

8. Részletes tervek

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. május 11.

8. Részletes tervek

8.1. Osztályok és metódusok tervei

8.1.1. Bullet

- Felelősség
A tornyok egy Bullet-et tárolnak, amit minden lövésnél átadnak a lövő függvénynek. Ennek a lövedéknek a feladata, hogy az ellenségnek megmondja mennyit sebez rajta.
 - Ősosztályok
Object
 - Interfészek
Nincs
 - Attribútumok
- damage: int** Az alapsebzés értéke. Ezt adják vissza a getDamage függvények, ha nincs fejlesztve az adott típusú Enemy-re a torony.
- enemyType: String** A torony itt tárolja, hogy melyik ellenség típusra erősebb a sebzése.
- Metódusok
- +Bullet(dmg: int, et: String)** Konstruktor
- +getHobbitDamage(): int** Ha hobbitot sebez, ezzel a függvénnyel kérdezi le a sebzés értékét.
- +getHumanDamage(): int** Ha embert sebez, ezzel a függvénnyel kérdezi le a sebzés értékét.
- +getDwarfDamage(): int** Ha törpöt sebez, ezzel a függvénnyel kérdezi le a sebzés értékét.
- +getElfDamage(): int** Ha tündét sebez, ezzel a függvénnyel kérdezi le a sebzés értékét.
- +setEnemy(e: String): void** Beállítja az ellenséget akire specializált (enemyType).
- +setDamage(dmg: int): void** Beállítja a lövedék sebzését (damage).

8.1.2. Enemy

- Felelősség
Az Enemy osztály 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 helyét értesítse a Game osztályt. Tudja, hogy milyen sebességgel haladt eredetileg, és milyen sebességgel halad most. Ez egy absztrakt ősosztály, ami összefogja a 4 ellenségtípust (Hobbit, Elf, Dwarf, Human).
 - Ősosztályok
Object
 - Interfészek
IPathPlaceable
 - Attribútumok
- speed: int** A két lépés között eltelt idő. static, final

-maxHp: int A maximális életerő értéke. static, final

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

-nextPath: Path A soron következő path címe.

#health: int Életerejét tárolja ebben. Hurt függvényben csökkenti.

#myPath: Path Az a mező, ahol tartózkodik.

#igame: IGame Ezen keresztül tudja módosítani a manát, amikor meghal, illetve ha elér a végzet hegyére módosítani a számlálót (Game.succeededE), hogy nőjön egyel.

- Metódusok

+hurt(b: Bullet): void Sebződést megvalósító metódus, abstract, minden Enemy-típusban máshogy implementálódik.

+move(): void mozog, a következő path-ra lép, cellát vált.

+Enemy(ig: IGame) konstruktor.

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

+setHealt(h: int): void A health változó settere.

+void cut(): void A kettévágásos lövésért felelős metódus. Létrehoz egy új, azonos típusú ellenséget (abstract), feleakkora életerővel. Beteszi az igame enemiesIn tömbjébe. Saját életerjét is felezi.

8.1.3. 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(b: Bullet): void Sebződik, a kapott Bullet alapján getDamage... metódus által visszaadott értékkel csökkenti az életerejét. Ha elfogyott az életerejé törli magát a pathról és az igame.enemiesIn-ből Ha 0-t kap, cut-ot hív.

8.1.4. Game

- Felelősség

A Game osztály 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.

- Őssosztályok

Object

- Interfészek
IGame
- Attribútumok

-map: Map játék térképe

-enemiesOut: List<Enemy> A pályára még be nem lépett ellenségek. Ezt töltjük meg újabb ellenség hullám generálásakor, majd fokozatosan kiürítjük és egyesével átírjuk a tartalmat az enemiesIn-be.

-enemiesIn: List<Enemy> A pályára már belépett ellenségek. Minden update hívásnál módosul(hat) a tartalom, pl. új Enemy lép a pályára vagy meghal egy-pár.

-towers: List<Tower> A tornyok, amik a pályán vannak. Update-enként végigmegyünk rajtuk és meghívjuk a shoot.

-mana: int A maradék varázserő.

-firstP: Path Az út kezdő cellája. Ennek segítségével indítjuk el az Enemy-ket a pályán, megadva nekik a címet.

-noEnemies: int Kezdeti hullámérték, amely folyamatosan nő, azt mutatja meg, hogy következő körben hány ellenséget kell létrehozni és beküldeni a pályára. Ebből megtudhatjuk azt is, hogyha a játékos teljesítette a pályát.

-succeededE: int A végzet hegyét elért enemy-k száma. Ha egy Enemy célba ér, az IGame-n keresztül tudja növelni. Ha elér egy maximum értéket, vége a játéknak.

-hazeTime: int Az az idő, ameddig a köd tart. Ha lejár a köd, nullázzuk, ameddig a ködnek tartania kell, növeljük.

- Metódusok

+update(): void Frissíti a modellt, grafikát. X időközönként folyamatosan hívjuk. Belül az enemiesIn-en végighaladva minden elemen move-ot hívunk, towers-en ugyanígy shoot-ot, illetve ha kell, enemiesOut-ból betesszük a következő elemet. Random módon a towers elemein haze-t is hív.

+initialize(String name): void Kiinduló állapot felvétele.

+Game() Konstruktor.

+makeEnemies(): void Létrehoz noEnemies db ellenséget, random típusút, ezeket beteszi az enemiesOut-ba.

+getMap(): Map map getter.

8.1.5. Controller

- Felelősség
A felhasználó és a játék közötti kommunikációnak véghezvitele a felelőssége. Kristályok, tornyok, akadályok vásárlását lehet rajta keresztül megcsinálni. Ellenőriznie kell, hogy van-e a játékosnak elég varázsereje a tranzakcióhoz.
- Ősosztályok
Object
- Interfészek
Nincs
- Attribútumok

- igame: IGame** Szükséges függvények elérésére szolgál, hogy tudja módosítani a Game-et.
- chosenField: Field** Ha Field-et választ ki, ebben a változóban tárolja.
- chosenPath: Path** Ha Path-t választ ki, ebben a változóban tárolja.
- chosenEnemy: String** A kiválasztott EnemyType.

- Metódusok

- +buyTower(): void** Torony vételére szolgál. Létrehoz egy tornyot, beregisztrálja a chosenField-re.
- +buyObstacle(): void** Lásd buyTower Obstacle-el és chosenPath-el.
- +buySpeed/Range/Damage/EnemyType/Intensiyty/RepairGem(): void** Kristályok vásárlása. A Tower/Obstacle-t amire vettük a chosenField/Path-en érjük el.
- +setField(f: Field): void** A chosenField-et állítja.
- +setPath(p: Path): void** A chosenPath-t állítja.
- +setEnemy(e: String): void** A chosenEnemy-t állítja.

8.1.6. 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

- +changeMana(): void** A manát megváltoztató metódus.
- +incSucceeded(): void** A succeededE értékét megváltoztató metódus.
- +addTower(t: Tower): void** A hozzáad egy tornyot a towers listához.
- +removeEnemyIn(e: Enemy): void** Az ellenség törlése enemiesIn-ből.
- +removeEnemyOut(e: Enemy): void** Az ellenség törlése enemiesOut-ból.
- +removeTower(t: Tower): void** A torony törlése towers-ből.
- +addEnemyIn(e: Enemy): void** Ellenség hozzáadása enemiesIn-hez.
- +getMana(): int** A mana lekérdezése.

8.1.7. Gem

- Felelősség

A Gem osztály felel a torony fejlesztéséért. Ha a játékos vesz a toronyra/akadályra valamilyen kristályt, akkor jön létre, megkapja a torony, és beépíti magába.

- Ősosztályok

Object

- Interfészek

Nincs

- Attribútumok

+price: int Az ár, amennyi varázserőbe kerül. static, final

- Metódusok

+Gem(price: int) Konstruktor.

8.1.8. IObstacle

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

+slow(intesity: int, p: Path): void Szól p-nek, hogy lassítsa le az ellenséget intensity-vel.

+amortization(): void Amortizál, csökkenti amort értékét.

+increaseIntesity(intens: int): void Megnöveli az intesity-t intens-el.

+addIOGem(IOGem: iog): void Az iog kristályt hozzáadja az akadályhoz.

+repair(): void Megjavítja az akadályt, amort értékét maximalizálja.

8.1.9. Obstacle

- Felelősség
Az Obstacle osztály 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.
- Ősosztályok
Nincs
- Interfészek
IObsacle, IPathPlaceable
- Attribútumok

-slowIntens: int A lassítás mértéke.

-myPath: Path A path, amin rajta van.

-amort: int Az elhasználódottság mértéke.

+price: int Az ára. static, final

-gems: ArrayList<IOGem> A megvett kristályok listája.

- Metódusok

+Obstacle() Konstruktor.

8.1.10. ITower

- Felelősség
A torony funkciói vannak benne. Egy olyan interfész, amin keresztül a tornyok kezelhetők.
- Metódusok
 - +setPaths(): void** A myField-ből kiindulva a range-el lefedett területen felkeresi, és beregisztrálja a paths listába a path cellákat.
 - +shoot(): void** A torony akkor lő, ha letelt az újratöltési idő, ekkor megnézi, hogy lőtávon belül van-e ellenség, és ha van meghívja a hurt függvényét, átadva paraméterként a bullet-et.
 - +addITGem(gem: ITGem): void** Paraméterként megkapja a kiválasztott kristályt, bulletet frissíti, ha kell, illetve a torony adott attribútumait.
 - +chooseEnemy(): Enemy** 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.
 - +haze(): void** Ködöt generál a tornyon mod range beállításával.
 - +clearUp(): void** Beállítja a modRange-et normál, köd nélküli értékre range alapján.
 - +getEnemyType(): String** Lekérdezi és visszaadja a Bulletben tárolt enemyType-ot.
 - sell(): void** Torony eladása.

8.1.11. Tower

- Felelősség
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 tornyokat, illetve el is adhatja őket. Ez a legfontosabb eszköz amivel a játékos meg tudja akadályozni az ellenségek célba jutását.
- Ősosztályok
Nincs
- Interfészek
ITower, IFieldPlaceable
- Attribútumok
 - +price: int** Az ára varázserőben. static, final
 - range: int** Lőtáv, hatókör.
 - modRange: int** Módosított hatókör.
 - speedCtr: int** A torony belső idő mérője. Ezt vizsgálja minden lövés előtt, hogy eltelt e elég idő.
 - speed: int** két lövés között eltelt minimális idő.
 - bullet: Bullet** A torony tárol egy lövedéket, mindig ezt lövi ki.
 - gems: ArrayList<ITGem>** A megvásárolt kristályokat tárolja.
 - paths: ArrayList<Path>** Hatósugarba eső út cellák.
 - myField: Field** Mező, amin áll.
 - igame: IGame** Egy interfész a játék logikára, amivel a bejutott ellenségek számát, és a varázserőt is lehet állítani.

- Metódusok

+Tower(igame: IGame) Konstruktor.

+upgradeSpeed(sp: int): void Fejleszti a lövési sebességét.

+upgradeRange(rng: int): void Fejleszti a lőtávot.

+upgradeEnemy(e: String): void Egy ellenségtípusra növeli a sebzést.

+upgradeDamage(dmg: int): void Növeli a sebzést.

8.1.12. Map

- Felelősség

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

- Interfészek
Nincs

- Attribútumok

-name: String A pálya neve, egyben az azonosítója.

-level: int A pálya szintje.

-grid: Array<Array<Cell> A cellákat tartalmazó 2 dimenziós tömb.

-igame: IGame Interfész Game eléréséhez.

- Metódusok

+Map(igame: IGame) Az osztály konstruktora.

+load(name: String) Megnyitja a paraméterként kapott nevű fájlt, és abból betölti a pálya celláinak tulajdonságait, felépíti a pályát.

+getLevel(): int Visszaadja a pálya szintjét.

+getName(): String Visszaadja a pálya nevét.

+getFirstPath(): Path Visszaadja a pálya belépési pontjának referenciáját.

+getCell(i: int, j: int): Cell Adott cella lekérése.

+getGrid(): ArrayList<ArrayList<Cell> : Játéktér lekérése.

+isLoading(): bool : Megadja, hogy be van-e töltve.

+getHeight(): int : Magasságot adja vissza.

+getWidth(): int : Szélességet adja vissza.

+getSize(): Dimension : A pálya méreteit adja vissza.

8.1.13. 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

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

- Metódusok

+Cell(+cells: Array<Cell>, igrade: IGame) Konstruktor, paraméterként kapja a szomszédos mezők referenciáit.

+isPath(): bool Olyan értékkel tér vissza amilyen típusú a cella (true/false megfelelés).

+getNeighbours(): Array<Cell> Getter a neighbours-höz.

setNeighbours(neig: Array<Cell>) Setter a neighbours-höz.

8.1.14. Field

- Felelősség
A Field osztály a Cell osztály leszármazottja. A nem út típusú cellákat (mező) reprezentálja. Egy mezőre egy torony helyezhető.
- Ősosztályok
Object → Cell
- Interfészek
Nincs
- Attribútumok

-itower: ITower A mezőn álló torony interfészű elem tárolása.

-igrade: IGame : Referencia Game interfészéhez.

- Metódusok

+registerIFieldPlaceable(ifp: IFieldPlaceable): void Egy új tornyot ad hozzá a mezőhöz, ifp-en meghívja a registerField-et önmagával, ami a registerITowert hívja.

+deleteIFieldPlaceable(ifiel: IFieldPlaceable): void Eltávolítja a tornyot a mezőről.

registerITower(itower: ITower): void Beteszi a fieldbe a kapott tornyot.

+hasTower(): bool Megadja, hogy szabad-e a mező tower elhelyezéséhez.

+getITower(): ITower Visszaadja az itowert.

+Field(igrade: IGame) Konstruktor.

8.1.15. Path

- Felelősség
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.
- Ősosztályok
Object → Cell
- Interfészek
Nincs
- Attribútumok

-iobstacle: IObstacle Az esetleg az úton levő akadályt tárolja.

-enemies: ArrayList<Enemy> Az éppen áthaladó ellenségek listája.

-paths: ArrayList<Path> Következő path-ok címei.

- Metódusok

+isPath(): bool Igaz értékkel tér vissza.

+deleteIPathPlaceable(ipath: IPathPlaceable): void Kitörli a tárolójából a paraméterként kapott referenciával megegyező tárolt referenciát.

+deleteEnemy(e: Enemy): bool Kitörli enemies-ből a kapott e-t.

+deleteIObstacle(io: IObstacle): bool Akadály törlése az útról.

+registerIPathPlaceable(ipath: IPathPlaceable): void Beregisztrálja a paraméterként kapott objektumot, mint saját magán tartózkodó ellenség vagy akadály. RegisterPath-t hív az ipath-on, ami a rá jellemző registert hívja vissza.

+hasEnemy(): bool Megmutatja, hogy van-e a cellán ellenség.

+registerEnemy(e: Enemy): void A kapott ellenséget beteszi az enemies-be.

+registerIObstacle(io: IObstacle): void A kapott akadály lesz az obstacle.

+getEnemies(): ArrayList<Enemy> Visszatér az enemies-el.

getIObstacle(): IObstacle Visszatér myIObstacle-el.

+getNext(): Path Paths-ből ad vissza egy random elemet.

+hasObstacle(): bool Megadja, hogy van-e rajta Obstacle.

+setNextPaths(np: ArrayList<Path>): void Következő utak címeinek beállítása, nextPaths setter.

8.1.16. Towerhez tartozó krsztályok: Range, Speed, Damage, EnemyType

- Felelősség
A torony lövésének hatósugarát, gyorsítását, sebzését, ellenféltípusra sebzés, növelő kristályok osztályai.
- Ősosztályok
Object → Gem
- Interfészek
ITGem

- Attribútumok

-range/ speed/ damage/ eType: int/ int/ int/ String A kristálynak megfelelő attribútum, amin fejlődést.

- Metódusok

Konstruktorok

8.1.17. Obstaclehez tartozó kristályok: Intensity, Repair

- Felelősség
Akadályra helyezhető kristályok, ami növeli a lassítás értékét vagy megjavítja az akadályt.
- Ősosztályok
Object → Gem
- Interfészek
IOGem
- Attribútumok

-intensity: int Az intensityGem-hez tartozik, intensity értékét állítja az obstacle-ben. A repairGem-nek nincs attribútuma.

- Metódusok

Konstruktorok

8.1.18. IOGem

- Felelősség
Akadályra helyezhető kristályok interfésze.
- Metódusok

+upgradeObstacle(o: Obstacle): void A kapott akadályt fejleszti. Meghívja a kristályra jellemző, obstacle-ben lévő fejlesztő függvényt.

8.1.19. ITGem

- Felelősség
Toronyra illeszthető kristályok interfésze.
- Metódusok

+upgradeTower(t: Tower): void Fejleszti a kapott t tornyot magával. Meghívja a rá jellemező upgrade... tower metódust.

+getValue(): int Visszaad egy, a torony árával képzett értéket, a torony eladásakor jóváírandó mana érték kiszámításához.

8.1.20. IFieldPlaceable

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

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

8.1.21. IPathPlaceable

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

+eliminate(p: Path): void Az útra helyezhető objektum eltávolítása az útról (p-ről).

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

8.2. A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1. Ellenség mozgás és sebzés

- Leírás
A tesztelő létrehoz 2 ellenséget és vesz 2 tornyot, egyiket a 3 kristállyal, a másikat Elf kristállyal. Ez után utukra indítja az ellenségeket, akik útközben meghalnak. Végén elad a játékos egy tornyot.
- Ellenőrzött funkcionalitás, várható hibahelyek
Ellenségek létrehozása, halála. Tornyok vétele, fejlesztése, tüzelés és eladást tesztel.
Várható hibahely: ellenség mozgatása, kristály vétel
- Tesztesethez használt térkép

```
testmap1.txt
x      x      x      x      x      x
en      r      r      r      r      ex
x      x      x      x      x      x
```

- Bemenet

```
loadmap testmap1.txt
buy tower 2-1
buy tower 2-4
buy gem 2-1 speed
buy gem 2-1 range
buy gem 2-1 damage
buy gem 2-4 enemy elf
make hobbit 1-1
```

```

update 2
make elf 1-1
update 2
sell 2-1
exit

```

- Elvárt kimenet

```

map testmap1.txt loaded
Tower0 created
Tower1 created
SpeedGem0 created
RangeGem0 created
DamageGem0 created
EnemyTypeGem0 created
Hobbit0 created
update
Hobbit0 moves
Tower0 shoots Hobbit0
Hobbit0 damage 40
update
Hobbit0 moves
Tower0 shoots Hobbit0
Hobbit0 damage 40
Tower1 shoots Hobbit0
Hobbit0 damage 20
Hobbit0 died
Elf0 created
update
Elf0 moves
Tower0 shoots Elf0
Elf0 damage 40
update
Elf0 moves
Tower0 shoots Elf0
Elf0 damage 40
Tower1 shoots Elf0
Elf0 damage 60
Elf0 died
Tower0 sold for 80 mana
VEGE

```

8.2.2. Akadály teszt

- Leírás

A pályán elhelyez a játékos egy akadályt, amit fejleszt egy intenzitás növelő kristállyal, majd bejut az ellenség a végzet hegyéhez. Amikor az utolsó ellenség is áthaladt rajta elhasználódik.

- Ellenőrzött funkcionalitás, várható hibahelyek

Ellenségek létrehozása, akadály létrehozása, fejlesztése, elhasználódása. Végzet hegyére jutás.

Várható hibahely: akadály elhasználódása, ellenség bejutása a Végzet Hegyéhez

- Tesztesethez használt térkép

```

testmap2.txt:
x           x           x           x

```

en	r	r	ex
x	x	x	x

- Bemenet

```
loadmap testmap2.txt
buy obstacle 1-1
buy gem 1-1 intensity
make hobbit 1-0
update 14
exit
```

- Elvárt kimenet

```
map testmap2.txt loaded
Obstacle0 created
IntensityGem0 created
Hobbit0 created
update
Hobbit0 moves
Obstacle0 slows Hobbit0 by 10
Obstacle0 worn out
update
update
update
update
update
update
update
update
update
update
update
update
update
Hobbit0 moves
update
Hobbit0 moves
update
Hobbit0 moves
Hobbit0 reached Mount Doom
VEGE
```

8.2.3. Kód, és ellenség kettészélés

- Leírás

A pályára feltesz a tesztelő egy Human ellenséget és egy tornyot. Frissíti a játékot, a torony lő az ellenségre. Beállítja a kódot, és hogy kettészélő lövedékeket lőjenek a tornyok. Kétszer frissíti a pályát, elsőre csak lép, mert kikerült a torony hatósugarából, másodiknak újra bekerült, kettészélő lövedékkel lövi meg a torony.

- Ellenőrzött funkcionalitás, várható hibahelyek

Kód, kettélövő lövedék.

Várható hibahely: hatósugár újrakalibrálása, ellenség kettészélése

- Tesztesethez használt térkép

```
testmap3.txt:
```

```
x      x      x      x      x
en      r      r      r      ex
x      x      x      x      x
```

- Bemenet

```
loadmap testmap3.txt
make human 1-0
buy tower 2-4
buy gem 2-4 range
update 1
haze on
cut on
update 2
exit
```

- Elvárt kimenet

```
map testmap3.txt loaded
Human0 created
Tower0 created
RangeGem0 created
update
Human0 moves
Tower0 shoots Human0
Human0 damage 20
Haze is turned on
Cut is turned on
update
Human0 moves
update
Human0 moves
Tower0 shoots Human0
Human0 damage 0
Human0 cut in half
Human1 created
VEGE
```

8.2.4. Ellenség elágazáshoz ér

- Leírás
A pályára feltesz a tesztelő 3 különböző ellenséget. A pálya kialakításából adódóan 1 lépés után elágazáshoz érnek. Mindegyiket különböző irányba küldjük.
- Ellenőrzött funkcionalitás, várható hibahelyek
Ellenségek elágazás kezelése.
Várható hibahely: elágazás
- Tesztesethez használt térkép

```
testmap4.txt:
```

x	x	x	x	x	x
x	x	r	r	d	x
en	r	rud	r	r	ex
x	x	r	r	u	x
x	x	x	x	x	x

- Bemenet

```
loadmap testmap4.txt
make elf 2-1
make human 2-1
make dwarf 2-1
update 1
route Elf0 1
route Human0 0
route Dwarf0 2
update 1
drawMap
exit
```

- Elvárt kimenet

```
map testmap4.txt loaded
Elf0 created
Human0 created
Dwarf0 created
update
Elf0 moves
Human0 moves
Dwarf0 moves
update
Elf0 moves
Elf0 crossroad
Human0 moves
Human0 crossroad
Dwarf0 moves
Dwarf0 crossroad
/-----\
|xx|xx|xx|xx|xx|xx|
|xx|xx|xx|xx|xx|xx|
|---+---+---+---+---|
|xx|xx|e | | |xx|
|xx|xx| | | |xx|
|---+---+---+---+---|
| | | |e | | |
| | | | | | |
|---+---+---+---+---|
|xx|xx|e | | |xx|
|xx|xx| | | |xx|
|---+---+---+---+---|
|xx|xx|xx|xx|xx|xx|
|xx|xx|xx|xx|xx|xx|
\-----/
VEGE
```


8.3. A tesztelést támogató programok tervei

Készülnek a prototípus program, illetve a szövegfájl összehasonlító segédprogram fordítását és futtatását megkönnyítő batch fájlok, melyek futtatásuk után automatizáltan elvégzik ezeket a feladatokat.

Az előre elkészített, és a 8.2-es pontban leírt tesztesetek amellet, hogy futtathatóak a parancssorból a be-menet átirányításával a megfelelő fájlból (pl.: java ProtoTester < teszt1.dat), reprodukálhatjuk őket az erre a célra készített batch fájl segítségével is: test.bat. A batch fájl működése a következő: futtatás után ellenőrzi, hogy a szükséges programok (prototípus, illetve szöveges fájl összehasonlító) rendelkezésre állnak-e a megfelelő helyen, ha nem, akkor felkéri a felhasználót azoknak a lefordítására, vagy áthelyezésére, majd kilép. Amennyiben rendelkezésre állnak, megkérdezi a tesztelőtől, hogy melyik tesztesetet kívánja lefuttatni, illetve a kimenetet megjelenítse-e a konzolablakban. Amennyiben a megjelenítés mellett dönt a tesztelő, az adott tesztet lefuttatása után keletkező kimenet a konzolban jelenik meg, és billentyűlenyomásig ott is marad, így az elemezhető, az elvárt kimenettel összevethető. Ellenkező esetben, tehát ha a tesztelő úgy dönt, hogy ne jelenjen meg a prototípus kimenete, akkor egy fájlba irányítja azt a batch fájl, és az így keletkező fájl összeveti az elvárt kimenettel a TxtComparer segédprogram segítségével (ennek specifikációja lentebb található), és megmondja, hogy egyezik-e, majd billentyűlenyomás után kilép.

A tesztelést tovább segíti a TxtComparer nevű, szöveges fájlok összehasonlítását végző program. A program Java nyelven készül, forráskódja a prototípushoz hasonlóan adott. A program a fordítás után a parancssorból futtatható, ahol két paramétert vár: a két összehasonlítandó fájl nevét (pl.: java TxtComparer file1 file2). Futtatás után, amennyiben a helyes paraméterek rendelkezésre állnak, illetve elérhetőek a megadott fájlok, a program soronként összehasonlítja a két fájlt, és amennyiben azok megegyeznek, értesíti erről a felhasználót. Azon esetben, amikor a fájlok tartalmában eltérést észlel, az eltérést tartalmazó sorok számai jelennek meg a kimeneten.

8.4. Napló

Kezdet	Időtartam	Résztvevők	Leírás
2014.04.03. 15:00	5 óra	Rédey	8.1 pont
2014.04.05	3 óra	Elekes Seres	Értekezlet 8.2 pont megírása
2014.04.05	45 perc	Elekes	Tevékenység Teszteset megírása Kimeneti nyelv leírása
2014.04.05 21:40	1 óra	Seres	8.3 pont megírása
2014.04.07. 00:30	1,5 óra	Fuksz	Tesztesetek pályáinak megvalósítása. Dokumentáció véglegesítése.