**4. Analízis modell kidolgozása 2**

93 – angolkarakterek

Konzulens:

Tüske Máté Róbert

**Csapattagok:**

|  |  |  |
| --- | --- | --- |
| Hegedüs Dániel | YBY8BK | dani940414@gmail.com |
| Nemeslaki Keve | X72642 | nemeslaki.keve@gmail.com |
| Rabinovits Jakov | FSR38D | jashok12@gmail.com |
| Bálint Kártyás | ZBRZFP | Bkartyas@gmail.com |
| Tuskó Gergely | DM04Q5 | t.gergely95@gmail.com |

# **4.Analízis modell kidolgozása 2**

## **4.1 Objektum katalógus**

### **4.1.1 Box (doboz)**

O’Neill fel tudja venni a dobozt és le tudja rakni. Ha egy mérlegre helyezi a játékos, akkor a mérleg állapota megváltozik, egy ajtót kinyitva.

**4.1.2 Chasm (mérleg)**

Olyan pályaelem, amelyre ha O’Neill rálép, akkor meghal, és a játéknak vége. Ha O’Neill egy dobozt rak egy szakadékra, az adott doboz is megsemmisül. Koordináta párral rendelkezik, ami jelöli a pozícióját a pályán.

### **4.1.3 Door (ajtó)**

Két állapottal rendelkezik, lehet csukott és nyitott. Csukott állapotban falként, nyitott állapotban padlóként viselkedik. Minden Ajtóhoz tartozik egy mérleg, amelyre ha súly kerül (ez lehet doboz vagy maga O’Neill ),akkor az ajtó nyitott állapotba kerül, míg ha a súly lekerül a Mérlegről,akkor az ajtó visszazárul

### **4.1.4 Floor (padló)**

O’Neill át tud sétálni rajta. Lehet, hogy van rajza ZPM, amit O’Neill autómatikusan felvesz, ha ilyen mezőre lép.

### **4.1.5 Level (pálya)**

Létrehozza és tartalmazza a játékos által látható pályaelemeket.

### **4.1.6 Missile (lövedék)**

O’Neill hozza létre, amikor lő. Lehet kék vagy sárga. Előre mozog abban az irányba, amerre O’Neill néz a lövés pillanatában, és addig megy, amíg falba vagy zárt ajtóba nem ütközik. Ha sima falba ütközik, akkor hatás nélkül megsemmisül a lövedék. Ha olyan falba ütközik melyen nyílhat portál, akkor a felületen létre jön egy, a lövedék színének megfelelő portál, és ha volt már korábban egy olyan színű portál a pályán, akkor az eltűnik.

### **4.1.7 ONeill (O’Neill)**

A játékos tudja irányítani. Rendelkezik kordinátapárral, ami jelöli az aktuális pozícióját a pályán. Tud lőni a fegyverével, ekkor létre jön egy lövedék. Ha rálép egy ZPM-et tartalmazó padlóra, akkor autómatikusan felveszi azt.

### 

### **4.1.8 Opener (mérleg)**

Minden ajtóhoz tartozik egy mérleg. Minden mérleg két állapottal rendelkezik, vagy van rajta súly vagy nincs. Amennyiben súly kerül rá, a hozzá tartozó ajtó kinyílik, és ameddig a súly ott van, nyitva is marad. Ha nincs rajta súly, a hozzá tartozó ajtó csukott állapotban van.

### **4.1.9 Portal (portál)**

Megfelelő falon lövedék hatására nyíló objektum. Két színű lehet, a pályán maximum 1 sárga és 1 kék lehet jelen. Ha mindkét színből van a pályán, akkor a két portál között féreglyuk nyílik, tehát O’Neill ennek segítségével tud teleportálni.

### **4.1.10 Wall (fal)**

A falak alap állapotban korlátozzák a O’neil mozgását, nem tud átsétálni rajtuk. Ha a fal olyan, amin nyílhat portál, és beleütközik egy lövedék, akkor ott a lövedéknek megfelelő színű portál nyílik.

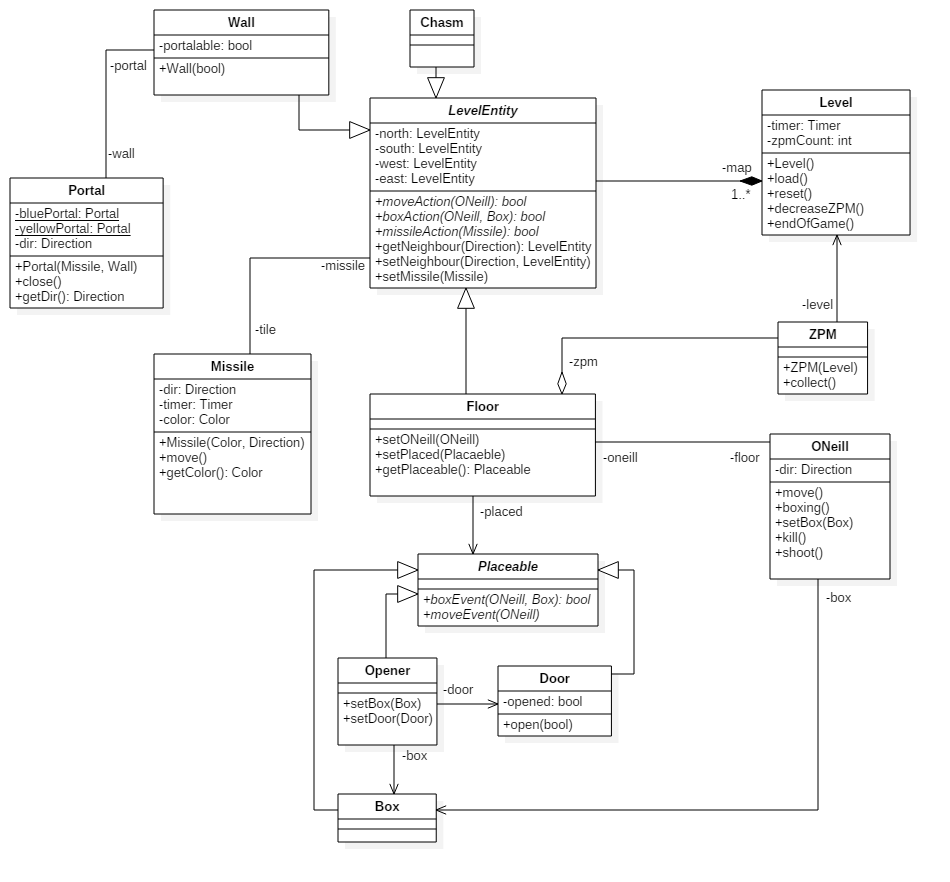
### **4.1.11 ZPM (ZPM)**

A ZPM-ek az ONeill által hőn áhított kincsek, amelyeket megszerezve megmentheti a világot. Padló mezőkön lehet, ha O’Neill rálép egyre, azt begyűjti.

## 

## 

## **4.2 Statikus struktúra diagramok**



## **4.3 Osztályok leírása**

### **4.3.1 Box**

#### **4.3.1.1 Felelősség**

Olyan elem leírása, melyet ONeill fel tud venni és le tud rakni. Ha mérlegre kerül, akkor kinyílik a mérleghez tartozó ajtó.

#### **4.3.1.2 Ősosztályok**

Placeable

#### **4.3.1.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.1.4 Attribútumok**

Nincs attribútuma.

#### **4.3.1.5 Metódusok**

* **bool boxEvent(ONeill, Box)**: Hamissal tér vissza, mert a dobozra nem lehet lerakni egy másik dobozt.
* **moveEvent(ONeill)**: Igazzal tér vissza, mivel ONeill rá tud sétálni.

### **4.3.2 Chasm**

#### **4.3.2.1 Felelősség**

Olyan pályaelem leírása, amely elpusztítja ONeill-t ha rá lép, illetve a dobozokat, amiket ráhelyeznek.

#### **4.3.2.2 Ősosztályok**

#### LevelEntity

#### **4.3.2.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.2.4 Attribútumok**

Csak leszármazott attribútumai vannak.

#### **4.3.2.5Metódusok**

* **bool moveAction(ONeill)**: Megöli paraméterül kapott ONeill-t. Visszatér hamissal.
* **bool boxAction(ONeill, Box)**: Visszatér igazzal. Elpusztul a doboz.
* **bool missileAction(Missile)**: Beállítja saját missile referenciát a paraméterül kapott referenciára. Visszatér igazzal, mivel a lövedék át tud haladni felette.

### **4.3.3. Door**

#### **4.3.3.1 Felelősség**

Olyan pályaelem leírása, melynek két állapota van, csukott állapotban olyan Wall-ként viselkedik, melyen nem nyitható portál, nyitott állapotban pedig Floor-ként.

#### **4.3.3.2 Ősosztályok**

Placeable

#### **4.3.3.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.3.4 Attribútumok**

* **opened: bool** : Azt tárolja, hogy az ajtó nyitva van-e, vagy sem.

#### 

#### **4.3.3.5 Metódusok**

* **bool boxEvent(ONeill, Box)**: Visszatér hamissal, mivel dobozt nem lehet rárakni.
* **moveEvent(ONeill)**: Ha nyitva van, igazzal tér vissza, ha zárva van, akkor hamissal..
* **bool open(bool)**: Az opened adattagot a paraméterként kapott értékre állítja.

### **4.3.4 Floor**

#### **4.3.4.1 Felelősség**

Olyan pályaelemet valósít meg, amelyen ONeill tud járkálni, illetve lehetnek rajta különböző Placeable játékelemek, Box, Opener, vagy Door.

#### **4.3.4.2 Ősosztályok**

LevelEntity

#### **4.3.4.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.4.4 Attribútumok**

* **placed: Placeable:** Olyan objektum, ami rajta lehet a padlón(Opener, Door, Box). Ha van rajta már egy Opener, és le akarunk rakni egy Box-ot, akkor az az Openerre kerül.

#### **4.3.4.5 Metódusok**

* **bool moveAction(ONeill)**: Ha nincs rajta ajtó, vagy az nyítva van, akkor igazzal tér vissza és átállítja saját oneill referenciáját a paraméterül kapott referenciára. Ha van rajta ZPM, akkor meghívjuk a collect() függvényét és nullra állítjuk a ZPM referenciáját.
* **bool boxAction(ONeill, Box)**: Ha nincs rajta placed objektum, akkor igazzal tér vissza. Különben meghívja a placed boxEvent().
* **bool missileAction(Missile)**: Ha van rajta egy zárt ajtó, akkor hamissal tér vissza. Különben igazzal tér vissza és átállítja a missile referenciáját a paraméterül kapottra, tehát oda tud mozogni a lövedék.
* **setONeill(ONeill):** Beállítja ONeill referenciáját a paraméterül kapottra.
* **setPlaced(Placeable)**: Placed referenciát átállítjuk a paraméterül kapottra.
* **Placeable** **getPlaceable()**: Visszaadja a placed objektum referenciát.

### **4.3.5 Level**

#### **4.3.5.1 Felelősség**

Betölti, eltárolja az egész pályát, és vezérli a játék végét, a timer értéke, és a ZPM-ek száma alapján.

**4.3.5.2 Ősosztályok**

Nincs őse.

#### **4.3.5.3 Interfészek**

Nem valósít meg interfészt.

#### 

#### **4.3.5.4 Attribútumok**

* **Timer:** timer : Az időzítőt valósítja meg.
* **int: zpmCount :** Tárolja, hogy hány zpm van még a pályán.

#### **4.3.5.5 Metódusok**

* **Level():** Level konstruktorja, inicializálja a timert, zpmCount-ot és meghívja a load() függvényt.
* **load():** Fileból betölti a pályát.
* **reset():** Újraindítja a játékot.
* **decreaseZPM():** Csökkenti eggyel a ZPM-ek számát. Ha nincs több zpm a pályán, akkor meghívja a reset() függvényt.

### **4.3.6 LevelEntity**

#### **4.3.6.1 Felelősség**

Egy pályaelemet ír le, ez lehet padló, fal, vagy szakadék, illetve elmenti az ezzel szomszédos pályaelemeket. Vezérli, hogy mi történjen, ha rá akarunk lépni, rakéta megy rá, vagy dobozt akarunk rá rakni.

#### **4.3.6.2 Ősosztályok**

Nincs ősosztálya.

#### **4.3.6.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.6.4 Attribútumok**

* **north: LevelEntity**: a pályaelemtől északra lévő pályaelem.
* **south: LevelEntity**: a pályaelemtől délre lévő pályaelem.
* **west : LevelEntity:** a pályaelemtől nyugatra lévő pályaelem.
* **east: LevelEntity**: a pályaelemtől keletre lévő pályaelem.
* **missile: Missile**: az objektum feletti lövedék referenciája.

#### **4.3.6.5 Metódusok**

* **bool moveAction(ONeill)**: Ez a függvény definiálja, hogy mit kell tenni, ha az objektumra ONeill rálépett. A visszatérési értéke akkor igaz, ha ONeill sikeresen rá tud lépni.
* **bool boxAction(ONeill, Box)**: Ez a függvény definiálja, hogy mit kell tenni, ha az objektumra ráraktak egy dobozt vagy felveszik a rajta lévő dobozt (ekkor a Box paraméter egy nullreferencia). A visszatérési értéke akkor igaz, ha sikeresen rá tudjuk rakni a dobozt.
* **bool missileAction(Missile)**: Ez a függvény definiálja, hogy mit kell tenni, ha az objektum felett megy el a lövedék. Visszatérési értéke akkor igaz, ha a lövedék el tud menni felette, különben hamis.
* **LevelEntity getNeighbour(Direction)**: Visszaadja irány alapján a szomszédos LevelEntity referenciát.
* **setNeighbour(Direction, LevelEntity)**: Beállítja irány alapján a szomszédos LevelEntity referenciát a paraméterül kapott referenciára.
* **setMissile(Missile)**: Beállítja a missile referenciát a paraméterül kapott referenciára.

### **4.3.7 Missile**

#### **4.3.7.1 Felelősség**

ONeill hozza létre, és utána arra mozog, amerre ONeill a létrehozásának pillanatában nézett, míg be nem csapódik valahova. Ha olyan falba csapódik, ahol nyílhat portál, akkor ott egy, a lövedéknek megfelelő színű portál nyílik.

#### **4.3.7.2 Ősosztályok**

Nincs ősosztálya

#### **4.3.7.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.7.4 Attribútumok**

* **dir: Direction:** Amerre néz a rakéta.
* **timer: Timer:** Időzítő ami a mozgását irányítja.
* **color: Color:**  A lövedék színe, ami meghatározza milyen színű portált nyit.

#### **4.3.7.5 Metódusok**

* **Missile(Color, Direction):** Konstruktor, amely beállítja a rakéta színét és irányát.
* **move():** Missile mozgásáért felelős függvény.
* **getColor() : Color:** Missile színét lehet lekérdezni ezzel a függvénnyel.

### **4.3.8 ONeill**

#### **4.3.8.1 Felelősség**

A felhasználó által irányítható játékelem leírása.

#### **4.3.8.2 Ősosztályok**

Nincs ősosztálya

#### **4.3.8.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.8.4 Attribútumok**

* **dir: Direction:** Amerre ONeill néz.
* **floor: Floor**: ONeill alatti padlóelem.
* **box: Box**: ONeill doboz referenciája.

#### **4.3.8.5 Metódusok**

* **move():** ONeil mozgását megvalósító függvény.
* **boxing():** ONeil dobozt letételét/felvételét megvalósító függvény.
* **setBox(Box):** Beállítja ONeil doboz referenciáját.
* **setFloor(Floor):** Beállítja ONeil padló referenciáját.
* **kill():** Megöli ONeill-t. Meghívja a Level reset() függvényét.

### **4.3.9 Opener**

#### **4.3.9.1 Felelősség**

Olyan pályaelem leírása, amely kinyitja a hozzátartozó ajtót, ha O’Neill rálép vagy ha doboz kerül rá, bezárja a hozzá tartozó ajtót ha O’Neill le lép róla vagy lekerül róla a doboz.

#### **4.3.9.2 Ősosztályok**

Placeable

#### **4.3.9.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.9.4 Attribútumok**

* **box: Box:** A doboz referenciája ami rákerülhet.
* **door: Door**: A hozzá tartozó ajtó.

#### **4.3.9.5 Metódusok**

* **bool boxEvent(ONeill, Box)**: Ha van már doboz az Opener-en, akkor hamissal tér vissza, nem engedi, hogy új doboz kerüljön rá. Ha még nincs rajta doboz, akkor kinyitja a hozzá tartozó ajtót, elmenti a doboz referenciáját. Ebben az esetben true-val tér vissza.
* **moveEvent(ONeill)**: Ha ONeill rálép a mérleghez tartozó Floor-ra, akkor hívódik meg. Kinyitja az ajtót.
* **setBox(Box):** Beállítja a Box referenciát.
* **setDoor(Door):** Beállítja a Door referenciát.

### **4.3.10 Placable**

#### **4.3.10.1 Felelősség**

Az osztály célja, hogy egységesítsük a Floor típusú objektumra rakható objektumokat.

#### **4.3.10.2 Ősosztályok**

Nincs ősosztálya.

#### **4.3.10.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.10.4 Attribútumok**

Nincs attribútuma.

#### **4.3.10.5 Metódusok**

* **bool boxEvent(ONeill, Box)**: A földre helyezhető dolgok(Box, Opener, Door) között azt az esetet valósítja meg, ha rá akarunk tenni egy dobozt. Visszatérési értéke akkor igaz, ha sikerült lerakni oda a dobozt.
* **moveEvent(ONeill)**: Az a Floor hívja meg, amin az adott Placeable van. Ha a visszatérési értéke true, akkor O’Neill rá tud lépni, ha false, akkor nem.

### **4.3.11 Portal**

#### **4.3.11.1 Felelősség**

A teleportálást elősegítő objektum. Letrőjöttekor létrehoz egy kapcsolatot a másik színű portállal, ha olyan már nyítva van.

#### **4.3.11.2 Ősosztályok**

Nincs ősosztálya.

#### **4.3.11.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.11.4 Attribútumok**

* **bluePortal: Portal:** Referencia a kék portálra.
* **yellowPortal: Portal:** Referencia a sárga portálra.
* **dir: Direction:** az adott portál iránya, hogy merre néz.

#### **4.3.11.5 Metódusok**

* **Portal(Missile, Wall):** Konstruktor. Beállítja a portálok közötti kapcsolatot, a portál színét, az irányát, amerre néz, eltörli a korábbi azonos színű portált.
* **close():** Becsukja a portált.

### **4.3.12 Wall**

#### **4.3.12.1 Felelősség**

A Wall objektum valósítja meg a játéktér határait. Egy olyan objektum amire nem lehet rásétálni, vagy dobozt rárakni. Megakadályozza, hogy bármilyen pályaelem elhagyja a pályát magát, továbbá bizonyos példányain portál nyitható.

#### **4.3.12.2 Ősosztályok**

LevelEntity

#### **4.3.12.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.12.4 Attribútumok**

* **bool portalable:** Azt határozza meg, hogy a falra lőhető-e portál.
* **portal: Portal**: A rajta lévő portál referenciája.

#### **4.3.12.5 Metódusok**

* **bool Wall(bool)**: A fal konstruktora, melyben meg kell adni, rajta nyílhat-e portál.
* **bool moveAction(ONeill)**: Visszatér hamissal, mivel nem lehet rálépni.
* **bool boxAction(ONeill, Box)**: Hamisan tér vissza, mivel nem lehet sem rárakni a dobozt, sem felvenni onnan.
* **bool missileAction(Missile)**: Ha a fal portálosítható, akkor létrehozzuk a portált. A lövedéket pedig minden esetben megsemmisítjük. A visszatérési értéke igaz.

### **4.3.13 ZPM**

#### **4.3.13.1 Felelősség**

Az a feladata, hogy, ha ONeill rálép egy olyan mezőre, amin van, akkor onnan eltűnjön, a ZPM számláló eggyel csökkenjen.

#### **4.3.13.2 Ősosztályok**

Nincs ősosztálya.

#### **4.3.13.3 Interfészek**

Nem valósít meg interfészt.

#### **4.3.13.4 Attribútumok**

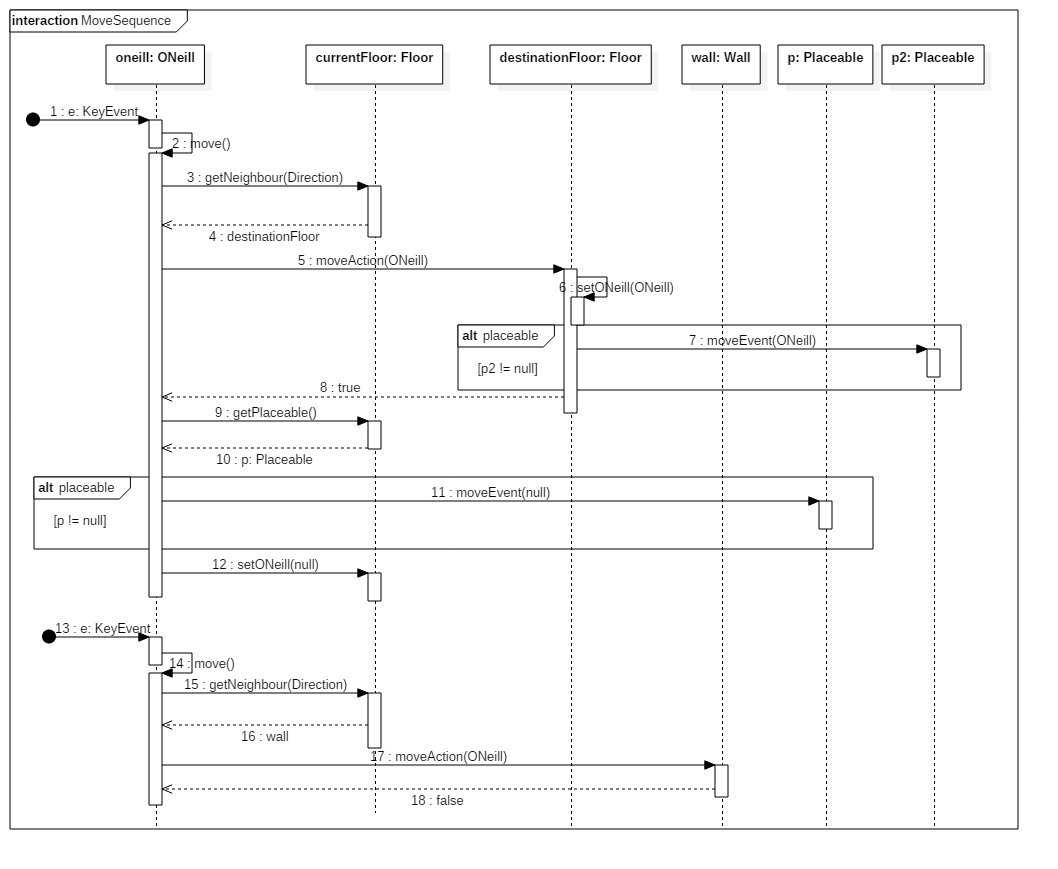
* **level: Level:** a pályára mutató referencia.

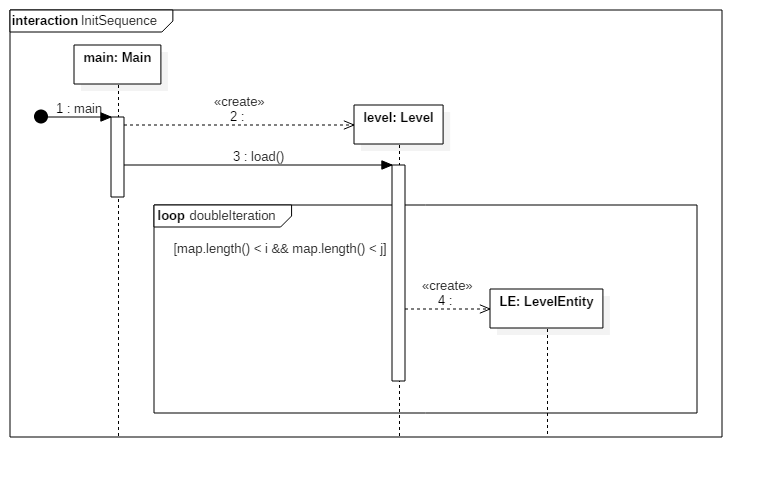
#### **4.3.13.5 Metódusok**

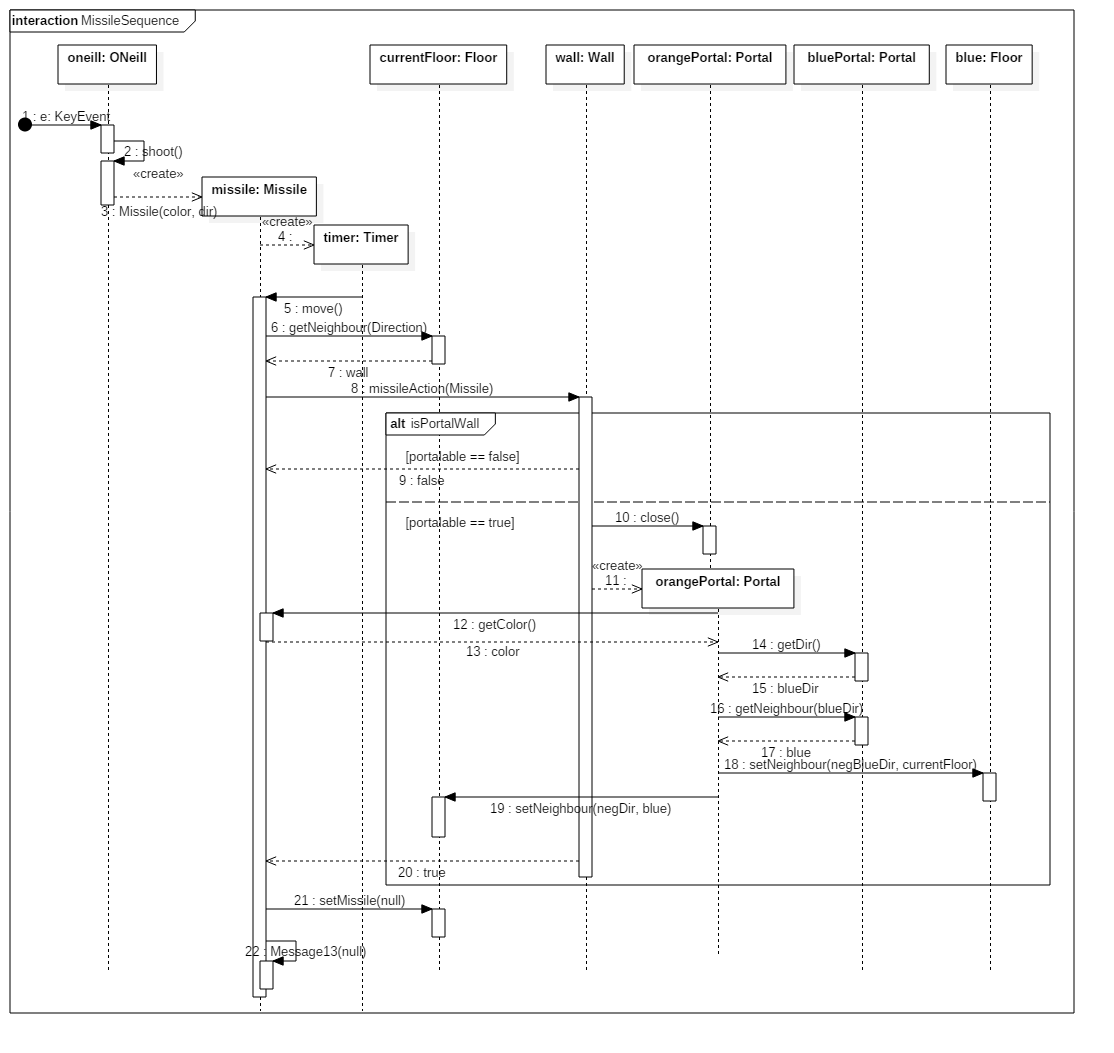
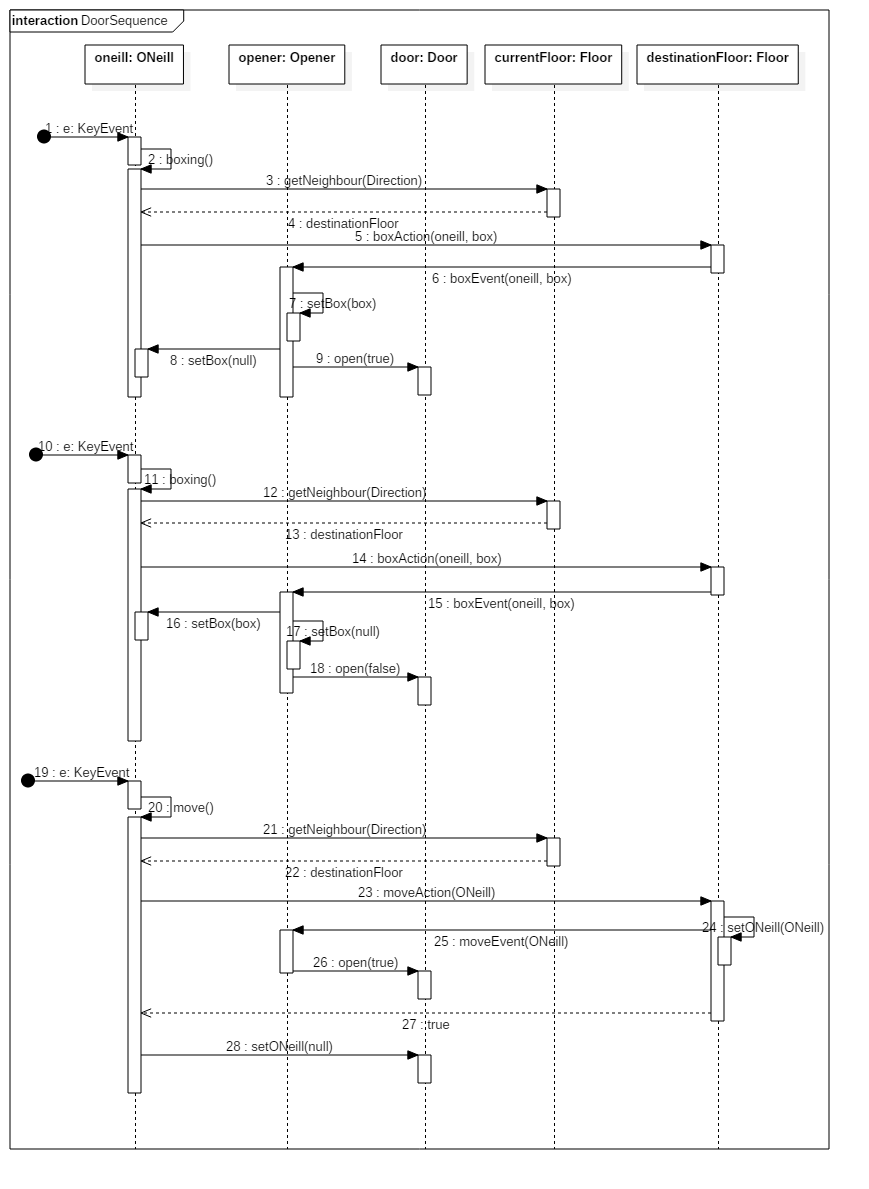
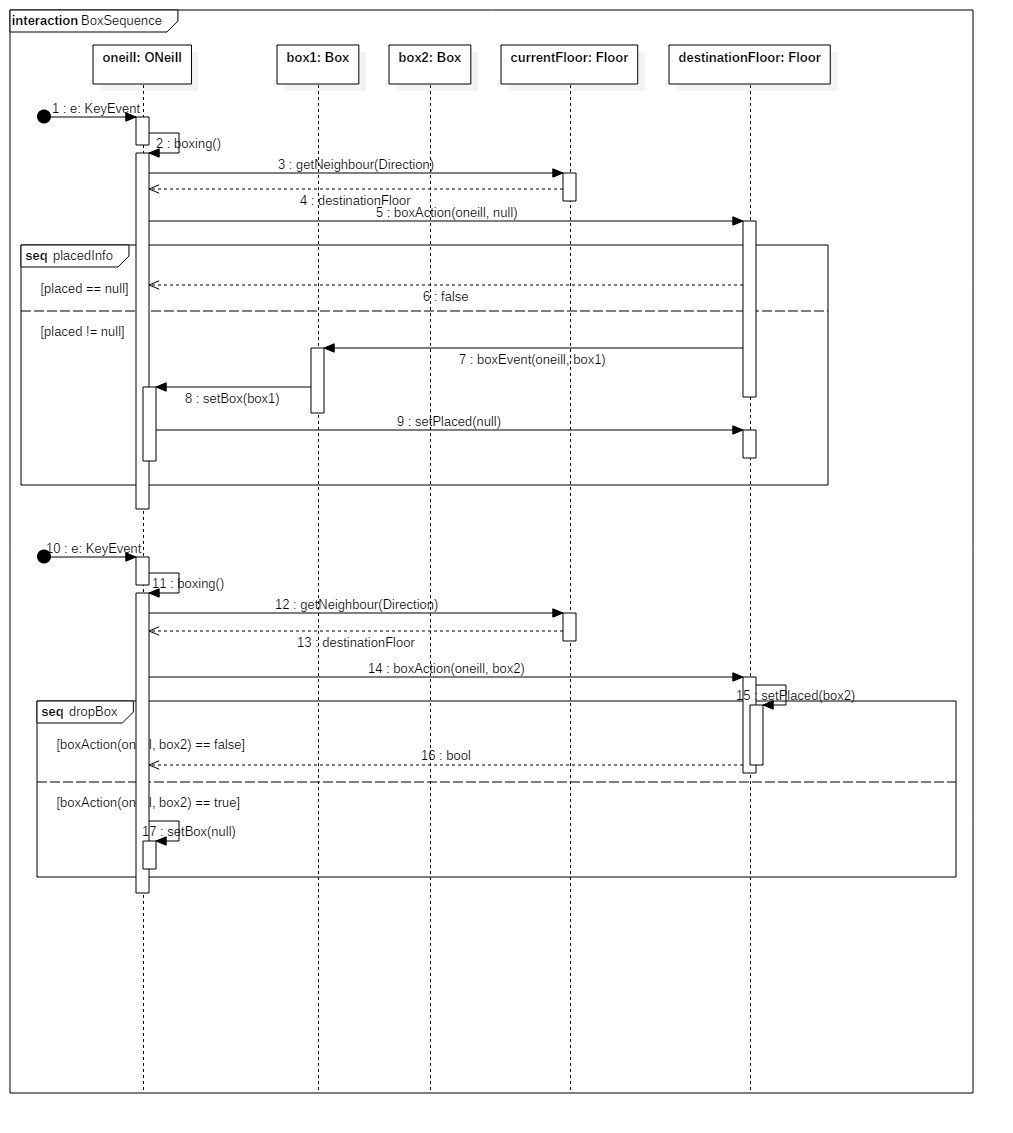
* **ZPM(Level):** A ZPM konstruktora, beállítja a level referenciáját.
* **collect():** Ha rálépünk arra a padlóra, amin ez a ZPM van, akkor hívódik meg ez a függvény. Ez meghívja a Level decreaseZPM() metódusát.

## 

## **4.4 Szekvencia diagramok**







## 

## 

## **4.5 State-chartok**

Egyik osztály sem tartalmaz olyan állapotokat, melyek szemléltetése célszerű lenne state-charttal.

## **4.6 Napló**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2016.03.02.  18:25 | 45 perc | Hegedüs Dániel | Dokumentumok elkészítése a munkához. |
| 2016.03.03.  13:00 | 1 óra 15 perc | Nemeslaki Keve Tuskó Gergely  Rabinovits Jakov | Objektumok felsorolása |
| 2016.03.03.  14:20 | 1 óra  45 perc | Kártyás Bálint Rabinovits Jakov | Osztálydiagram elkezdése |
| 2016.03.03.  20:20 | 1 óra | Rabinovits Jakov | Diagram átültetése StarUML-be |
| 2016.03.05.  14:20 | 50 perc | Nemeslaki Keve | Objektumok leírása |
| 2016.03.05.  16:00 | 2 óra 30 perc | Kártyás Bálint | Osztályok leírása |
| 2016.03.05.  19:45 | 30 perc | Kártyás Bálint | Osztályok leírása |
| 2016.03.05.  17:15 | 3 óra | Nemeslaki Keve Tuskó Gergely  Hegedüs Dániel Rabinovits Jakov | Osztálydiagram készítése |
| 2016. 03.05.  20:50 | 2 óra | Tuskó Gergely  Nemeslaki Keve | Osztálydiagram leírása |
| 2016.03.05.  21:00 | 2 óra | Hegedüs Dániel Rabinovits Jakov | Játékmotor kidolgozása |
| 2016.03.05. 23:15 | 30 perc | Hegedüs Dániel | Osztálydiagram formázása |
| 2016.03.05.  23:25 | 30 perc | Hegedüs Dáneil | InitSequence szekvenciadiagram |
| 2016.03.06.  00:00 | 30 perc | Nemeslaki Keve Rabinovits Jakov | Osztályok módosítása |
| 2016.03.06.  00:00 | 1 óra | Tuskó Gergely | State-chartok leírása |
| 2016.03.06.  00:25 | 25 perc | Rabinovits Jakov  Nemeslaki Keve | State-chartok leírása |
| 2016.03.06.  00:30 | 30 perc | Hegedüs Dániel | Osztálydiagram leírása |
| 2016.03.06. 11:10 | 1 óra | Rabinovits Jakov Hegedüs Dániel | Osztálydiagram módosítása |
| 2016.03.06. 12:40 | 1 óra | Rabinovits Jakov | Osztályok leírása |
| 2016.03.06.  12:15 | 30 perc | Hegedüs Dániel | ResetSequence szekvenciadiagram létrehozása |
| 2016.03.06. 14:00 | 1 óra | Rabinovits Jakov | Osztályleírások kiegészítése |
| 2016.03.06. 15:30 | 1 óra 15 perc | Kártyás Bálint | Osztályok leírásának átnézése, rendezése, számozása |
| 2016.03.06. 15:40 | 25 perc | Tuskó Gergely | Egy State-chart hozzáadása |
| 2016.03.06. 15:45 | 45 perc | Nemeslaki Keve  Hegedüs Dániel | MissileSequence  szekvenciadiagram készítése |
| 2016.03.06. 16:30 | 30 perc | Rabinovits Jakov | State-chartok javítás |
| 2016.03.06. 17:00 | 2 óra | Rabinovits Jakov Hegedüs Dániel | Szekvenciadiagramok készítése |
| 2016.03.06. 19:30 | 1 óra | Kártyás Bálint | Formázás, számozások korrigálása |
| 2016.03.06. 22:00 | 2 óra | Tuskó Gergely | Javítások |

|  |  |  |  |
| --- | --- | --- | --- |
| 2016.03.11.  16:00 | 2 óra | Keve Nemeslaki  Rabinovits Jakov  Tuskó Gergely  Hegedüs Dániel | Az osztálydiagram átdolgozása. Átgondolása. |
| 2016.03.11.  18:00 | 1 óra | Keve Nemeslaki  Rabinovits Jakov  Tuskó Gergely | Az osztálydiagram átdolgozása. Átgondolása. |
| 2016.03.11.  23:10 | 1 óra  20 perc | Hegedüs Dániel | Doksi létrehozása, Github kezelése, osztálydiagram elkezdése. |
| 2016.03.12.  11:30 | 1óra  30 perc | Tuskó Gergely | Funkciók összegyűjtése |
| 2016.03.12.  12:50 | 1 óra  30 perc | Kártyás Bálint  Rabinovits Jakov  Hegedüs Dániel | Osztálydiagram alakítása. |
| 2016.03.12.  14:30 | 3 óra | Kártyás Bálint  Rabinovits Jakov  Hegedüs Dániel  Tuskó Gergely | Osztálydiagram alakítása. |
| 2016.03.12. 16:00 | 1 óra 15 perc | Nemeslaki Keve | Osztálydiagram alakítása. |
| 2016.03.13.  14:30 | 2 óra | Hegedüs Dániel  Kártyás Bálint  Rabinovits Jakov | Osztálydiagram. |
| 2016.03.13.  16:30 | 2 óra | Rabinovits Jakov  Nemeslaki Keve  Tuskó Gergely  Hegedüs Dániel | Osztálydiagram |
| 2016.03.13.  19:15 | 1 óra | Hegedüs Dániel  Tuskó Gergely  Nemeslaki Keve | Szekvencia diagramok kialakítása |
| 2016. 03.13  20:15 | 1 óra | Hegedüs Dániel  Tuskó Gergely | Szekvencia diagram |
| 2016.03.13  21:15 | 45 perc | Hegedüs Dániel | Szekvencia diagram |
| 2016.03.13.  21:30 | 29 perc | Rabinovits Jakov | Szekvencia diagram |
| 2016.03.13.  22:15 | 2 óra  45 perc | Tuskó Gergely | Osztályok leírása |
| 2016.03.13.  23:00 | 2 óra  15 perc | Hegedüs Dániel  Rabinovits Jakov | Szekvencia diagram, osztály diagram |
| 2016.03.14.  6:30 | 3 óra  30 perc | Kártyás Bálint | Objektum katalógus, osztályok leírása |
| 2016.03.14 11:00 | 1 óra | Hegedüs Dániel | Osztályok leírása |
| 2016.03.14 11:00 | 2 óra | Rabinovits Jakov  Tuskó Gergely | Osztályok leírása |
| 2016.03.14 13:00 | 40 perc | Rabinovits Jakov  Tuskó Gergely Nemeslaki Keve | Formázás |