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. április 9.

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 Az alapsebzés értéke. Ezt adják vissza a getDamage függvények, ha nincs fejlesztve az adott típusú Enemy-re a torony. -, int

  enemyType A torony itt tárolja, hogy melyik ellenség típusra erősebb a sebzése. -, String
- 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(String e) Beállítja az ellenséget akire specializált (enemyType). +

  void setDamage(int damage) 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 hegyét 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

  static final speed A két lépés között eltelt idő. -, int

static final maxHP A maximális életerő értéke. -, int

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

nextPath A soron következő path címe. -, Path

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

myPath Az a mező, ahol tartózkodik. #, Path

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

- Metódusok

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

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

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

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

setHealt(int h) A health változó settere. +

void cut() A kettévágásos lövésért felelős metódus. Létrehoz egy új, azonos típusú ellenséget (abstract), feleakkora életerővel. Beteszi az igam 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 ősosztály hurt metódusát.
- Ősosztályok
  Object → Enemy
- Interfészek
  IPathPlaceable
- Metódusok

hurt(Bullet) Sebződik, a kapott Bullet alapján getDamage... metódus által visszaadott értékkel csökkenti az életerejét. Ha elfogyott az életereje 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.
- Ősosztályok
  Object

- Interfészek
  IGame
- Attribútumok

  Map map játék térképe

  enemiesOut 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. -, List<Enemy>

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

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

  mana A maradék varázserő. -, int

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

  noEnemies 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. -, int

  succeededE 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. -, int

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

- Metódusok

  void update() 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. +

  void initialize(String name) Kiinduló állapot felvétele. +

  Game() Konstruktor. +

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

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 Szükséges függvények elérésére szolgál, hogy tudja módosítani a Game-et. -, int

chosenField Ha Field-et választ ki, ebben a változóban tárolja. -, Field

chosenPath Ha Path-t választ ki, ebben a változóban tárolja. -, Path

chosenEnemy A kiválasztott EnemyType. -, String

- Metódusok

void buyTower() Torony vételére szolgál. Létrehoz egy tornyot, beregisztrálja a chosenField-re. +

void buyObstacle() Lásd buyTower Obstacle-el és chosenPath-el. +

void buySpeed/Range/Damage/EnemyType/Intensiyty/RepairGem() Kristályok vásárlása. A Towert/Obstacle-t amire vettük a chosenField/Path-en érjük el. +

void setField(Field f) A chosenField-et állítja. +

void setPath(Path p) A chosenPath-t állítja. +

void setEnemy(String e) 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

void changeMana() A manát megváltoztató metódus. +

void incSucceeded() A succeededE értékét megváltoztató metódus. +

void addTower(Tower t) A hozzáad egy tornyot a towers listához. +

void removeEnemyIn(Enemy e) Az ellenség törlése enemiesIn-ből. +

void removeEnemyOut(Enemy e) Az ellenség törlése enemiesOut-ból. +

void removeTower(Tower t) A torony törlése towers-ből. +

void addEnemyIn(Enemy e) Ellenség hozzáadása enemiesIn-hez. +

int getMana() 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

static price Az ár, amennyi varázserőbe kerül. -, int

- Metódusok

Gem(int price) Konstruktor. +

8.1.8. IObstacle

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

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

void amortization() Amortizál, csökkenti amort értékét. +

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

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

repair() 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 A lassítás mértéke. -, int

myPath A path, amin rajta van. -, Path

amort Az elhasználódottság mértéke. -, int

static final price Az ára. +, int

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

- Metódusok

Obstacle(int intens, Path p, int amort, int price, bool up) Konstruktor. +

8.1.10. ITower

- Felelősség
  A torony funkciói vannak benne. Egy olyan interfész, amin keresztül a tornyok kezelhető.
- Metódusok

  void setPaths() A myField-ből kiindulva a range-el lefedett területen felkeresi, és beregisztrálja 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 hurt függvényét, átadva paraméterként a bullet-et. +

  void addITGem(ITGem gem) Paraméterként megkapja a kiválasztott kristályt, bulletet frissíti, ha kell, illetve a torony adott attrimútumait. +

  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. +

  void haze() Ködöt generál a tornyon mod range beállításával. +

  void clearUp() Beállítja a modRange-et normál, köd nélküli értékre range alapján. +

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

  static final price Az ára varázserőben. +, int

  range lőtáv, hatókör. -, int

  modRange Módosított 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. -, int

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

  ArrayList<ITGem> gems A megvásárolt kristályokat tárolja. -, ArrayList<ITGem>

  myField A mezőt tárolja amin áll. -, Field

  paths Hatósugárba eső út cellák. -, ArrayList<Path>

  myField Mező, amin áll. -, Field

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

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

  level A pálya szintje. -, int

  grid A cellákat tartalmazó 2 dimenziós tömb. -, Array<Array<Cell»

- Metódusok

  Map(Map map) Az osztály konstruktora. +

  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. +

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

- Metódusok

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

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

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 A mezőn álló torony interfészű elem tárolása. -, ITower

igame IGame referencia, hogy tudja módosítani a Game-et. -, IGame

- Metódusok

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

void deleteIFieldPlaceable(IFieldPlaceable ifield) Eltávolítja a tornyot a mezőről. +

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

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

ITower getITower() Visszaadja az itowert. +

Field() 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 Az esetleg az úton levő akadályt tárolja. -, IObstacle

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

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

- Metódusok

ArrayList<Enemy> hasEnemy() Visszaadja a rajta lévő ellenségek listáját (enemies-t). +

bool isPath() Igaz értékkel tér vissza. +

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

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

void registerIPathPlaceable(IPathPlaceable ipath) 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. +

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

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

void registerObstacel(Obstacle o) A kapott akadály lesz az obstacle. +

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

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

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

8.1.16. Towerhez tartozó krsitályok: 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 A kristálynak megfelelő attribútum, amin fejleszt. -, az első 3 int, eType String

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

- Metódusok

Konstruktorok

8.1.18. IOGem

- Felelősség
  Akadályra helyezhető kristályok interfésze.
- Metódusok

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

8.1.19. ITGem

- Felelősség
  Toronyra illeszthető kristályok interfésze.
- Metódusok

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

int getValue() 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

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. +

8.1.21. 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 (p-ről). +

void registerPath(Path p) 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 2-5
buy gem 2-1 speed
buy gem 2-1 range
buy gem 2-1 damage
buy gem 2-5 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 move
Tower0 shoots Hobbit0
Hobbit0 damage 40
update
```

```
Hobbit0 move
Tower0 shoots Hobbit0
Hobbit0 damage 40
Tower1 shoots Hobbit0
Hobbit0 damage 20
Hobbit0 died
Elf0 created
update
Elf0 move
Tower0 shoots Elf0
Elf0 damage 40
update
Elf0 move
Tower0 shoots Elf0
Elf0 damage 40
Tower1 shoots Elf0
Elf0 damage 60
Elf0 died
Tower0 sold 80 mana
```

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 4
exit
```

- Elvárt kimenet

```
map testmap2.txt loaded
Obstacle0 created
IntensityGem0 created
Hobbit0 created
```

```
update
Hobbit0 moves
Obstacle0 slows Hobbit0 by 1
Obstacle0 worn out
update
update
Hobbit0 moves
update
Hobbit0 reached Mount Doom
```

8.2.3. Köd, és ellenség kettészelés

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

- Ellenőrzött funkcionalitás, várható hibahelyek
  Köd, kettélövő lövedék.

  Várható hibahely: hatósugár újrakalibrálása, ellenség kettészelése

- 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
make tower 2-3
update 1
haze on
cut on
update 2
exit
```

- Elvárt kimenet

```
map testmap3.txt loaded
Human0 created
Tower0 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
```

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 3
update 1
drawMap
exit
```

- Elvárt kimenet

```
map testmap4.txt loaded
Elf0 created
Human0 created
Dwarf0 created
update
Elf0 moves
Elf0 crossroad
Human0 moves
Human0 crossroad
```

```
Dwarf0 moves
Dwarf0 crossroad
update
Elf0 moves
Human0 moves
Dwarf0 moves


    0   1   2   3   4   5
0   x   x   x   x   x   x
1   x   x   hob r   r   x
2   en  r   r   hum r   ex
3   x   x   elf r   r   x
4   x   x   x   x   x   x
```

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 amellett, hogy futtathatóak a parancssorból a bemenet 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 tesztteset 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 fájlt 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.