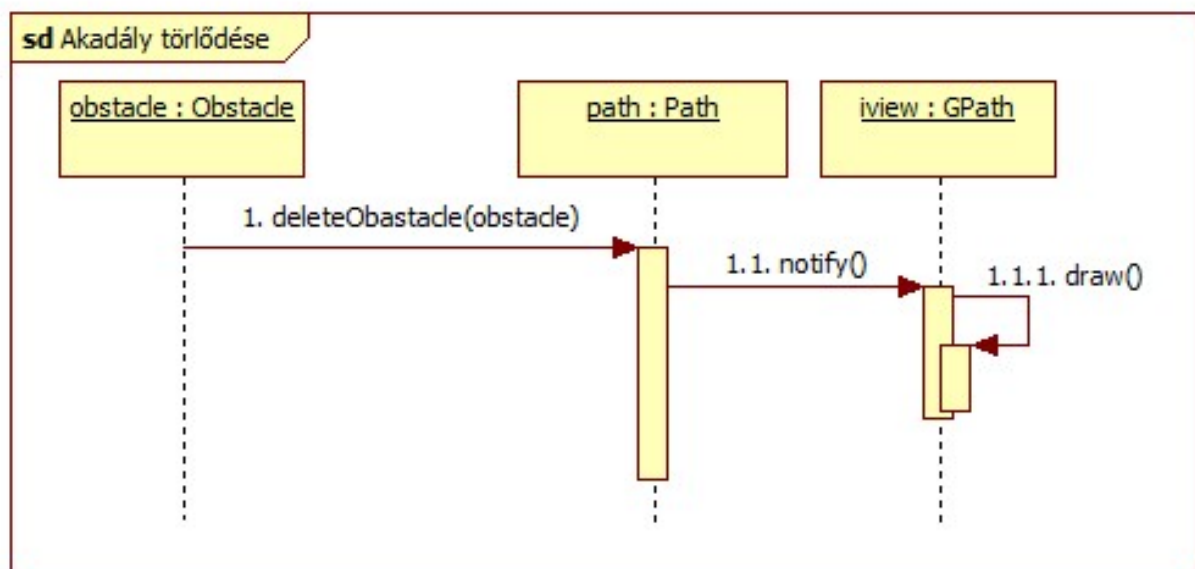
11.4. ábra. Akadály elhelyezése szekvenciadiagram

## 11.4.2. Akadály javítása



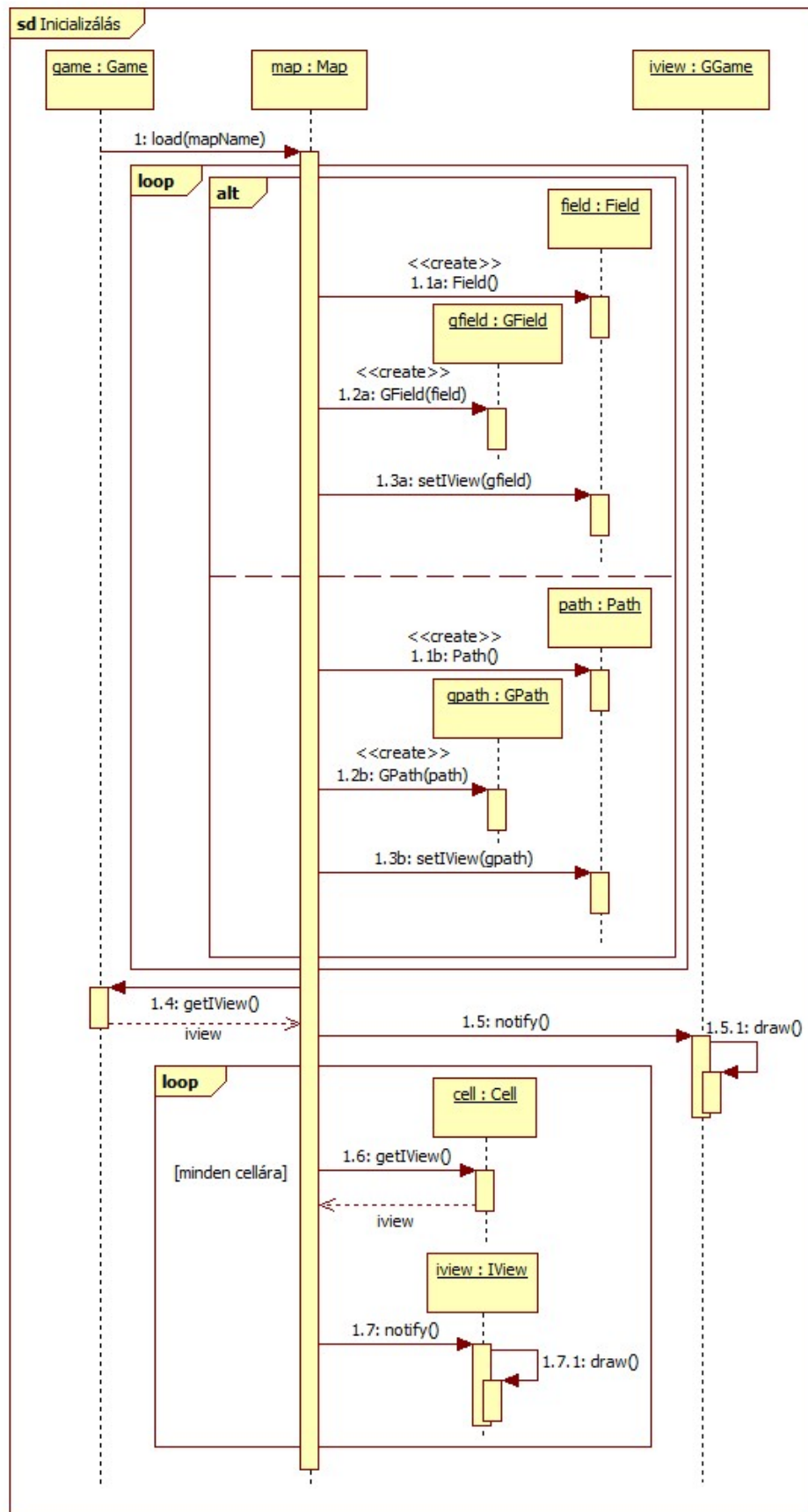
11.5. ábra. Akadály javítása szekvenciadiagram

## 11.4.3. Akadály törlődése



11.6. ábra. Akadály törlődése szekvenciadiagram

## 11.4.4. Betöltés

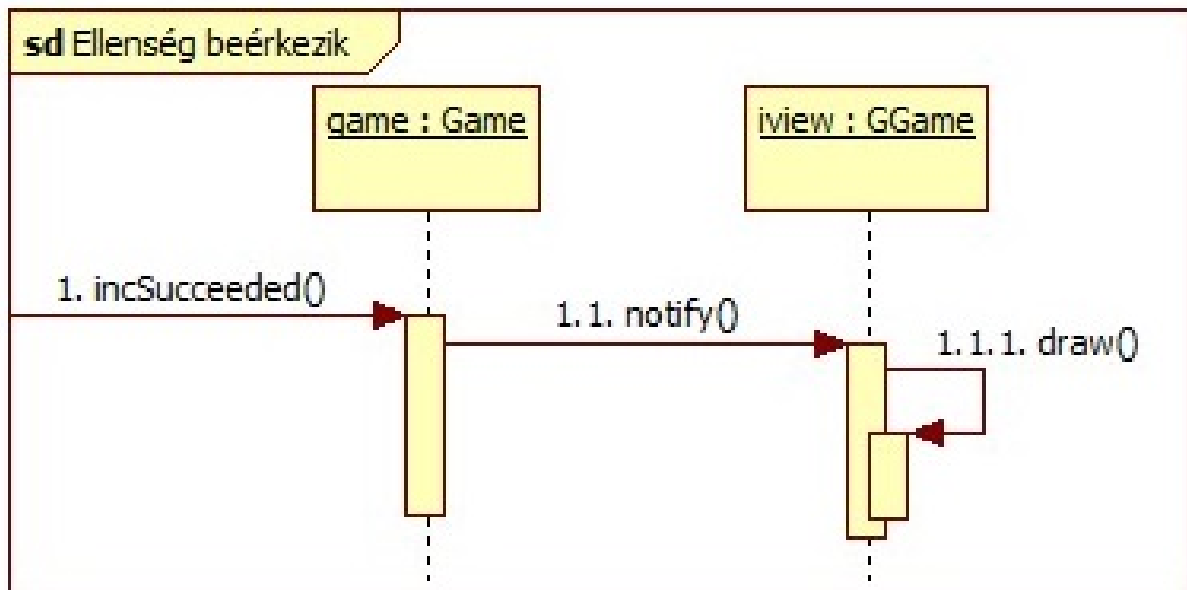


2014. április 28.

11

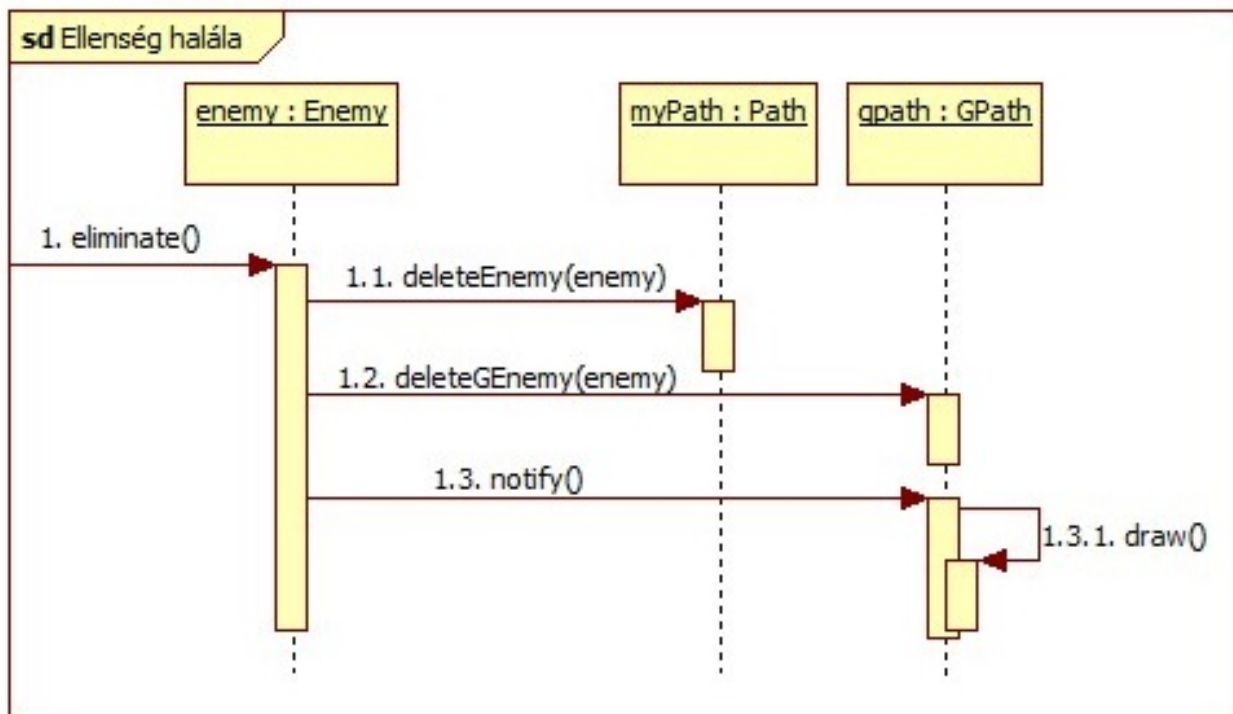
11.7. ábra. Betöltés szekvenciadiagram

## 11.4.5. Ellenség beérkezik



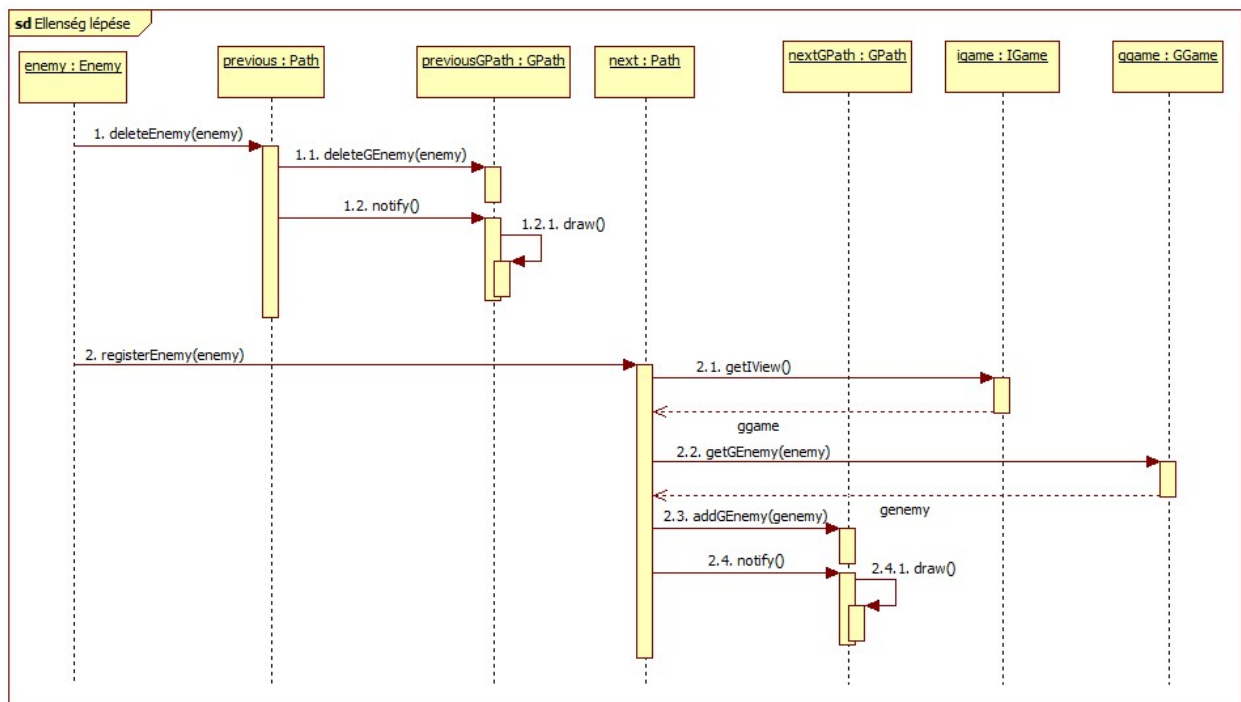
11.8. ábra. Ellenség beérkezik szekvenciadiagram

## 11.4.6. Ellenség halála



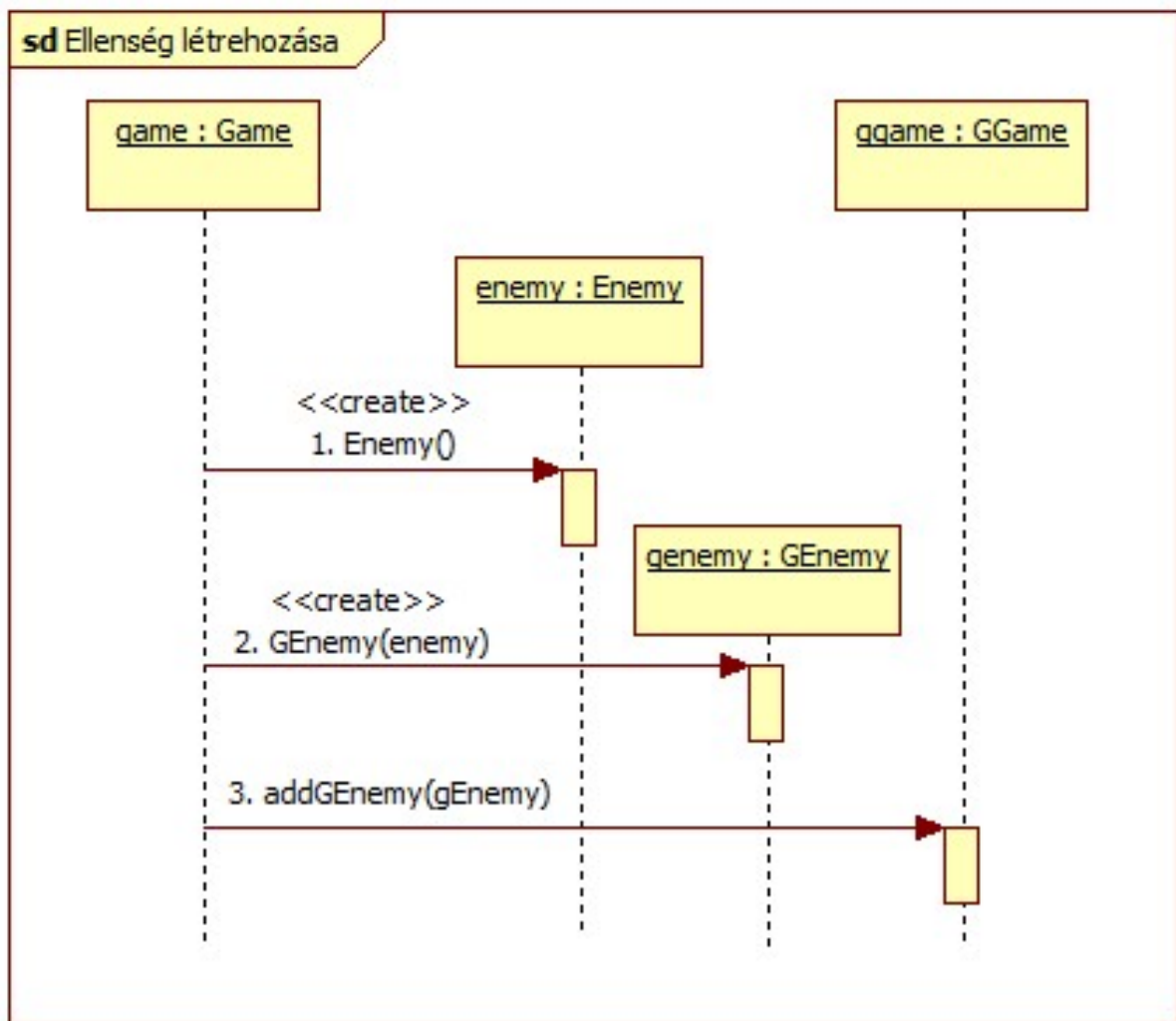
11.9. ábra. Ellenség halála szekvenciadiagram

## 11.4.7. Ellenség lépése



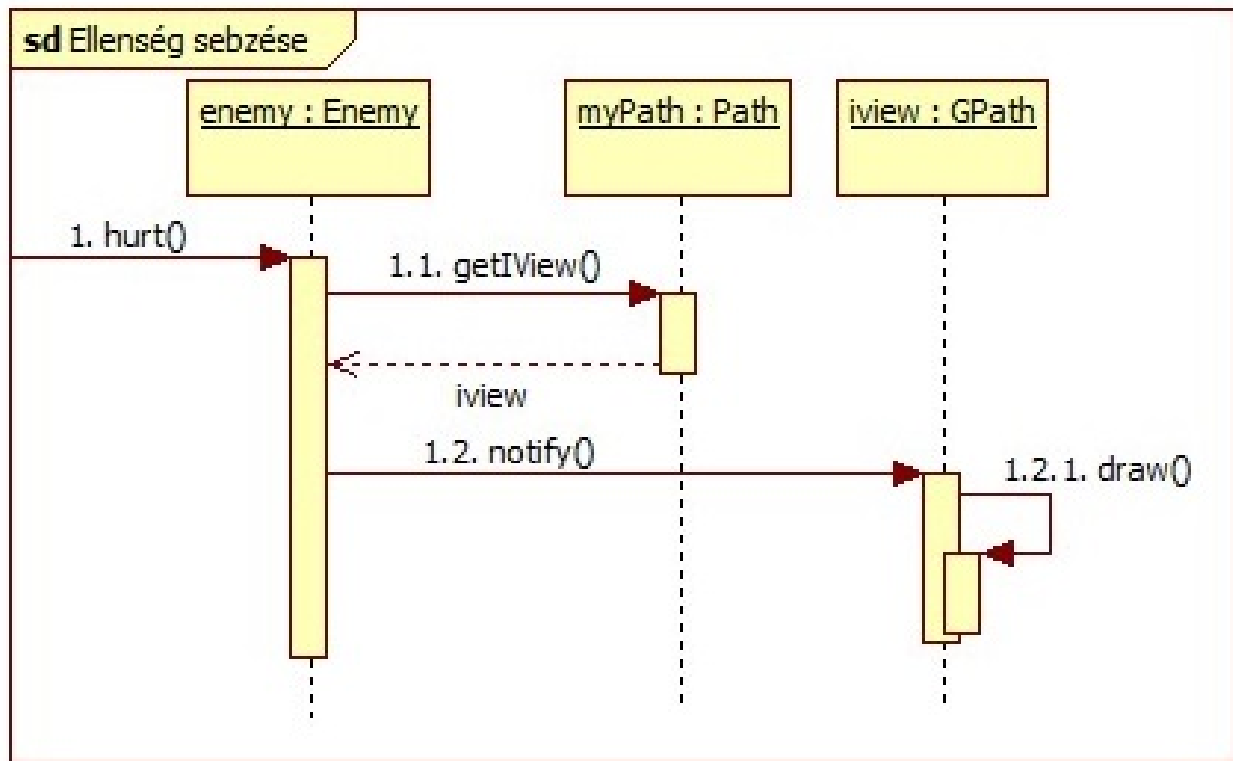
11.10. ábra. Ellenség lépése szekvenciadiagram

## 11.4.8. Ellenség létrehozása



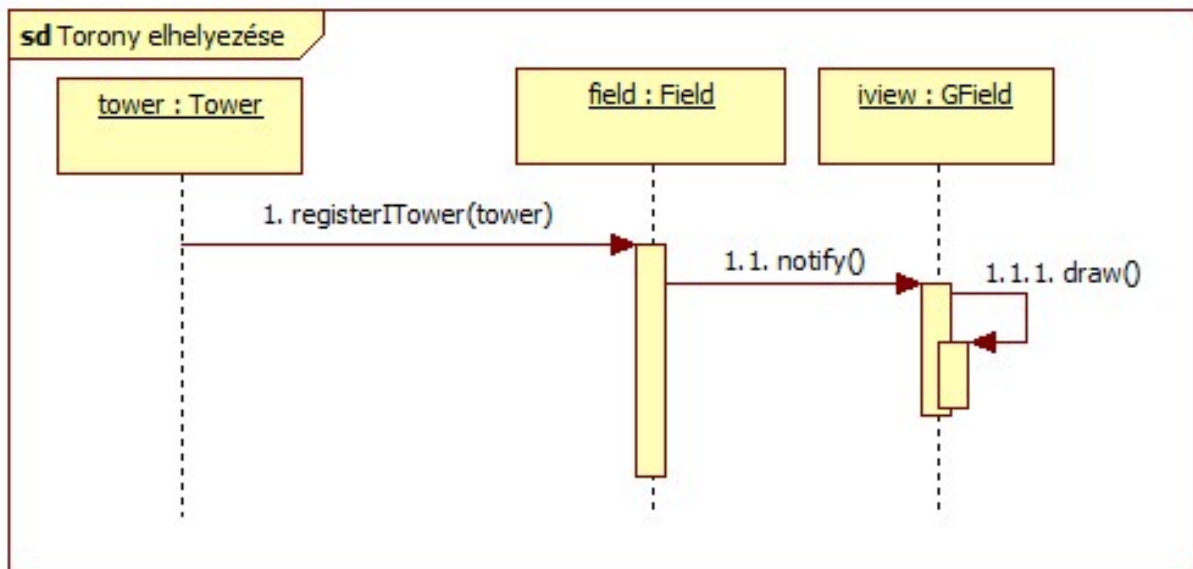
11.11. ábra. Ellenség létrehozása szekvenciadiagram

## 11.4.9. Ellenség sebzése



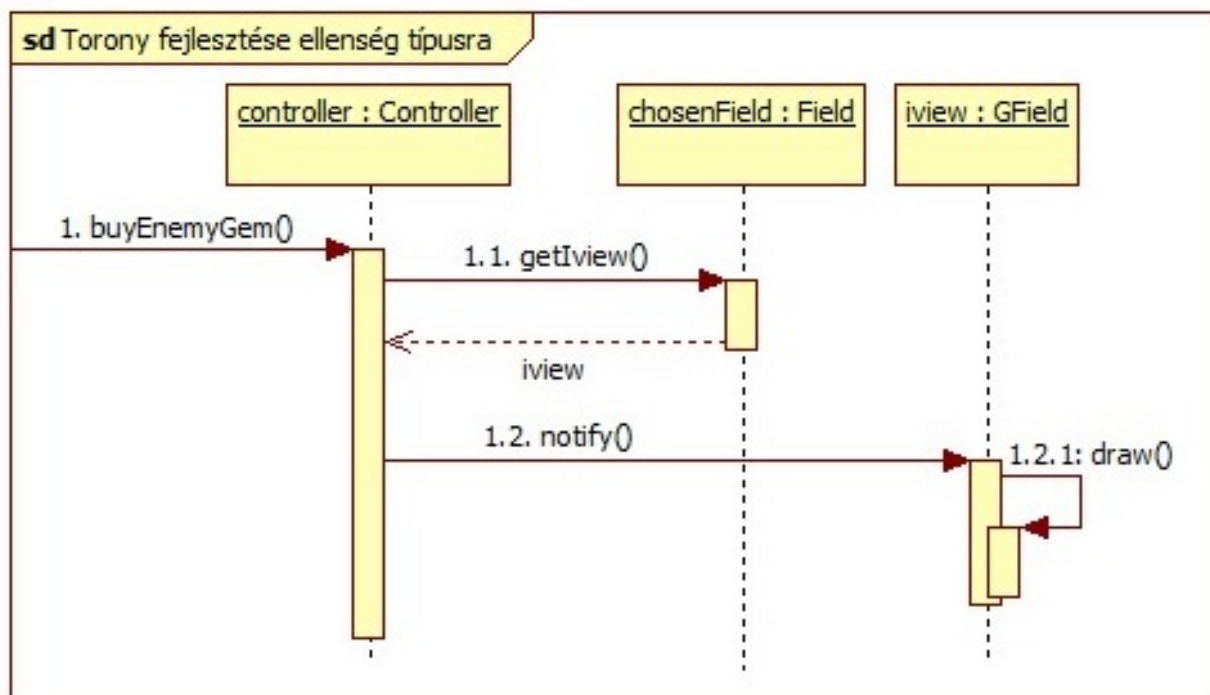
11.12. ábra. Ellenség sebzése szekvenciadiagram

## 11.4.10. Torony elhelyezése



11.13. ábra. Torony elhelyezése szekvenciadiagram

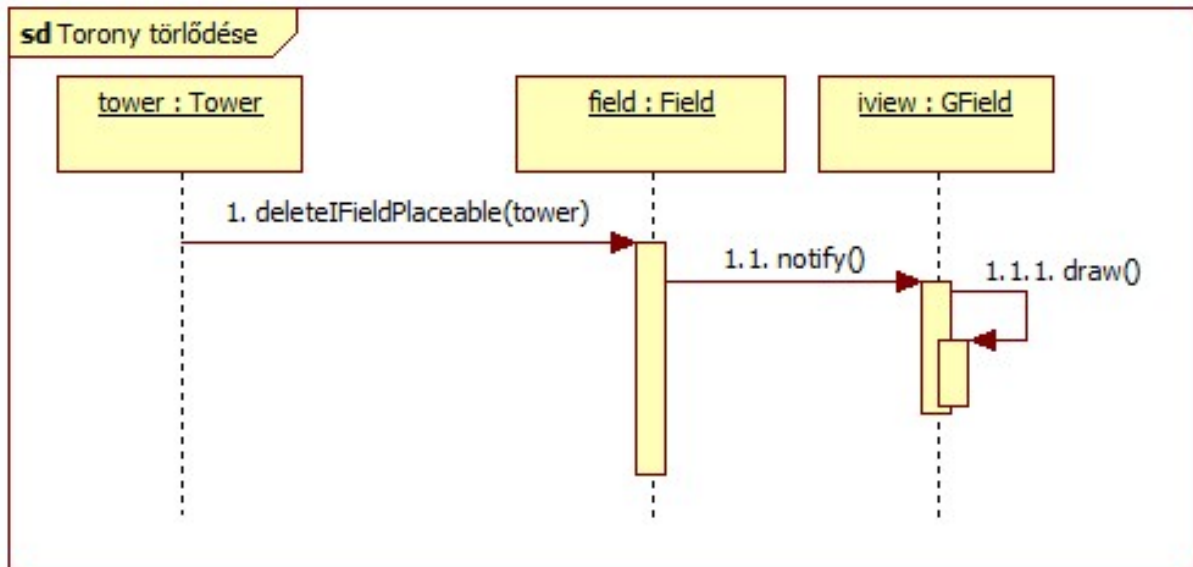
## 11.4.11. Torony fejlesztése ellenség típusra



11.14. ábra. Torony fejlesztése ellenség típusra szekvenciadiagram

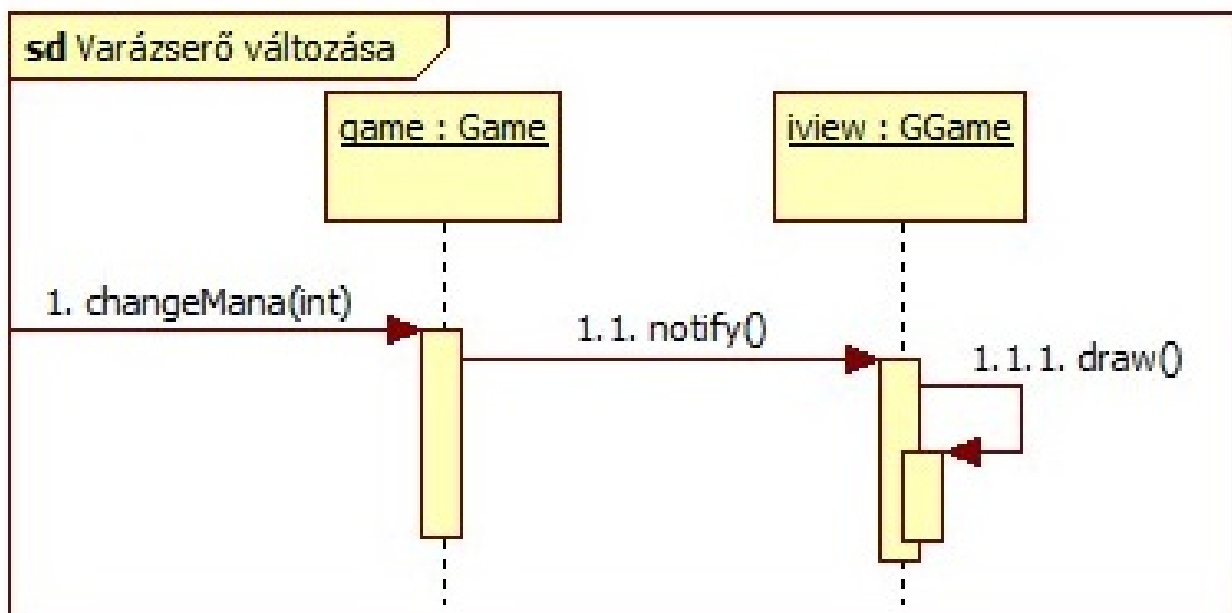


## 11.4.12. Torony törlődése



11.15. ábra. Torony törlődése szekvenciadiagram

## 11.4.13. Varázserő változása



11.16. ábra. Varázserő változása szekvenciadiagram

## 11.5. Napló

2014. április 28.

<b>Kezdet</b>	<b>Időtartam</b>	<b>Résztevők</b>	<b>Leírás</b>
2014.04.23. 08:15	1,5 óra	<b>Elekes Fuksz Nagy Rédey Seres</b>	Konzultáció
2014.04.26. 17:00	3 óra	<b>Seres</b>	Grafikus vázlat készítése a játékról
2014.04.26 11:00	4 óra	<b>Rédey</b>	Grafikus osztályok, MVC kapcsolatok megal- kotása, 11.2.2
2014.04.27 14:00	2 óra	<b>Rédey</b>	Grafikus osztályok finomítása, 11.3 objektum katalógus
2014.04.27. 22:00	4 óra	<b>Seres</b>	Grafikus működés szekvencia diagramjai
2014.04.28. 03:00	2 óra	<b>Fuksz</b>	Dokumentáció véglegesítése