**5. Szkeleton tervezése**

98 – The Uninterruptables

Konzulens:

Tóth Márton

Csapattagok

|  |  |  |
| --- | --- | --- |
| Gaál Soma | BQ4777 | [galso5@gmail.com](mailto:galso5@gmail.com) |
| Szekeres Imre | GDF6HY | [bushi21@freemail.hu](mailto:bushi21@freemail.hu) |
| Jobbágy Vince | D2ZCY2 | [Jobbagy.vince@gmail.com](mailto:Jobbagy.vince@gmail.com) |
| Buzás Krisztián | KUS0ZS | [ccester07@gmail.com](mailto:ccester07@gmail.com) |
| Danyik Dávid | Q5YZIP | [Danyik.david@freemail.hu](mailto:Danyik.david@freemail.hu) |

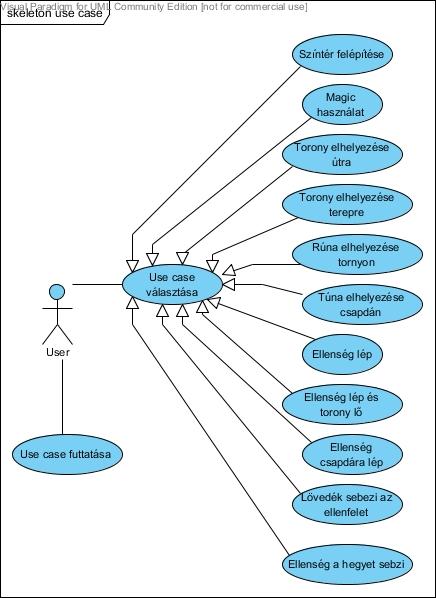
2014.03.16

# **5.** **Szkeleton tervezése**

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

### 5.1.1 Use-case diagram

**A felhasználóhoz tartozó Use-case diagram**



### 5.1.2 Use-case leírások

|  |  |
| --- | --- |
| **Use-case neve** | **Cast-rune-trap** |
| **Rövid leírás** | Ez a use-case azt ellenőrzi, amikor a játékos egy csapda-rúnát szeretne egy csapdára helyezni. A bemutatáshoz gyengítő rúnát használ, a többi ugyanilyen, csak más státuszmódosítóval. |
| **Aktorok** | csapda, útcella, státuszmódosító (gyengítő), rúna (gyengítő) |
| **Forgatókönyv** | Először meghívjuk a rúnára (activeObject) a canCastOn(grid) metódust. Ezután a rúna, a cella és a csapda párbeszéde következik: a rúna megkérdezi a cellát, készen áll-e rá. A grid elkéri a saját InjectionTarget-ét, ami esetünkben egy csapda lesz. Ezután, annak átadva a rúnát, megkérdezi, ő készen áll-e a beültetésre. A csapda megnézi, van-e még hely rajta (itt válik el a teszteset annak megfelelően, hogy milyen bemenet érkezett a skeleton felhasználójától). Ha van, megkérdi a rúnát, mit tud vele kezdeni - azaz meghívja a rúna canInjectOn(trap) metódusát. Erre, mivel csapdarúnáról van szó, az létrehoz valamilyen státuszmódosítót, amelyet elhelyez a csapda listájában. Ha már nincs hely új rúnára, akkor false-al tér vissza. |

|  |  |
| --- | --- |
| **Use-case neve** | **Cast-rune-tower** |
| **Rövid leírás** | Ez a use-case azt ellenőrzi, amikor a játékos egy csapda-rúnát szeretne egy toronyra helyezni. A bemutatáshoz gyengítő rúnát használ, a többi ugyanilyen, csak más státuszmódosítóval. |
| **Aktorok** | torony, földcella, rúna (gyengítő) |
| **Forgatókönyv** | Először meghívjuk a rúnára (activeObject) a canCastOn(grid) metódust. Ezután a rúna, a cella és a torony párbeszéde következik: a rúna megkérdezi a cellát, készen áll-e rá. A cella elkéri a saját InjectionTarget-ét, ami esetünkben egy torony lesz. Ezután, annak átadva a rúnát, megkérdezi, ő készen áll-e a beültetésre. A torony megnézi, van-e még hely rajta (itt válik el a teszteset annak megfelelően, hogy milyen bemenet érkezett a skeleton felhasználójától). Ha van, megkérdi a rúnát, mit tud vele kezdeni - azaz meghívja a rúna canInjectOn(tower) metódusát. Erre, mivel csapdarúnáról van szó, false-al tér vissza, hiszen csapda rúnát nem lehet toronyra helyezni. Ha már nincs hely új rúnára, akkor false-t ad vissza. |

|  |  |
| --- | --- |
| **Use-case neve** | **Troop-lép** |
| **Rövid leírás** | Ez a use-case az ellenfél lépésének meghatározását végzi. |
| **Aktorok** | ellenfél, útkereső |
| **Forgatókönyv** | Az ellenség cselekvésének sorra kerülésével a controlI-t metódus először azt deríti ki, merre tud lépni. Ezt a feladatot végzi a PathFinder utility class, mely megkapja, hogy melyik útcellán áll az ellenfél, és vissza ad egy útcellát ahova lépnie érdemes. Mindig olyan cellát keres ami a cél felé haladást elősegíti, tehát visszafele nem léphet az ellenség. Ha talál olyan cellát ami üres, akkor oda irányítja, ha nem akkor megpróbál olyat vissza adni, amin barikád van, ha olyan sincs, akkor a jelenlegi celláját adja vissza az ellenfélnek, ezzel jelezve, hogy nem tud semerre se mozdulni. |

|  |  |
| --- | --- |
| **Use-case neve** | **Troop-lép-trap** |
| **Rövid leírás** | Ez a use-case az ellenfél csapdára lépését fedi le. |
| **Aktorok** | ellenfél, csapda, útkereső, útcella, sebezhető(barikád) |
| **Forgatókönyv** | Az ellenség lépése során az útkeresőtől kéri a következő cellát. Amennyiben ez a cella üres, akkor odalép, ha nem akkor interakcióba lép az ott lévő sebezhetővel. Minden esetben megnézi, hogy van-e a kapott cellán csapda, és ha van, természetesen ráteszi a hatását az ellenfélre. |

|  |  |
| --- | --- |
| **Use-case neve** | **Lövedék- sebez** |
| **Rövid leírás** | Ezen use-case az egyes ellenfelek sebződését fedi le, amely magában foglalja a töltet mozgását, érkezését, alapállapotba kerülését. |
| **Aktorok** | lövedék, ellenfél, torony, útcella |
| **Forgatókönyv** | A lövedék sorra kerül cselekvésre, meghívják a controlI-t metódusát, amelyben szabadon dönt, saját állapotának függvényében, mit is csináljon. Amennyiben még pihennie kell, akkor csökkenti a pihenés hátramaradó időszeleteinek számát, s ha az eléri a zérust, akkor tüzelésre késznek ítéli magát. Egyébként pedig cselekszik, amennyiben nem várakozik célpontra. Elsőként előrébb halad, majd megvizsgálja, elérte-e a célját. Amennyiben igen, akkor interakcióra hívja a célpontot, amely a kommunikáció során lekérdezi a töltet faj-specifikus sebzését, s ezzel csökkenti a hátramaradt életpontjait. Ha egy ellenfél éltét veszíti, akkor szól az őt tartalmazó cellának, hogy engedje el.  A töltet, miután leadta sebzését, pihenő állást vesz fel, visszatér a torony pozíciójára és beállítja várakozási idejét. |

|  |  |
| --- | --- |
| **Use-case neve** | **Troop-lép + Torony-lő** |
| **Rövid leírás** | Az ellenfelek egy dedikált, már kiválasztott cellára való áthelyezkedését, s az ezzel járó adminisztrációkat, valamint azon konfliktus-helyzetet, amikor is a célpont cella már foglalt, mutatja be ez a use-case. Ezen esetben a cellát foglaló sebezhető egy barikád, hogy szemléltessük azt a kapcsolatba lépést, amely közöttük zajlik. |
| **Aktorok** | troop/ellenfél, útcella, torony, töltény, sérülékeny(barikád) |
| **Forgatókönyv** | Az ellenfélnek cselekvésre támad lehetősége, s amennyiben pihenőideje lejárt, lépést kezdeményez a már előre kiválasztott cellára, amelyet a PathFinder helper osztály szolgáltatott neki (ennek kifejtésére külön use-case-t teszünk elérhetővé). Elsőként lekérdezi a cellától, hogy tartózkodnak-e már rajta sérülékeny egységek, s amennyiben igen, akkor interakcióra hívja azt, azaz meghívja annak interactWith metódusát, átadva magát. Esetünkben barikád van a másik oldalon, ezért az majd lekéri az ellenséges egység sebzését, és elszenvedi azt, majd, ha elfogytak életpontjai, az őt tartalmazó cellából kiregisztráltatja magát. Amennyiben (már) nem foglalt a célpont cella, akkor figyelmezteti azt, hogy Ő bizony rálépett. Ekkor a cella beregisztrálja magához, s értesít minden tornyot, aki beregisztrálta magát hozzá ellenfélVan eseményre. A tornyok erre lekérdezik a töltetüket, hogy képesek-e lőni, és ha igen, akkor ellövik, amijük van. Ennek során átadják célpontnak az ellenfelet a töltet részére, a töltet beállítja, hova is kell szállnia, és átállítja az állapotát. |

|  |  |
| --- | --- |
| **Use-case neve** | **Cast-Tower-Ground** |
| **Rövid leírás** | A tornyok egy bizonyos cellára történő lehelyezése, amely vizsgálja, hogy a kiválasztott cella (ground) alkalmas-e a torony ‘befogadására’. |
| **Aktorok** | torony, földcella, user, logikai színtér |
| **Forgatókönyv** | A user által kiválasztott toronytípust az egér segítségével az általa kiválasztott cella fölé helyezi. Amennyiben ez a cella foglalt, vagy útcella, abban az esetben a rendszer nem engedélyezi a cselekvést. Amennyiben földcella került kiválasztásra - ezt a logikai színtér vizsgálja - létrejön a torony, amely a játéktéren kerül elhelyezésre, a kiválasztott cellát lefoglalja magának, ezután a torony beregisztrálja magát a hatókörében lévő összes cellához, mint ‘előfizető’. |

|  |  |
| --- | --- |
| **Use-case neve** | **Cast-Tower-Road** |
| **Rövid leírás** | A Tornyokat cellákra helyezhetjük, ez a use-case vizsgálja, hogy a kiválasztott útcella(road) alkalmas-e, hogy tornyot építsünk rá. |
| **Aktorok** | user, torony, (road)útcella, logikai színtér |
| **Forgatókönyv** | A user/felhasználó által építendő toronyfajtát az egér segítségével, a kívánt cellák fölé húzza. Tornyot csak nem foglalt, szabad ground cellára lehet építeni. Így ha (road) útcella fölé húzzuk a tornyot, a torony nem rakható le, semmilyen esetben sem. Így piros marad a torony képe és a user/felhasználó nem lesz képes megépíteni. |

|  |  |
| --- | --- |
| **Use-case neve** | **Cast-magic** |
| **Rövid leírás** | Ezen use-case egy varázslat ellövését szimulálja. Esetünkben egy Nazghul-t idézünk meg, amely lebénítja a pályán mozgó ellenfeleket. |
| **Aktorok** | ellenfél, színtér, Mordorframe, varázslat(Nazghul), státuszmódosító, úti cella |
| **Forgatókönyv** | A szkeleton létrehozza a varázslatot, majd lekérdezi annak árát. Ezután megtudakolja a színtértől, hogy mennyi mana áll még a felhasználó rendelkezésére, s amennyiben elegendő, akkor elkezdődik az idézés, ellenkező esetben nem történik semmi. Élesben az idézés abból állna, hogy végigszaladunk a scene-tér ellenfél listáján, s azoknak átadjuk a státusz módosítót, amit a varázslatból nyerünk ki, ez esetben egyetlen gird egyetlen ellenfelét kérjük le, és adjuk át neki a már lekérdezett módosítót. |

|  |  |
| --- | --- |
| **Use-case neve** | **Troop-Sebez-Hegyet** |
| **Rövid leírás** | Az ellenséges egységek a pálya végére érve a Végzet Hegyével találják magukat szemben, ezt a találkozást hivatott modellezni ez a use-case |
| **Aktorok** | játék-színtér, Végzet Hegye, ellenfél, útcella |
| **Forgatókönyv** | A még élő EnemyTroopok a térképen végighaladva, az utolsó cellára lépve lekérdezik, hogy mi tartózkodik a következő cellán. Ennek az információnak a birtokában interakcióba lépnek a MountainOfDoom-al, melynek nincs élete, csupán egy találat elég az EnemyTroop-tól, hogy megsemmisüljön. Ekkor a hegy meghívja a Scene EndGame() metódusát, melynek hatására a játék véget ér és a felhasználó veszített. |

|  |  |
| --- | --- |
| **Use-case neve** | **MordorFrame-builds** |
| **Rövid leírás** | Ezen use-case-en keresztül szimuláljuk a tényleges színtér felépítését.  Sarkalatos pontjai a Builder pattern, amellyel a színtér épül, valamint az a mechanizmus, amellyel, egy file-ból olvasva adatokat, jönnek létre az egyes cellák, s azok “összekötött” kollekciója. |
| **Aktorok** | MordorFrame, main, ContentPane, Scene.Builder, Scene |
| **Forgatókönyv** | A program fő metódusa (main) létrehozza a keretet (MordorFrame), amely ezzel inicializálja a GUI-t, annak minden kütyüjét (widget), ami a felhasználó számára vezérlési felületet képez. Ezen közben létrehoz egy építőt a Scene-en keresztül, azt felparaméterezi, hogy mely file-ból olvassa be az adatokat a színtér felépítéséhez, majd meghívja annak a build() metódusát. A metódus egy belső mechanizmusnak, buildScene() továbbítja a hívást, amely a már beállított filePath alapján összeállítja a cellák kollekcióját, s ezzel visszatér. A gyűjtemény paraméterül szolgál a Scene privát konstruktorának, amelyet maga a build() hív s ad majd vissza. E “közben” a frame lekéri saját ContentPane objektumát és beleteszi a visszakapott Scene objektumot, ezzel létrehozva a színteret. |

## 5.2 A szkeleton kezelői felületének terve, dialógusok

Az általunk tervezett szkeleton egy, a standard konzollal vezérelhető tesztalkalmazás lesz. Feladata, hogy biztosítsa, hogy a létrehozott specifikáció az implementációval összhangban lesz.  
  
A fontosabb tesztesetekhez külön futtatható állományok lesznek. Az azokon belül előforduló, kisebb döntések (adott változó értéke valaminél nagyobb, stb…) kérdésként fognak megjelenni a standard kimeneten, melyet követően a felhasználó által megadott válaszoknak megfelelően fog folytatódni a program futása.

A program indításakor a standard kimenetre kiírja a különböző use-case-eket, amiket ki lehet választani. A felhasználó ezután az azok mellett található számokkal navigálhat.  
  
Egy teszteset lefutása során kiírja a meghívott függvények nevét (pl:->castOn(grid)), azok visszatérése során a visszatérési értéket és megint csak a függvény nevét, jelezve, hogy épp kilép (pl.: return contains(xy) : true)

## 

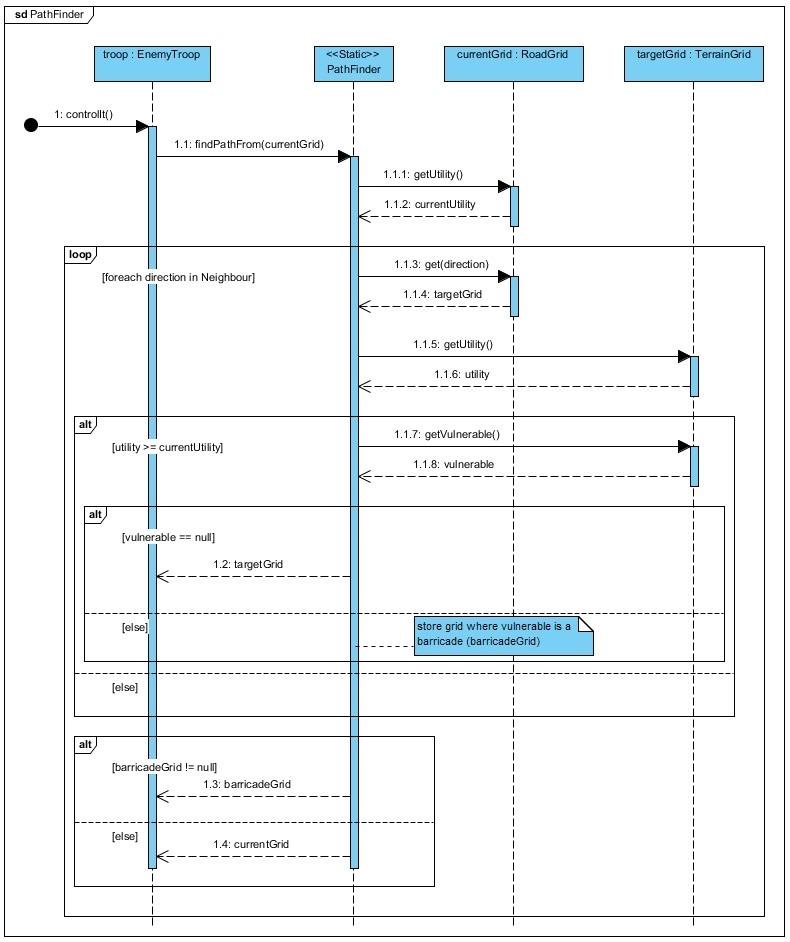
## 5.3 Szekvencia diagramok a belső működésre

[A szkeletonban implementált szekvencia diagramok. Tipikusan egy use-case egy diagram. Ezek megegyezhetnek a korábban specifikált diagramokkal, de az egyes életvonalakat (lifeline) egyértelműen a szkeletonban példányosított objektumokhoz kell tudni kötni. Azt kell megjeleníteni, hogy a szkeletonban létrehozott objektumok egymással hogyan fognak kommunikálni.]

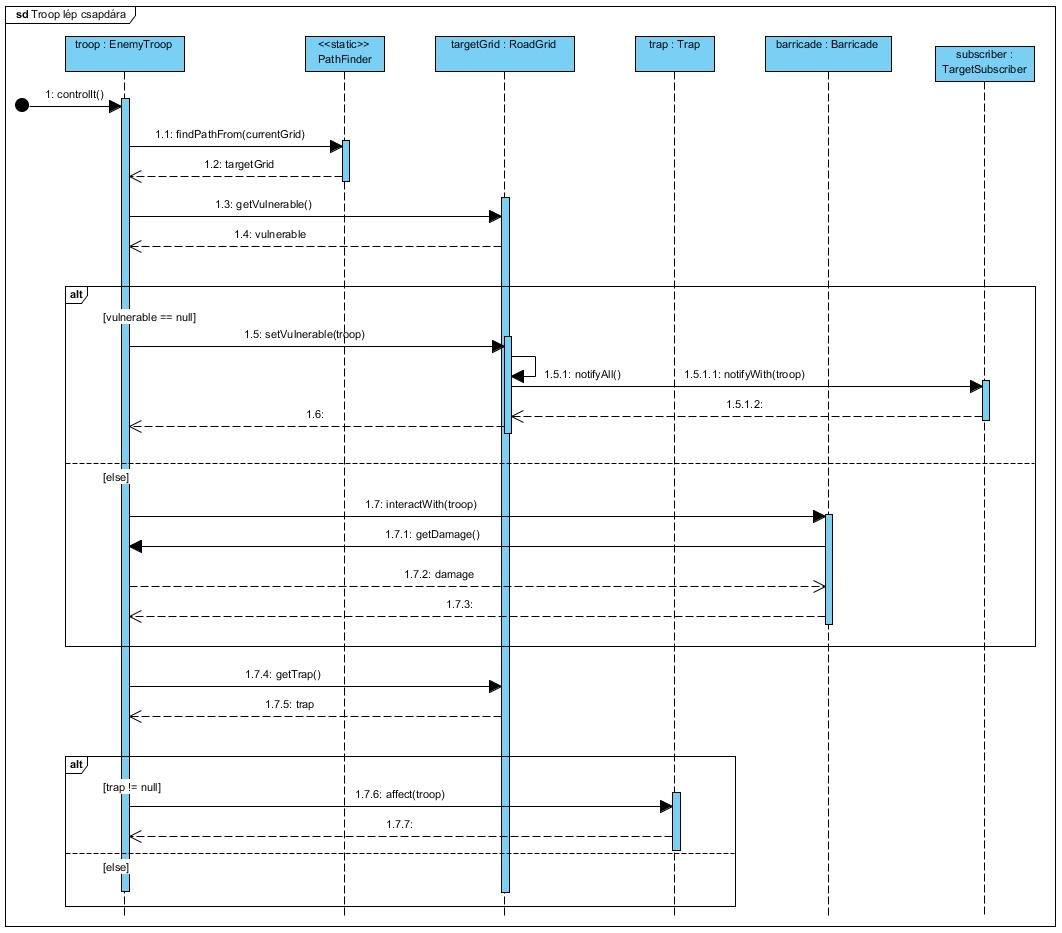
**MordorFrame-builds szekvencia**

|  |
| --- |
| A description... |

**Troop-lép szekvencia**



**Troop-lép-trap szekvencia**



**Troop lép, torony lő szekvencia**

**[Első Rész]**

|  |
| --- |
| A description... |

**Troop lép, torony lő szekvencia**

**[Második Rész]**

|  |
| --- |
| A description... |

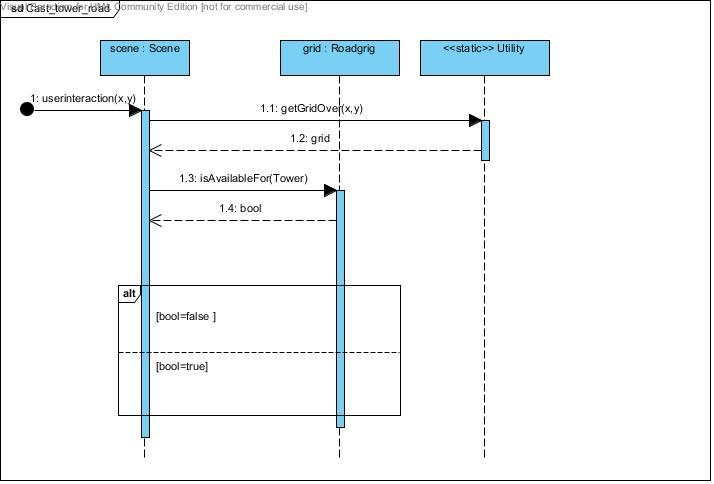
**Lövedék sebez szekvencia**

|  |
| --- |
| A description... |

## cast-tower-ground szekvencia diagram

|  |
| --- |
| A description... |

## cast-tower-road szekvencia diagram



**Troop-sebez-hegyet szekvencia**

|  |
| --- |
| A description... |

**Csapdarúna elhelyezése csapdán**

|  |
| --- |
| A description... |

## Csapdarúna elhelyezése tornyon

|  |
| --- |
| A description... |

**Cast-magic szekvencia**

|  |
| --- |
| A description... |

## 

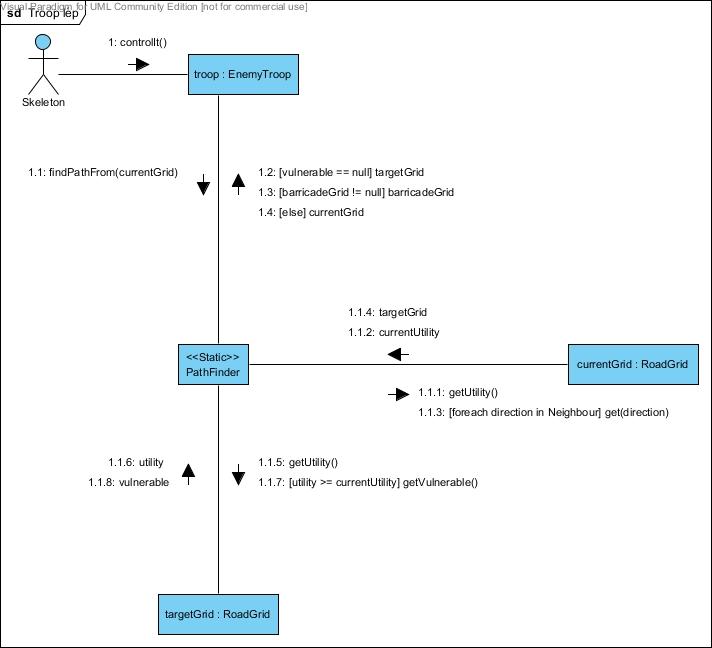
## 

## 

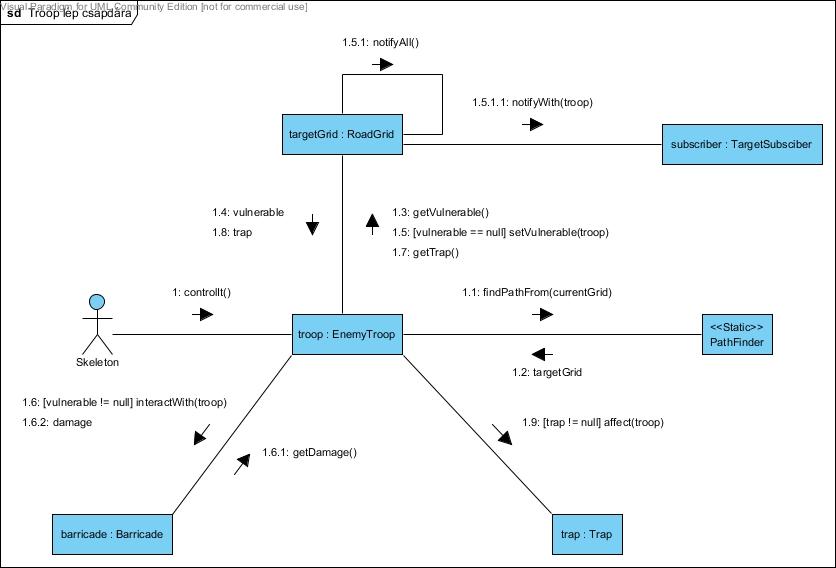
## 5.4 Kommunikációs diagramok

[A szkeletonban, az egyes szkeleton-use-case-ek futása során létrehozott objektumok és kapcsolataik bemutatására szolgáló diagramok. Ezek alapján valósítják meg a szkeleton fejlesztői az inicializáló kódrészleteket.]

**Troop lép**



**Troop lép csapdára**



**Lövedék sebez**

|  |
| --- |
| A description... |

**Troop lép, torony lő**

|  |
| --- |
| A description... |

## cast-tower-ground kommunikációs diagram

|  |
| --- |
| A description... |

**Troop-sebez-hegyet kommunikációs diagram**

|  |
| --- |
| A description... |

**MordorFrame-builds kommunikáció**

|  |
| --- |
| A description... |

**Csapdarúna elhelyezése csapdán kommunikáció**

|  |
| --- |
| A description... |

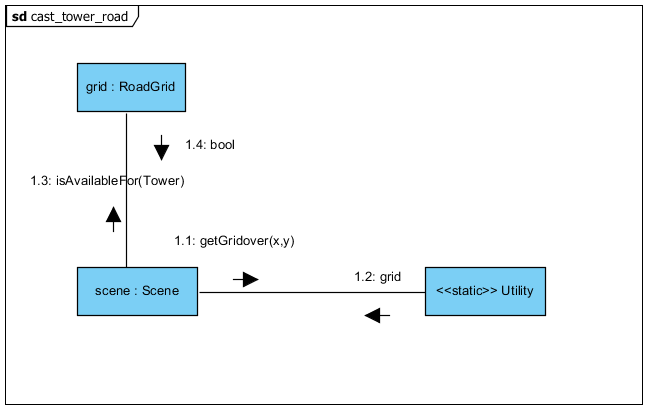
**Csapdarúna elhelyezése tornyon kommunikáció**

|  |
| --- |
| A description... |

**Cast-magic kommunikáció**

|  |
| --- |
| A description... |

**cast\_tower\_road kommunikációs diagram**



**5.5** **Napló**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2014.03.13 14.00 | 2 óra | Gaál  Danyik  Buzás  Jobbágy  Szekeres | Tevékenység:  konferencia, feladatkiosztás |
| 2010.03.14. 14.00 | 3.5 | Szekeres | Tevékenység:  troop lép torony lő  szekvencia diagram, kommunikációs diagram, use-case leírása |
| 2014.03.14. 22:00 | 2 óra | Gaál | Tevékenység:  use-case diagram és leírás, szekvencia diagram |
| 2014:03:15. 10:00 | 2 óra | Gaál | Tevékenység:  use-case leírás, szekvencia diagram, kommunikációs diagram |
| 2014. 03.15 12:00 | 4 óra | Buzás | Tevékenység:  cast\_tower\_ground szekvencia/kommunikációs diagram/use-case leírás  troop-hit-mountain  szekvencia/kommunikációs diagram/use-case leírás |
| 2014.03.16. 12:00 | 1.5 óra | Jobbágy | Tevékenység: kezelői felület megírása, rúna torony és csapdára helyezés szekvenciája, valamint use-case-e |
| 2014.03.16. 13.00 | 1.5 óra | Szekeres | Tevékenység:  a színtér felépítése, inicializálása szekvencia, kommunikációs diagramok felvétele, valamint a use-case kitöltése |
| 2014.03.16 18.00 | 1 óra | Jobbágy | Szekvencia diagram javítás, kommunikációs diagramok |
| 2014.03.16 18.00 | 2 óra | Danyik | Tevékenység: Use-case leírás,Szekvencia diagram, kommunikációs diagram |
| 2014.03.16 23.00 | 1 óra | Szekeres | Tevékenység:  cast-magic szekvencia és kommunikáció modellezése diagramokon, finomítások + use-case leírás |