

# Szoftver projekt laboratórium összefoglaló

## 91 – zoldseggkommando

Konzulens:

Tóth Márton

### Csapattagok

Bujdosó Dániel  
Bukta Bence  
Horváth Balázs  
Karácsony Zsolt  
Kelemen András  
Vincze Zsolt

RYOP48  
LLXB5S  
VM83FI  
BQV0X1  
IOYO4K  
JA6WNL

bujdoo@gmail.com  
bbence89@gmail.com  
hobamolvan@gmail.com  
me@karacsonyzsolt.hu  
andrew.kelemen@gmail.com  
vincze001@gmail.com

2016.05.19

## Tartalomjegyzék

Tartalomjegyzék.....	2
Ábrajegyzék .....	10
2. Követelmény, projekt, funkcionalitás .....	13
2.1 Bevezetés .....	13
2.1.1 Cél .....	13
2.1.2 Szakterület.....	13
2.1.3 Definíciók, rövidítések .....	13
2.1.4 Hivatkozások .....	13
2.1.5 Összefoglalás.....	14
2.2 Áttekintés.....	14
2.2.1 Általános áttekintés .....	14
2.2.2 Funkciók.....	14
2.2.3 Felhasználók.....	15
2.2.4 Korlátozások.....	16
2.2.5 Feltételezések, kapcsolatok .....	16
2.3 Követelmények .....	16
2.3.1 Funkcionális követelmények.....	16
2.3.2 Erőforrásokkal kapcsolatos követelmények.....	18
2.3.3 Átadással kapcsolatos követelmények .....	18
2.3.4 Egyéb nem funkcionális követelmények.....	19
2.4 Lényeges use-case-ek .....	19
2.4.1 Use-case leírások.....	19
2.4.2 Use-case diagram .....	21
2.5 Szótár .....	21
2.6 Projekt terv .....	22
2.6.1 Csapattagok és feladatkörök.....	22
2.6.2 Kommunikáció a fejlesztés során.....	22
2.6.3 Alkalmazott eszközök a fejlesztés során .....	23
2.6.4 Mérföldkövek és határidők.....	23
2.7 Napló .....	25
3. Analízis modell kidolgozása .....	27
3.1 Objektum katalógus.....	27
3.1.1 Game .....	27
3.1.2 GamePanel .....	27

3.1.3	GameStateManager .....	27
3.1.4	GameState .....	27
3.1.5	TileMap .....	27
3.1.6	Tile .....	27
3.1.7	MapObject .....	28
3.1.8	Entity .....	28
3.1.9	Wormhole .....	28
3.1.10	Stargate .....	28
3.2	Statikus struktúra diagramok .....	29
3.3	Osztályok leírása .....	30
3.3.1	Box .....	30
3.3.2	Door .....	30
3.3.3	Entity .....	30
3.3.4	Floor .....	31
3.3.5	Game .....	31
3.3.6	GamePanel .....	32
3.3.7	GameState .....	32
3.3.8	GameStateManager .....	33
3.3.9	Gap .....	34
3.3.10	LevelState .....	34
3.3.11	MapObject .....	35
3.3.12	MenuState .....	35
3.3.13	ONeill .....	35
3.3.14	Projectile .....	36
3.3.15	Scale .....	36
3.3.16	Stargate .....	37
3.3.17	Tile .....	37
3.3.18	TileMap .....	37
3.3.19	Wall .....	38
3.3.20	Wormhole .....	39
3.3.21	ZPM .....	39
3.3.22	Szekvencia diagramok .....	40
3.4	State-chartok .....	48
3.4.1	Játékállapotok .....	48
3.4.2	O'Neill állapotai .....	48
3.4.3	Féregjárat állapotai .....	49
3.5	Ajtó állapotai .....	49

3.6	Napló .....	50
4.	Analízis modell kidolgozása 2 .....	53
4.1	Objektum katalógus .....	53
4.1.1	Game .....	53
4.1.2	GamePanel .....	53
4.1.3	GameStateManager .....	53
4.1.4	GameState .....	53
4.1.5	TileMap .....	53
4.1.6	Tile .....	53
4.1.7	MapObject .....	53
4.1.8	Entity .....	54
4.1.9	Wormhole .....	54
4.1.10	Stargate .....	54
4.2	Statikus struktúra diagramok .....	55
4.3	Osztályok leírása .....	56
4.3.1	Box .....	56
4.3.2	Door .....	56
4.3.3	Entity .....	56
4.3.4	Floor .....	57
4.3.5	Game .....	58
4.3.6	GamePanel .....	58
4.3.7	GameState .....	59
4.3.8	GameStateManager .....	59
4.3.9	Gap .....	60
4.3.10	LevelState .....	60
4.3.11	MapObject .....	61
4.3.12	MenuState .....	61
4.3.13	ONeill .....	61
4.3.14	Projectile .....	62
4.3.15	Scale .....	62
4.3.16	Stargate .....	63
4.3.17	Tile .....	63
4.3.18	TileMap .....	64
4.3.19	Wall .....	64
4.3.20	Wormhole .....	65
4.3.21	ZPM .....	65
4.4	Szekvencia diagramok .....	66

4.5	State-chartok .....	76
4.5.1	Játékállapotok .....	76
4.5.2	O'Neill állapotai .....	76
4.5.3	Féregjárat állapotai .....	76
4.5.4	Ajtó állapotai .....	76
4.6	Napló .....	77
5.	Szkeleton tervezése .....	78
5.1	A szkeleton modell valóságos use-case-ei .....	78
5.1.1	Use-case diagram .....	78
5.1.2	Use-case leírások .....	79
5.2	A szkeleton kezelői felületének terve, dialógusok .....	82
5.3	Szekvencia diagramok a belső működésre .....	83
5.4	Kommunikációs diagramok .....	95
5.5	Napló .....	107
6.	Szkeleton beadás .....	108
6.1	Fordítási és futtatási útmutató .....	108
6.1.1	Fájllista .....	108
6.1.2	Fordítás .....	108
6.1.3	Futtatás .....	108
6.2	Értékelés .....	109
6.3	Napló .....	109
7.	Prototípus koncepciója .....	110
7.0	Osztálydiagram .....	110
	Szekvencia diagramok .....	111
7.1	Prototípus interface-definíciója .....	115
7.1.1	Az interfész általános leírása .....	115
7.1.2	Bemeneti nyelv .....	115
7.1.3	Kimeneti nyelv .....	116
7.2	Összes részletes use-case .....	120
7.3	Tesztelési terv .....	123
7.4	Tesztelést támogató segéd- és fordítóprogramok specifikálása .....	124
7.5	Napló .....	125
8.	Részletes tervek .....	126
8.1	Osztályok és metódusok tervei .....	126
8.1.1	Game .....	126
8.1.2	GamePanel .....	126
8.1.3	GameState .....	127

8.1.4	GameStateManager .....	127
8.1.5	MenuState.....	128
8.1.6	LevelState.....	129
8.1.7	TileMap .....	129
8.1.8	Tile .....	130
8.1.9	Floor .....	131
8.1.10	Gap .....	132
8.1.11	Wall .....	132
8.1.12	MapObject.....	133
8.1.13	Box .....	133
8.1.14	ZPM.....	134
8.1.15	Door.....	134
8.1.16	Scale .....	135
8.1.17	Wormhole.....	135
8.1.18	Stargate.....	136
8.1.19	Entity .....	136
8.1.20	Replicator .....	137
8.1.21	Projectile.....	137
8.1.22	Player.....	138
8.1.23	O'Neill.....	139
8.1.24	Jaffa .....	139
8.2	A tesztek részletes tervei, leírásuk a teszt nyelven .....	140
8.2.1	Teszteset1: A játékos padló csempére lép .....	140
8.2.2	Teszteset2: A játékos szakadékba lép .....	140
8.2.3	Teszteset3: A játékos falba próbál lépni .....	141
8.2.4	Teszteset4: A játékos mérlegre lép.....	141
8.2.5	Teszteset5: A játékos zárt ajtónak nekimegy .....	142
8.2.6	Teszteset6: A játékos átmegy egy nyitott ajtón.....	142
8.2.7	Teszteset7: A játékos dobozra próbál lépni.....	143
8.2.8	Teszteset8: A játékos ZPM-et vesz fel. ....	143
8.2.9	Teszteset9: A játékos teleportál.....	144
8.2.10	Teszteset10: A játékos dobozt vesz fel.....	145
8.2.11	Teszteset11: A játékos dobozt tesz le.....	145
8.2.12	Teszteset12: A játékos szakadékba lép .....	146
8.2.13	Teszteset13: A játékos dobozt tesz egy szakadékba .....	147
8.2.14	Teszteset14: A játékos dobozt próbál tenni fal csempére .....	148
8.2.15	Teszteset15: A játékos dobozt próbál tenni ajtóra.....	148

8.2.16	Teszteset16: A játékos dobozt próbál letenni, míg nincs nála .....	149
8.2.17	Teszteset17: A játékos dobozt tesz le egy ZPM-re .....	150
8.2.18	Teszteset18: A játékos lövedéket lő ki .....	151
8.2.19	Teszteset19: A lövedék egy szakadék fölött mozog .....	151
8.2.20	Teszteset20: A lövedék falnak ütközik .....	152
8.2.21	Teszteset21: A lövedék doboznak ütközik .....	152
8.2.22	Teszteset22: A lövedék speciális falba ütközik .....	153
8.2.23	Teszteset23: A lövedék replikátorba ütközik .....	153
8.2.24	Teszteset24: A lövedék zárt ajtóba ütközik .....	154
8.2.25	Teszteset25: A lövedék nyitott ajtón át halad .....	154
8.2.26	Teszteset26: A lövedék ZPM-nek ütközik .....	155
8.2.27	Teszteset27: A replikátor szakadékba mozog .....	156
8.3	A tesztelést támogató programok tervei .....	156
8.4	Napló .....	157
10.	Prototípus beadása .....	158
10.1	Fordítási és futtatási útmutató .....	158
10.1.1	Fájllista .....	158
10.1.2	Fordítás .....	158
10.1.3	Futtatás .....	158
10.2	Tesztek jegyzőkönyvei .....	159
10.2.1	Teszteset1 .....	159
10.2.2	Teszteset2 .....	159
10.2.3	Teszteset3 .....	159
10.2.4	Teszteset4 .....	159
10.2.5	Teszteset5 .....	159
10.2.6	Teszteset6 .....	159
10.2.7	Teszteset7 .....	159
10.2.8	Teszteset8 .....	159
10.2.9	Teszteset19 .....	160
10.2.10	Teszteset10 .....	160
10.2.11	Teszteset11 .....	160
10.2.12	Teszteset12 .....	160
10.2.13	Teszteset13 .....	160
10.2.14	Teszteset14 .....	160
10.2.15	Teszteset15 .....	161
10.2.16	Teszteset16 .....	161
10.2.17	Teszteset17 .....	161

10.2.18	Teszteset18 .....	161
10.2.19	Teszteset19 .....	161
10.2.20	Teszteset20 .....	161
10.2.21	Teszteset21 .....	161
10.2.22	Teszteset22 .....	161
10.2.23	Teszteset23 .....	161
10.2.24	Teszteset24 .....	162
10.2.25	Teszteset25 .....	162
10.2.26	Teszteset26 .....	162
10.2.27	Teszteset27 .....	162
10.3	Értékelés.....	162
10.4	Napló.....	163
11.	Grafikus felület specifikációja.....	165
11.1	A grafikus interfész.....	165
11.1.1	Felhasznált képek .....	165
11.1.2	Minta összeállítás .....	167
11.2	A grafikus rendszer architektúrája .....	167
11.2.1	A felület működési elve.....	167
11.2.2	A felület osztály-struktúrája .....	167
11.3	A grafikus objektumok felsorolása .....	168
11.3.1	GamePanel .....	168
11.3.2	GameStateManager .....	168
11.3.3	LevelState.....	169
11.3.4	TileMap .....	169
11.3.5	Tile .....	169
11.3.6	Wall .....	170
11.3.7	Entity .....	170
11.3.8	Player.....	171
11.3.9	Wormhole.....	171
11.3.10	Stargate .....	172
11.3.11	Replicator.....	172
11.4	Kapcsolat az alkalmazói rendszerrel.....	173
11.5	Napló.....	175
13.	Grafikus változat beadása.....	176
13.1	Fordítási és futtatási útmutató .....	176
13.1.1	Fájllista .....	176
13.1.2	Fordítás és telepítés .....	179



13.1.3	Futtatás .....	179
13.2	Értékelés.....	179
13.3	Napló.....	180
14.	Összefoglalás.....	181
14.1	Projektösszegzés .....	181
14.2	Kérdések, válaszok .....	181
14.2.1	Bujdosó Dániel.....	181
14.2.2	Bukta Bence .....	182
14.2.3	Horváth Balázs .....	182
14.2.4	Karácsony Zsolt.....	183
14.2.5	Kelemen András .....	183
14.2.6	Vincze Zsolt .....	184

## Ábrajegyzék

2.1. ábra Use-case diagram .....	21
3.1. ábra Osztálydiagram.....	29
3.2. ábra GamePanel inicializálása.....	40
3.3. ábra Ablak inicializálása .....	40
3.4. ábra Játékindítás .....	41
3.5. ábra Főmenü kezelése .....	41
3.6. ábra Mozgás .....	42
3.7. ábra Falhoz érkezés és teleportálás kezelése .....	43
3.8. ábra Lövés .....	44
3.9. ábra Doboz felvétele.....	44
3.10. ábra Doboz lerakása .....	45
3.11. ábra Padlón mozgás kezelése .....	45
3.12. ábra Szakadék kezelése .....	46
3.13. ábra Mérleg kezelése .....	46
3.14. ábra Speciális fal kezelése.....	47
3.15. ábra Normál falba lőtt lövedék kezelése .....	47
3.16. ábra ZPM felvétel kezelése .....	48
3.17. ábra Menü állapotváltozása .....	48
3.18. ábra O'Neill állapotváltozása.....	48
3.19. ábra Féregjárat állapotváltozása .....	49
3.20. ábra Ajtó állapotváltozása .....	49
4.1. ábra Osztálydiagram.....	55
4.2. ábra Játéklablak beállítása .....	66
4.3. ábra Játékindítás .....	66
4.4. ábra GamePanel inicializálása.....	67
4.5. ábra Főmenü kezelése .....	67
4.6. ábra Mozgás kezelése.....	68
4.7. ábra Mérleg és ajtó kezelése.....	69
4.8. ábra Padlóra lépés kezelése .....	70
4.9. ábra Mérlegre lépés illetve ZPM kezelése .....	70
4.10. ábra Falra lépés kezelése .....	71
4.11. ábra Doboz felvétel kezelése.....	72
4.12. ábra Doboz lerakás kezelése .....	73
4.13. ábra Lövés kezelése.....	73
4.14. ábra Csillagkapuk kezelése .....	74
4.15. ábra Speciális falba érkezés kezelése .....	74
4.16. ábra Szakadék kezelése .....	75
4.17. ábra Padlóra lépés kezelése .....	75
4.18. ábra Lövedék falba ütközése .....	75
4.19. ábra A játék állapotváltozásai .....	76
4.20. ábra O'Neill állapotváltozása.....	76
4.21. ábra A féregjárat állapotváltozása .....	76
4.22. ábra Ajtó állapotváltozása .....	76
5.1. ábra Szkeleton use-case-ek.....	78
5.2. ábra Doboz felvétele.....	83
5.3. ábra Doboz lerakása csempére .....	84
5.4. ábra Doboz lerakása dobozra .....	84
5.5. ábra Doboz lerakása falba .....	85

5.6. ábra Doboz lerakása mérlegre .....	85
5.7. ábra Doboz lerakása szakadékba .....	86
5.8. ábra Doboz lerakása ZPM-re.....	86
5.9. ábra Dobozra lépés .....	87
5.10. ábra Falba lépés .....	87
5.11. ábra Lövedék doboznak ütközése .....	87
5.12. ábra Lövedék falnak csapódik.....	88
5.13. ábra Lövedék nyitott ajtóhoz érkezik .....	88
5.14. ábra Lövedék haladása a csempén .....	89
5.15. ábra Lövedék speciális falnak csapódik .....	89
5.16. ábra Lövedék szakadék felett áthalad.....	90
5.17. ábra Lövedék zárt ajtónak csapódik .....	90
5.18. ábra Lövedék ZPM-nek ütközik.....	91
5.19. ábra Mérlegre lépés és ajtónyitás .....	91
5.20. ábra Nyitott ajtóba lépés.....	92
5.21. ábra Padlóra lépés.....	93
5.22. ábra Szakadékba lépés.....	93
5.23. ábra Zárt ajtóbalépés .....	94
5.24. ábra ZPM-re lépés .....	94
5.25. ábra Teleportálás .....	95
5.26. ábra Doboz felvétele.....	95
5.27. ábra Doboz lerakása csempére .....	96
5.28. ábra Doboz lerakása dobozra .....	96
5.29. ábra Doboz lerakása falba .....	96
5.30. ábra Doboz lerakása mérlegre .....	97
5.31. ábra Doboz lerakása szakadékba .....	97
5.32. ábra Doboz lerakása ZPM-re.....	98
5.33. ábra Dobozra lépés .....	98
5.34. ábra Falba lépés .....	99
5.35. ábra Lövedék dobozra lép .....	99
5.36. ábra Lövedék falnak csapódik.....	99
5.37. ábra Lövedék ajtóhoz érkezik .....	100
5.38. ábra Lövedék padló fölött mozog.....	100
5.39. ábra Lövedék speciális falnak csapódik .....	101
5.40. ábra Lövedék szakadék felett áthalad.....	101
5.41. ábra Lövedék zárt ajtónak csapódik .....	102
5.42. ábra Lövedék ZPM-nek ütközik.....	102
5.43. ábra Mérlegre lépés ajtónyitással .....	103
5.44. ábra Nyitott ajtóba lépés.....	103
5.45. ábra Padlóra lépés.....	104
5.46. ábra Szakadékba lépés.....	104
5.47. ábra Zárt ajtóba lépés .....	105
5.48. ábra ZPM-re lépés .....	105
5.49. ábra Teleportálás .....	106
7.1. ábra Megváltozott osztálydiagram .....	110
7.2. ábra Lövedék padló felett repülése.....	111
7.3. ábra Replikátor szakadékba lép .....	111
7.4. ábra ZPM felvétel.....	112
7.5. ábra Mérlegre lépés ajtónyitással .....	113
7.6. ábra Doboz lerakása csempére .....	114

7.7. ábra Doboz lerakása mérlegre .....	115
11.1. ábra A játék indító képernyője .....	165
11.2. ábra Padló képe .....	165
11.3. ábra Zárt ajtó .....	166
11.4. ábra Nyitott ajtó.....	166
11.5. ábra Lövedékek .....	166
11.6. ábra Doboz képe.....	166
11.7. ábra O'Neill és Jaffa képe.....	166
11.8. ábra Replikátor képe.....	166
11.9. ábra Aktív és inaktív mérleg képe .....	166
11.10. ábra Csillagkapuk képei .....	166
11.11. ábra Fal képe .....	167
11.12. ábra Speciális fal képe.....	167
11.13. ábra Szakadék képe .....	167
11.14. ábra ZPM képe .....	167
11.15. ábra Játék mintaösszeállítása.....	167
11.16. ábra Grafikus felület működése.....	173
11.17. ábra Menü kirajzolása .....	173
11.18. ábra Csempék kirajzolása.....	173
11.19. ábra Tilemap kirajzolása .....	174
11.20. ábra Csempék inicializálása .....	174

## 2. Követelmény, projekt, funkcionalitás

### 2.1 Bevezetés

#### 2.1.1 Cél

A játék specifikációjának elkészítése a megrendelő számára, a projekttel szemben felállított követelmények körül határolása és az alapvető funkcionalitás ismertetése. Ezeket figyelembe véve a projekt menetének meghatározása, melyeket a fejlesztés során követni fogunk.

#### 2.1.2 Szakterület

A feladatkiírás szerint az elkészítendő szoftver egy számítógépes játék, melynek az azt használó játékos szórakoztatása az alapvető célja.

#### 2.1.3 Definíciók, rövidítések

- **BME** - A Budapesti Műszaki Egyetem rövidítése.
- **Eclipse** - Főként Java nyelvhez kialakított fejlesztőkörnyezet.
- **fejlesztőkörnyezet** - Fejlesztési munkát lehetővé tevő, segítő számítógépes program vagy programok összessége.
- **Git** - Verziókezelő rendszer.
- **GitHub** - A Git verziókezelő rendszerre épülő online szolgáltatás.
- **Google Groups** - A Google egy szolgáltatása levelezőlisták kezelésére.
- **HSZK** - Hallgatói Számítógép Központ rövidítése.
- **JDK** - Java Development Kit rövidítése; Java programozási nyelv fejlesztői környezete.
- **JRE** - Java Runtime Environment rövidítése; Java programozási nyelven írt programok használatához szükséges.
- **prototípus** - A program olyan állapota, amikor a funkciók már működnek, de még nem teljes, a grafikus felület még nincs kész.
- **szkeleton** - A program olyan állapota, mikor a felépítése, váza készen van, de az egyes funkciók még nem működnek.
- **szoftver** - Számítógépen futtatható program.
- **Trello** - Online projektmenedzsment szoftver.
- **use-case** - A felhasználó(k) és az adott rendszer közti interakciók leírása.
- **Visual Paradigm** - Diagramok szerkesztését segítő szoftver.

#### 2.1.4 Hivatkozások

- Tárgyhonlap: <https://www.iit.bme.hu/~projlabs>
- GitHub: <https://github.com/>
- Trello: <https://trello.com/>
- Eclipse: <https://eclipse.org/>
- Google Drive: <https://drive.google.com/>
- Google Groups: <https://groups.google.com/>
- Visual Paradigm: <http://www.visual-paradigm.com/>
- Microsoft Office 2016: [https://en.wikipedia.org/wiki/Microsoft\\_Office\\_2016](https://en.wikipedia.org/wiki/Microsoft_Office_2016)
- TeamSpeak: <https://www.teamspeak.com/teamspeak3>

### 2.1.5 Összefoglalás

- **Áttekintés:** A szoftver bemutatása nagyvonalakban, funkcióinak és korlátozásainak áttekintése.
- **Követelmények:** A projekttel kapcsolatos követelmények áttekintése, külön részletezve a funkcionális, erőforrásokkal és átadással kapcsolatos és egyéb nem funkcionális követelményeket.
- **Lényeges use-case-ek:** A lényegesebb use-case-ek felsorolása, ábrázolása.
- **Szótár:** A projekttel kapcsolatos egyedi szótár, mely tartalmazza a szoftverrel kapcsolatos és annak fejlesztése során használt nem hétköznapi szavak értelmezését, leírását.
- **Projekt terv:** A projekt végrehajtásának tervét tartalmazza feladatokra lebontva és az egyes feladatokhoz határidőket, felelősöket rendelve.
- **Napló:** Az elvégzett feladatokat tartalmazó napló, a ráfordított idő és a feladat elvégzését végrehajtó személyek megjelölésével.

## 2.2 Áttekintés

### 2.2.1 Általános áttekintés

A fejlesztendő szoftver egy játékprogram. A játék célja, hogy a játékos a karakterét irányítva, egy labirintusban manőverezve megszerezze a pályán található összes ZPM-et, ezzel teljesítve az adott pályát.

A megvalósításhoz a következő alrendszerek szükségesek:

- **Grafikus felület:** a játéktér megjelenéséért felelő alrendszer
- **Input kezelő:** a felhasználó interakcióit kezelő alrendszer
- **Pálya betöltő:** az aktuális pályát olvassa be egy megfelelő fájlból
- **Pálya kezelő:** a világban lévő objektumok tulajdonságainak változtatása egymástól és a felhasználói inputtól függően

A program nem igényel hálózati kapcsolatot.

### 2.2.2 Funkciók

Anubis, a gonosz goa'uld le akarja igázni a Földet. Ennek megakadályozásához O'Neill ezredesnek össze kell gyűjtenie a Földön található összes ZPM-et, hogy az atlantiszi űrhajó energiaellátását meg lehessen oldani, és Anubis támadását vissza lehessen verni.

A játék egy két dimenziós, felülnézetes logikai játék, ahol a játékos egyetlen célja, hogy az adott pályán található összes ZPM modul összeszedje a pályán található elemek (ajtó, doboz, mérleg) illetve a fegyver segítségével.

A pálya, egy csempékből álló labirintus, aminek előre meghatározott méretei illetve határai vannak így a játékos nem képes elhagyni azt. Minden csempe egy egység méretű. A játékos a labirintus egy előre meghatározott pontján kezd. A labirintus teljes egészében látható, így a felhasználó tisztában van annak felépítésével.

A ZPM modulok a labirintusban vannak elrejtve. Egyes helyek közvetlenül nem megközelíthetők, így valamilyen más módszer segítségével kell oda eljutni. Szerencsére O'Neill ezredesnek van egy speciális fegyvere, amellyel főregiáratokat tud nyitni a labirintus

különböző pontjai között. A fegyver kétféle töltényt tud egyenes vonalban kilőni: sárgát és kéket. A töltények sosem fogynak el. A labirintus egyes falai speciálisak, és amennyiben egy lövedék ilyen speciális falba csapódik, akkor a falon a lövedék színének megfelelő csillagkapu nyílik. Az ettől eltérő objektumokkal érintkezve, (doboz, ajtó, nem speciális fal) nem történik változás. Egy adott színű csillagkapuból mindig csak legfeljebb egy létezhet, így ha egy újabb ugyanolyan színű csillagkapu nyílik, akkor a régi eltűnik. Amikor mindkét színű csillagkapu nyitva van, közöttük egy féregjárat alakul ki, amely mindkét irányban átjárható. A féregjáraton O'Neill ezredes is át tud menni, de akár tárgyakat is átvihet rajta.

A ZPM modulok elszórtan helyezkednek el a labirintusban, az ezredes feladata, hogy ezeket a ZPM modulokat mind összegyűjtse. Amint ez megtörtént, a játékos sikeresen teljesíti az adott pályát. A dobozok is elszórtan helyezkednek el a labirintusban. Az ezredes fel tud venni egy dobozt, ha az közvetlenül ön maga előtt található, majd el tudja vinni azt a labirintus más részeibe (akár egy féregjáraton keresztül is), végül le tudja tenni a dobozt saját maga elé.

Az ajtók alaphelyzetben zárva vannak és rajtuk semmi nem tud áthatolni, vagyis falként viselkednek. Minden ajtóhoz tartozik egy mérleg, amely nem feltétlenül az ajtó közelében található (lehet akár a labirintus egy távoli pontján is). A mérlegek nem mozgathatók, azonban amikor egy mérlegre súly kerül (egy doboz vagy maga az ezredes), akkor a mérleghez tartozó ajtó kinyílik, és mindaddig nyitva marad, amíg a súly a mérleget nyomja. Ha a súly lekerül a mérlegről, az ajtó visszazáródik.

A labirintusban lehetnek szakadékok, amelyeken az ezredes közvetlenül nem tud átmenni, de át tud lőni felettük. Ha az ezredes szakadékba esik, akkor meghal. Ha egy tárgy szakadékba esik, akkor megsemmisül.

A játékos O'Neill ezredest irányítja, más szavakkal a játékos avatárja az ezredes akit irányíthat a játékon belül. O'Neill öt különböző cselekedetre képes, ezek pedig a következők:

- **Forgás:** A játékos forgathatja a karaktert így az más irányba néz, illetve más irányba lö.
- **Mozgás:** A játékos mozgathatja is az ezredest, egyszerre 1 egységnyi mozoghat egy felhasználó által megadott irányba. Csak a megfelelő csempékre megengedett a mozgás (falba nem lehet besétálni) viszont a szakadékba bele lehet mozogni ami O'Neill (és a Föld) csúfos végét jelenti.
- **Lövés:** A karakter kilő egy lövedéket abba az irányba amerre éppen néz. Ekkor csillagkapu nyílik, a lövedék színétől függően sárga vagy kék. Ha egyszerre van sárga illetve kék kapu is nyitva, közöttük egy féregjárat nyílik.
- **Dobozfelvétel:** Ha a karakter éppen egy doboz melletti csempén áll és a doboz irányába néz akkor azt felveheti, amit ilyenkor a karakter maga előtt visz és letételig nála marad.
- **Dobozletétel:** Ha a karakternél van doboz, akkor azt a karakter leteszi maga elé (amennyiben ez lehetséges), viszont ha a dobozt szakadékra próbálja rátenni, a doboz megsemmisül.

### 2.2.3 Felhasználók

A szoftver felhasználóinak nem kell rendelkezniük semmilyen különleges előképzettséggel, csupán a billentyűzet kezelésével kell tisztában lenniük.

### 2.2.4 Korlátozások

- A szoftver kizárólag Java nyelven, annak is a Standard Edition jelzéssel ellátott eszközkészletével készülhet.
- A kész szoftvernek futnia kell a BME Hallgatói Számítógép Központ (továbbiakban HSZK) gépein, amelyeken a Java futtatókörnyezet (JRE) 1.8-as verziója áll rendelkezésre
- Az előző pontból következően legfeljebb az 1.8-as verziójú Java Development Kit-ben szereplő Java eszközök használhatóak

### 2.2.5 Feltételezések, kapcsolatok

A játék futtatásához a fent felsoroltakon kívül a következő környezet szükséges:

- Számítógép
- Operációs rendszer: legalább Windows 7
- Monitor
- Billentyűzet

## 2.3 Követelmények

### 2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
REQ01	A játékot el lehet indítani	Bemutató	Fontos	Csapat	Indítás	
REQ02	A játékból ki lehet lépni	Bemutató	Fontos	Csapat	Kilépés	
REQ03	A pálya egy labirintus	Bemutató	Alapvető	Megrendelő		
REQ04	A labirintus nem minden része elérhető el közvetlenül	Bemutató	Fontos	Megrendelő		
REQ05	A labirintusban ajtók vannak	Bemutató	Alapvető	Megrendelő		
REQ06	A labirintusban ZPM-ek vannak	Bemutató	Alapvető	Megrendelő		
REQ07	A labirintusban dobozok vannak	Bemutató	Alapvető	Megrendelő		
REQ08	A labirintusban mérlegek vannak	Bemutató	Alapvető	Megrendelő		
REQ09	Minden ajtóhoz tartozik egy mérleg	Bemutató	Alapvető	Megrendelő		



REQ10	Az ajtók alapesetben zárva vannak	Bemutató	Alapvető	Megrendelő		
REQ11	A zárt ajtó falként viselkedik	Bemutató	Alapvető	Megrendelő	Lépés	
REQ12	A mérlegek ajtókat nyitnak	Bemutató	Alapvető	Megrendelő		
REQ13	A mérlegek nem mozgathatók	Bemutató	Alapvető	Megrendelő		
REQ14	Dobozt lehet helyezni a mérlegre	Bemutató	Alapvető	Megrendelő	Doboz lerakás	
REQ15	Rá lehet állni a mérlegre	Bemutató	Alapvető	Megrendelő	Mérlegre lépés	
REQ16	Mozogni lehet a labirintusban	Bemutató	Alapvető	Megrendelő	Lépés, Forgás	
REQ17	A játékos tud forogni	Bemutató	Opcionális	Csapat	Forgás	
REQ18	Lövedéket lehet lőni bizonyos falakra	Bemutató	Alapvető	Megrendelő	Lövés	
REQ19	Narancssárga és kék töltény van	Bemutató	Alapvető	Megrendelő	Kék csillagkapu, Narancs csillagkapu	
REQ20	Végtelen lőszer van	Ellenőrzés	Alapvető	Megrendelő	Lövés	
REQ21	Egyszerre csak egy csillagkapu létezik ugyanolyan színnel	Bemutató	Alapvető	Megrendelő	Kék csillagkapu, Narancs csillagkapu	
REQ22	ZPM-eket lehet felvenni	Bemutató	Alapvető	Megrendelő	ZPM felvétel	
REQ23	Dobozokat lehet felemelni és letenni	Bemutató	Alapvető	Megrendelő	Doboz felvétel, Doboz lerakás	
REQ24	Csak az ezredes előtt lévő doboz vehető fel	Bemutató	Fontos	Megrendelő	Doboz felvétel	
REQ25	Ha egy doboz szakadékba esik, megsemmisül	Bemutató	Fontos	Megrendelő	Doboz lerakás	
REQ26	A labirintusban lehetnek szakadékok	Bemutató	Fontos	Megrendelő		

REQ27	Szakadékbá lépve meghalunk	Bemutató	Alapvető	Megrendelő	Halál	
REQ28	A főregjékerek mindkét irányba átjárhatók	Bemutató	Fontos	Megrendelő	Teleportálás	

### 2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
RES01	Git	Nincs	Fontos	Csapat	Elosztott verziókezelő
RES02	GitHub	Nincs	Fontos	Csapat	Git alapú online rendszer
RES03	JRE 8	Bemutató	Alapvető	Megrendelő	
RES04	Visual Paradigm	Nincs	Opcionális	Csapat	UML modellező
RES05	Eclipse	Nincs	Opcionális	Csapat	Java IDE
RES06	HSZK-ban található vagy annál erősebb számítógép	Bemutató	Alapvető	Megrendelő	
RES07	Monitor	Nincs	Alapvető	Csapat	
RES08	Billentyűzet	Nincs	Alapvető	Csapat	
RES09	Google Drive	Nincs	Opcionális	Csapat	Dokumentum megosztás, szerkesztés
RES10	Microsoft Office 2016	Nincs	Opcionális	Csapat	Szöveg- és táblázatszerkesztő

### 2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
HO01	Skeleton átadás	Bemutató	Alapvető	Megrendelő	2016.03.29
HO02	Prototípus átadás	Bemutató	Alapvető	Megrendelő	2016.04.25
HO03	Teljes program átadása	Bemutató	Alapvető	Megrendelő	2016.05.17
HO04	A program külső segítség nélkül futtatható, útmutatás alapján	Bemutató	Fontos	Megrendelő	
HO05	A programnak futnia kell a HSZK-ban rendelkezésre álló gépeken	Bemutató	Alapvető	Megrendelő	

### 2.3.4 Egyéb nem funkcionális követelmények

Nincs egyéb nem funkcionális követelmény.

## 2.4 Lényeges use-case-ek

### 2.4.1 Use-case leírások

Use-case neve	Indítás
Rövid leírás	A Felhasználó elindítja a programot.
Aktorok	Játékos (O'Neill)
Forgatókönyv	A program elindul, a pálya betöltődik, kezdődik a játék.

Use-case neve	Kilépés
Rövid leírás	A felhasználó bezárja a programot.
Aktorok	Játékos (O'Neill)
Forgatókönyv	A játék véget ér és a Program bezárul.

Use-case neve	Lövés
Rövid leírás	A Felhasználó egy csillagkaput lő ki. Két lehetőség létezik: kék vagy narancs csillagkapu.
Aktorok	Játékos (O'Neill)
Forgatókönyv	A megfelelő falra lőjük ki a lövedéket, egy megfelelő színű csillagkapu nyílik a falon.

Use-case neve	Kék csillagkapu
Rövid leírás	A „Lövés” alete, ez egy kék csillagkaput nyit.
Aktorok	Játékos (O'Neill)
Forgatókönyv	A megfelelő falra lőjük ki a lövedéket, egy kék színű csillagkapu nyílik a falon.

Use-case neve	Narancs csillagkapu
Rövid leírás	A „Lövés” alete, ez egy narancsszínű csillagkaput nyit.
Aktorok	Játékos (O'Neill)
Forgatókönyv	A megfelelő falra lőjük ki a lövedéket, egy narancsszínű csillagkapu nyílik a falon.

Use-case neve	Lépés
Rövid leírás	A felhasználó előre lép egy mezőt.
Aktorok	Játékos (O'Neill)
Forgatókönyv	O'Neill lép egy mezőt, ha nem fal vagy (zárt) ajtó van előtte.

Use-case neve	Forgás
Rövid leírás	A felhasználó forog, megváltoztatja az irányát.
Aktorok	Játékos (O'Neill)
Forgatókönyv	O'Neill a megadott irányba fordul.

Use-case neve	Doboz felvétel
Rövid leírás	O'Neill felvesz egy dobozt.
Aktorok	Játékos (O'Neill)
Forgatókönyv	O'Neill felveszi a dobozt, tehát a földről eltűnik, majd ezután

	O'Neill-hez kerül és képes lesz lerakni azt.
--	--

<b>Use-case neve</b>	Doboz lerakás
<b>Rövid leírás</b>	O'Neill lerak egy dobozt.
<b>Aktorok</b>	Játékos (O'Neill)
<b>Forgatókönyv</b>	O'Neill lerakja a dobozt, így az megjelenik O'Neill előtt. O'Neill-től eltűnik a doboz.

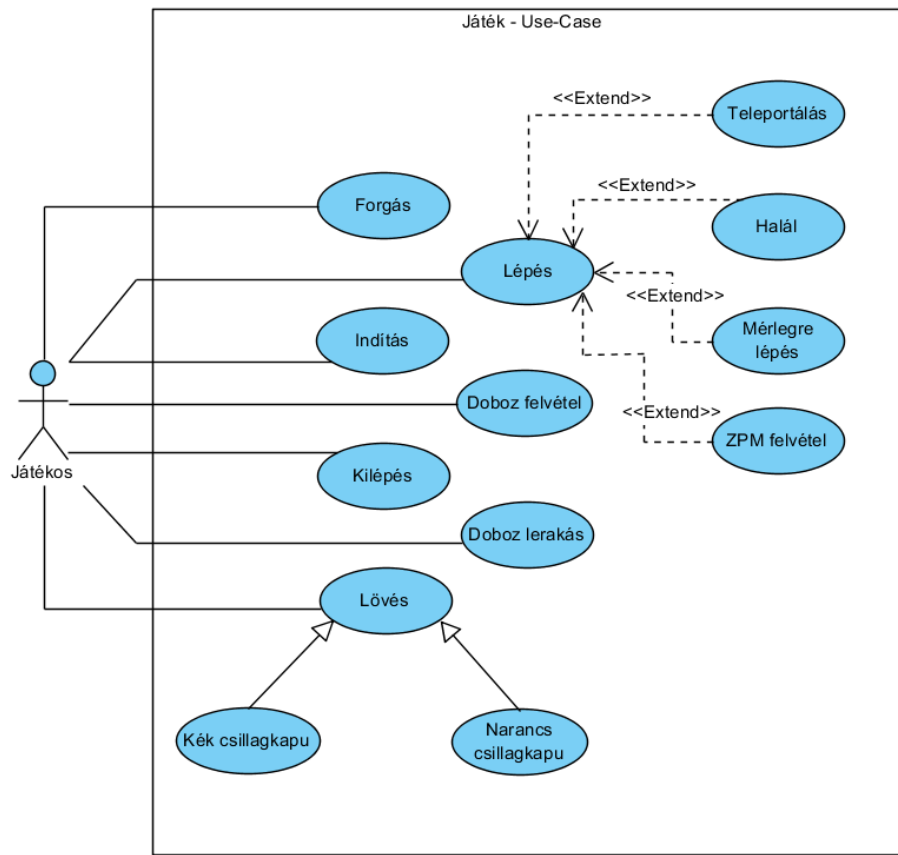
<b>Use-case neve</b>	Teleportálás
<b>Rövid leírás</b>	O'Neill belép a csillagkapuba. A „Lépés” egy speciális esete.
<b>Aktorok</b>	Játékos (O'Neill)
<b>Forgatókönyv</b>	O'Neill belép a csillagkapuba és a másik csillagkapu elé kerül.

<b>Use-case neve</b>	Mérlegre lépés
<b>Rövid leírás</b>	O'Neill rálép egy mérlegre, ami kinyit egy ajtót. A „Lépés” egy speciális esete.
<b>Aktorok</b>	Játékos (O'Neill)
<b>Forgatókönyv</b>	O'Neill rálép egy mérlegre, ami kinyit egy ajtót.

<b>Use-case neve</b>	ZPM felvétel
<b>Rövid leírás</b>	O'Neill felveszi a ZPM-et, miután rálépett. A „Lépés” egy speciális esete.
<b>Aktorok</b>	Játékos (O'Neill)
<b>Forgatókönyv</b>	O'Neill felveszi a ZPM-et, így az eltűnik a földről és O'Neill-hez kerül.

<b>Use-case neve</b>	Halál
<b>Rövid leírás</b>	O'Neill szakadékba esik és meghal. A „Lépés” egy speciális esete.
<b>Aktorok</b>	Játékos (O'Neill)
<b>Forgatókönyv</b>	O'Neill szakadékba esik és meghal.

## 2.4.2 Use-case diagram



2.1. ábra Use-case diagram

## 2.5 Szótár

- **ajtó** - A labirintus egyes szobáit választja el. Ha be van zárva, akkor nem lehet átmenni rajta. Ahhoz hogy kinyissuk, az ajtóhoz tartozó megfelelő mérlegre egy sújt kell elhelyezni, ami lehet egy doboz, vagy maga O'Neill ezredes. Ha a súly lekerül a mérlegről, az ajtó újra bezáródik.
- **Anubis** - A játék főgonosza. Az a terve hogy leigázza a Földet.
- **atlantiszi úrhajó** - O'Neill ezredes úrhajója. Ezzel lehet megakadályozni hogy a gonosz Anubis leigázhassa a földet.
- **csempe** - A pálya elemei (falak, folyosók, szakadékok, stb.) ezekből az előre meghatározott egység méretű elemekből épülnek fel. Minden csempehez tartozik egy tulajdonság, hogy rá tudunk e lépni és ha igen, akkor mi történjen a játékban.
- **csillagkapu** - O'Neill ezredes fegyverével speciális falakon létrehozott kapu. Ezen áthaladva tudunk egy főregjáraton keresztül a másik csillagkapuhoz eljutni.
- **doboz** - Egy olyan tárgy, aminek a súlya elegendő, hogy a mérlegre helyezve azt lent tartsa, ahhoz hogy az ajtó kinyíljon. A dobozt O'Neill ezredessel fel tudjuk venni és odébb tudjuk vinni, akár a főregjáratokon keresztül is.
- **fegyver** - O'Neill ezredes fegyvere, mellyel egy speciális falra löve egy csillagkaput tud nyitni. Két féle kaput tud nyitni, sárgát és kéket, melyek egyazon főregjárat két végének a bejáratai. Egy színű csillagkapuból csak egy létezhet egyszerre. A fegyver töltényei soha nem fogynak el.
- **főregjárat** - O'Neill ezredes fegyverével létrehozott két csillagkapu közötti járat. A két csillagkapu között szabadon mozoghatunk ezen a járaton keresztül mindkét

irányba úgy, hogy a csillagkapuk legyenek bármilyen távol is egymástól, a járaton keresztül nem kell idő, hogy eljussunk az egyik bejáratról a másikig. Így egy pillanat alatt a labirintus egyik pontjáról egy másik távoli pontjára is el tudunk jutni, melyet egyébként nem tudnánk megközelíteni.

- **goa'uld** - Gonosz földönkívüli kukac, aki átveszi a gazdatest tudatát.
- **mérleg** - Ez egy kapcsoló az ajtókhöz. Ha súly kerül rá, a hozzá tartozó ajtó kinyílik és amíg az rajta van, nyitva marad. Ha lekerül róla a súly, akkor az ajtó újra bezáródik.
- **O'Neill ezredes** - A játék főhőse, aki meg szeretné akadályozni Anubis gonosz tervét. A játék folyamán a játékos őt irányítja a labirintusokban.
- **pálya** - Az a tér amin a játék zajlik. Ezt a játék során felülnézetben látjuk.
- **speciális fal** - Olyan fal amire O'Neill fegyverével rálöve meg tud jelenni egy csillagkapu. Nem speciális falakon nem tud csillagkaput létrehozni.
- **szakadék** - A pályán található olyan terület, amire ha rálép O'Neill ezredes, akkor lezuhan és meghal, illetve ha egy doboz kerül rá, akkor az is leesik, és megsemmisül. A szakadékok felett át lehet lőni a fegyverrel.
- **ZPM** - Az atlantiszi űrhajó energiaellátásához szükséges kristály. A játék célja hogy O'Neill ezredest irányítva a labirintusban ezeket összegyűjtsük, hogy az űrhajó energiaellátását meg lehessen oldani, és Anubis támadását vissza lehessen verni.

## 2.6 Projekt terv

### 2.6.1 Csapattagok és feladatkörök

A csapat hat főből áll. A nagy létszámra való tekintettel a feladatokat igyekeztünk úgy szétosztani a projektben résztvevők közt, hogy a tagok preferenciáját szem előtt tartsuk, ugyanakkor fontosnak tartjuk kiemelni, hogy a csapat összes tagja kiveszi a részét a projekt leszállítandóinak elkészítésében. Ennek fényében - a projektre általánosan értelmezett felosztás - a következő:

Név	Feladatkör
Bujdosó Dániel	Dokumentáció, forráskód
Bukta Bence	Dokumentáció, forráskód
Horváth Balázs	Dokumentáció, diagramok
Karácsony Zsolt	Csapatvezetés, dokumentáció, tesztek
Kelemen András	Dokumentáció, diagramok
Vincze Zsolt	Dokumentáció, diagramok, tesztek, grafikus elemek

### 2.6.2 Kommunikáció a fejlesztés során

A projekt sikere, véleményünk szerint, az összehangolt munkavégzésen és a jó kommunikáción múlik. Ezt több módon is támogatjuk a fejlesztés során.

**Heti rendszeres státusz:** Heti rendszerességgel (amennyiben lehet, többször is) 1-1,5 órás személyes státuszt tartunk a projekt tagjaival, ahol átbeszéljük az aktuális hét feladatait, megbeszéljük a nehézségeket. Előre meghatározott ütemterv szerint szeretnénk haladni, így itt szeretnénk biztosítani, hogy erőforráshiány esetén ad-hoc megoldást tudjunk adni a problémára, felelőssel és határidővel együtt.

**Heti rendszeres konferencia:** Egnél több személyes találkozóra nem mindig jut lehetőségünk, így minden héten (főként a hét végén) 1-1,5 órás VoIP konferenciát tartunk, ahol a leadás előtti utolsó teendőket, elvégzendő feladatokat egyeztetjük. Ezt a lehetőséget már csak a minimális pontosításokra szeretnénk fenntartani, ügyelve a határidő közelségére.

**Google Groups:** Létrehoztunk egy csoport levelező listát, amire a tagok levelet tudnak küldeni, ezt mindenki megkapja. Sürgős feladat vagy probléma esetén könnyebbnek tartjuk az e-mailes kommunikációt. A csapat tagjai mind rendszeresen olvassák emailjeiket, így valaki biztos reagál egy-egy beérkező levélre.

**Trello:** A fejlesztés során használjuk a Trello nevű online projekt menedzsment szoftvert, amely jól támogatja kisebb fejlesztő csapatok együttműködését. A rendszer teljesen testre szabható, így mi munkánk során Kanban boardként használjuk. A feladatokat négy fő kategóriára osztjuk (Elvégzendő, Folyamatban, Tesztelés alatt, Kész) és mindegyikhez határidőt, valamint felelőst rendelünk. A felelősök rendszeres email értesítést kapnak az elvégzendő feladról. A csapatvezető napi szinten ellenőrzi a feladatok állapotát, ahol szükség van, beavatkozik.

### 2.6.3 Alkalmazott eszközök a fejlesztés során

**GitHub:** Az összehangolt munkavégzés érdekében a Git elosztott verziókezelő szoftvert választottuk a fejlesztésünk támogatására. A fejlesztéshez a GitHub rendszert használjuk. A választásunk azért esett erre a megoldásra, mert rendkívül elterjedt, könnyen kezelhető és jó supporttal bíró programról van szó. A csapat tagjai mind megismertekedtek korábban a Git kezelésének alapjaival, aminek segítségével a hatékony munkavégzés biztosítható.

**Google Drive:** Mivel a csapat minden tagja részt vesz a dokumentáció elkészítésében, a változások követésére a Google Drive rendszerét fogjuk használni. Ennek a változás követés funkcióján kívül nagy előnye, hogy kommentezhetők a dokumentum tetszőleges részei, így egymásnak kérdéseket, pontosítási utasításokat tudunk jelezni.

**Microsoft Office 2016:** A formázások pontosságának ellenőrzésére pedig Microsoft Office rendszert használunk, a leadásra kerülő anyagok minden esetben átesnek ezen a korrektúrázáson, hogy ne legyen például sérült címsor, rossz formázás. A sablonok kialakításán értelemszerűen nem változtatunk.

**Eclipse:** A fejlesztésünkhöz az Eclipse nevű Java IDE-t fogjuk használni, egészen pontosan a szoftver Mars (4.5.0) verzióját. Azért esett erre a választásunk, mert munkánk és korábbi tanulmányaink során megismertük a működését illetve egy jól konfigurálható fejlesztő eszköznek tartjuk.

**Visual Paradigm:** A diagramok szerkesztésére a Visual Paradigm nevű programot (a szoftver Community Edition verzióját) választottuk. Korábbi tapasztalataink alapján jól kezelhető, sok funkcióval bíró alkalmazásról van szó, mely hatékonyan tudja támogatni az analízis modellben szállítandó diagramok elkészítését.

### 2.6.4 Mérföldkövek és határidők

A feladat főbb mérföldköveit, leadási és bemutatási határidőit az alábbi táblázatban foglaltuk össze.

#	Dátum	Feladat	Ellenőrzés módja
1	2016.02.29	Követelmény, projekt, funkcionalitás	Beadás
2	2016.03.07	Analízis modell kidolgozása 1.	Beadás
3	2016.03.14	Analízis modell kidolgozása 2.	Beadás
4	2016.03.21	Szkeleton tervezése	Beadás
5	2016.03.29	Skeleton	Bemutatás

6	2016.04.04	Prototípus koncepciója	Beadás
7	2016.04.11	Részletes tervek	Beadás
<b>8</b>	<b>2016.04.25</b>	<b>Prototípus</b>	<b>Bemutató</b>
9	2016.05.02	Grafikus felület specifikációja	Beadás
<b>10</b>	<b>2016.05.17</b>	<b>Grafikus változat</b>	<b>Bemutató</b>
11	2016.05.20	Összefoglalás	Beadás

A feladatot három lépcsőben fogjuk megoldani. Ezek a főbb lépcsők (mérföldkövek) a fenti táblázatba vastagon, dőlttel szedve szerepelnek.

A **szkeleton** változat célja annak bizonyítása, hogy az objektum és dinamikus modellek a definiált feladat egy modelljét alkotják. A szkeleton egy program, amelyben már valamennyi, a végső rendszerben is szereplő business objektum szerepel. Az objektumoknak csak az interfésze definiált. Valamennyi metódus az indulás pillanatában az ernyőre szöveges változatban kiírja a saját nevét, majd meghívja azon metódusokat, amelyeket a szolgáltatás végrehajtása érdekében meg kell hívnia. Amennyiben a metódusból valamely feltétel fennállása esetén hívunk meg más metódusokat, akkor a feltételre vonatkozó kérdést interaktívan az ernyőn fel kell tenni és a kapott válasz alapján kell a továbbiakban eljárni. A szkeletonnak alkalmasnak kell lenni arra, hogy a különböző forgatókönyvek és szekvencia diagramok ellenőrizhetők legyenek. Csak karakteres ernyőkezelés fogadható el, mert ez biztosítja a rendszer egyszerűségét.

A **prototípus** program célja annak demonstrálása, hogy a program elkészült, helyesen működik, valamennyi feladatát teljesíti. A prototípus változat egy elkészült program kivéve a kifejlett grafikus interfészt. A változat tervezési szempontból elkészült, az ütemezés, az aktív objektumok kezelése megoldott. A business objektumok - a megjelenítésre vonatkozó részeket kivéve - valamennyi metódusa a végleges algoritmusokat tartalmazza. A megjelenítés és működtetés egy alfanumerikus ernyőn követhető, ugyanakkor a megjelenítés fájlban is logolható, ezzel megteremtve a rendszer tesztelésének lehetőségét. Különös figyelmet kell fordítani az interfész logikájára, felépítésére, valamint arra, hogy az mennyiben tükrözi és teszi láthatóvá a program működését, a beavatkozások hatásait.

A teljes (**grafikus**) **változat** a prototípustól elvileg csak a kezelői felület minőségében különbözhet. Ennek változatnak az értékelésekor a hangsúlyt sokkal inkább a megvalósítás belső szerkezetére, semmint a külalakra kell helyezni.



## 2.7 Napló

Kezdet	Időtartam	Résztevők	Leírás
2016.02.23. 18:30	2 óra	Bujdosó Bukta Horváth Karácsony Kelemen Vincze	Értekezlet. Döntés: A csapat tagjai a rájuk kiosztott fejezetek megírását elvégzik a hét végéig. A csapat egyeztetést tart vasárnap, ahol áttekinti az elkészült anyagot. A szerdai konzultáción a csapat tagjai mind részt vesznek.
2016.02.24. 15:00	3 óra	Bujdosó Bukta	Értekezlet. Döntés: A program általános áttekintésének és funkciói a megbeszélés bekerülnek a dokumentumba.
2016.02.24. 15:00	1 óra	Bujdosó	Tevékenység: 2.2.1-2.2.2 fejezetek
2016.02.24. 15:00	1 óra	Bukta	Tevékenység: 2.2.3-2.2.5 fejezetek
2016.02.25. 12:30	1,5 óra	Kelemen	Tevékenység: A 2.1.1, 2.1.2, 2.1.5 pontok megírása, a 2.1.3 első felének kigyűjtése, megírása.
2016.02.25. 17:30	1,5 óra	Karácsony	Tevékenység: 2.3.1-2.3.3 fejezetek megírása. 2.6-os fejezet megírása, véglegesítése.
2016.02.26. 11:00	0,5 óra	Kelemen	Tevékenység: 2.1.3 folytatása, 2.1.4. elkészítése.
2016.02.26. 19:00	2 óra	Horváth	Tevékenység: 2.4-es feladat kidolgozása, és a 2.3.1-es követelményekhez Use-Case-ek csatolása.

2016.02.27. 22:00	2 óra	Vincze	Tevékenység: Helyesírási hibák ellenőrzése, kijavítása. Táblázatok és szövegek formázása. 2.5 Szótár fejezet kidolgozása
2016.02.28. 11:00	2 óra	Bujdosó Bukta Horváth Karácsony Kelemen Vincze	Értekezlet. Döntés: Horváth elkészíti a megbeszélésen pontosított use-case diagramot és frissíti a táblázatos formában leírt use-case-eket. Karácsony elkészíti a dokumentum végleges változatát és megküldi a résztvevőknek véleményezésre. A tagok a nap végéig véleményezik az anyagot. Karácsony elvégzi a nyomtatást és átadja az anyagot Vinczének, akii 02.29-én leadja a kész verziót a meghirdetett időpontban.
2016.02.28 13:30	0,5 óra	Horváth	Tevékenység: 2.4.1, 2.4.2 fejezet véglegesítése.
2016.02.28 14:00	2,5 óra	Karácsony	Tevékenység: Dokumentum véglegesítése, formázások, táblázatok javítása, egységesítés, korrektúra.

### 3. Analízis modell kidolgozása

#### 3.1 Objektum katalógus

##### 3.1.1 Game

A játék belépési pontja. Felelőssége, hogy megjeleníti az ablakot, amiben a játék fut.

##### 3.1.2 GamePanel

Ez az objektum felel a játékalak kirajzolásáért, a keret feltöltéséért. Felelőssége továbbá, hogy figyeli a felhasználótól érkező parancsokat (a billentyűzetről.)

##### 3.1.3 GameStateManager

A játék állapotait kezelő objektum. Meghatározza, hogy melyik az éppen aktív játékalap és azt vezérli a GamePaneltől kapott bemenet segítségével.

##### 3.1.4 GameState

A játékalapok osztálya.

##### 3.1.4.1 MenuState

A játék főmenüjét vezérlő objektum. Az előforduló menüpontok, azok műveleteinek elvégzése tartozik a feladatai közé.

##### 3.1.4.2 LevelState

A játék vezérléséért felelős objektum.

##### 3.1.5 TileMap

A pályát fájlból beolvasó és felépítő objektum. Feladata hogy a megadott térképfájl alapján a játékosnak megjelenítse a pályát, amin végig kell haladnia. Ezen kívül ő tárolja a Tile objektumokat koordináták szerint.

##### 3.1.6 Tile

Csempe objektum. Ismeri az adott csempe x,y koordinátáját és a hozzá tartozó textúrát. A hozzá tartozó (rajta lévő) MapObject objektumokat is tárolja, kezeli.

##### 3.1.6.1 Floor

A padló elem, ezekre lépkedhet a játékos és lehetnek rajta egyéb speciális entitások.

##### 3.1.6.2 Gap

A szakadék elem a pályán. A játékos belelépve meghal, a dobozt erre lerakva megsemmisül, a lövedék áthalad felette.

##### 3.1.6.3 Wall

A pályán lévő falak. Erre az elemre sem a játékos nem tud rálépni, sem lövedék nem haladhat át rajta. Dobozt, ZPM-et nem tartalmazhat ez a mező. Speciális változata is létezik, azon csillagkapu nyílhat.

### 3.1.7 MapObject

A pályán található entitások. A Box, ZPM, Door, Scale őssosztálya.

#### 3.1.7.1 Box

A feladatkiírásban dobozként említett objektumot reprezentálja. A játékos fel tudja venni és másik helyre tudja cipelni. Felelőssége, hogy mérlegre helyezve lenyomja azt és így a mérleg ajtót tud nyitni.

#### 3.1.7.2 ZPM

A játékos által összegyűjtendő energiamodult reprezentáló objektum. Külön felelőssége nincs, a játékos ezeket gyűjti össze a játék során.

#### 3.1.7.3 Door

A feladatkiírásban ajtóként említett objektumot reprezentálja. Egy ajtó egy mérleggel (Scale) van összekapcsolva. Amennyiben a mérlegen megfelelő súly van, az ajtó nyitott állapotú lesz.

#### 3.1.7.4 Scale

A feladatkiírásban mérlegként említett objektumot reprezentálja. Egy mérleg egy ajtóval (Door) van összekapcsolva. Amennyiben a mérleg lenyomott állapotban van (azaz megfelelő súly van rajta), a hozzá tartozó ajtó nyitott állapotba kerül. A játékos rá tud állni vagy rá tud helyezni egy dobozt. Felelőssége, hogy vele lehet kinyitni az ajtót, amin O'Neill átmehet.

### 3.1.8 Entity

A mozgásra képes objektumok őssosztálya. Belőle származik le a játékos (O'Neill) és a lövedék (Projectile). Az Entity osztály oldja meg a csempékkel való kommunikációt.

#### 3.1.8.1 O'Neill

A játékost reprezentáló osztály. Felvehet dobozt illetve az összegyűjtött ZPM-ek számát tárolja. Ő felel a kilőhető narancssárga és kék lövedékekért.

#### 3.1.8.2 Projectile

O'Neill által kilőhető lövedék. Kék vagy narancssárga verzióban létezhet. Kilövés után a pályán halad. Speciális falba csapódva csillagkapu jön létre a becsapódás helyén a megfelelő színben.

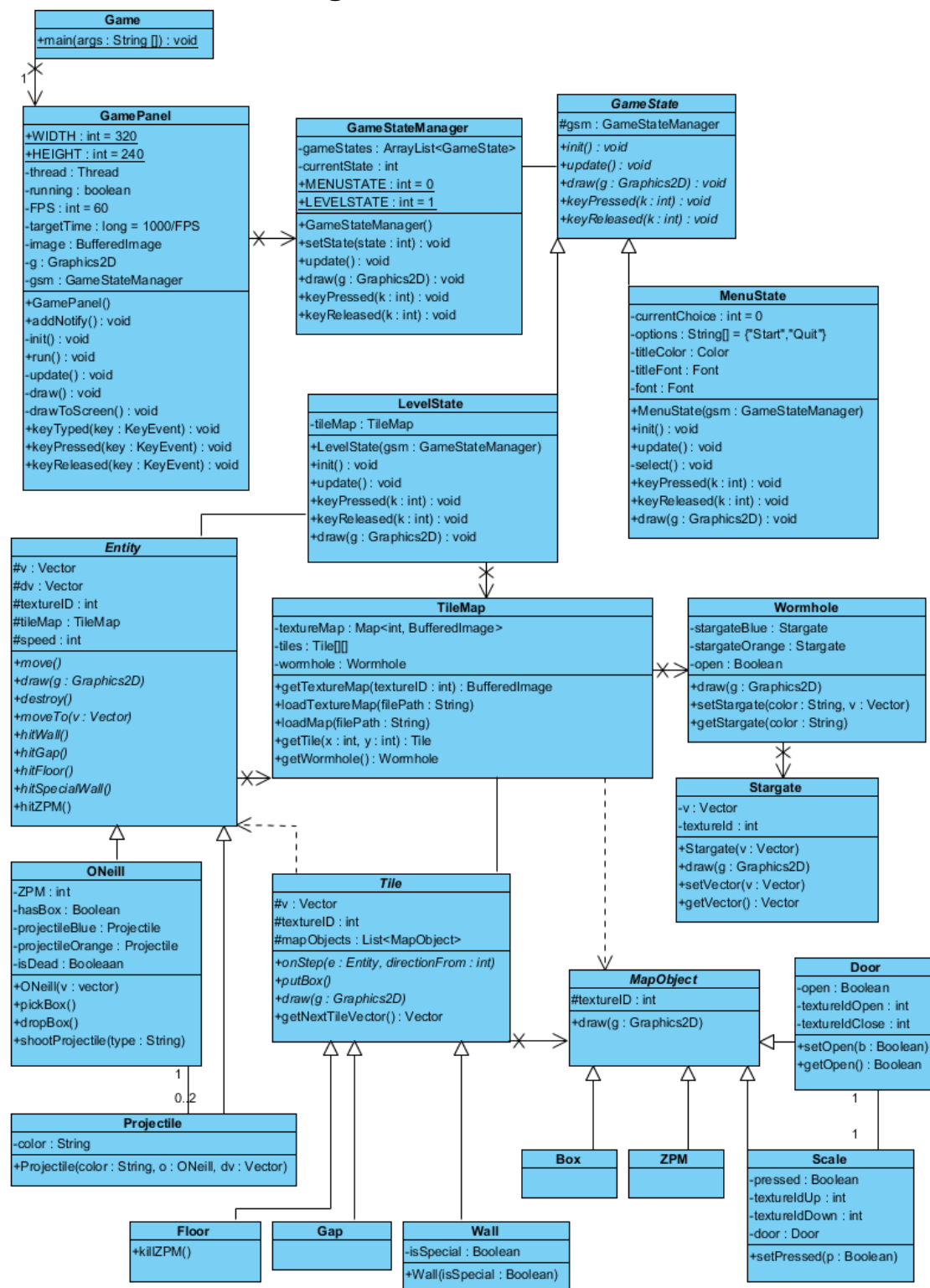
### 3.1.9 Wormhole

A feregjáratot kezelő objektum. Ez kezeli a feregjárat két végén lévő csillagkapukat (Stargate).

#### 3.1.10 Stargate

Egy csillagkaput és annak paramétereit tároló objektum.

### 3.2 Statikus struktúra diagramok



### 3.1. ábra Osztálydiagram

### 3.3 Osztályok leírása

#### 3.3.1 Box

- **Felelősség**  
A játékos által fel és letehető doboz objektumot leíró osztály. A dobozt a játékos felveheti majd azt egy másik alkalmas csempére ráteheti. A dobozon átsétálni és átlőni sem lehet.
- **Ősosztályok**  
MapObject
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**  
Nem értelmezhető.

#### 3.3.2 Door

- **Felelősség**  
Az ajtót leró osztály, feladata hogy nyilvántartsa hogy az adott ajtó nyitva van e, és ezáltal átjárható e O'Neill illetve a lövedékek számára
- **Ősosztályok**  
MapObject
- **Attribútumok**
  - **boolean open:** Változó amiben azt tároljuk hogy az ajtó nyitva van vagy sem.
  - **int textureIDOpen:** A nyitott ajtóhoz tartozó grafika.
  - **int textureIDClosed:** A zárt ajtóhoz tartozó grafika.
- **Metódusok**
  - **void setOpen(boolean b):** Beállítja az ajtó nyitvatartását.
  - **boolean isOpen():** Visszaadja, hogy az ajtó nyitva van-e.

#### 3.3.3 Entity

- **Felelősség**  
A játékos által „irányított” és animált objektumok őssosztálya.
- **Ősosztályok**  
Nem értelmezhető.
- **Attribútumok**
  - **Vector v:** Az Entity koordinátáit tartalmazó változó.
  - **Vector dv:** Az Entity abszolút koordinátáit tartalmazó változó.
  - **int textureID:** Az Entityhez tartozó grafika azonosítója.
  - **TileMap tileMap:** A játéktérteret tartalmazó TileMap referenciája.
  - **int speed:** Az Entitás mozgási sebessége.

- **Metódusok**

- **void move()**: Mozgatja az adott entitást, lekérdezi az éppen aktuális csempétől hogy mi történik ha rálép.
- **void draw(Graphics2D g)**: Kirajzolja az entitást.
- **void destroy()**: Megsemmisíti az entitást.
- **void moveTo(Vector v)**: Az entitást a megadott helyre helyezi.
- **void hitWall()**: Akkor hívódik, ha az entitás falhoz ér.
- **void hitGap()**: Akkor hívódik, ha az entitás szakadékhoz ér.
- **void hitFloor()**: Akkor hívódik, ha az entitás padlóra lép.
- **void hitSpecialWall()**: Akkor hívódik meg, ha az entitás speciális falra lép(ne).
- **void hitZPM()**: Akkor hívódik meg, ha az entitás ZPM-re lép.

### 3.3.4 Floor

- **Felelősség**

A labirintus egyik alapeleme. Ezen a csempén a játékos szabadon mozoghat, illetve vehet fel vagy tehet le dobozokat. A padlón szabadon sétálni s átlőni is lehetséges.

- **Őszosztályok**

Tile

- **Attribútumok**

Nem értelmezhető.

- **Metódusok**

- **void killZPM()**: A ZPM felvételét kezelő metódus.

### 3.3.5 Game

- **Felelősség**

A játék futtatásáért felelős osztály.

- **Őszosztályok**

Nem értelmezhető.

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

Nem értelmezhető.

- **Metódusok**

- **void main(String args[])**: Ez a játék main metódusa, ami a játék futtatását végzi.

### 3.3.6 GamePanel

- **Felelősség**

A GamePanel osztály indítja a játékot futtató szálát, valamint tartalmazza játékciklust (gameloop). Ez az osztály tölti ki a Game osztályban létrehozott ablakot, valamint fogadja a bemenetet a billentyűzetről és továbbadja azokat.

- **Ősosztályok**

java.lang.Object -> java.awt.Component -> java.awt.Container ->  
javax.swing.JComponent

- **Interfészek**

Runnable, KeyListener

- **Attribútumok**

- **int WIDTH**: A játéklablak szélessége. Default értéke 320.
- **int HEIGHT**: A játéklablak magassága. Default értéke 240.
- **Thread thread**: A játékot futtató szálát ebben az objektumban tároljuk.
- **boolean running**: Logikai érték, mely azt mutatja, hogy a játék fut vagy sem.
- **int FPS**: A játék FPS értékét tartalmazó egész szám. Default értéke 60.
- **long targetTime**: Az újrarajzoltatás frekvenciáját tároló érték. Default értéke 1000 ms/FPS.
- **BufferedImage image**: A játék képét tartalmazó mutató.
- **Graphics2D g**: A kirajzolásért felelős objektum példánya.
- **GameStateManager gsm**: A játékmenet kezelését végző GameStateManager objektum egy példánya.

- **Metódusok**

- **GamePanel()**: Az osztály konstruktora.
- **void addNotify()**: Kapcsolatot létesít a natív kimenet és a program között
- **void run()**: A játékot futtató szál elindításáért felelős metódus.
- **void keyTyped(KeyEvent key)**: Billentyűzeten érkező karaktersorozatot figyelő metódus.
- **void keyPressed(KeyEvent key)**: Egy billentyű leütését figyelő metódus.
- **void keyReleased(KeyEvent key)**: Egy billentyű felengedését figyelő metódus.

### 3.3.7 GameState

- **Felelősség**

A játékállapotok absztrakt ősosztálya, tartalmazza a megvalósítandó metódusokat valamint egy GameStateManager-t.

- **Attribútumok**

- **GameStateManager gsm**: A játék állapotát kezelő GameStateManager objektum példánya.

- **Metódusok**

- **abstract void init()**: Játékállapot inicializálására szolgáló metódus.
- **abstract void update()**: A játékállapot frissítését végző metódus.



- **abstract void draw(Graphics2D g):** Ez a metódus rajzolja ki a játékos által látott képet a játéklablakba.
- **abstract void keyPressed(int k):** Billentyű lenyomását figyelő metódus.
- **abstract void keyReleased(int k):** Billentyű felengedését figyelő metódus.

### 3.3.8 GameStateManager

- **Felelősség**

A játék különböző állapotai közötti váltásért felelős osztály. Ez az osztály kapja meg a GamePanel osztálytól a leütött billentyűket, majd továbbadja őket az éppen aktuális játékállapotnak

- **Attribútumok**

- **ArrayList<GameState> gameStates:** A játék során előforduló állapotokat tároló lista.
- **int currentState:** A játék jelenlegi állapotát reprezentáló változó.
- **int MENUSTATE:** A játék menüjének állapotát reprezentáló változó. Default értéke 0.
- **int LEVELSTATE:** A választott pálya állapotát reprezentáló változó. Default értéke 1.

- **Metódusok**

- **GameStateManager():** Az osztály konstruktora.
- **void setState(int state):** A játék állapotának beállítása ezen a metóduson keresztül történik.
- **void update():** A játékállapot frissítését végző metódus. Az aktuális játékállapot update() metódusát hívja.
- **void draw(Graphics2D g):** Ez a metódus rajzolja ki a játékos által látott képet a játéklablakba.
- **void keyPressed(int k):** Billentyű lenyomását figyelő metódus. Továbbítja az aktuális játékállapot felé
- **void keyReleased(int k):** Billentyű felengedését figyelő metódus. Továbbítja az aktuális játékállapot felé

### 3.3.9 Gap

- **Felelősség**  
Egy olyan speciális csempe amire ha a O'Neill rálép akkor meghal, valamint ha egy tárgy kerül egy ilyen csempére az megsemmisül.
- **Ősosztályok**  
Tile
- **Metódusok**  
Nem értelmezhető.

### 3.3.10 LevelState

- **Felelősség**  
A játékmenetért felelős játékalapot (GameState) osztály. Ő felel a labirintusért illetve a labirintusban szereplő elemekért, mint például O'Neill ezredes.
- **Ősosztályok**  
GameState
- **Attribútumok**
  - **TileMap tileMap**: A pálya alapját adó csempéket tartalmazó objektum.
- **Metódusok**
  - **LevelState(GameStateManager gsm): A pálya konstruktora.**
  - **void init()**: A pálya inicializálását végző metódus.
  - **void update()**: A pálya frissítését végző metódus.
  - **void draw(Graphics2D g)**: Feladata hogy meghívja a kirajzolandó objektumok draw metódusát.
  - **void keyPressed(int k)**: Billentyű lenyomását figyelő metódus valamint kezelő metódus
  - **void keyReleased(int k)**: Billentyű lenyomását figyelő metódus valamint kezelő metódus

### 3.3.11 MapObject

- **Felelősség**  
Ezen osztály alá a játékkossal interakcióba lépésre képes osztályok tartoznak
- **Attribútumok**
  - **int textureID**: Az adott textúra-elem sorszámát tartalmazza.
- **Metódusok**
  - **void draw(Graphics2D texture)**: Ez a metódus rajzolja ki a játékos által látott textúrát az adott elemhez.

### 3.3.12 MenuState

- **Felelősség**  
A játék főmenüjéért felelős osztály, ez felelőssége hogy a képernyőre a megfelelő menüpontok legyenek kirajzolva, majd a megfelelő billentyűk leütésére leütése után elindítja a játékot vagy kilép belőle.
- **Össztályok**  
GameState
- **Attribútumok**
  - **int currentChoice**: A felhasználó által akutálisan kiválasztott menüpont értéke. Default értéke 0.
  - **String[] options**: A menü elérhető állapotait tartalmazza. Lehetséges értékei "Start" és "Quit"
  - **Color titleColor**: A főcím színét tartalmazó változó.
  - **Font titleFont**: A főcím betűtípusát tartalmazó változó.
  - **Font font**: A menüben előforduló betűtípust tartalmazó változó.
- **Metódusok**
  - **MenuState(GameStateManager gsm)**: A menü konstruktora.
  - **void init()**: A menü inicializálását végző metódus.
  - **void update()**: A menü frissítését végző metódus.
  - **void draw(Graphics2D g)**: Ez a metódus rajzolja ki a menüpontokat a képernyőre valamint jelöli a játékos által épp kiválasztott menüpontot
  - **void keyPressed(int k)**: Billentyű lenyomását figyelő metódus.
  - **void keyReleased(int k)**: Billentyű felengedését figyelő metódus.

### 3.3.13 ONeill

- **Felelősség**  
A játékost megszemélyesítő osztály, feladata hogy kövesse a játékos utasításait és annak megfelelően cselekedjen, lépjen, forduljon, meghaljon, stb.
- **Össztályok**  
Entity

- **Attribútumok**
  - **int ZPM**: Az összegyűjtött ZMP-ek száma.
  - **boolean hasBox**: Megmondja hogy van e éppen O'Neill kezében doboz.
  - **Projectile projectileBlue**: A kilőtt kék lövedék referenciája (ha van ilyen)
  - **Projectile projectileOrange**: A kilőtt sárga lövedék referenciája (ha van ilyen)
- **Metódusok**
  - **ONeill(Vector v)**: Konstruktor, beállítja a kezdőpozíciót.
  - **void pickBox()**: Doboz felvétele a következő mezőről.
  - **void dropBox()**: Doboz letétele a következő mezőre.
  - **void shootProjectile(String type)**: Lövedék útnak indítása.

### 3.3.14 Projectile

- **Felelősség**

A játékos által kilőtt lövedék.
- **Össztályok**

Entity
- **Attribútumok**
  - **String color**: A lövedék színe szövegesen ("blue", "orange").
- **Metódusok**
  - **Projectile(String color, O'Neill o, Vector dv)**: Konstruktor, ami megkapja a lövedék színét, O'Neill-t, és a kezdő pozícióját.

### 3.3.15 Scale

- **Felelősség**

A mérleget leíró osztály, feladata hogy kinyissa a hozzá tartozó ajtót ha sújt tesznek rá.
- **Össztályok**

MapObject
- **Attribútumok**
  - **boolean pressed**: Ez a változó jelzi hogy a mérlegen van-e súly.
  - **int textureIDUp**: A súlyozatlan mérleg grafikája.
  - **int textureIDDown**: A súlyozott mérleg grafikája.
  - **Door door**: A mérleghez tartozó ajtó referenciája.
- **Metódusok**
  - **void setPressed(boolean b)**: Beállítja a pressed változó értékét.

### 3.3.16 Stargate

- **Felelősség**  
A csillagkapuk adatainak tárolását végző osztály, tárolja a kapuk helyzetet valamint irányát.
- **Attribútumok**
  - **Vector v**: A csillagkapuk helyzetére és irányára vonatkozó adatot tartalmazó segédosztály.
  - **int textureID**: A kapuk grafikájának azonosítója.
- **Metódusok**
  - **void draw(Graphics2D g)**: kirajzolja a kapukat a megfelelő helyen.
  - **Vector getVector()**: Visszaadja a kapu helyzetét és irányát tartalmazó objektumot.
  - **void setVector(Vector v)**: beállítja a kapu helyzetét és irányát.

### 3.3.17 Tile

- **Felelősség**  
A játékkeret felépítő labirintus alap építőegysége.
- **Attribútumok**
  - **Vector v**: csempe labirintusbeli koordinátáit tároló segédosztály
  - **int textureID**: A csempehez tartozó grafika azonosítóját tartalmazó változó.
  - **List<MapObject> mapObjects**: A csempén található MapObject-ek egy listában tárolva.
- **Metódusok**
  - **void onStep()**: Ez a metódus hívódik abban az esetben ha a egy Entity (O'Neill vagy egy lövedék) „rálép” a csempepre
  - **void putBox()**: Ez a metódus hívódik meg abban az esetben ha a játékos egy dobozt akar rátenni
  - **void draw(Graphics2D g)**: Ez a metódus lekérdezi a grafikákat tartalmazó Map-tól a hozzá tartozó grafikát majd kirajzolja azt illetve a vele egy koordinátán szereplő MapObjectek draw metódusait is meghívja

### 3.3.18 TileMap

- **Felelősség**  
Ez az osztály tárolja magában a labirintust, illetve az ahhoz szükséges grafikai elemeket. A labirintus valamint a hozzá tartozó csempegrafika beolvasás is az ő feladata.
- **Attribútumok**
  - **Map<int, BufferedImage> textureMap**: Ez az attribútum tárolja a pályát alkotó csempek képeit és azok sorszáma kulcs-érték párokban.
  - **Tile[][] tiles**: A csempek ebben a tömbben vannak eltárolva.
  - **Wormhole wormhole**: A pályán előforduló féregjáratot ebben tároljuk.

- **Metódusok**

- **BufferedImage getTextureMap(int textureID):** Ez a függvény visszaadja az adott ID-hoz tartozó csempe képét.
- **void loadTextureMap(String filePath):** A pályán található csempék képét ez a függvény olvassa be egy fájlból.
- **void loadMap(String filePath):** A pályát(labirintust) egy fájlból beolvasó metódus.
- **Tile getTile(int x, int y):** A megadott koordinátákon található csempét adja meg.
- **Wormhole getWormhole():** Megadja a pályán működő féregjáratot (ha van ilyen.)

### 3.3.19 Wall

- **Felelősség**

Speciális csempe, feladata hogy elválasztóelemként működjön a játékban, korlátozva a játékost a mozgásban valamint bizonyos falakon lehet csillagkaput is nyitni. A falon sem átsétálni se átlőni nemlehet.

- **Ősosztályok**

Tile

- **Attribútumok**

- **boolean isSpecial:** Ez az attribútum jelöli, ha egy fal speciális, azaz lehet rajta csillagkaput nyitni.

- **Metódusok**

- **void onStep(Entity e, int directionFrom):** Ez a metódus kezeli, ha falba próbálunk belelépni, amit a feladat értelmezése szerint nem tudunk megtenni.
- **void putBox():** Ez a metódus kezeli, ha a nálunk lévő dobozt a falba akarnánk elhelyezni.
- **void draw(Graphics2D texture):** Ez a metódus rajzolja ki a játékos által látott textúrát az adott falhoz.

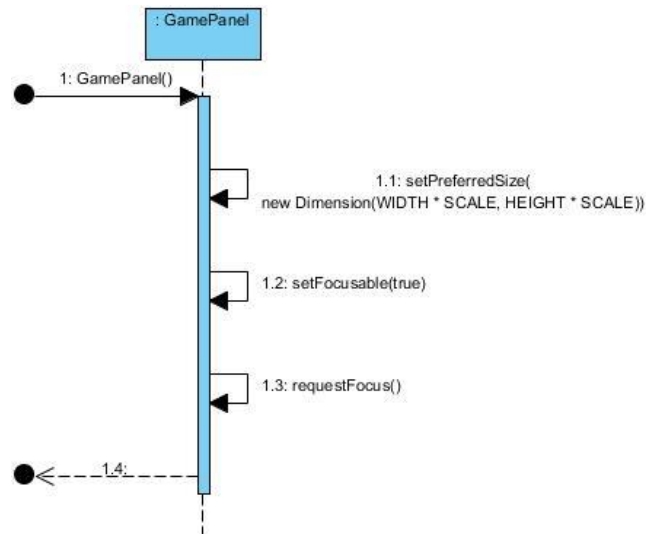
### 3.3.20 Wormhole

- **Felelősség**  
A pályán található csillagkapuk és féregjárat működéséért felelős osztály. Ő tárolja a kapuk helyét és ügyel arra, hogy egy fajta kapuból csak 1 darab létezzen.
- **Attribútumok**
  - **StarGate starGateBlue**: A kék csillagkapu adatait tartalmazó osztály referenciája.
  - **StarGate starGateOrange**: A sárga csillagkapu adatait tartalmazó osztály referenciája.
  - **boolean open**: Jelzi hogy a féregjárat nyitva van e (van e 2 kapu a játékban).
- **Metódusok**
  - **draw(Graphics2D g)**: Kirajzolja a kapukat a megfelelő helyre.
  - **setStarGate(String color, Vector v)**: Beállítja a megfelelő színű csillagkapu helyzetét.

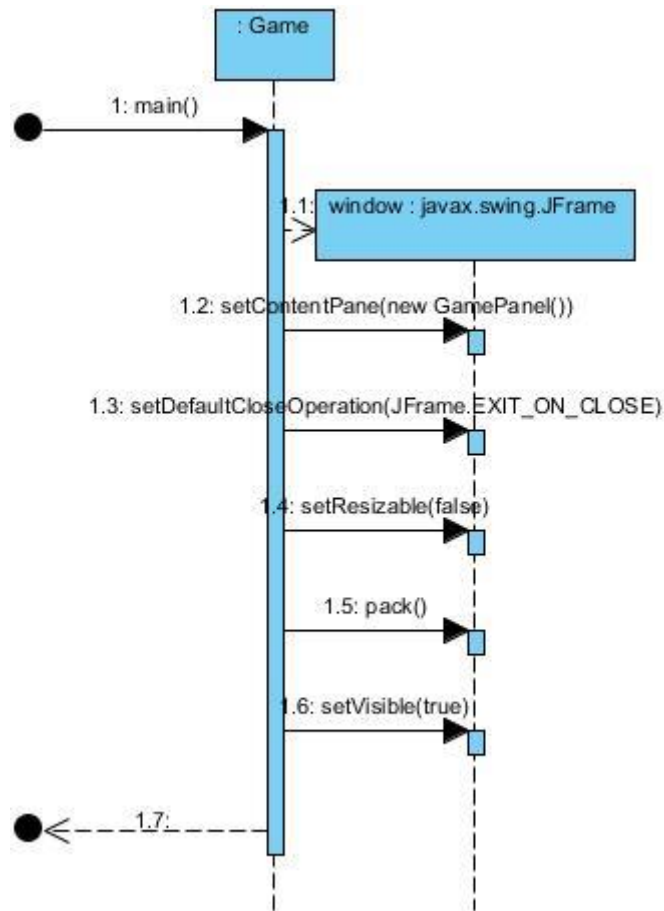
### 3.3.21 ZPM

- **Felelősség**  
A játék célját jelentő Zero Point Module. A játékos felszedheti majd amint már egy ZPM sem maradt a pályán a játékos teljesítette a feladatát.
- **Ősosztályok**  
MapObject
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**  
Nem értelmezhető.

### 3.3.22 Szekvencia diagramok

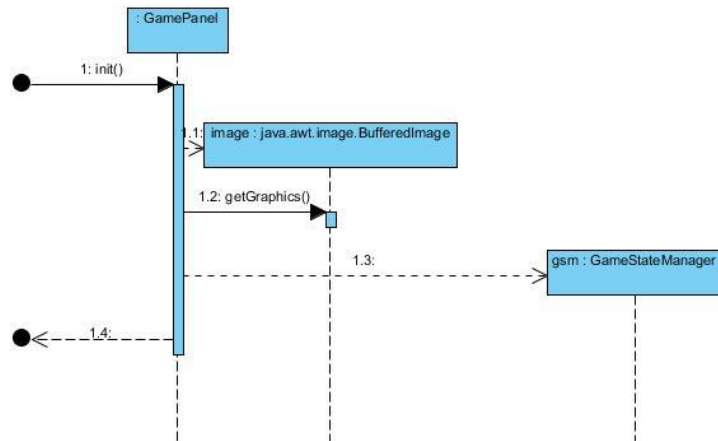


3.2. ábra `GamePanel` inicializálása

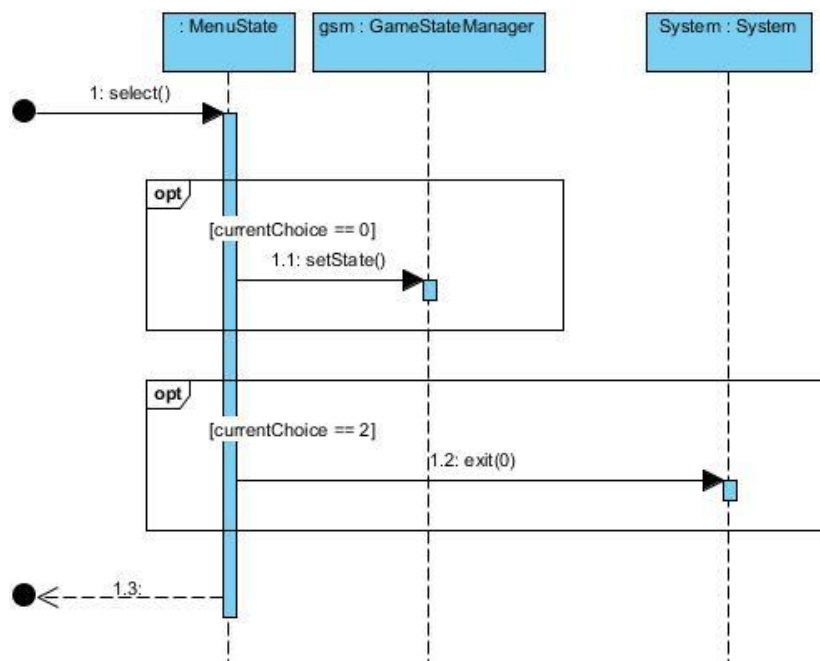


3.3. ábra Ablak inicializálása

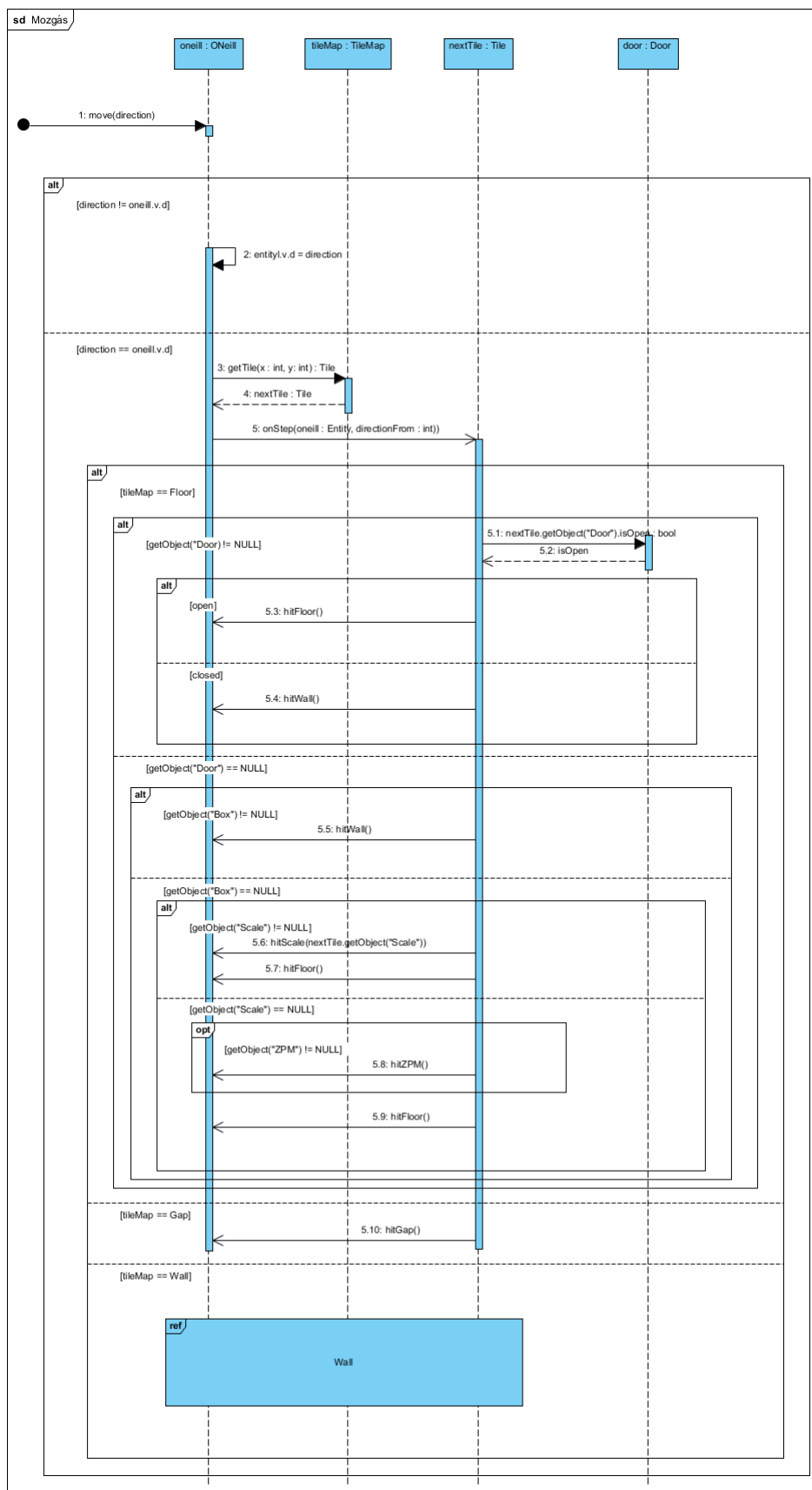


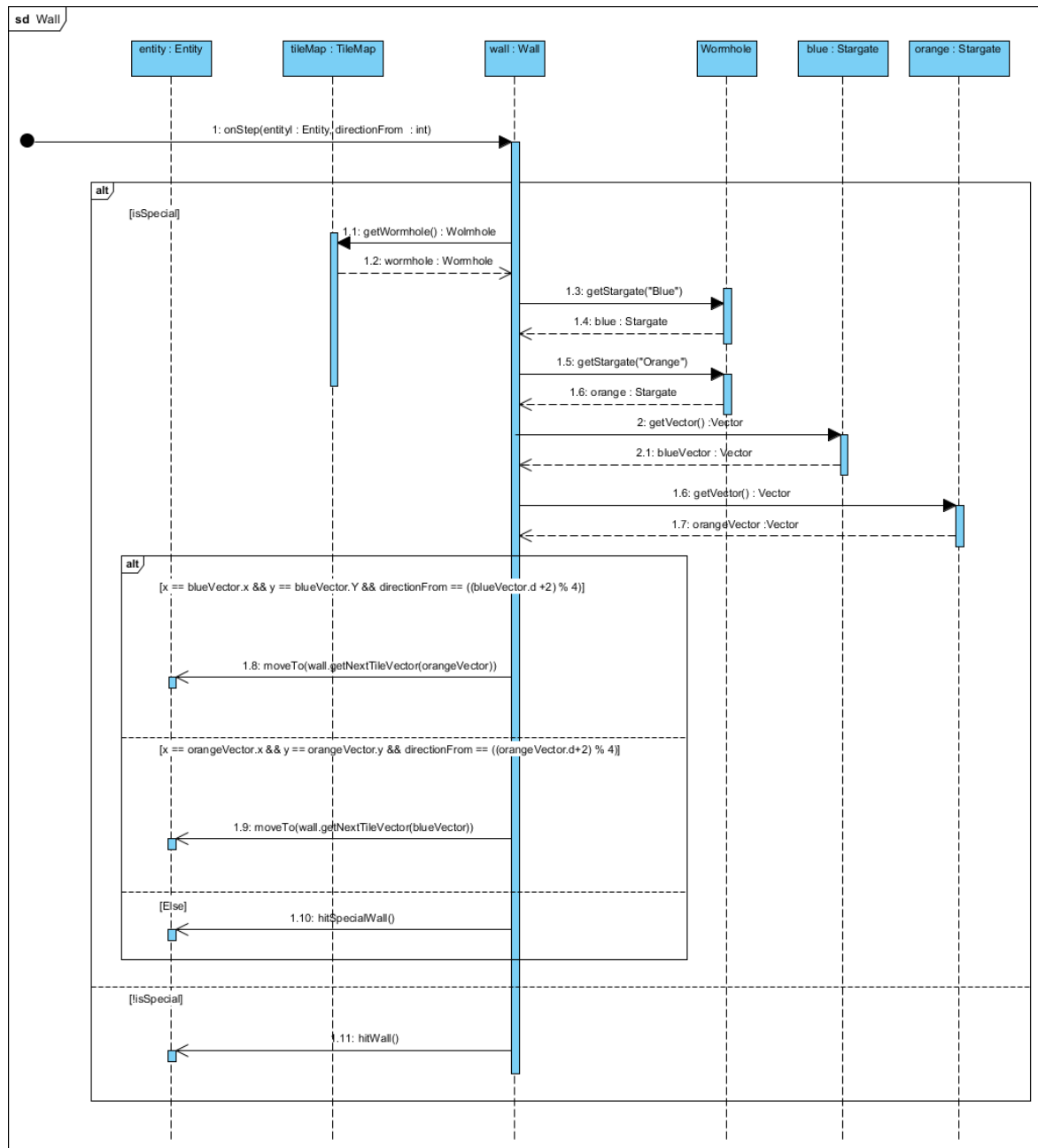


3.4. ábra Játékindítás

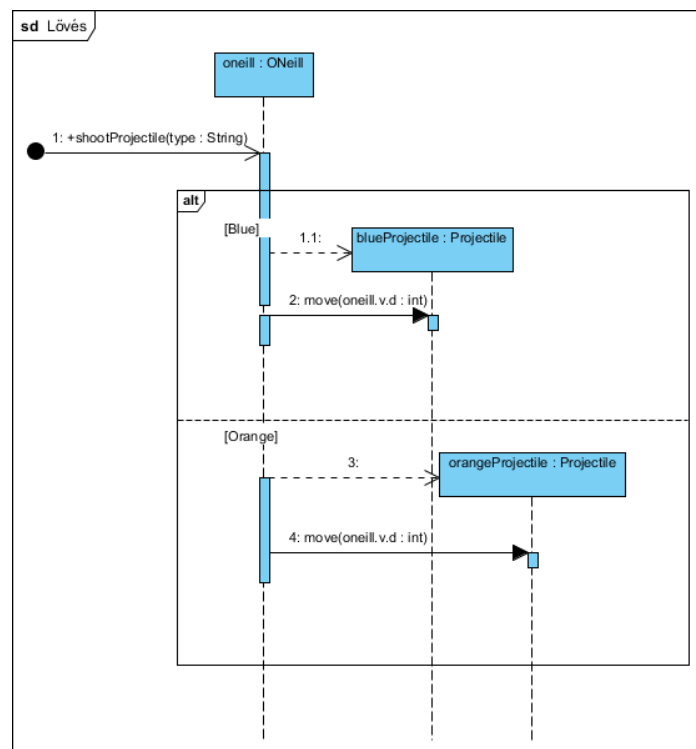


3.5. ábra Főmenü kezelése

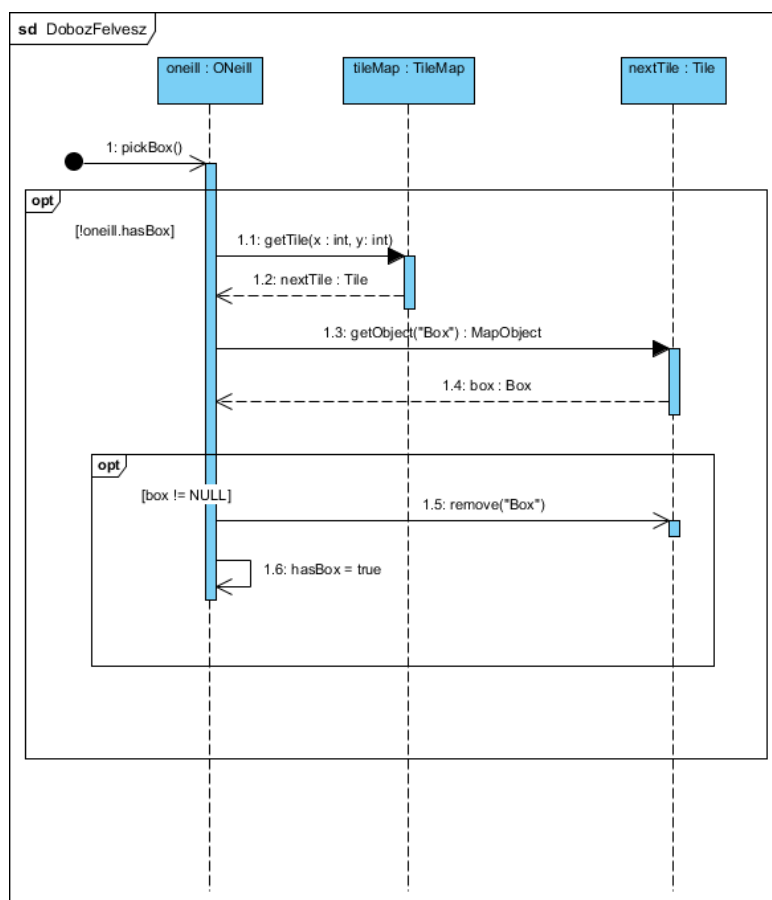




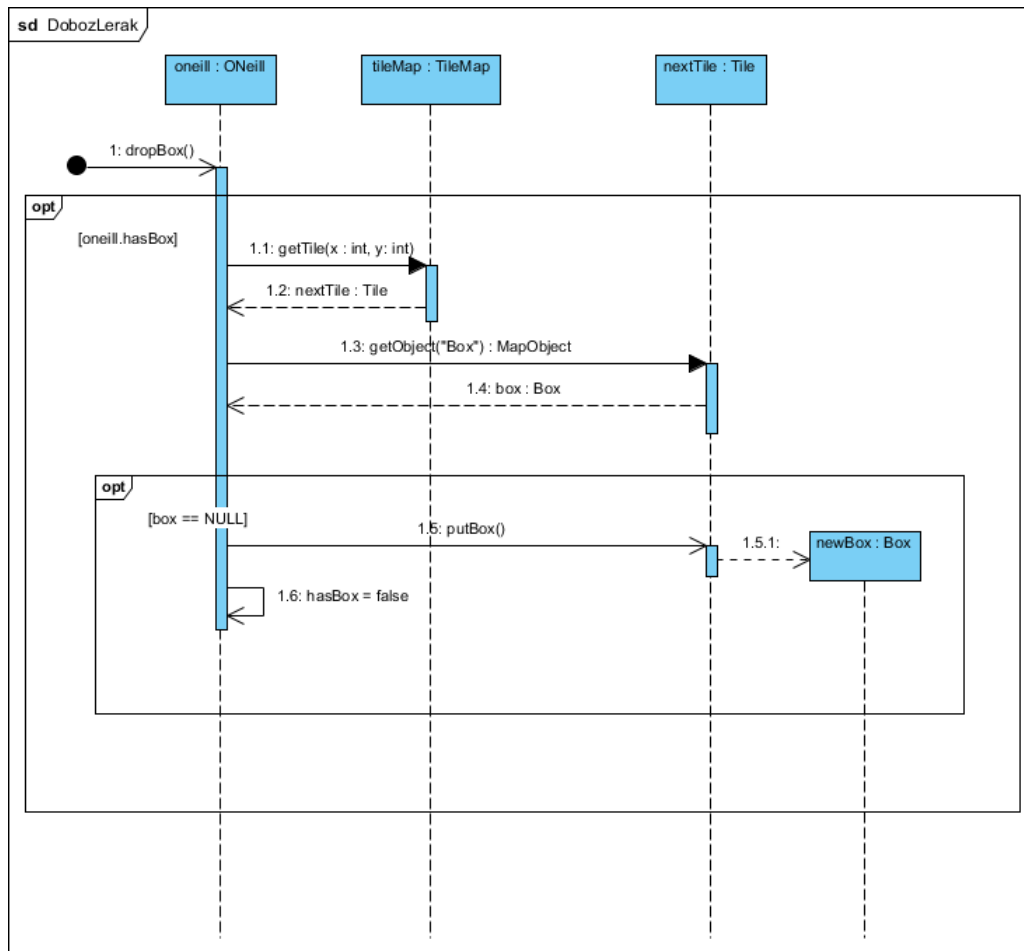
3.7. ábra Falhoz érkezés és teleportálás kezelése



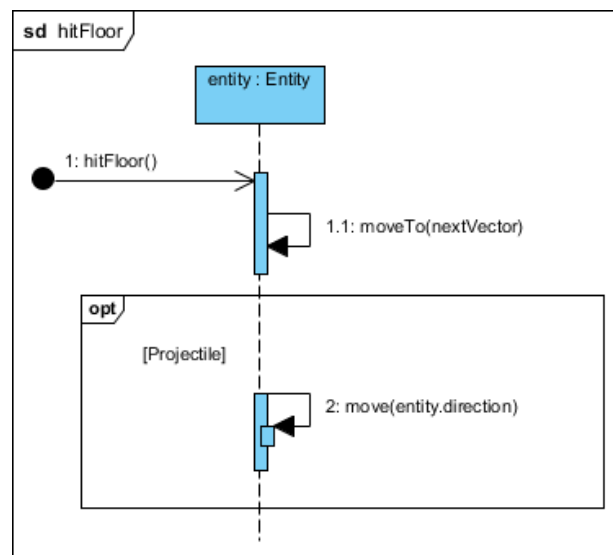
3.8. ábra Lövés



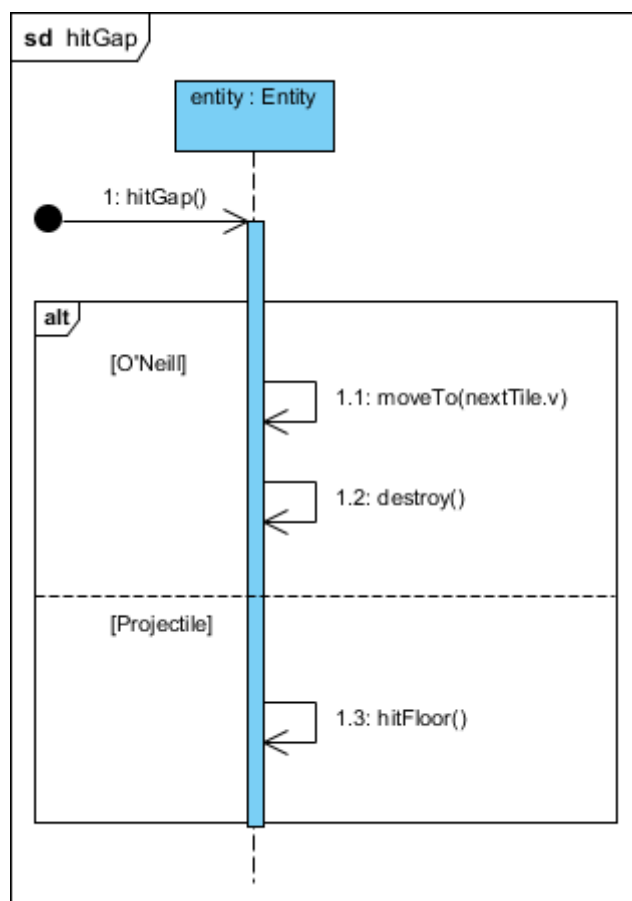
3.9. ábra Doboz felvétele



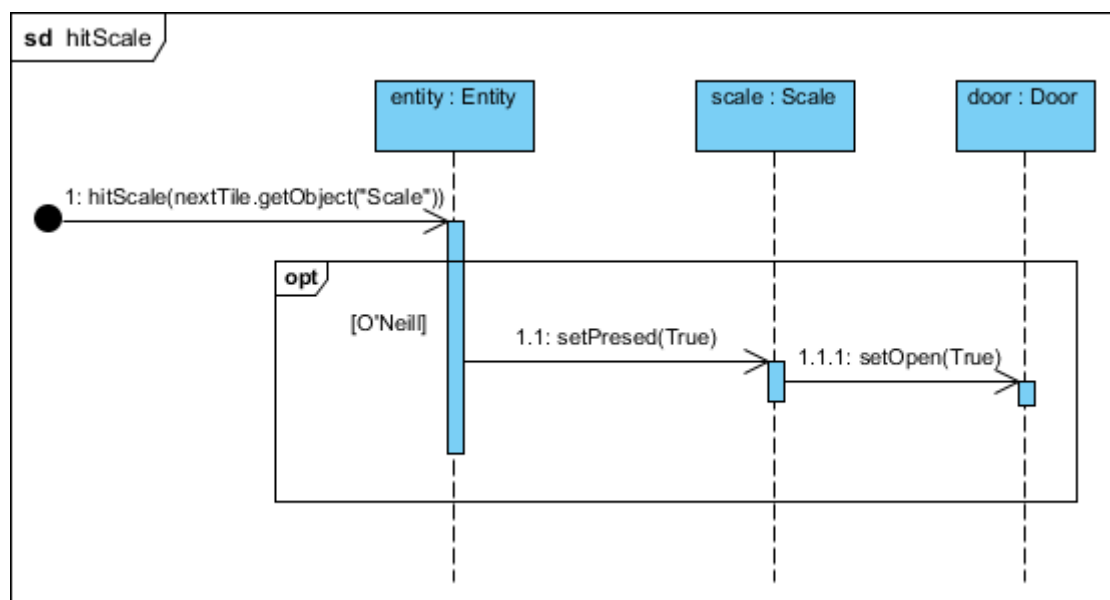
3.10. ábra Doboz lerakása



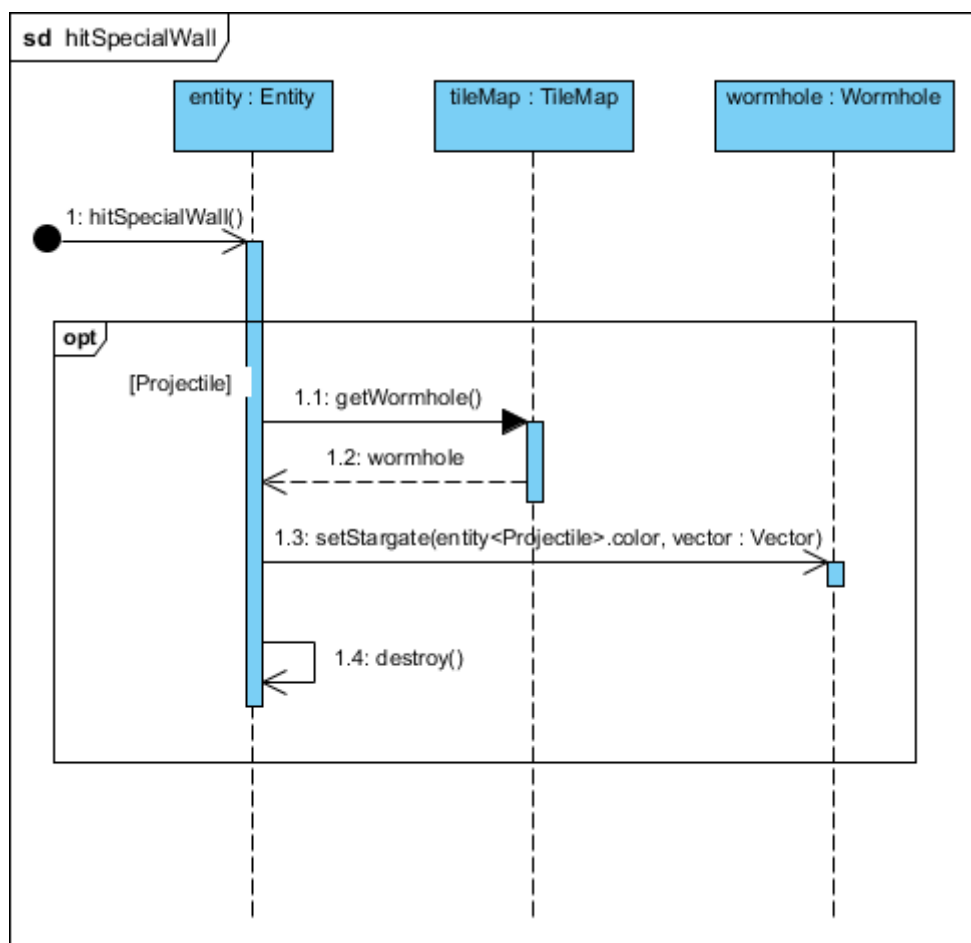
3.11. ábra Padlón mozgás kezelése



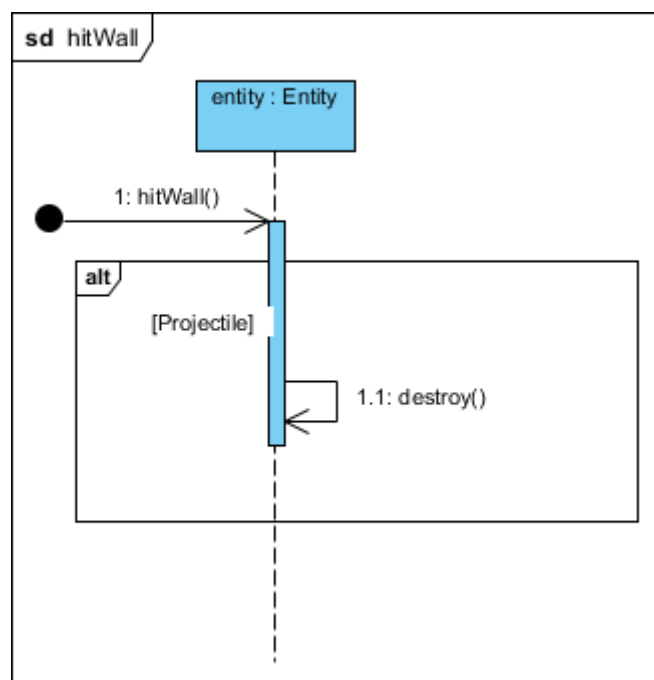
3.12. ábra Szakadék kezelése



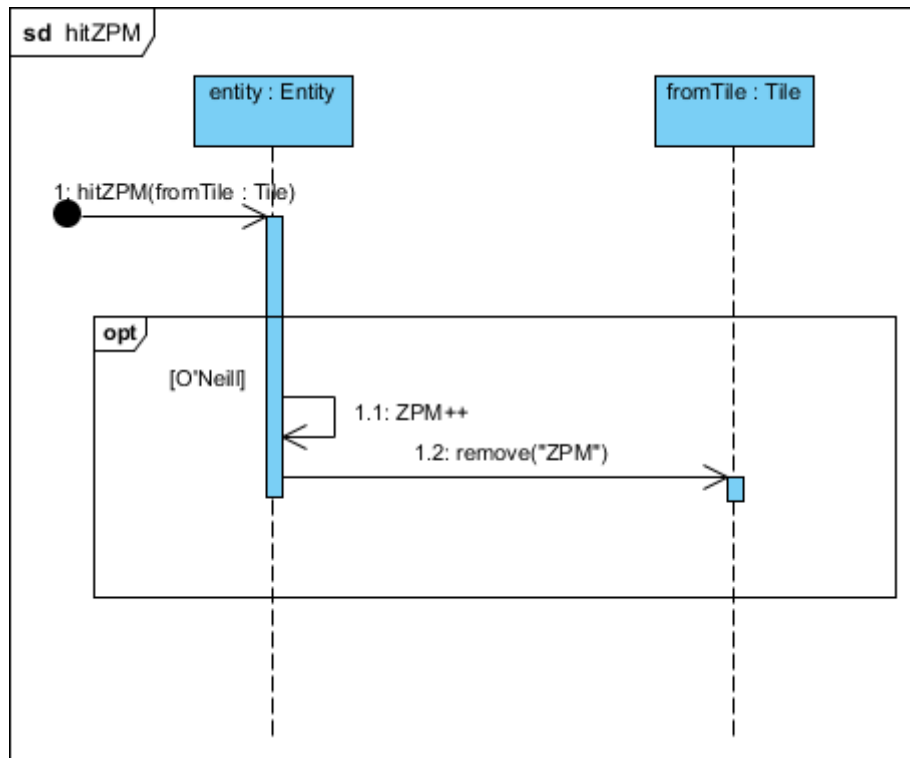
3.13. ábra Mérleg kezelése



3.14. ábra Speciális fal kezelése



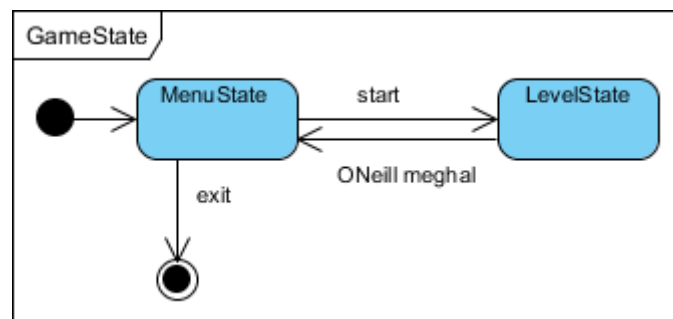
3.15. ábra Normál falba lőtt lövedék kezelése



3.16. ábra ZPM felvétel kezelése

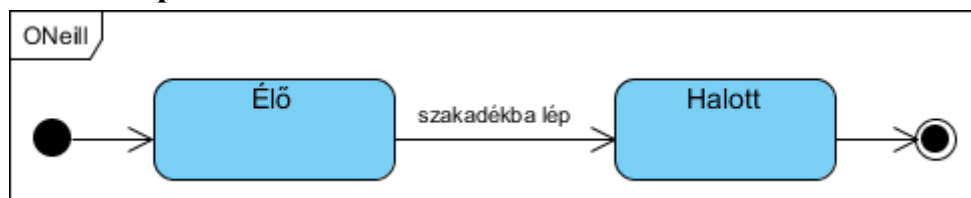
### 3.4 State-chartok

#### 3.4.1 Játékállapotok



3.17. ábra Menü állapotváltozása

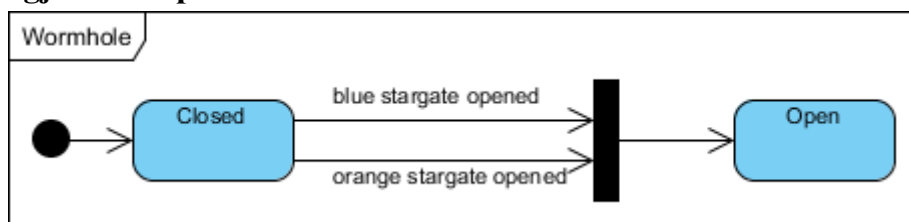
#### 3.4.2 O'Neill állapotai



3.18. ábra O'Neill állapotváltozása

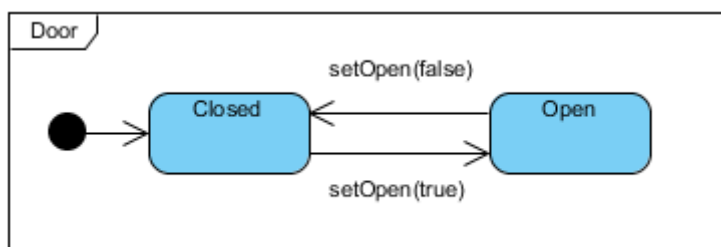


### 3.4.3 Féregjárat állapotai



3.19. ábra Féregjárat állapotváltozása

### 3.5 Ajtó állapotai



3.20. ábra Ajtó állapotváltozása

### 3.6 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2016.03.02. 19:30	2 óra	Bujdosó Bukta Horváth Karácsony Kelemen Vincze	<p>Értekezlet.</p> <p>Döntés: Bujdosó elkészíti az Indítás, Kilépés szekvenciadiagramját. Bukta elkészíti a Mérlegre lépés és a ZPM felvétel szekvencia diagramját. Horváth elkészíti a Forgás és a Teleportálás szekvenciadiagramját. Karácsony elkészíti a Doboz felvétel és Doboz lerakás szekvenciadiagramját. Kelemen elkészíti a Lépés és Halál szekvenciadiagramját. Vincze elkészíti a Lövés, a Narancs csillagkapu és a Kék csillagkapu szekvenciadiagramját.</p> <p>Karácsony a megbeszélésen eldöntöttek alapján megírja a 3.1 fejezetet. A tagok 2016.03.04-én, 03.05-én és 03.06-án megbeszélést tartanak a dokumentum véglegesítésére. A 03.05-i találkozón Karácsony személyesen nem vesz részt, csak telefonon keresztül.</p>

2016.03.02. 23:30	1 óra	Vincze	Tevékenység: Tiled map editor működésének tanulmányozása hogy milyen formátumban és hogyan lehet kimenteni vele pályákat.
2016.03.03. 12:15	2 óra	Vincze	Tevékenység: Osztályok, és szekvenciák előzetes tervezése.
2016.03.03. 20:30	5,5 óra	Vincze	Tevékenység: Osztályok és szekvenciák előzetes tervezése. Egyszerű magyarázó szekvenciadiagramok elkészítése.
2016.03.03. 21:00	0,5 óra	Karácsony	Tevékenység: 3.1 fejezet alapjainak megírása, dokumentum feltöltése közös tárbba.
2016.03.05 11:00	2,5 óra	Bujdosó Bukta Vincze	Tevékenység: Osztályok és szekvenciák tervezése.
2016.03.05 16:00	1,5 óra	Bujdosó Vincze	Tevékenység: Osztályok és szekvenciák tervezése.
2016.03.06 11:30	2,5 óra	Bujdosó Bukta Vincze	Tevékenység: Osztályok és szekvenciák tervezése.
2016.03.06 14:00	6 óra	Bujdosó Bukta Horváth Kelemen Vincze	Értekezlet. Döntés: A program működésének finomítása.

2016.03.06 20:00	3 óra	Bujdosó	Tevékenység: Szekvencia diagramok szerkesztése, az osztálydiagram véglegesítése.
2016.03.06 20:00	3 óra	Bukta	Tevékenység: Osztály diagram és állapot diagram szerkesztése.
2016.03.06 20:00	2,5 óra	Kelemen	Tevékenység: Objektum katalógus véglegesítése, statechartok megrajzolása.
2016.03.06 20:00	6 óra	Horváth	Tevékenység: Szekvenciadiagramok elkészítése.
2016.03.06 24:00	1,5 óra	Karácsony	Tevékenység: Osztálydiagramok leírása.
2016.03.06 24:00	4 óra	Bujdosó	Tevékenység: Osztálydiagramok leírása.
2016.03.07 2:00	2 óra	Bukta	Tevékenység: Osztálydiagramok leírása, osztálydiagram szerkesztése.
2016.03.07 8:00	2 óra	Karácsony	Tevékenység: Dokumentum formázása, hiányzó részek pótlása, pontosítások.

## 4. Analízis modell kidolgozása 2

### 4.1 Objektum katalógus

#### 4.1.1 Game

A játék belépési pontja. Felelőssége, hogy megjeleníti az ablakot, amiben a játék fut.

#### 4.1.2 GamePanel

Ez az objektum felel a játékalak kirajzolásáért, a keret feltöltéséért. Felelőssége továbbá, hogy figyeli a felhasználótól érkező parancsokat (a billentyűzetről.)

#### 4.1.3 GameStateManager

A játék állapotait kezelő objektum. Meghatározza, hogy melyik az éppen aktív játékalap és azt vezérli a GamePaneltől kapott bemenet segítségével.

#### 4.1.4 GameState

A játékalapok osztálya.

##### 4.1.4.1 MenuState

A játék főmenüjét vezérlő objektum. Az előforduló menüpontok, azok műveleteinek elvégzése tartozik a feladatai közé.

##### 4.1.4.2 LevelState

A játék vezérléséért felelős objektum.

#### 4.1.5 TileMap

A pályát fájlból beolvasó és felépítő objektum. Feladata hogy a megadott térképfájl alapján a játékosnak megjelenítse a pályát, amin végig kell haladnia. Ezen kívül ő tárolja a Tile objektumokat koordináták szerint.

#### 4.1.6 Tile

Csempe objektum. Ismeri az adott csempe x,y koordinátáját és a hozzá tartozó textúrát. A hozzá tartozó (rajta lévő) MapObject objektumokat is tárolja, kezeli.

##### 4.1.6.1 Floor

A padló elem, ezekre lépkedhet a játékos és lehetnek rajta egyéb speciális entitások.

##### 4.1.6.2 Gap

A szakadék elem a pályán. A játékos belelépve meghal, a dobozt erre lerakva megsemmisül, a lövedék áthalad felette.

##### 4.1.6.3 Wall

A pályán lévő falak. Erre az elemre sem a játékos nem tud rálépni, sem lövedék nem haladhat át rajta. Dobozt, ZPM-et nem tartalmazhat ez a mező. Speciális változata is létezik, azon csillagkapu nyílhat.

#### 4.1.7 MapObject

A pályán található entitások. A Box, ZPM osztálya.

#### **4.1.7.1 Box**

A feladatkiírásban dobozként említett objektumot reprezentálja. A játékos fel tudja venni és másik helyre tudja cipelni. Felelőssége, hogy mérlegre helyezve lenyomja azt és így a mérleg ajtót tud nyitni.

#### **4.1.7.2 ZPM**

A játékos által összegyűjtendő energiamodult reprezentáló objektum. Külön felelőssége nincs, a játékos ezeket gyűjti össze a játék során.

#### **4.1.7.3 Door**

A feladatkiírásban ajtóként említett objektumot reprezentálja. Egy ajtó egy mérleggel (Scale) van összekapcsolva. Amennyiben a mérlegen megfelelő súly van, az ajtó nyitott állapotú lesz.

#### **4.1.7.4 Scale**

A feladatkiírásban mérlegként említett objektumot reprezentálja. Egy mérleg egy ajtóval (Door) van összekapcsolva. Amennyiben a mérleg lenyomott állapotban van (azaz megfelelő súly van rajta), a hozzá tartozó ajtó nyitott állapotba kerül. A játékos rá tud állni vagy rá tud helyezni egy dobozt. Felelőssége, hogy vele lehet kinyitni az ajtót, amin O'Neill átmehet.

### **4.1.8 Entity**

A mozgásra képes objektumok őssztálya. Belőle származik le a játékos (O'Neill) és a lövedék (Projectile). Az Entity osztály oldja meg a csempékkel való kommunikációt.

#### **4.1.8.1 O'Neill**

A játékost reprezentáló osztály. Felvehet dobozt illetve az összegyűjtött ZPM-ek számát tárolja. Ő felel a kilőhető narancssárga és kék lövedékekért.

#### **4.1.8.2 Projectile**

O'Neill által kilőhető lövedék. Kék vagy narancssárga verzióban létezhet. Kilövés után a pályán halad. Speciális falba csapódva csillagkapu jön létre a becsapódás helyén a megfelelő színben.

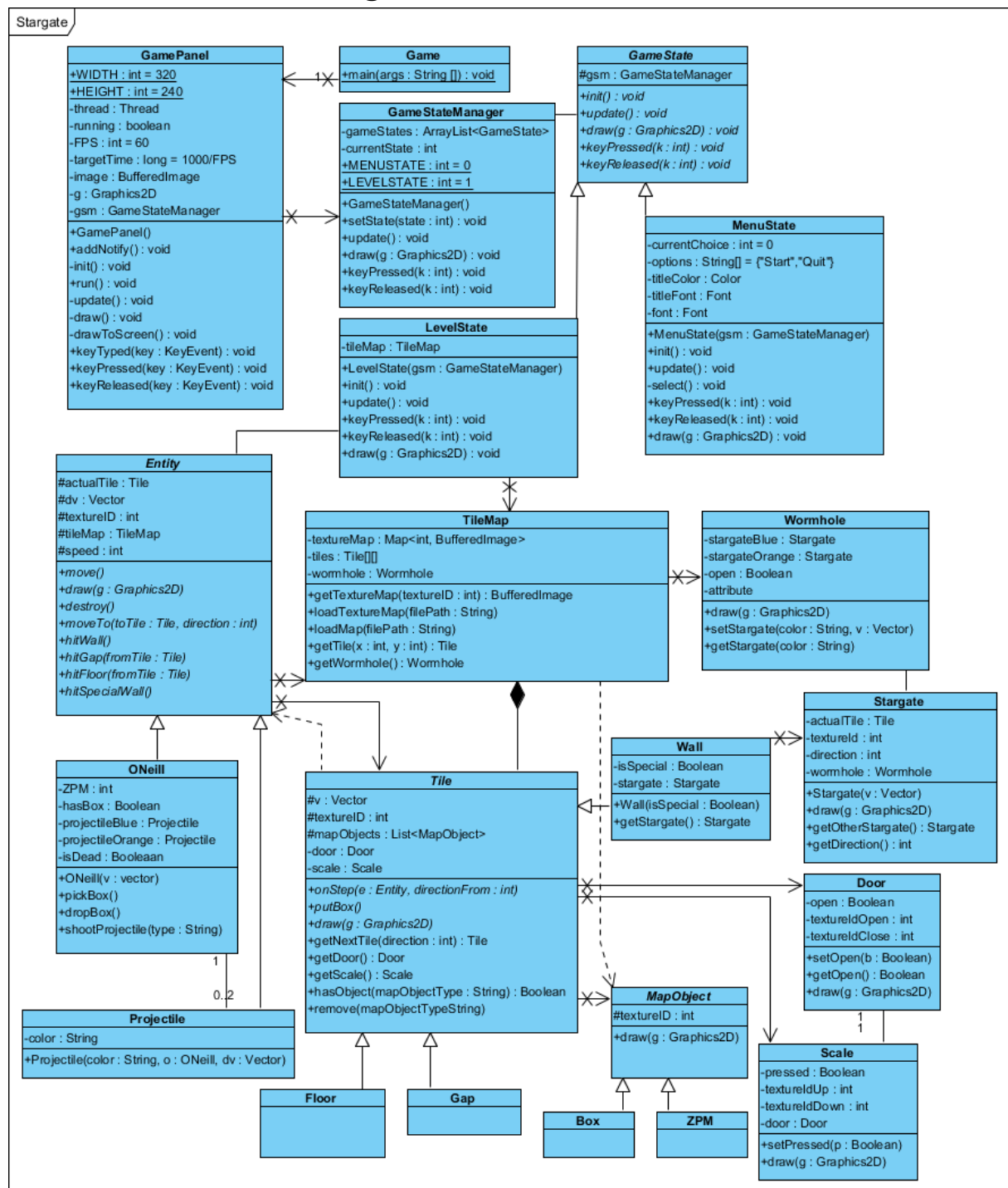
### **4.1.9 Wormhole**

A főregjáratot kezelő objektum. Ez kezeli a főregjárat két végén lévő csillagkapukat (Stargate).

#### **4.1.10 Stargate**

Egy csillagkaput és annak paramétereit tároló objektum.

## 4.2 Statikus struktúra diagramok



#### 4.1. ábra Osztálydiagram

## 4.3 Osztályok leírása

### 4.3.1 Box

- **Felelősség**  
A játékos által fel és letehető doboz objektumot leíró osztály. A dobozt a játékos felveheti majd azt egy másik alkalmas csempére ráteheti. A dobozon átsétálni és átlőni sem lehet.
- **Ősosztályok**  
MapObject
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**  
Nem értelmezhető.

### 4.3.2 Door

- **Felelősség**  
Az ajtót leró osztály, feladata, hogy nyilvántartja, hogy az adott ajtó nyitva van-e, és ezáltal átjárható-e O'Neill illetve a lövedékek számára.
- **Ősosztályok**  
MapObject
- **Attribútumok**
  - **boolean open:** Változó, amiben azt tároljuk, hogy az ajtó nyitva van vagy sem.
  - **int textureIDOpen:** A nyitott ajtóhoz tartozó grafika.
  - **int textureIDClosed:** A zárt ajtóhoz tartozó grafika.
- **Metódusok**
  - **void setOpen(boolean b):** Beállítja az ajtó nyitott állapotát.
  - **boolean getOpen():** Visszaadja, hogy az ajtó nyitva van-e.
  - **void draw(Graphics2d g):** Kirajzolja az adott állapotú ajtót (nyitott vagy zárt.)

### 4.3.3 Entity

- **Felelősség**  
A játékos által „irányított” és animált objektumok ősosztálya.
- **Ősosztályok**  
Nem értelmezhető.
- **Attribútumok**
  - **Tile actualTile:** Az Entity aktuális csempéjét jelző változó.
  - **Vector dv:** Az Entity abszolút koordinátáit tartalmazó változó.
  - **int textureID:** Az Entityhez tartozó grafika azonosítója.



- **TileMap tileMap:** A játékeret tartalmazó TileMap referenciája.
- **int speed:** Az Entitás mozgási sebessége.
- **Metódusok**
  - **void move():** Mozgatja az adott entitást, lekérdezi az éppen aktuális csempétől hogy mi történik ha rálép.
  - **void moveTo(Tile toTile, int direction):** Az adott csempére mozgatja az entitást és a megadott irányba állítja.
  - **void draw(Graphics2D g):** Kirajzolja az entitást.
  - **void destroy():** Megsemmisíti az entitást.
  - **void hitWall():** Akkor hívódik, ha az entitás falhoz ér.
  - **void hitGap():** Akkor hívódik, ha az entitás szakadékhoz ér.
  - **void hitFloor():** Akkor hívódik, ha az entitás padlóra lép.
  - **void hitSpecialWall():** Akkor hívódik meg, ha az entitás speciális falra lép(ne).

#### 4.3.4 Floor

- **Felelősség**

A labirintus egyik alapeleme. Ezen a csempén a játékos szabadon mozoghat, illetve vehet fel vagy tehet le dobozokat. A padlón szabadon sétálni s átlőni is lehetséges.
- **Össztályok**

Tile
- **Attribútumok**

Nem értelmezhető.
- **Metódusok**
  - **void onStep(Entity e, int directionFrom):** Ez a metódus hívódik abban az esetben, ha egy Entity (O'Neill vagy egy lövedék) „rálép” a csempéje. Ezt minden csempetípus felülírja, hogy a számára megfelelő működést valósítsa meg. Floor esetén az aktuális csempére lépteti a játékost vagy a lövedéket.
  - **void putBox():** Ez a metódus kezeli, ha a nálunk lévő dobozt a padlóra helyezzük.
  - **void draw(Graphics2D texture):** Ez a metódus rajzolja ki a játékos által látott textúrát az adott padlóhoz.
  - **void killZPM():** A ZPM felvételét kezelő metódus.

### 4.3.5 Game

- **Felelősség**  
A játék futtatásáért felelős osztály.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**
  - **void main(String args[]):** Ez a játék main metódusa, ami a játék futtatását végzi.

### 4.3.6 GamePanel

- **Felelősség**  
A GamePanel osztály indítja a játékot futtató szálát, valamint tartalmazza játékciklust (gameloop). Ez az osztály tölti ki a Game osztályban létrehozott ablakot, valamint fogadja a bemenetet a billentyűzetről és továbbadja azokat.
- **Ősosztályok**  
java.lang.Object -> java.awt.Component -> java.awt.Container -> javax.swing.JComponent
- **Interfészek**  
Runnable, KeyListener
- **Attribútumok**
  - **int WIDTH:** A játéklablak szélessége. Default értéke 320.
  - **int HEIGHT:** A játéklablak magassága. Default értéke 240.
  - **Thread thread:** A játékot futtató szálát ebben az objektumban tároljuk.
  - **boolean running:** Logikai érték, mely azt mutatja, hogy a játék fut vagy sem.
  - **int FPS:** A játék FPS értékét tartalmazó egész szám. Default értéke 60.
  - **long targetTime:** Az újrarajzoltatás frekvenciáját tároló érték. Default értéke 1000 ms/FPS.
  - **BufferedImage image:** A játék képét tartalmazó mutató.
  - **Graphics2D g:** A kirajzolásért felelős objektum példánya.
  - **GameStateManager gsm:** A játékmenet kezelését végző GameStateManager objektum egy példánya.
- **Metódusok**
  - **GamePanel():** Az osztály konstruktora.
  - **void addNotify():** Kapcsolatot létesít a natív kimenet és a program között.
  - **void run():** A játékot futtató szál elindításáért felelős metódus.

- **void keyTyped(KeyEvent key)**: Billentyűzeten érkező karaktersorozatot figyelő metódus.
- **void keyPressed(KeyEvent key)**: Egy billentyű leütését figyelő metódus.
- **void keyReleased(KeyEvent key)**: Egy billentyű felengedését figyelő metódus.

#### 4.3.7 GameState

- **Felelősség**

A játékalapok absztrakt osztálya, tartalmazza a megvalósítandó metódusokat valamint egy GameStateManager-t.

- **Attribútumok**

- **GameStateManager gsm**: A játék állapotát kezelő GameStateManager objektum példánya.

- **Metódusok**

- **abstract void init()**: Játékalapot inicializálására szolgáló metódus.
- **abstract void update()**: A játékalapot frissítését végző metódus.
- **abstract void draw(Graphics2D g)**: Ez a metódus rajzolja ki a játékos által látott képet a játékalapba.
- **abstract void keyPressed(int k)**: Billentyű lenyomását figyelő metódus.
- **abstract void keyReleased(int k)**: Billentyű felengedését figyelő metódus.

#### 4.3.8 GameStateManager

- **Felelősség**

A játék különböző állapotai közötti váltásért felelős osztály. Ez az osztály kapja meg a GamePanel osztálytól a leütött billentyűket, majd továbbadja őket az éppen aktuális játékalapnak.

- **Attribútumok**

- **ArrayList<GameState> gameStates**: A játék során előforduló állapotokat tároló lista.
- **int currentState**: A játék jelenlegi állapotát reprezentáló változó.
- **int MENUSTATE**: A játék menüjének állapotát reprezentáló változó. Default értéke 0.
- **int LEVELSTATE**: A választott pálya állapotát reprezentáló változó. Default értéke 1.

- **Metódusok**

- **GameStateManager()**: Az osztály konstruktora.
- **void setState(int state)**: A játék állapotának beállítása ezen a metóduson keresztül történik.
- **void update()**: A játékalapot frissítését végző metódus. Az aktuális játékalapot update() metódusát hívja.
- **void draw(Graphics2D g)**: Ez a metódus rajzolja ki a játékos által látott képet a játékalapba.
- **void keyPressed(int k)**: Billentyű lenyomását figyelő metódus. Továbbítja az aktuális játékalap felé.

- **void keyReleased(int k):** Billentyű felengedését figyelő metódus. Továbbítja az aktuális játékállapot felé.

#### 4.3.9 Gap

- **Felelősség**

Egy olyan speciális csempe amire ha O'Neill rálép, akkor meghal, valamint ha egy tárgy kerül egy ilyen csempére az megsemmisül.

- **Össztályok**

Tile

- **Metódusok**

- **void onStep(Entity e, int directionFrom):** Ez a metódus kezeli, ha szakadékra lépünk O'Neill ezredessel (aminek következtében meghalunk.)
- **void putBox():** Szakadékba rakott dobozt lekezelő metódus (aminek következtében a doboz megsemmisül.)
- **void draw(Graphics2D texture):** Ez a metódus rajzolja ki a játékos által látott textúrát az adott szakadékhoz.

#### 4.3.10 LevelState

- **Felelősség**

A játékmenetért felelős játékállapot (GameState) osztály. Ő felel a labirintusért illetve a labirintusban szereplő elemekért, mint például O'Neill ezredes.

- **Össztályok**

GameState

- **Attribútumok**

- **TileMap tileMap:** A pálya alapját adó csempéket tartalmazó objektum.

- **Metódusok**

- **LevelState(GameStateManager gsm):** A pálya konstruktora.
- **void init():** A pálya inicializálását végző metódus.
- **void update():** A pálya frissítését végző metódus.
- **void draw(Graphics2D g):** Feladata hogy meghívja a kirajzolandó objektumok draw metódusát.
- **void keyPressed(int k):** Billentyű lenyomását figyelő metódus valamint kezelő metódus.
- **void keyReleased(int k):** Billentyű lenyomását figyelő metódus valamint kezelő metódus.

#### 4.3.11 MapObject

- **Felelősség**  
Ezen osztály alá a játékkossal interakcióba lépésre képes osztályok tartoznak
- **Attribútumok**
  - **int textureID**: Az adott textúra-elem sorszámát tartalmazza.
- **Metódusok**
  - **void draw(Graphics2D texture)**: Ez a metódus rajzolja ki a játékos által látott textúrát az adott elemhez.

#### 4.3.12 MenuState

- **Felelősség**  
A játék főmenüjéért felelős osztály, ezen osztály felelősségem hogy a képernyőre a megfelelő menüpontok legyenek kirajzolva, majd a megfelelő billentyűk leütése után elindítja a játékot vagy kilép belőle.
- **Ősosztályok**  
GameState
- **Attribútumok**
  - **int currentChoice**: A felhasználó által akutálisan kiválasztott menüpont értéke. Default értéke 0.
  - **String[] options**: A menü elérhető állapotait tartalmazza. Lehetséges értékei "Start" és "Quit"
  - **Color titleColor**: A főcím színét tartalmazó változó.
  - **Font titleFont**: A főcím betűtípusát tartalmazó változó.
  - **Font font**: A menüben előforduló betűtípust tartalmazó változó.
- **Metódusok**
  - **MenuState(GameStateManager gsm)**: A menü konstruktora.
  - **void init()**: A menü inicializálását végző metódus.
  - **void update()**: A menü frissítését végző metódus.
  - **void draw(Graphics2D g)**: Ez a metódus rajzolja ki a menüpontokat a képernyőre valamint jelöli a játékos által épp kiválasztott menüpontot
  - **void keyPressed(int k)**: Billentyű lenyomását figyelő metódus.
  - **void keyReleased(int k)**: Billentyű felengedését figyelő metódus.

#### 4.3.13 ONeill

- **Felelősség**  
A játékost megszemélyesítő osztály, feladata hogy kövesse a játékos utasításait és annak megfelelően cselekedjen, lépjen, forduljon, meghaljon, stb.
- **Ősosztályok**  
Entity

- **Attribútumok**
  - **int ZPM:** Az összegyűjtött ZMP-ek száma.
  - **boolean hasBox:** Megmondja hogy van e éppen O'Neill kezében doboz.
  - **Projectile projectileBlue:** A kilőtt kék lövedék referenciája (ha van ilyen)
  - **Projectile projectileOrange:** A kilőtt sárga lövedék referenciája (ha van ilyen)
  - **boolean isDead:** Megmondja, hogy O'Neill ezredes halott-e.
- **Metódusok**
  - **ONeill(Vector v):** Konstruktor, beállítja a kezdőpozíciót.
  - **void pickBox():** Doboz felvétele a következő mezőről.
  - **void dropBox():** Doboz letétele a következő mezőre.
  - **void shootProjectile(String type):** Lövedék útnak indítása.
  - **hitWall():** Akkor hívódik, ha az entitás falhoz ér, jelen esetben nem engedi továbblépni a karaktert.
  - **hitGap():** Akkor hívódik, ha az entitás szakadékhoz ér, jelen esetben megöli O'Neill-t.
  - **hitFloor():** Akkor hívódik, ha az entitás padlóra lép, jelen esetben tovább lépteti a karaktert és figyel, hogy mérlegre vagy ZPM-re lépett-e.
  - **hitSpecialWall():** Akkor hívódik, ha az entitás speciális falra lép(ne), jelen esetben ugyan az a funkciója mint a hitWall-nak.

#### 4.3.14 Projectile

- **Felelősség**

A játékos által kilőtt lövedék.
- **Ősosztályok**

Entity
- **Attribútumok**
  - **String color:** A lövedék színe szövegesen ("blue", "orange").
- **Metódusok**
  - **Projectile(String color, O'Neill o, Vector dv):** Konstruktor, ami megkapja a lövedék színét, O'Neill-t, és a kezdő pozícióját.
  - **hitWall():** Akkor hívódik, ha az entitás falhoz ér, jelen esetben a lövedék megsemmisül.
  - **hitGap():** Akkor hívódik, ha az entitás szakadékhoz ér, jelen esetben a lövedék tovább lép.
  - **hitFloor():** Akkor hívódik, ha az entitás padlóra lép, jelen esetben a lövedék tovább lép.
  - **hitSpecialWall():** Akkor hívódik, ha az entitás speciális falra lép(ne), jelen esetben a lövedék megsemmisül és egy kaput nyit az adott falon.

#### 4.3.15 Scale

- **Felelősség**

A mérleget leíró osztály, feladata hogy kinyissa a hozzá tartozó ajtót ha sújt tesznek rá.
- **Ősosztályok**

## MapObject

- **Attribútumok**
  - **boolean pressed:** Ez a változó jelzi hogy a mérlegen van-e súly.
  - **int textureIDUp:** A súlyozatlan mérleg grafikája.
  - **int textureIDDown:** A súlyozott mérleg grafikája.
  - **Door door:** A mérleghez tartozó ajtó referenciája.
- **Metódusok**
  - **void setPressed(boolean b):** Beállítja a pressed változó értékét.
  - **void draw(Graphics2D g):** Kirajzolja a mérleget a megfelelő állapotában.

### 4.3.16 Stargate

- **Felelősség**

A csillagkapuk adatainak tárolását végző osztály, tárolja a kapuk helyzetet valamint irányát.
- **Attribútumok**
  - **Tile actualTile:** A csillagkapu aktuális pozíciója.
  - **int textureID:** A kapuk grafikájának azonosítója.
  - **int direction:** Az irány, amerre a kapu néz.
  - **Wormhole wormhole:** Az adott csillagkapuhoz tartozó féregjárat referenciáját tárolja.
- **Metódusok**
  - **Stargate (Vector v):** A csillagkaput létrehozza a megadott pozícióban.
  - **void draw(Graphics2D g):** kirajzolja a kapukat a megfelelő helyen.
  - **int getDirection():** Megadja a kapu irányát.
  - **Stargate getOtherStargate():** Visszaadja a pályán található másik csillagkaput, ha van.

### 4.3.17 Tile

- **Felelősség**

A játékkeret felépítő labirintus alap építőegysége.
- **Attribútumok**
  - **Vector v:** csempe labirintusbeli koordinátáit tároló segédosztály
  - **int textureID:** A csempehez tartozó grafika azonosítóját tartalmazó változó.
  - **List<MapObject> mapObjects:** A csempén található MapObject-ek egy listában tárolva.
  - **Door door:** A csempehez tartozó ajtót tárolja.
  - **Scale sacle:** A csempehez tartozó mérleget tárolja.
- **Metódusok**
  - **void onStep(Entity e, int directionFrom):** Ez a metódus hívódik abban az esetben ha egy Entity (O'Neill vagy egy lövedék) „rálép” a csempére a megadott irányból.

- **void putBox()**: Ez a metódus hívódik meg abban az esetben, ha a játékos egy dobozt akar rátenni a csempére.
- **void draw(Graphics2D g)**: Ez a metódus lekérdezi a grafikákat tartalmazó Map-tól a hozzá tartozó grafikát majd kirajzolja azt illetve a vele egy koordinátán szereplő MapObjectek draw metódusait is meghívja.
- **Tile getNextTile(int direction)**: A megadott irányban lévő következő csempét adja vissza.
- **Door getDoor()**: Visszaadja az ajtót az adott csempén.
- **Scale getScale()**: Visszaadja a méreteget az adott csempén.
- **boolean hasObject(String mapObjectType)**: Visszaadja, hogy az adott csempén van-e valamilyen objektum (doboz vagy ZPM.)
- **void remove(String mapObjectType)**: Törli az adott csempéről az objektumot (pl. ZPM felvételekor.)

#### 4.3.18 TileMap

- **Felelősség**

Ez az osztály tárolja magában a labirintust, illetve az ahhoz szükséges grafikai elemeket. A labirintus valamint a hozzá tartozó csempegrafika beolvasás is az ő feladata.

- **Attribútumok**

- **Map<int, BufferedImage> textureMap**: Ez az attribútum tárolja a pályát alkotó csempék képeit és azok sorszámár kulcs-érték párokban.
- **Tile[][] tiles**: A csempék ebben a tömbben vannak eltárolva.
- **Wormhole wormhole**: A pályán előforduló féregjáratot ebben tároljuk.

- **Metódusok**

- **BufferedImage getTextureMap(int textureID)**: Ez a függvény visszaadja az adott ID-hoz tartozó csempe képét.
- **void loadTextureMap(String filePath)**: A pályán található csempék képét ez a függvény olvassa be egy fájlból.
- **void loadMap(String filePath)**: A pályát(labirintust) egy fájlból beolvasó metódus.
- **Tile getTile(int x, int y)**: A megadott koordinátákon található csempét adja meg.
- **Wormhole getWormhole()**: Megadja a pályán működő féregjáratot (ha van ilyen.)

#### 4.3.19 Wall

- **Felelősség**

Speciális csempe, feladata hogy elválasztóelemként működjön a játékban, korlátozva a játékost a mozgásban valamint bizonyos falakon lehet csillagkaput is nyitni. A falon sem átsétálni se átlőni nemlehet.

- **Össztályok**

Tile

- **Attribútumok**

- **boolean isSpecial**: Ez az attribútum jelöli, ha egy fal speciális, azaz lehet rajta csillagkaput nyitni.



- **Stargate stargate:** Az adott falhoz tartozó csillagkapu referenciáját tárolja.
- **Metódusok**
  - **Wall(boolean isSpecial):** Fal konstruktora (speciális fal esetén máshogy néz ki a fal.)
  - **Stargate getStargate():** A falhoz tartozó csillagkaput adja vissza.
  - **void onStep(Entity e, int directionFrom):** Ez a metódus kezeli, ha falba próbálunk belemenni, amit a feladat értelmezése szerint nem tudunk megtenni.
  - **void putBox():** Ez a metódus kezeli, ha a nálunk lévő dobozt a falba akarnánk elhelyezni.
  - **void draw(Graphics2D texture):** Ez a metódus rajzolja ki a játékos által látott textúrát az adott falhoz.

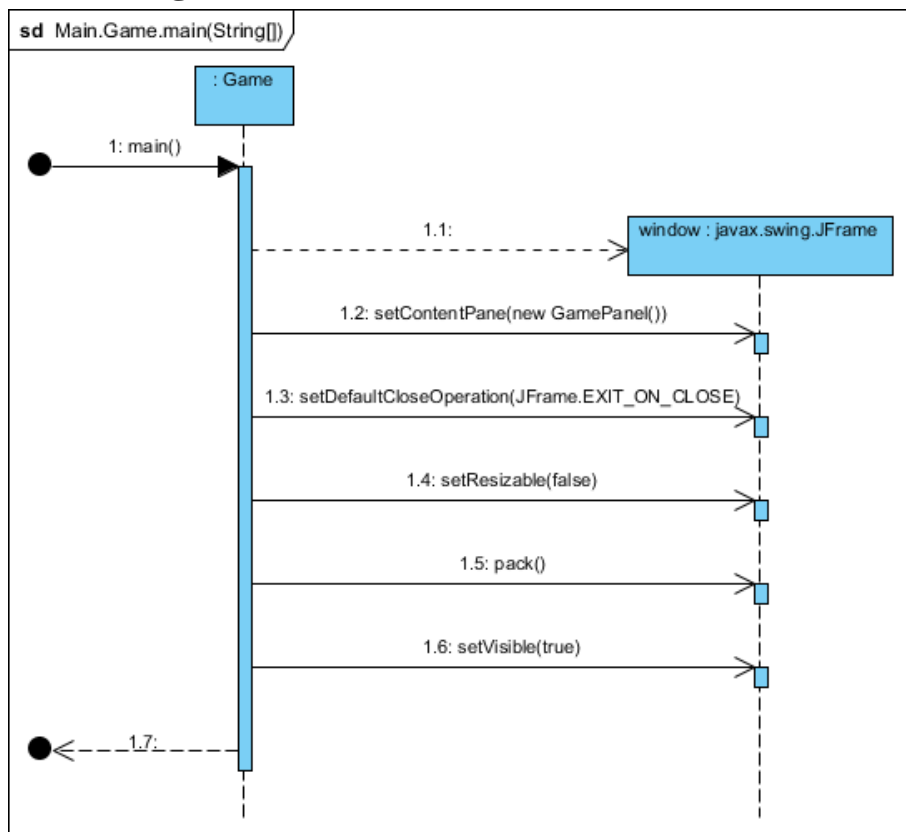
#### 4.3.20 Wormhole

- **Felelősség**  
A pályán található csillagkapuk és féregjárat működéséért felelős osztály. Ő tárolja a kapuk helyét és ügyel arra, hogy egy fajta kapuból csak 1 darab létezzen.
- **Attribútumok**
  - **StarGate starGateBlue:** A kék csillagkapu adatait tartalmazó osztály referenciája.
  - **StarGate starGateOrange:** A sárga csillagkapu adatait tartalmazó osztály referenciája.
  - **boolean open:** Jelzi hogy a féregjárat nyitva van e (van e 2 kapu a játékban).
- **Metódusok**
  - **void draw(Graphics2D g):** Kirajzolja a kapukat a megfelelő helyre.
  - **void setStarGate(String color, Vector v):** Beállítja a megfelelő színű csillagkapu helyzetét.
  - **void getStargate(String color):** Megadja a megadott csillagkaput.

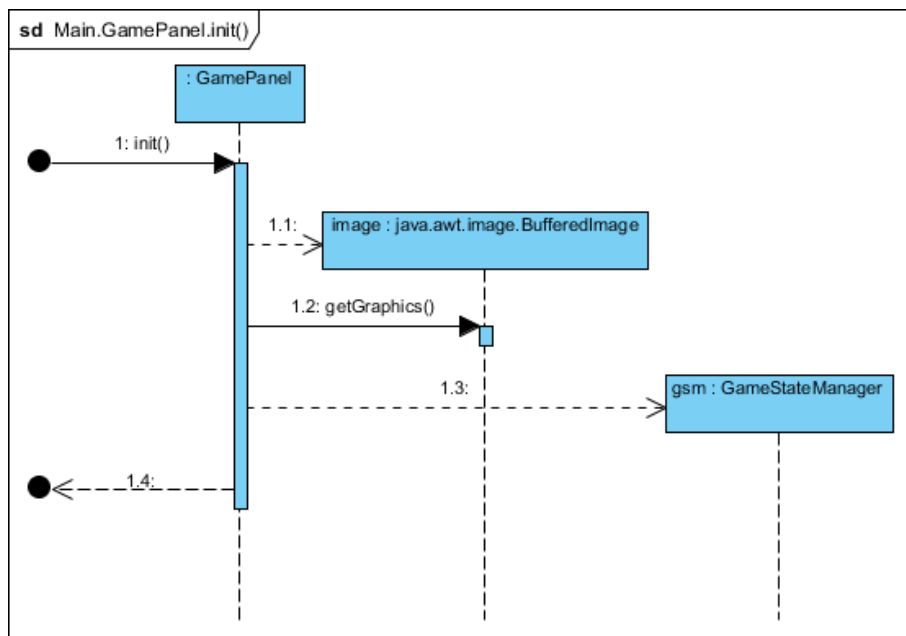
#### 4.3.21 ZPM

- **Felelősség**  
A játék célját jelentő Zero Point Module. A játékos felszedheti majd amint már egy ZPM sem maradt a pályán a játékos teljesítette a feladatát.
- **Össztályok**  
MapObject
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**  
Nem értelmezhető.

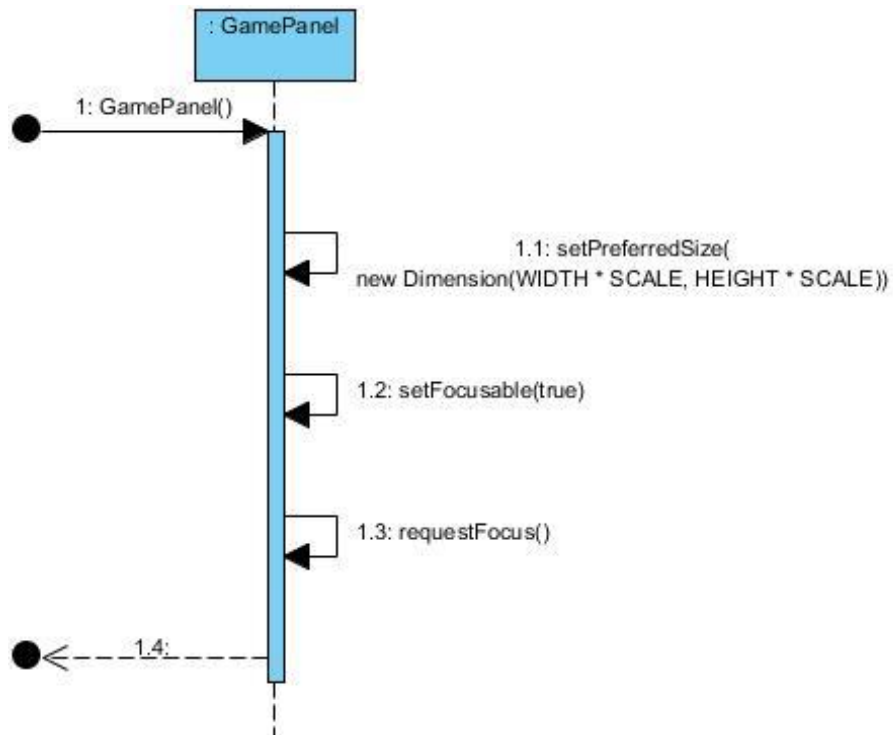
## 4.4 Szekvencia diagramok



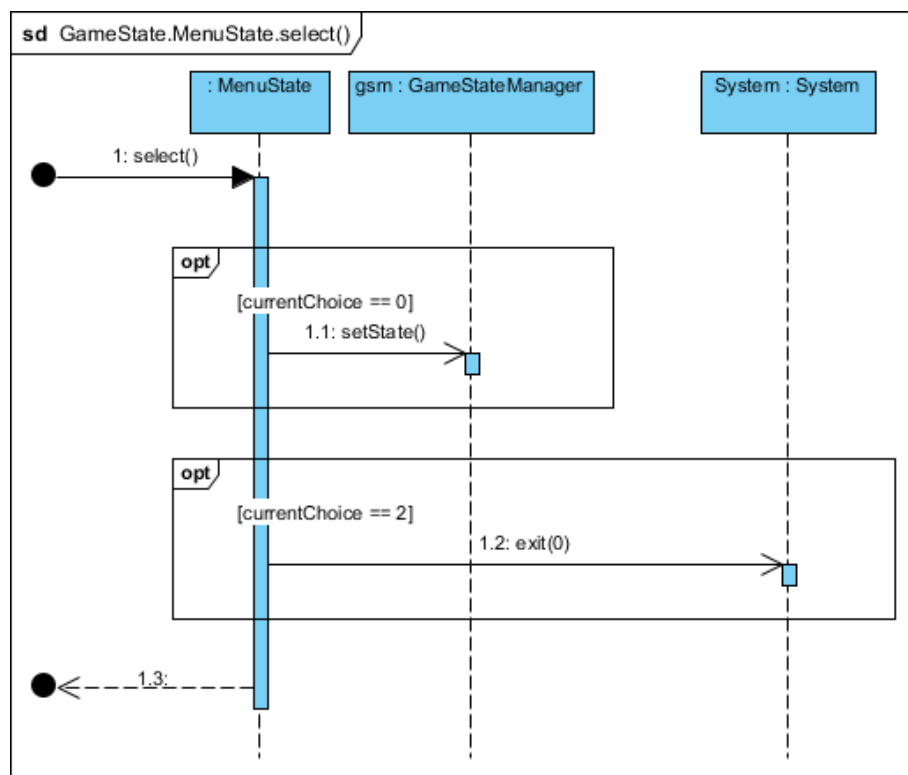
4.2. ábra Játéklablak beállítása



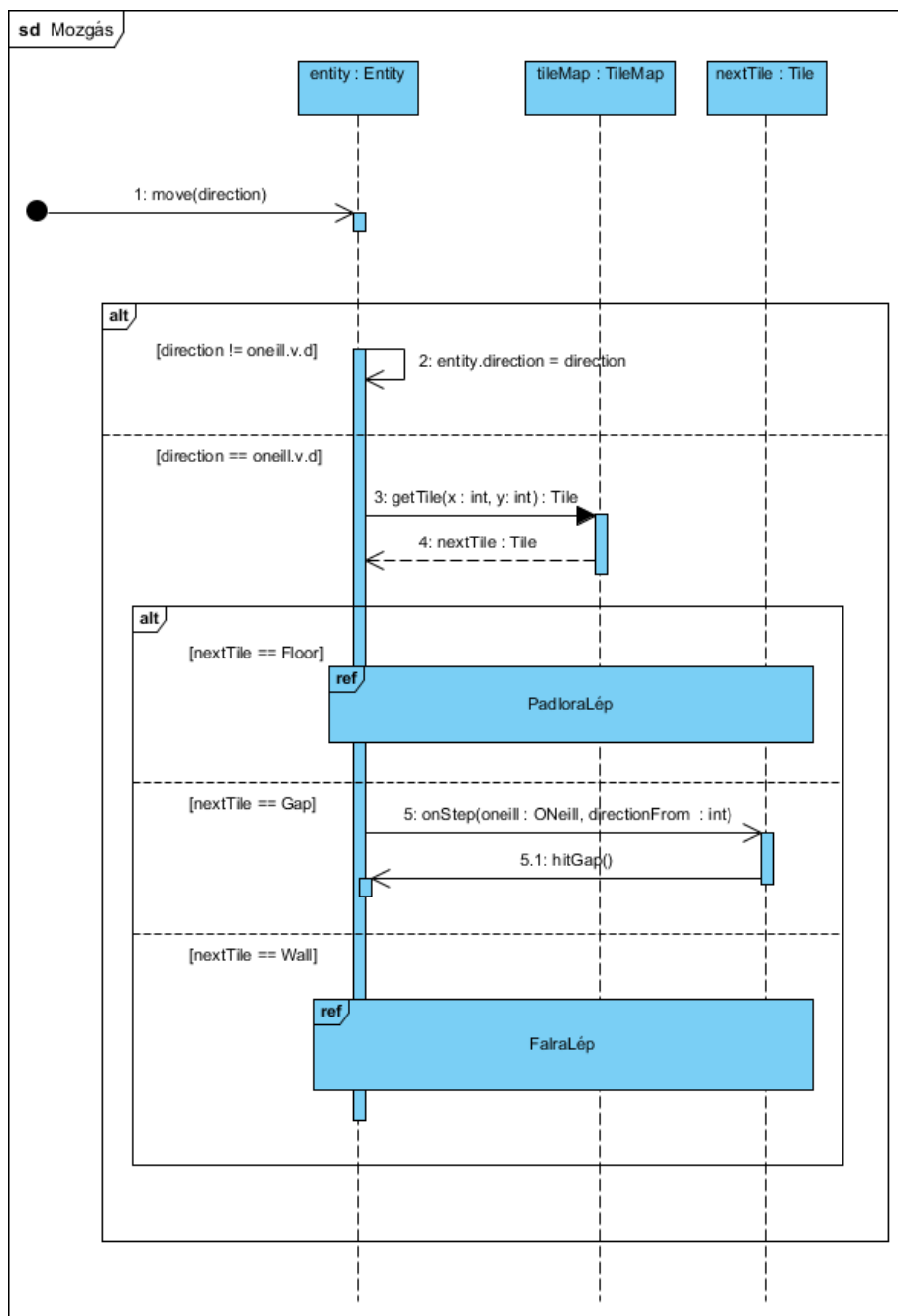
4.3. ábra Játékindítás



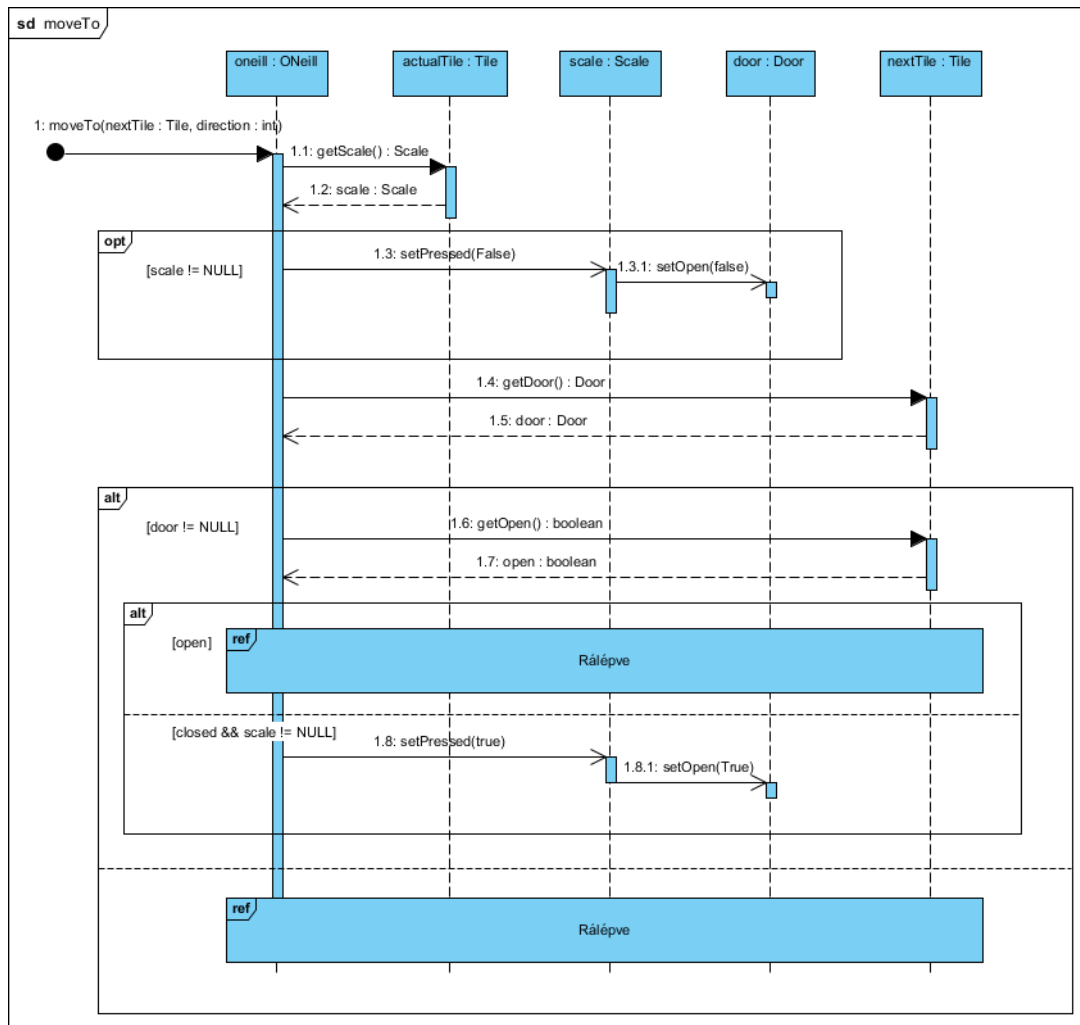
4.4. ábra GamePanel inicializálása



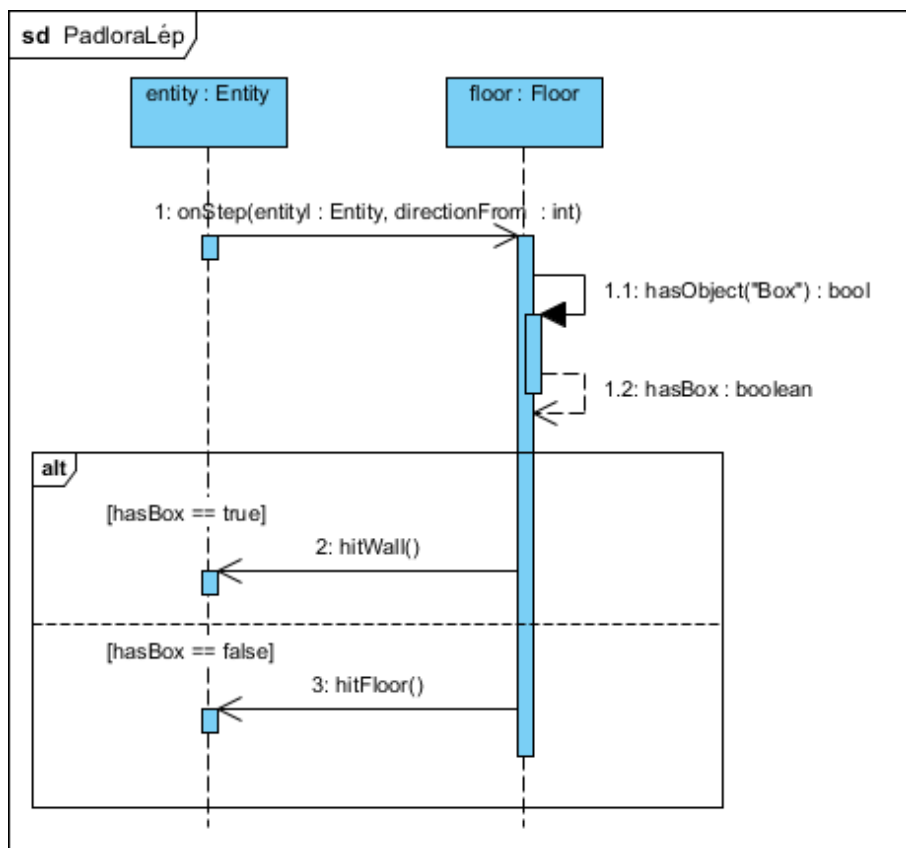
4.5. ábra Főmenü kezelése



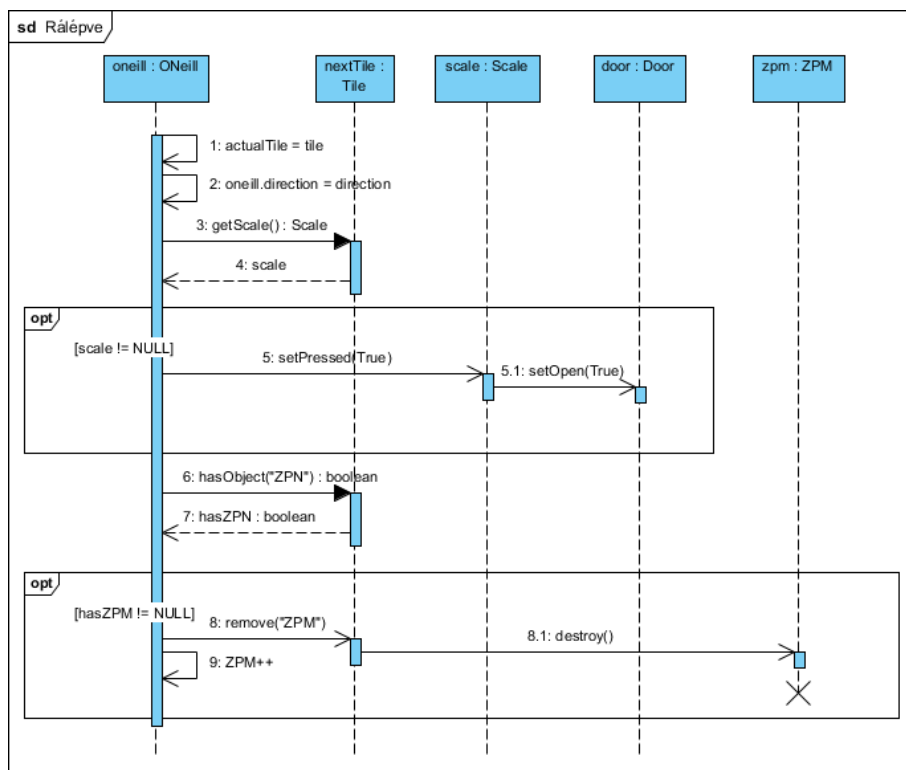
4.6. ábra Mozgás kezelése



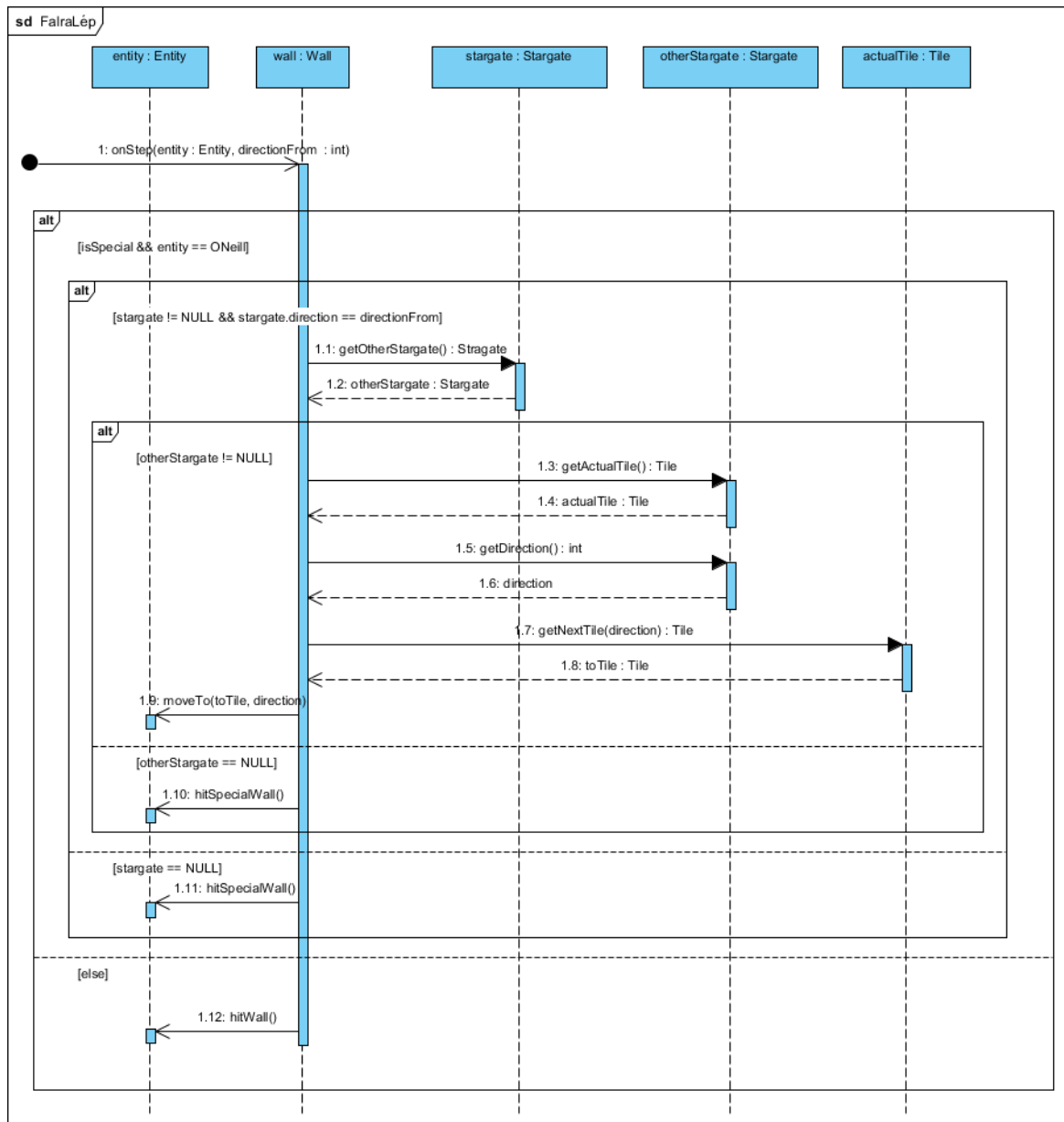
4.7. ábra Mérleg és ajtó kezelése



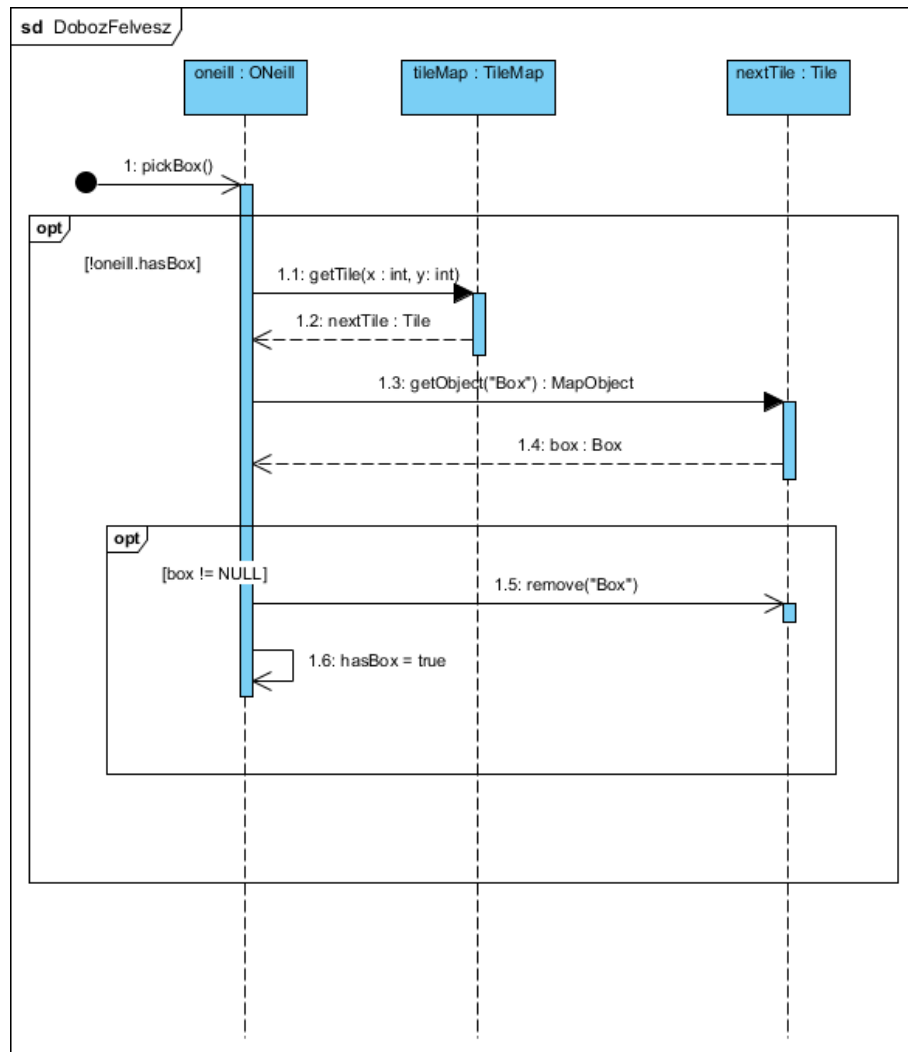
4.8. ábra Padlora lépés kezelése



4.9. ábra Mérlegre lépés illetve ZPM kezelése

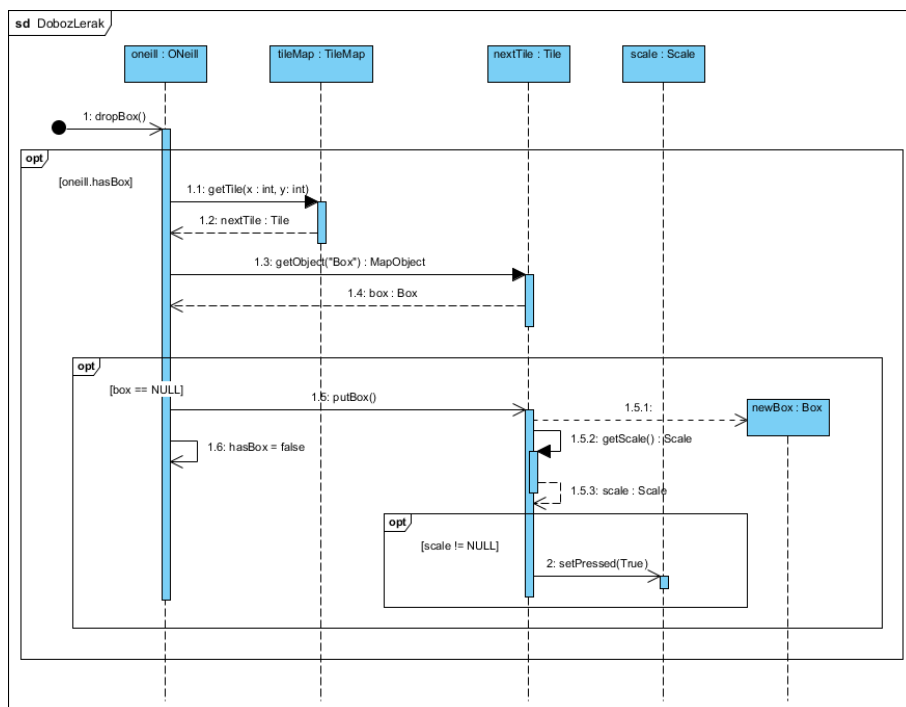


4.10. ábra Falra lépés kezelése

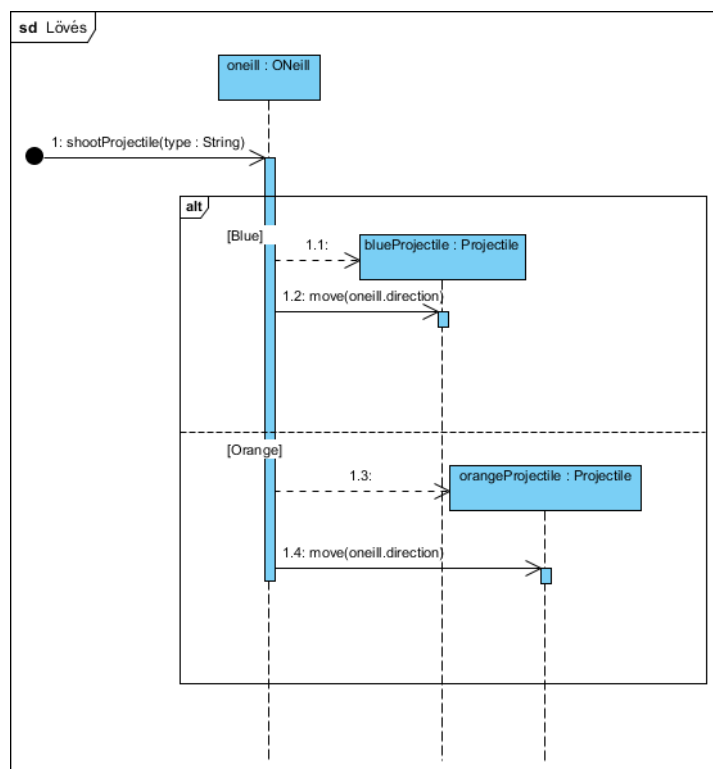


4.11. ábra Doboz felvétel kezelése

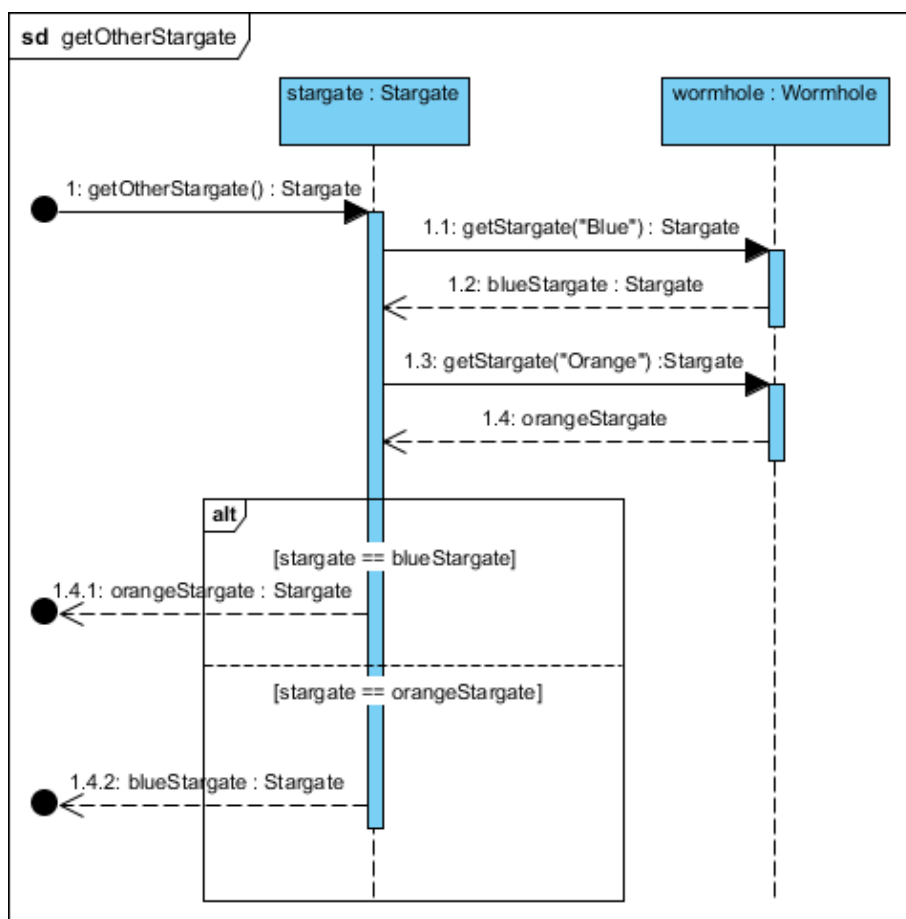




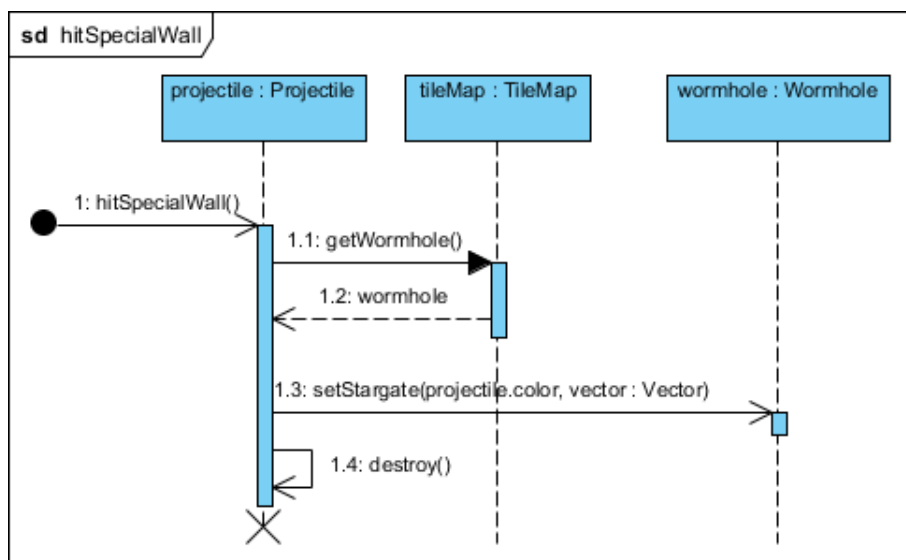
4.12. ábra Doboz lerakás kezelése



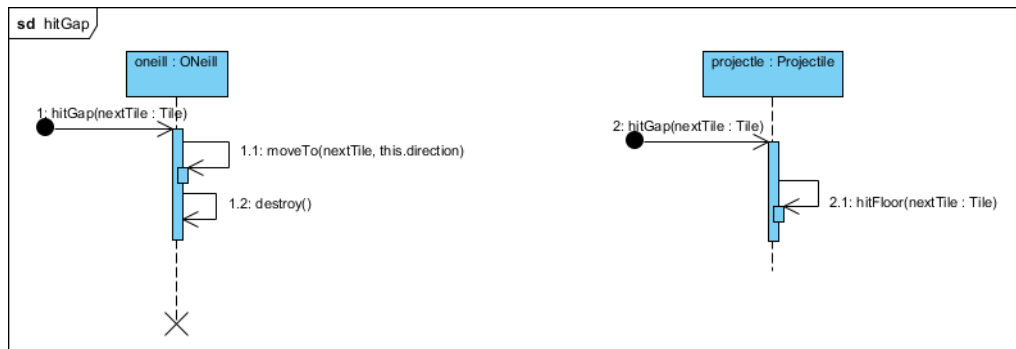
4.13. ábra Lövés kezelése



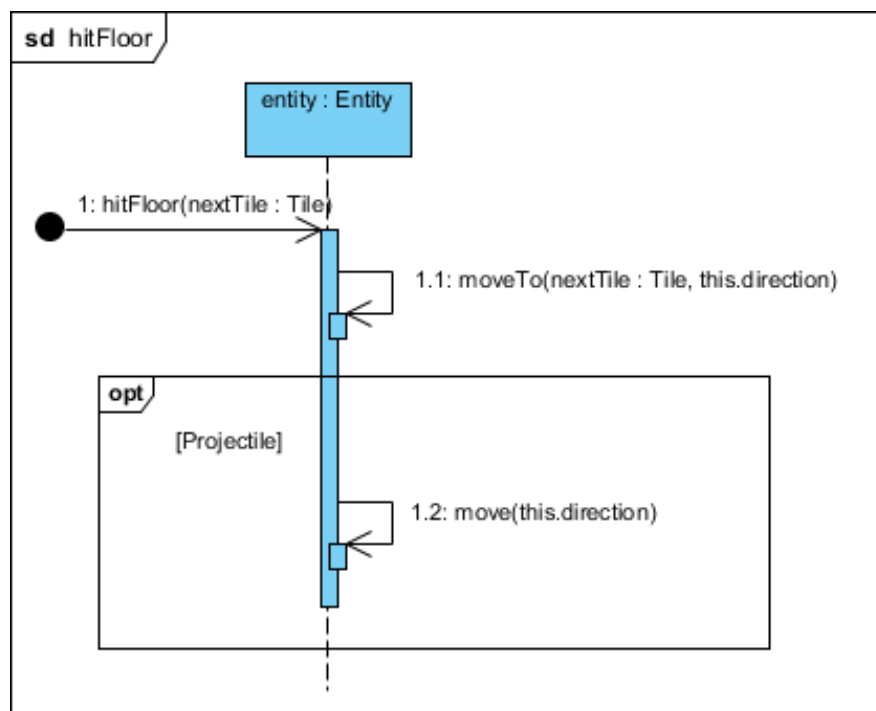
4.14. ábra Csillagkapuk kezelése



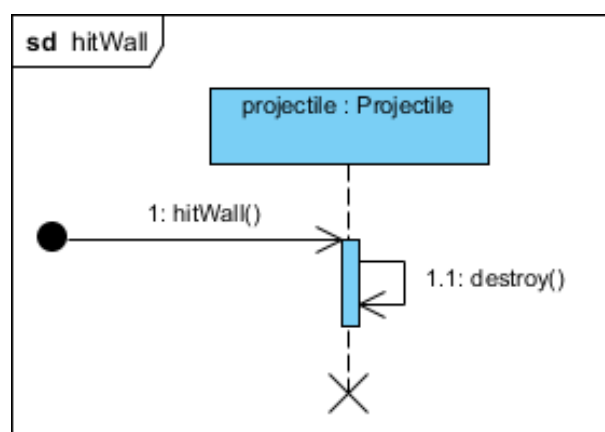
4.15. ábra Speciális falba érkezés kezelése



4.16. ábra Szakadék kezelése



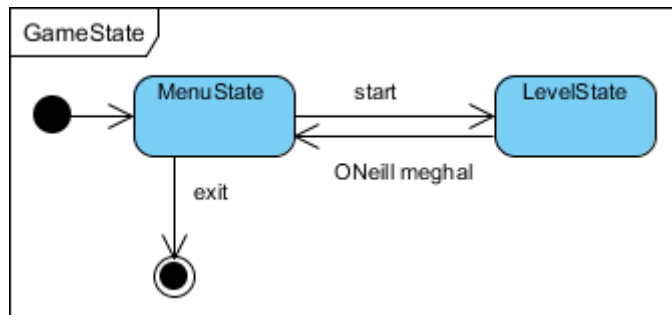
4.17. ábra Padlóra lépés kezelése



4.18. ábra Lövedék falba ütközése

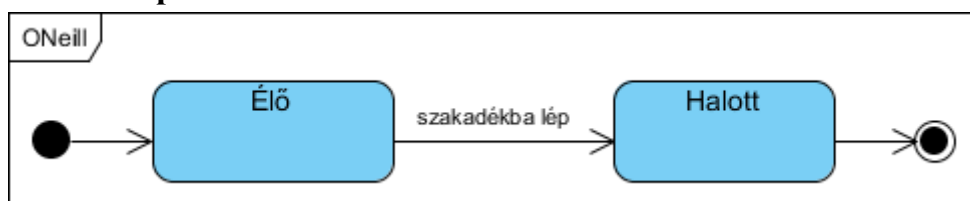
## 4.5 State-chartok

### 4.5.1 Játékállapotok



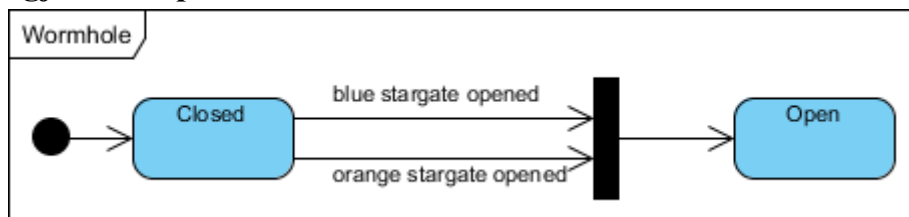
4.19. ábra A játék állapotváltozásai

### 4.5.2 O'Neill állapotai



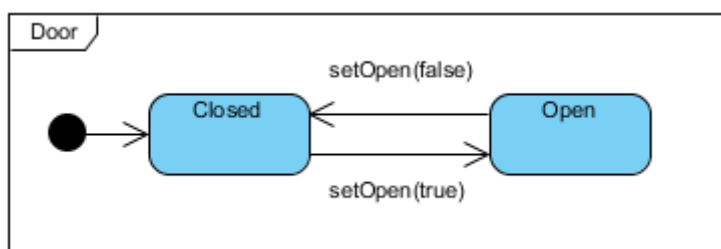
4.20. ábra O'Neill állapotváltozása

### 4.5.3 Féregjárat állapotai



4.21. ábra A féregjárat állapotváltozása

### 4.5.4 Ajtó állapotai



4.22. ábra Ajtó állapotváltozása

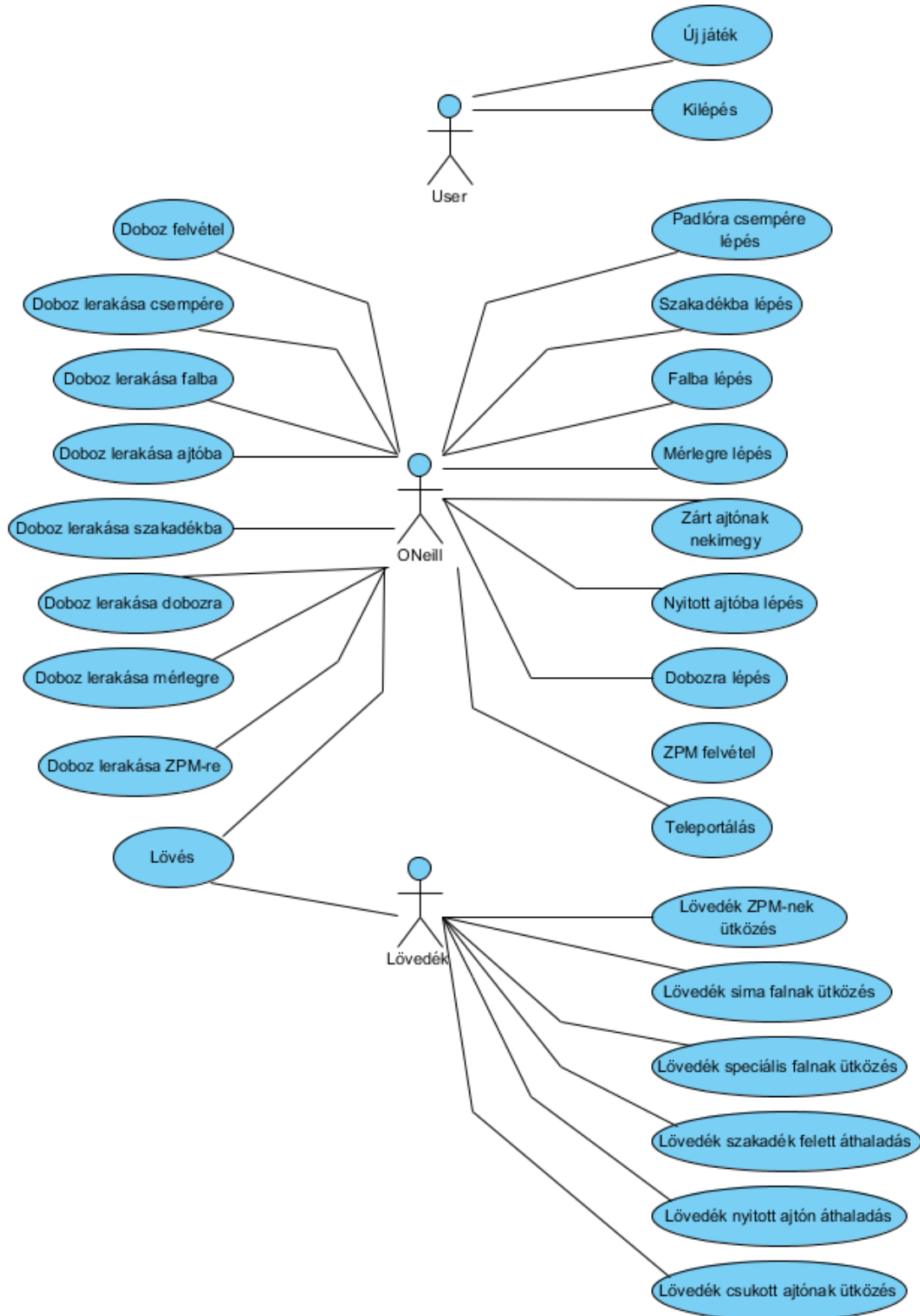
## 4.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2016.03.13 11:00	4,5 óra	Bujdosó	Tevékenység: Osztálydiagram javítása, osztály leírások javítása.
2016.03.14 00:00	2 óra	Horváth	Tevékenység: Szekvencia diagramok javítása.
2016.03.14 07:30	4,5 óra	Horváth	Tevékenység: Szekvencia diagramok javítása.
2016.03.14 10:00	1 óra	Bujdosó	Tevékenység: Osztálydiagram javítása, osztály leírások javítása.
2016.03.14 12:30	0,5 óra	Horváth	Tevékenység: Osztálydiagram pontosítása.
2016.03.14 10:30	3 óra	Karácsony	Tevékenység: Dokumentum véglegesítése, feltöltése a portálra.

## 5. Szkeleton tervezése

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

#### 5.1.1 Use-case diagram



5.1. ábra Szkeleton use-case-ek

### 5.1.2 Use-case leírások

<b>Use-case neve</b>	Menü - új játék
<b>Rövid leírás</b>	Új játék indul
<b>Aktorok</b>	User
<b>Forgatókönyv</b>	A "New Game" gombra kattintva, a program beolvassa a pályát és inicializálja az entitásokat/változókat.

<b>Use-case neve</b>	Menü - kilépés
<b>Rövid leírás</b>	Kilépés a játékból
<b>Aktorok</b>	User
<b>Forgatókönyv</b>	Az "Exit" gombra kattitva, a program futása megszakad.

<b>Use-case neve</b>	Padló csempére lépés
<b>Rövid leírás</b>	ONeill egy padló csempére lép amin nincs semmi
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	ONeill helyzete megváltozik az új csempére.

<b>Use-case neve</b>	Szakadékba lépés
<b>Rövid leírás</b>	ONeill szakadékba lép
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	Szakadékba lépés hatására O'Neill meghal és vége szakad a játéknak.

<b>Use-case neve</b>	Falba lépés
<b>Rövid leírás</b>	ONeill megpróbál falba lépni.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	A fal alapvetően nem átjárható, ezért, O'Neill az eredeti helyén marad.

<b>Use-case neve</b>	Mérlegre lépés
<b>Rövid leírás</b>	ONeill méregre lépéssel. kinyit egy ajtót.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	ONeill pozíciója megváltozik a mérlegére, ezzel lenyomva azt, aminek a hatására a mérleghez tartozó ajtó kinyílik.

<b>Use-case neve</b>	Zárt ajtónak nekimegy
<b>Rövid leírás</b>	ONeill megpróbál zárt ajtóba lépni.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	A zárt ajtó nem átjárható, ezért, O'Neill az eredeti helyén marad.

<b>Use-case neve</b>	Nyitott ajtóba lép
<b>Rövid leírás</b>	ONeill nyitott ajtóba lép.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	ONeill látja, hogy nyitva van az ajtó ezért a pozíciója megváltozik, az ajtót tartalmazó csempére.

<b>Use-case neve</b>	Dobozra lépés
<b>Rövid leírás</b>	ONeill dobozra próbál lépni.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	ONeill látja, hogy az előtte lévő csempén doboz van és mivel ez egy nem átjárható objektum, az eredeti helyén marad.

<b>Use-case neve</b>	ZPM felvétel
<b>Rövid leírás</b>	ONeill felvesz egy ZPM-et
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	ONeill egy olyan padló csempére lép, amin ZPM van. A helyzete megváltozik a ZPM eltűnik és növekszik a megszerzett ZPM-ek száma.

<b>Use-case neve</b>	Teleportálás
<b>Rövid leírás</b>	ONeill egyik csillagkapu elől a másik elé teleportál.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	Egy speciális fal mindkét oldalán létezik egy csillagkapu, amik előtt egy padló csempe található.

<b>Use-case neve</b>	Doboz felvétel
<b>Rövid leírás</b>	ONeill felvesz egy dobozt.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	ONeill és a doboz, két szomszédos, padló csempén vannak.

<b>Use-case neve</b>	Doboz lerakása csempére
<b>Rövid leírás</b>	ONeill leteszi a dobozt egy csempére.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	A doboz lerakás akciót, a csempe felé fordulva végrehajtva, megjelenik egy doboza célon.

<b>Use-case neve</b>	Doboz lerakása falba
<b>Rövid leírás</b>	ONeill megpróbál dobozt letenni falba.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	A doboz lerakás akciót, a fal felé fordulva végrehajtva, nem történik semmi, mivel ide nem lehet letenni.

<b>Use-case neve</b>	Doboz lerakása ajtóba
<b>Rövid leírás</b>	ONeill megpróbál dobozt letenni ajtóba.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	A doboz lerakás akciót, az ajtó felé fordulva végrehajtva, nem történik semmi, mivel ide nem lehet letenni.

<b>Use-case neve</b>	Doboz lerakása szakadékba
<b>Rövid leírás</b>	ONeill dobozt tesz egy szakadékba.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	A doboz lerakás akciót, az szakadék felé fordulva végrehajtva, a doboz megsemmisül.



<b>Use-case neve</b>	Doboz lerakása dobozra
<b>Rövid leírás</b>	ONeil megpróbál dobozt letenni egy másik dobozra.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	A doboz lerakás akciót, a padlón lévő doboz felé fordulva végrehajtva, nem történik semmi, mivel dobozokat nem lehet egymásra tenni.

<b>Use-case neve</b>	Doboz lerakása mérlegre
<b>Rövid leírás</b>	ONeill dobozt tesz egy mérlegre.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	ONeill mellett egy ajtó és egy padló található, amin egy mérleg van. O'Neill a mérleg felé néz és végrehajtja a doboz lerakás akciót. A doboz megjelenik a mérleg fölöt, ezzel lenyomva azt, aminek a hatására a hozzá tartozó ajtó kinyílik.

<b>Use-case neve</b>	Doboz lerakása ZPM-re
<b>Rövid leírás</b>	ONeill dobozt tesz egy ZPM-re.
<b>Aktorok</b>	ONeill
<b>Forgatókönyv</b>	ONeill mellett egy padló és azon egy ZPM található. O'Neill a ZPM felé néz és végrehajtja a doboz lerakás akciót. A dobozt nem tudja a ZPM-re elhelyezni és erről a játék értesíti őt.

<b>Use-case neve</b>	Lövés
<b>Rövid leírás</b>	ONeill kilő egy lövedéket.
<b>Aktorok</b>	ONeill, Lövedék
<b>Forgatókönyv</b>	Egy padló csempe felé O'Neill a lövés akció hatására kilő egy lövedéket, ami az alatta lévő csempén megjelenik és továbbhalad a következőre.

<b>Use-case neve</b>	Lövedék doboznak ütközés
<b>Rövid leírás</b>	Egy lövedék doboznak ütközik
<b>Aktorok</b>	Lövedék
<b>Forgatókönyv</b>	A lövedék a dobozt tartalmazó csempére mozdulva megsemmisül.

<b>Use-case neve</b>	Lövedék ZPM-nek ütközés
<b>Rövid leírás</b>	Egy lövedék ZPM-nek ütközik
<b>Aktorok</b>	Lövedék
<b>Forgatókönyv</b>	A lövedék a ZPM-et tartalmazó csempére mozdulva megsemmisül.

<b>Use-case neve</b>	Lövedék sima falnak ütközés
<b>Rövid leírás</b>	Egy lövedék sima falnak ütközik
<b>Aktorok</b>	Lövedék
<b>Forgatókönyv</b>	A lövedék a sima falat tartalmazó csempére mozdulva megsemmisül.

<b>Use-case neve</b>	Lövedék speciális falnak ütközés
<b>Rövid leírás</b>	Egy lövedék speciális falnak ütközik.
<b>Aktorok</b>	Lövedék
<b>Forgatókönyv</b>	A lövedék egy speciális falba ütközve, egy csillagkaput hoz létre azon és eltűnik.

<b>Use-case neve</b>	Lövedék szakadék felett haladás
<b>Rövid leírás</b>	Egy lövedék szakadék felett halad
<b>Aktorok</b>	Lövedék
<b>Forgatókönyv</b>	A lövedék egy padló-szakadék-padló csempéket tartalmazó egyenes vonalon mozog végig. A szakadék nem állítja meg a lövedéket.

<b>Use-case neve</b>	Lövedék nyitott ajtón áthaladás
<b>Rövid leírás</b>	Egy lövedék nyitott ajtón áthalad
<b>Aktorok</b>	Lövedék
<b>Forgatókönyv</b>	A lövedék egy padló-padló(nyitott ajtóval)-padló csempéket tartalmazó egyenes vonalon mozog végig. A nyitott ajtóval rendelkező padló csempe, nem állítja meg a lövedéket.

<b>Use-case neve</b>	Lövedék csukott ajtónak ütközés
<b>Rövid leírás</b>	Egy lövedék csukott ajtónak ütközik
<b>Aktorok</b>	Lövedék
<b>Forgatókönyv</b>	A lövedék a csukott ajtót tartalmazó csempére mozdulva megsemmisül.

## 5.2 A szkeleton kezelői felületének terve, dialógusok

A felhasználó a program indításakor egy menüt lát, amely tartalmazza az elvégezhető műveleteket. Ezek a műveletek előre leprogramozott teszteseteknek felelnek meg. A felhasználó a menüpont begépelése után (amit a program külön kér tőle) Entert nyom, aminek hatására a képernyőn megjelenik az egyes esetekhez tartozó hívási lánc (a hívásban részt vevő metódusok, a hívás iránya, megjegyzések stb.) A felhasználó ez alapján látja, hogy egy-egy esetben mi történik a programon belül.

A menü felépítése a következő:

- 1 Menü - új játék**
- 2 Menü - kilépés**
- 3 Lépéssel kapcsolatos esetek**
  - 3.1 Padló csempére lépés
  - 3.2 Szakadékba lépés
  - 3.3 Falba lépés
  - 3.4 Mérlegre lépés
  - 3.5 Zárt ajtónak nekimegy
  - 3.6 Nyitott ajtóba lép
  - 3.7 Dobozra lépés
  - 3.8 ZPM felvétel
  - 3.9 Teleportál

#### 4 Dobozzal kapcsolatos esetek

- 4.1 Doboz felvétel
- 4.2 Doboz lerakása csempére
- 4.3 Doboz lerakása falba
- 4.4 Doboz lerakása ajtóba
- 4.5 Doboz lerakása szakadékba
- 4.6 Doboz lerakása dobozra
- 4.7 Doboz lerakása mérlegre
- 4.8 Doboz lerakása ZPM-re

#### 5 Lövéssel kapcsolatok esetek

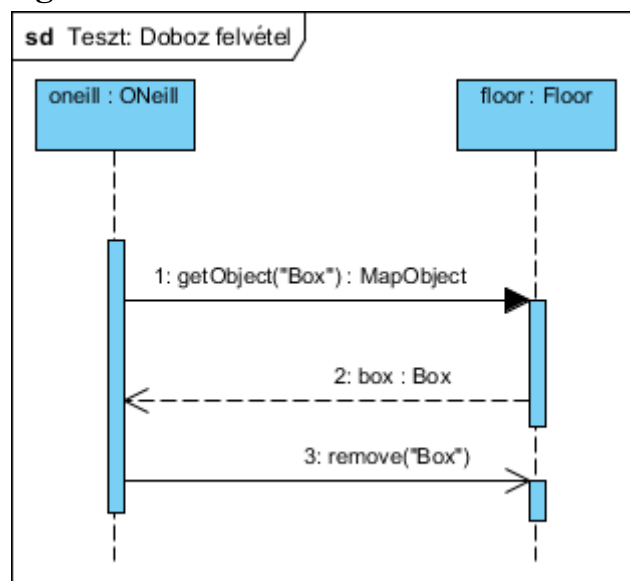
- 5.1 Lövedék doboznak ütközés
- 5.2 Lövedék ZPM-nek ütközés
- 5.3 Lövedék sima falnak ütközés
- 5.4 Lövedék speciális falnak ütközés
- 5.5 Lövedék szakadék felett haladás
- 5.6 Lövedék nyitott ajtónak áthaladás
- 5.7 Lövedék csukott ajtónak ütközés

A dialógusra egy példa a doboz szakadékba rakására, amit az alábbiaként képzelünk el:

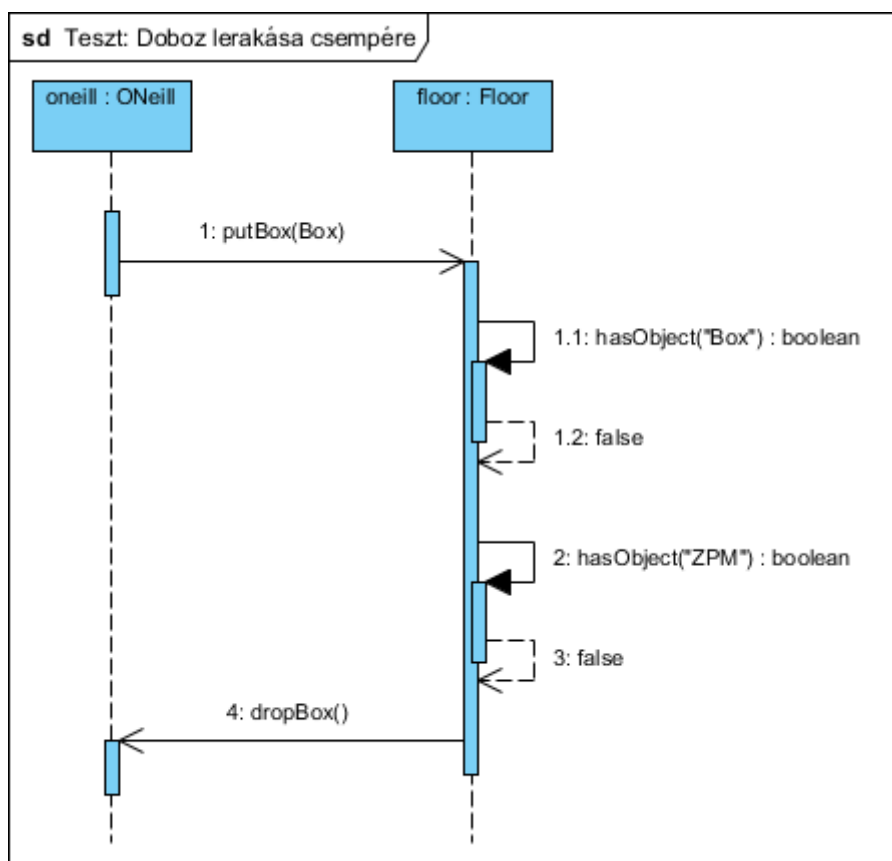
- DOBOZ LERAKÁSA SZAKADÉKBA
- > [:Oneill] oneill.putBox(Box)
- < [:Gap] gap.dropBox()
- > [:Gap] Box.destroy()
- A doboz megsemmisült

A dialógus képernyőkön a felhasználó választását egy mondat jelzi („Kérlek válassz teszt esetet:”). A > karakter kimenő hívást a < karakter visszaérkező hívást vagy adatot, a – karakter megjegyzést reprezentál a képernyőn.

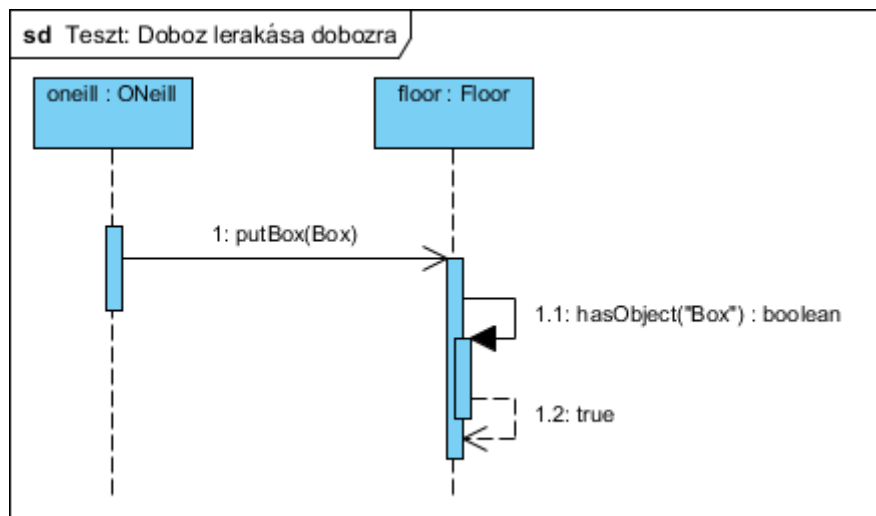
### 5.3 Szekvencia diagramok a belső működésre



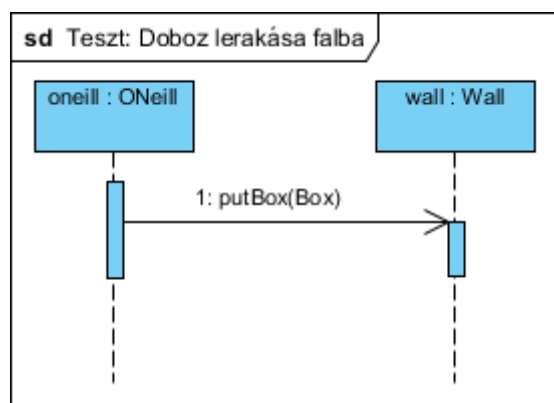
5.2. ábra Doboz felvétele



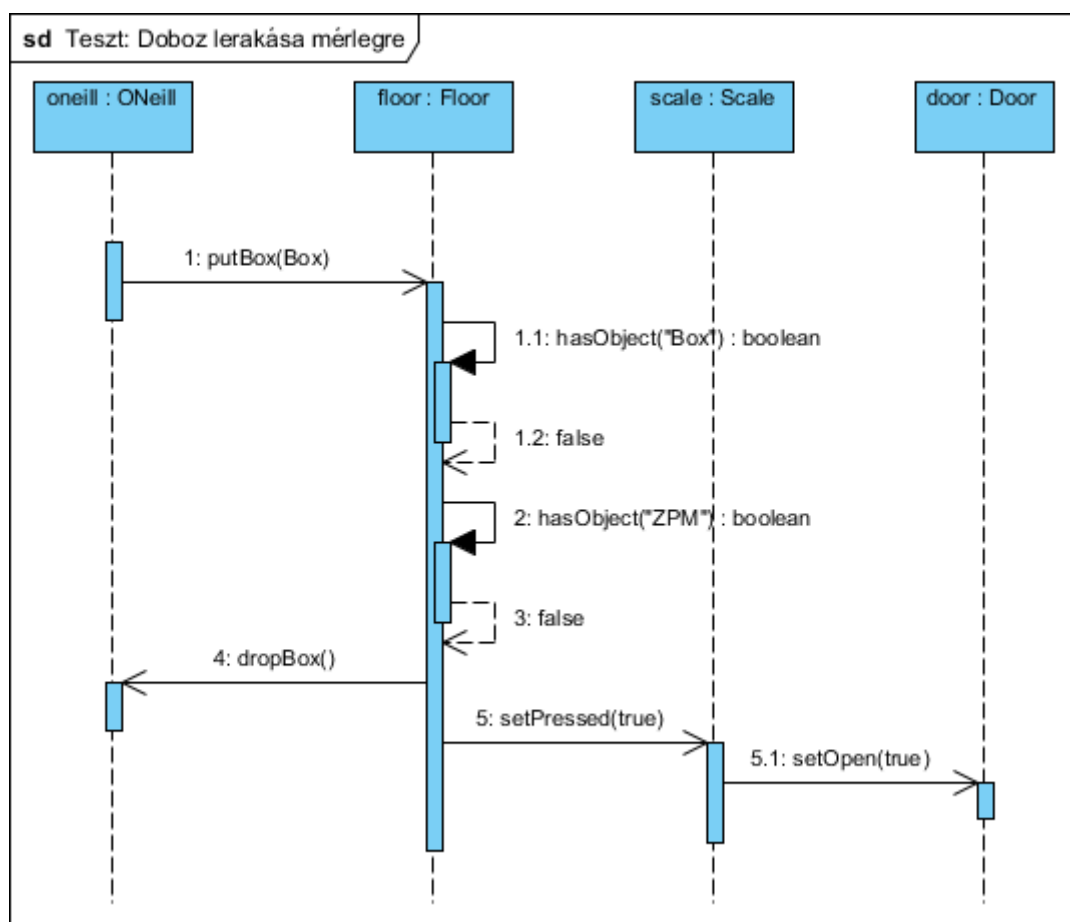
5.3. ábra Doboz lerakása csempére



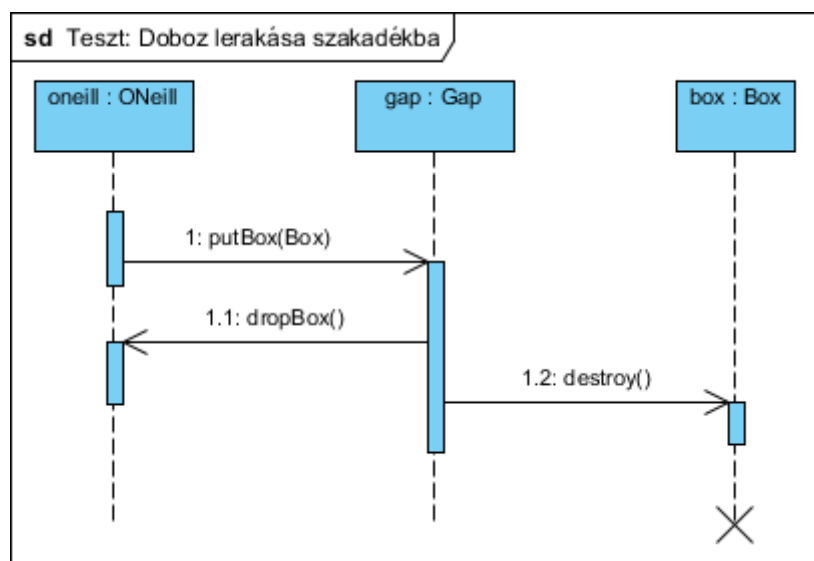
5.4. ábraDoboz lerakása dobozra



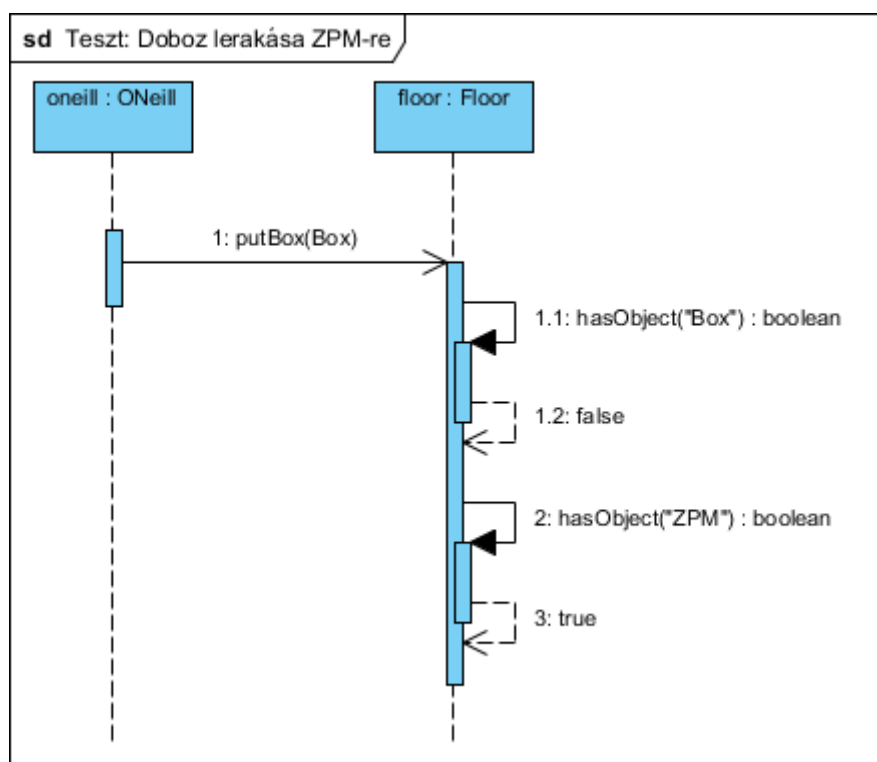
5.5. ábra Doboz lerakása falba



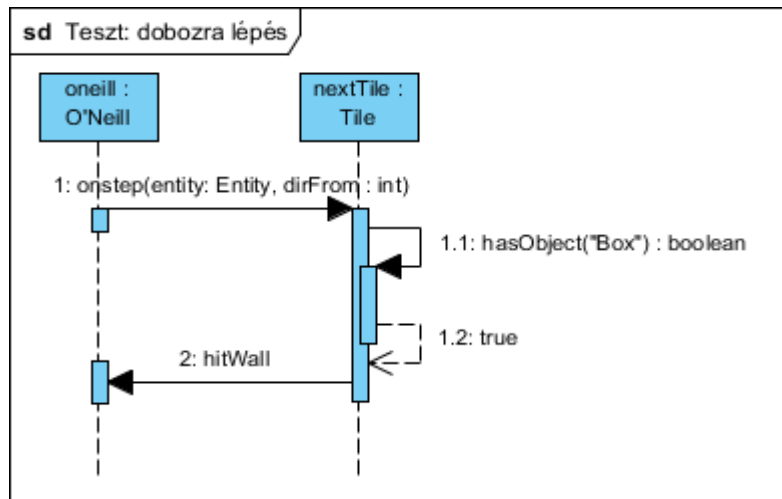
5.6. ábra Doboz lerakása mérlegre



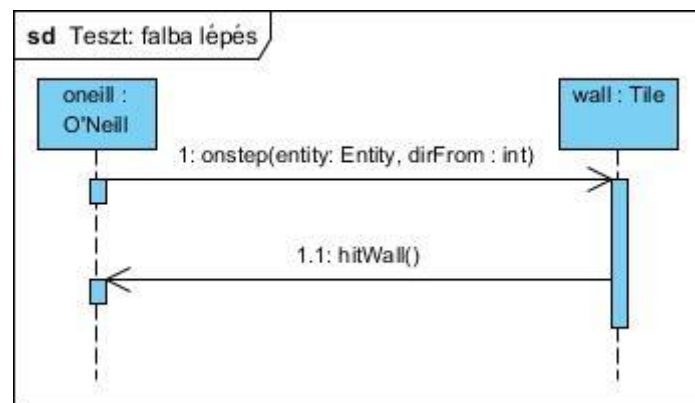
5.7. ábra Doboz lerakása szakadékba



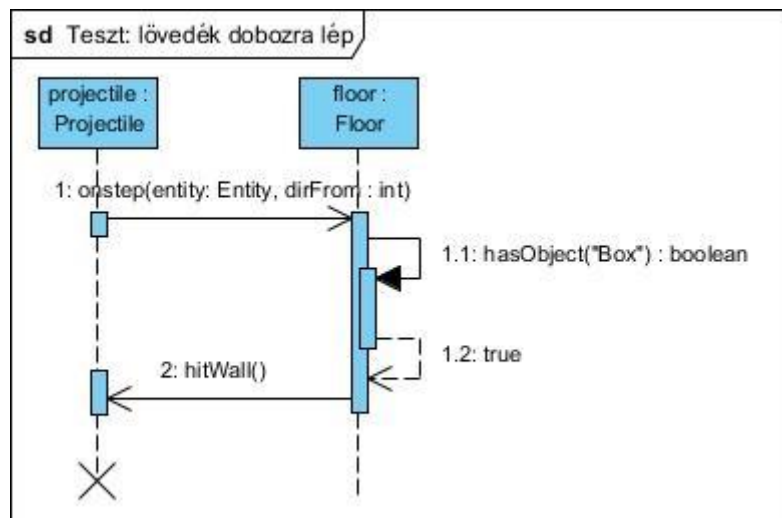
5.8. ábra Doboz lerakása ZPM-re



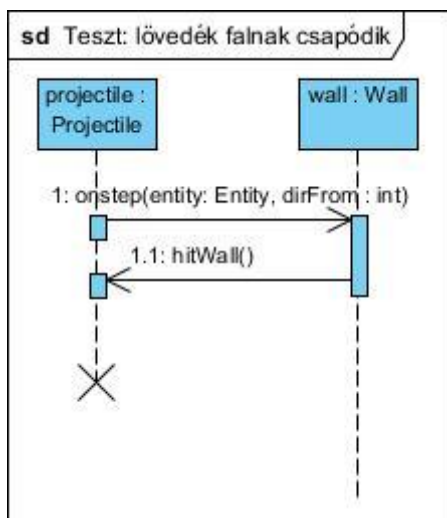
5.9. ábra Dobozra lépés



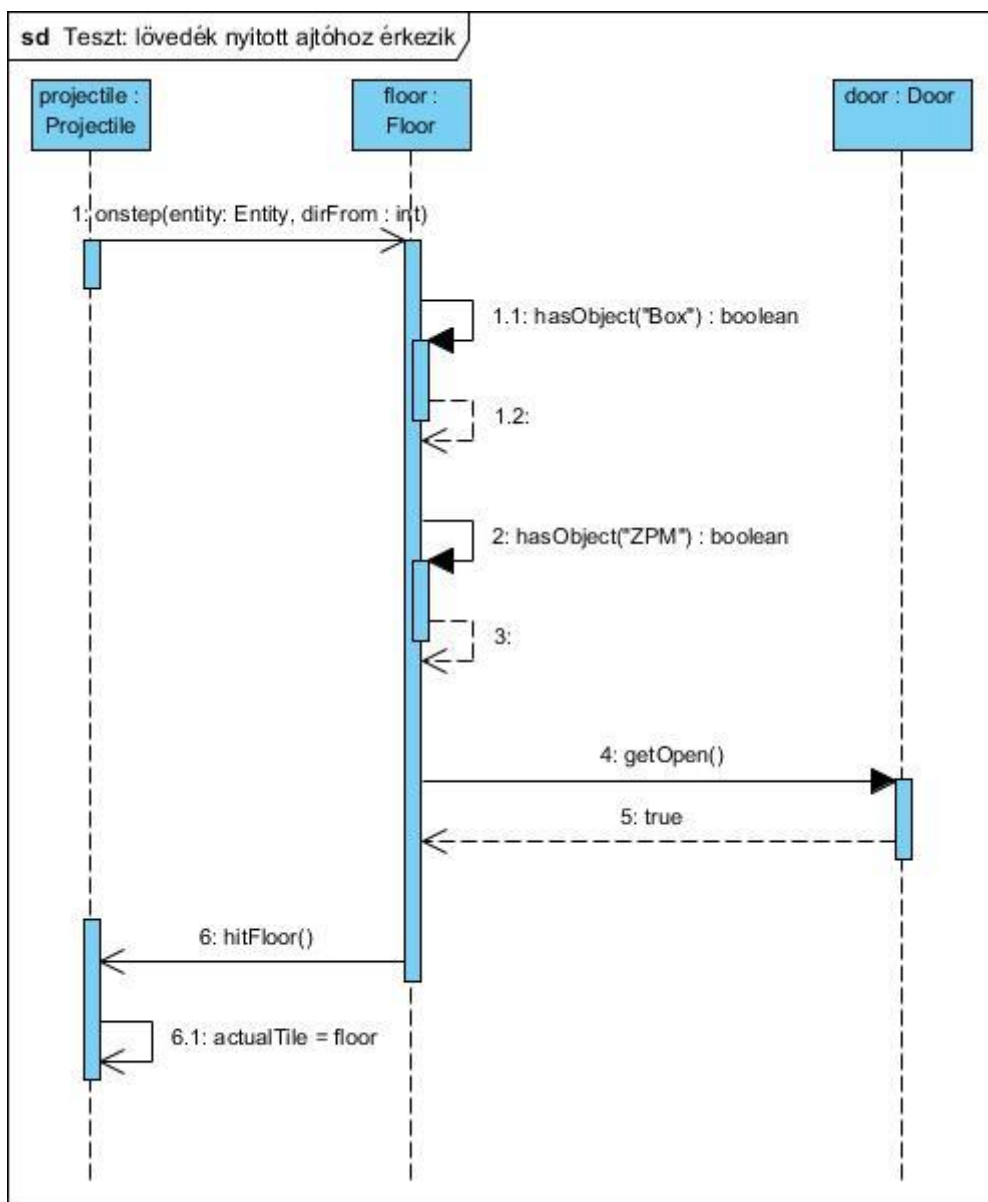
5.10. ábra Falba lépés



5.11. ábra Lövedék doboznak ütközése

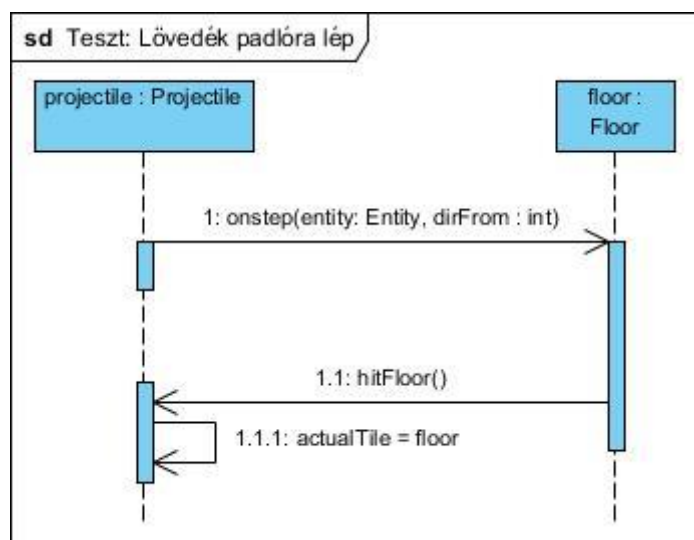


5.12. ábra Lövedék falnak csapódik

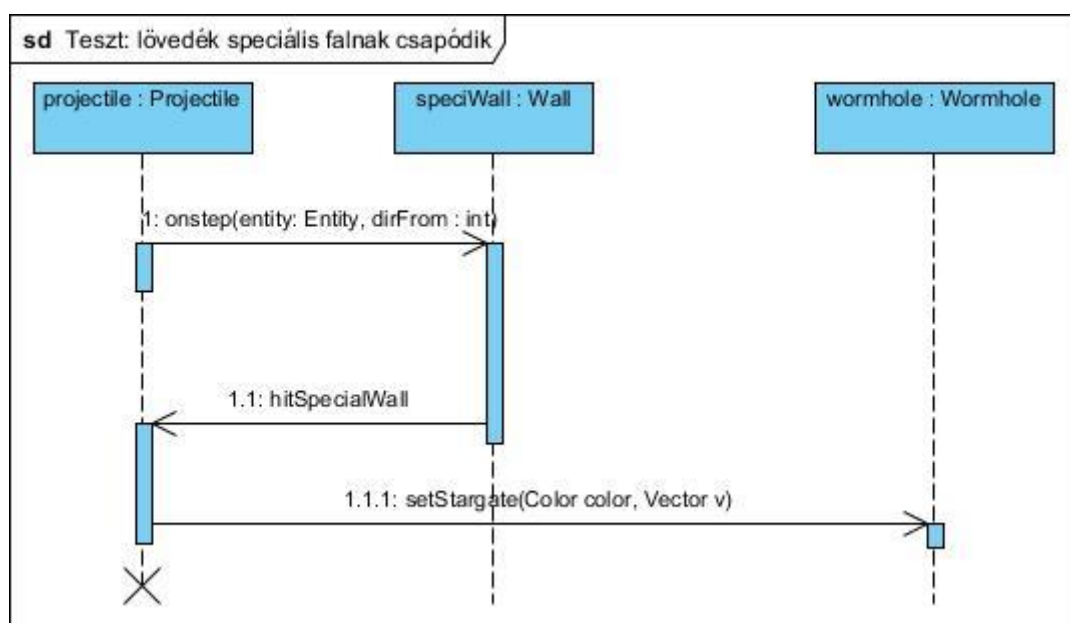


5.13. ábra Lövedék nyitott ajtóhoz érkezik

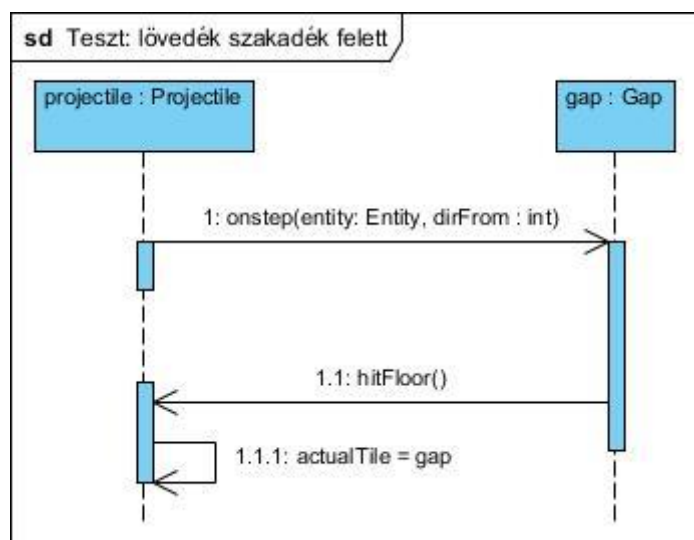




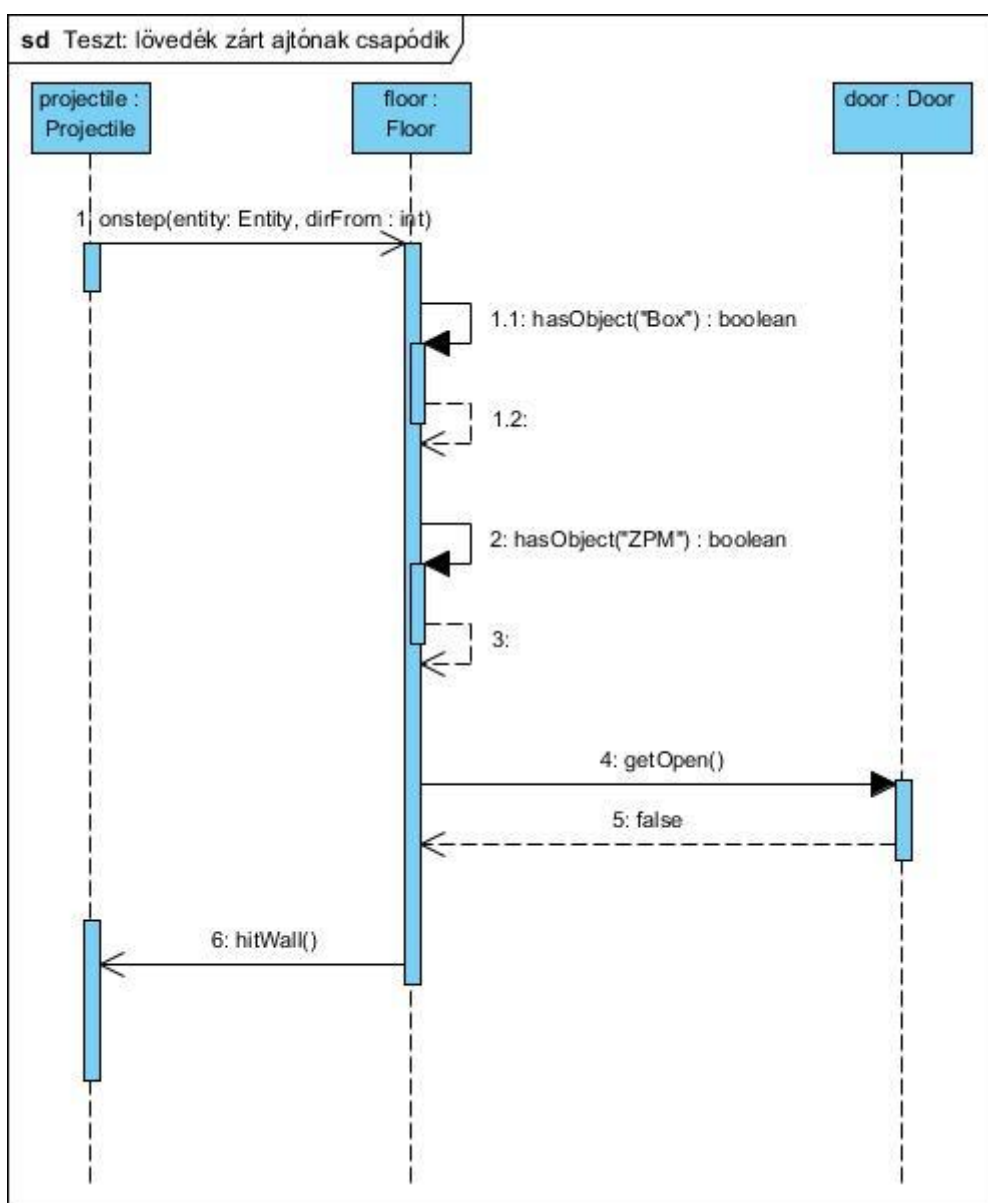
5.14. ábra Lövedék haladása a csempén



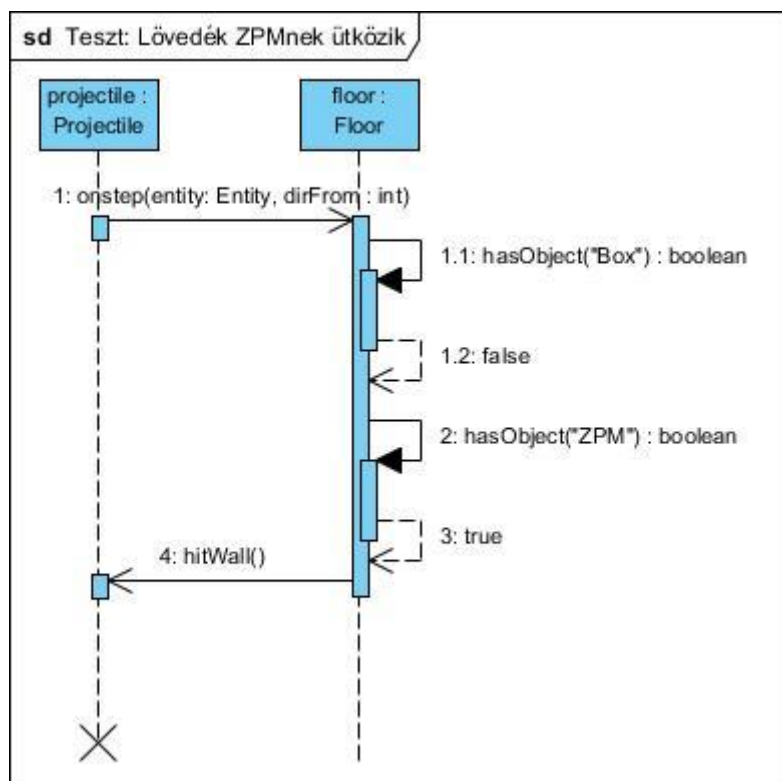
5.15. ábra Lövedék speciális falnak csapódik



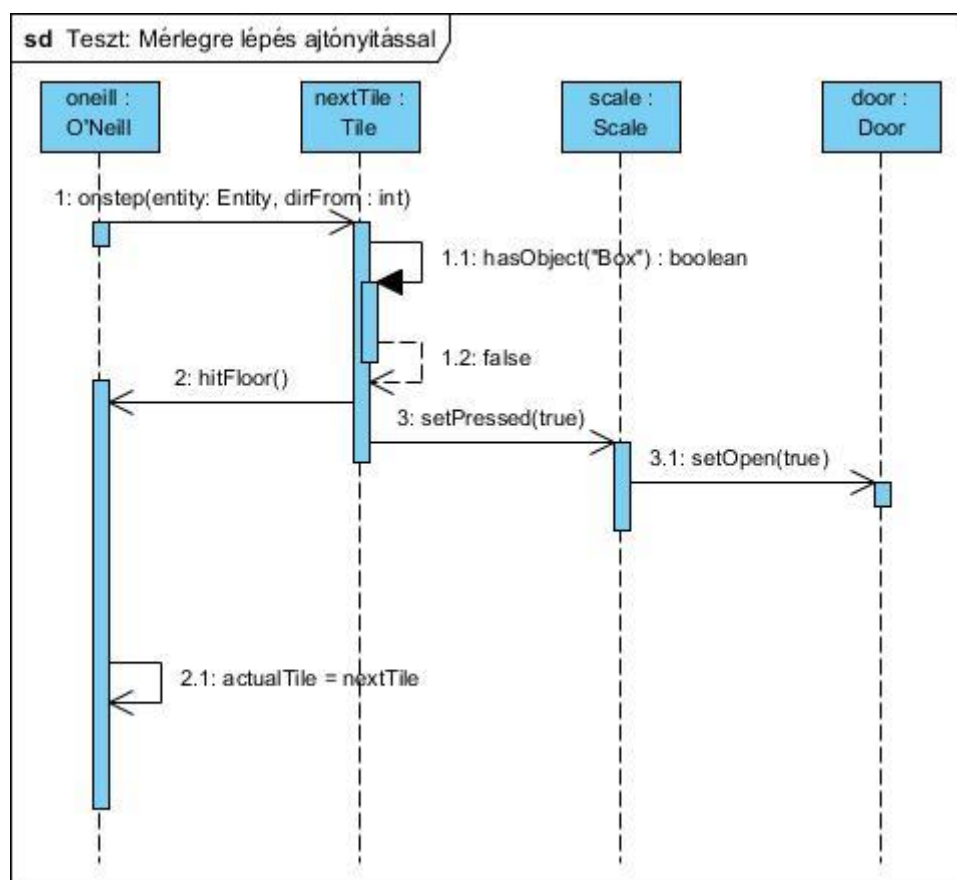
5.16. ábra Lövedék szakadék felett áthalad



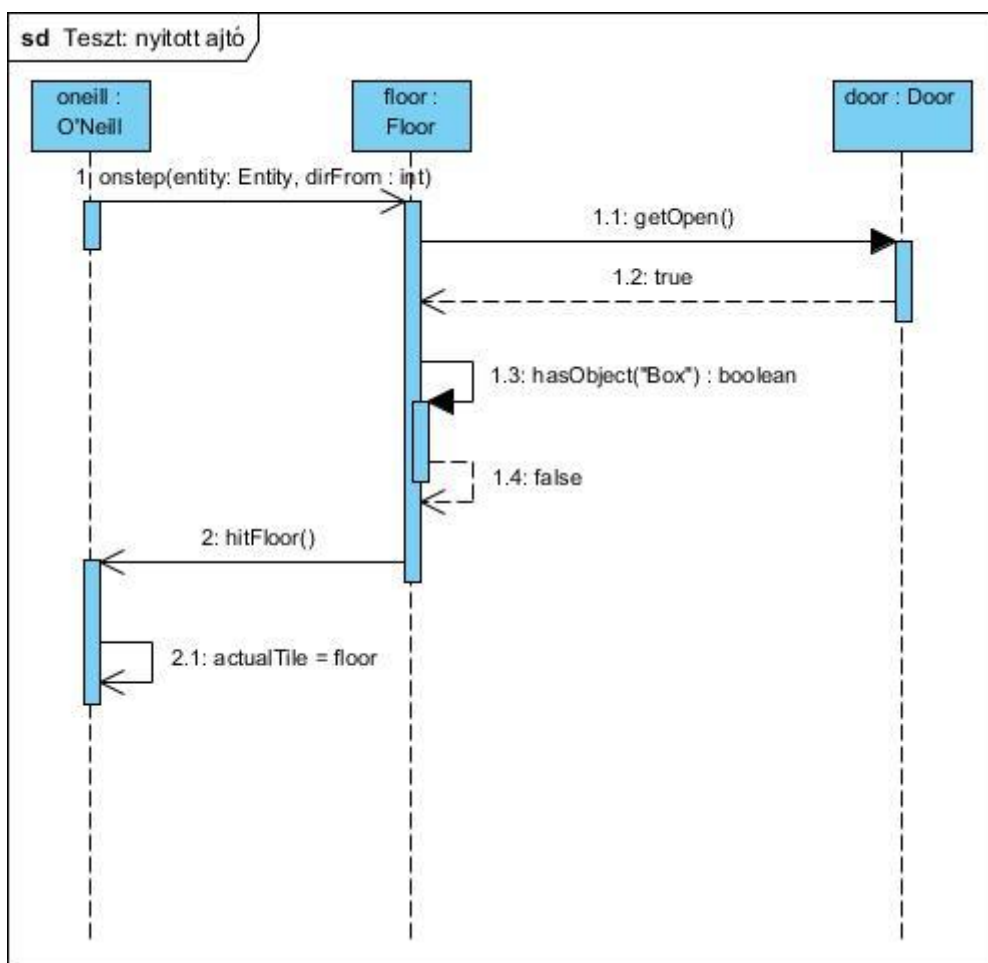
5.17. ábra Lövedék zárt ajtónak csapódik



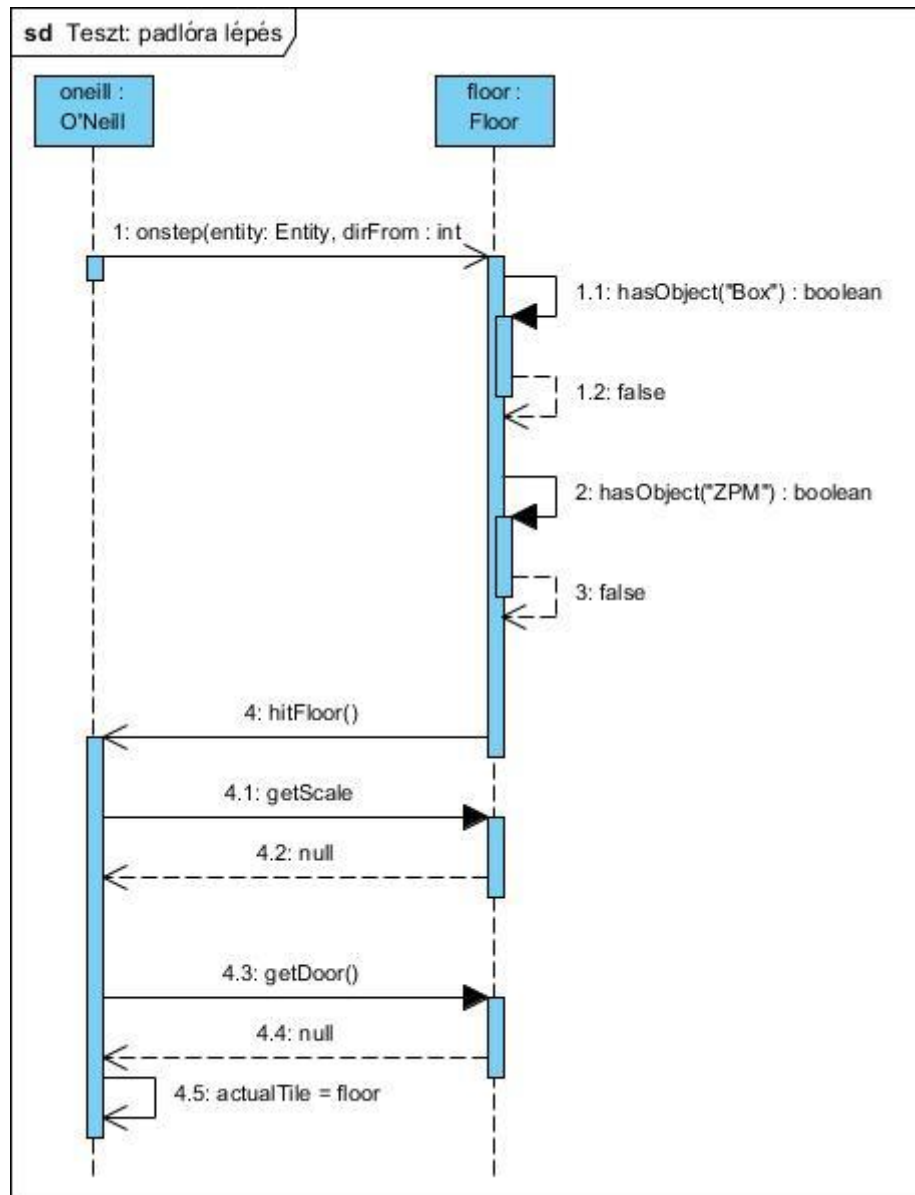
5.18. ábra Lövedék ZPM-nek ütközik



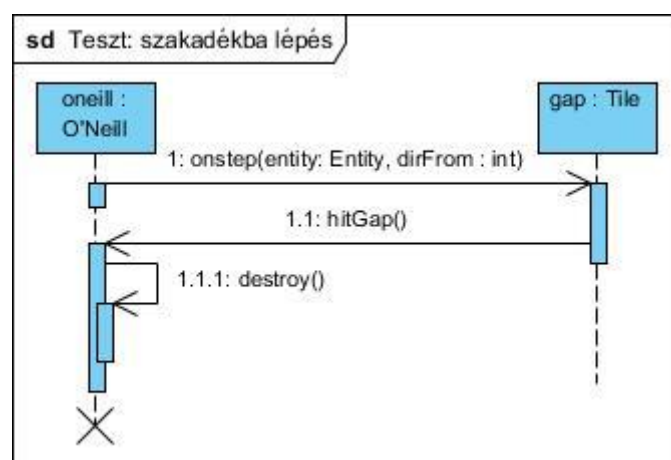
5.19. ábra Mérlegre lépés és ajtónyitás



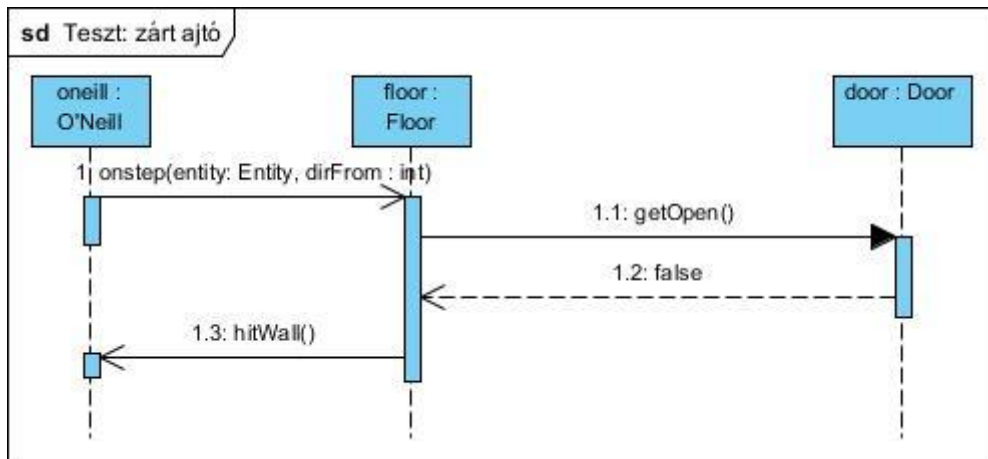
5.20. ábra Nyitott ajtóba lépés



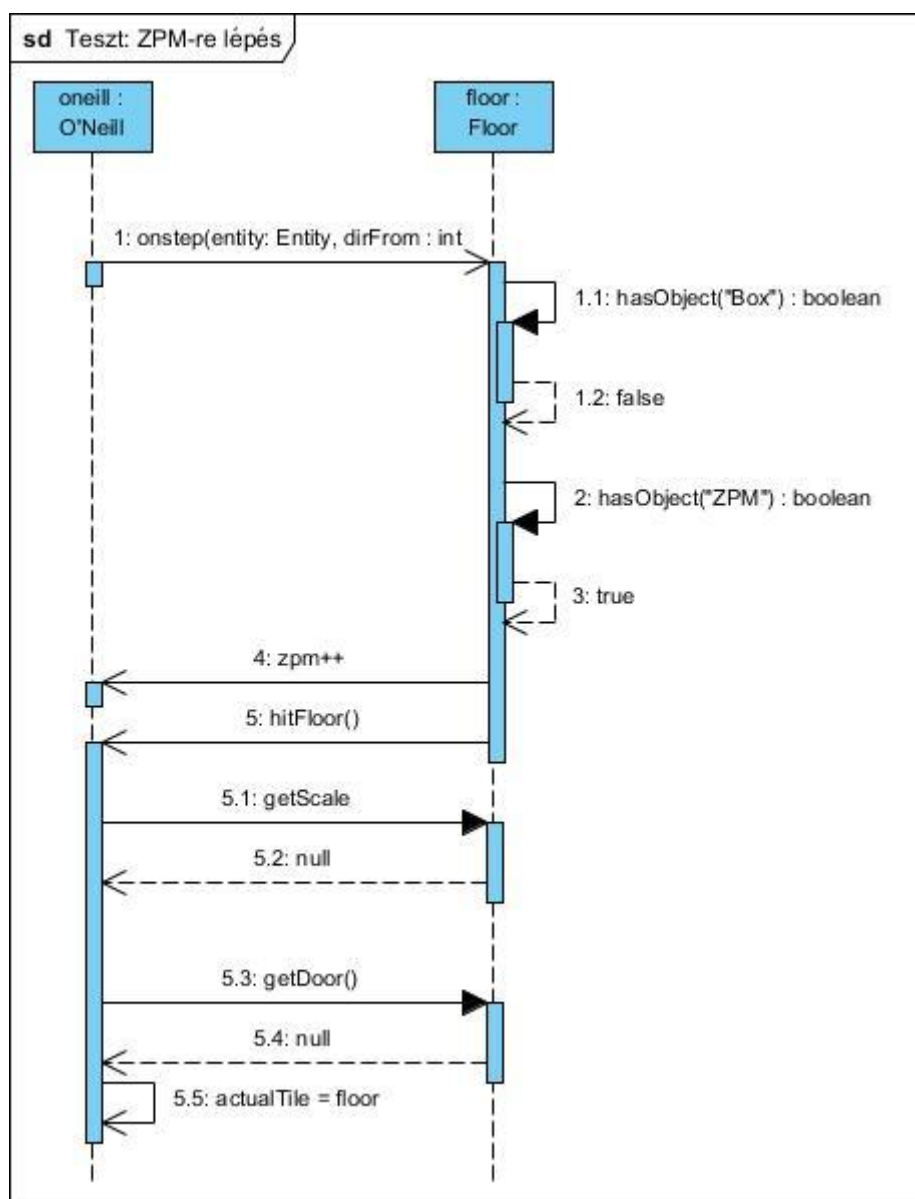
5.21. ábra Padlóra lépés



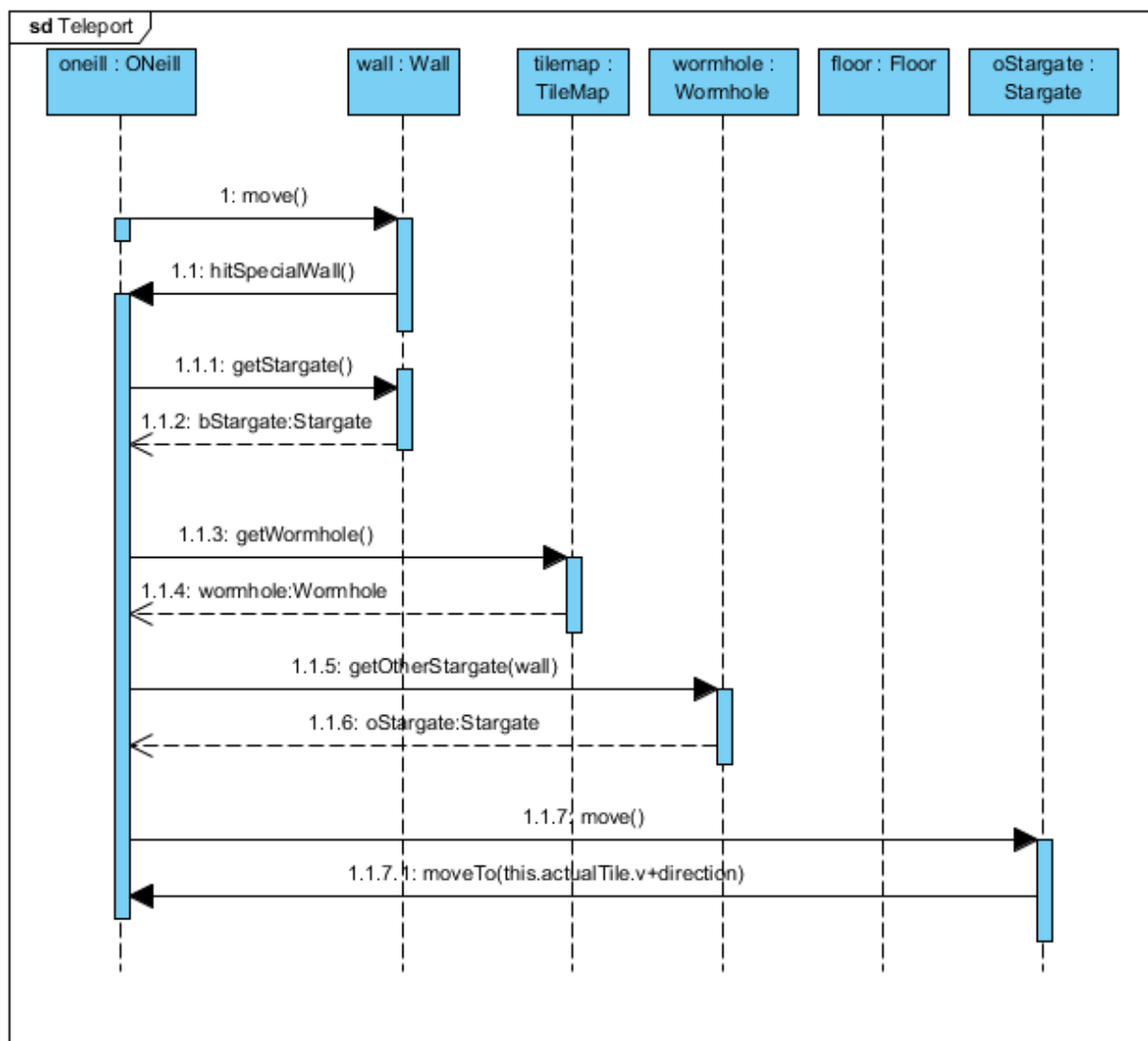
5.22. ábra Szakadékba lépés



5.23. ábra Zárt ajtóbalépés

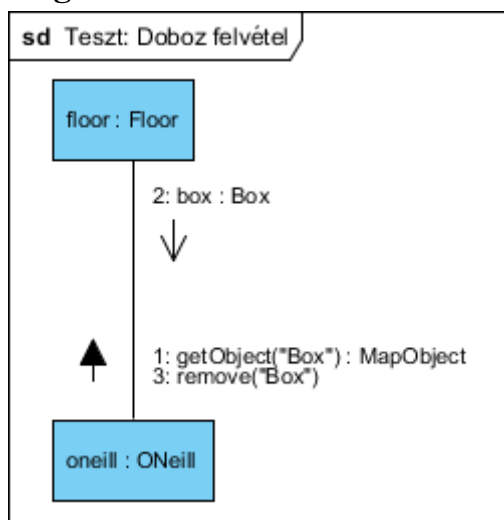


5.24. ábra ZPM-re lépés

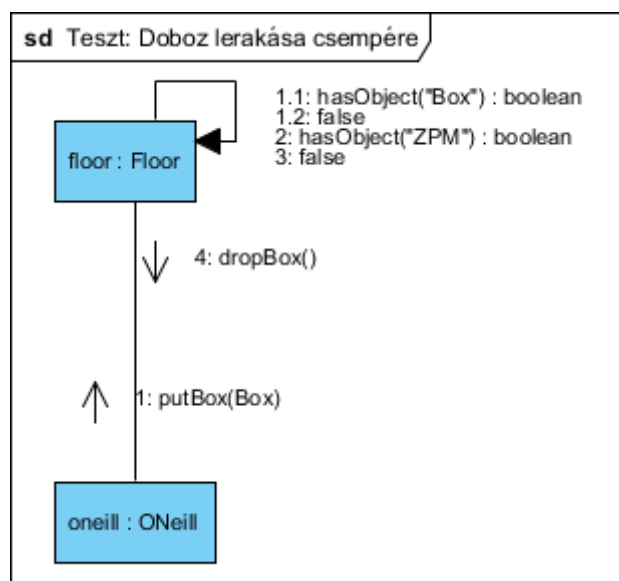


5.25. ábra Teleportálás

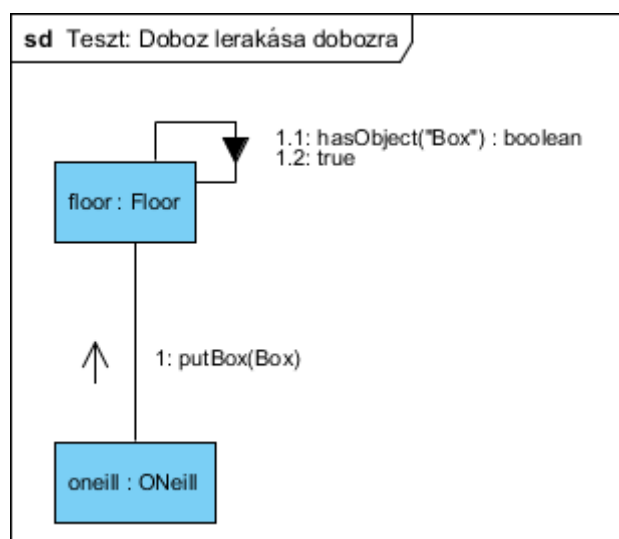
## 5.4 Kommunikációs diagramok



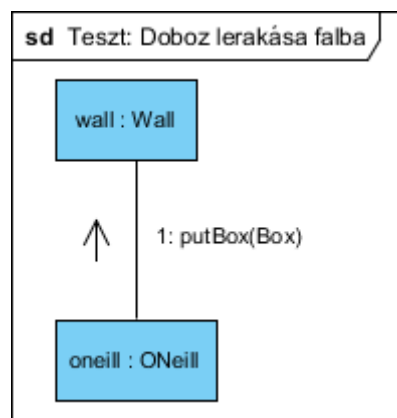
5.26. ábra Doboz felvétele



5.27. ábra Doboz lerakása csempére

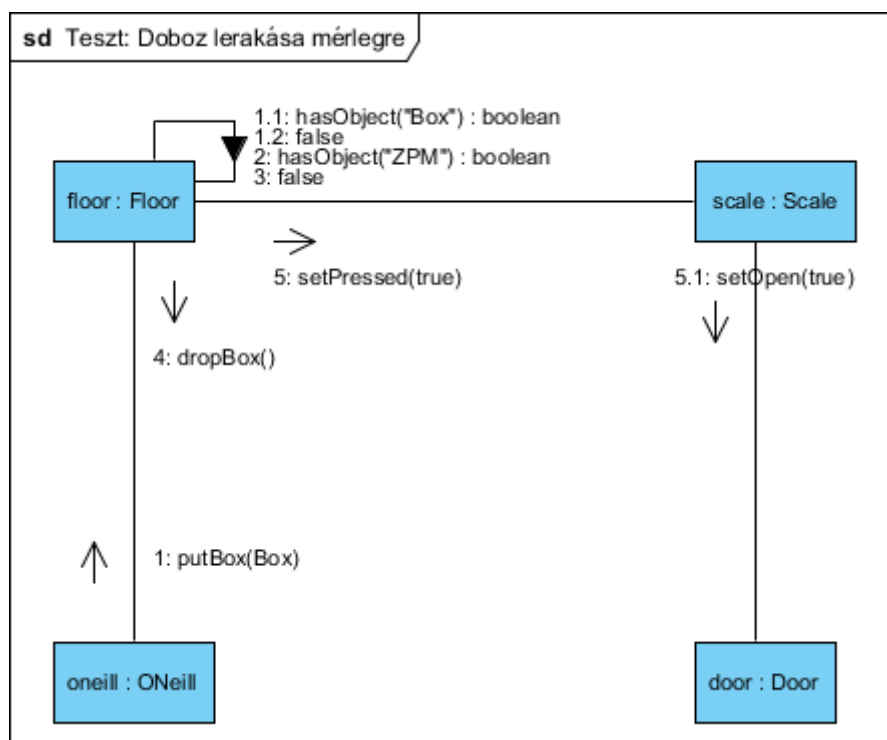


5.28. ábra Doboz lerakása dobozra

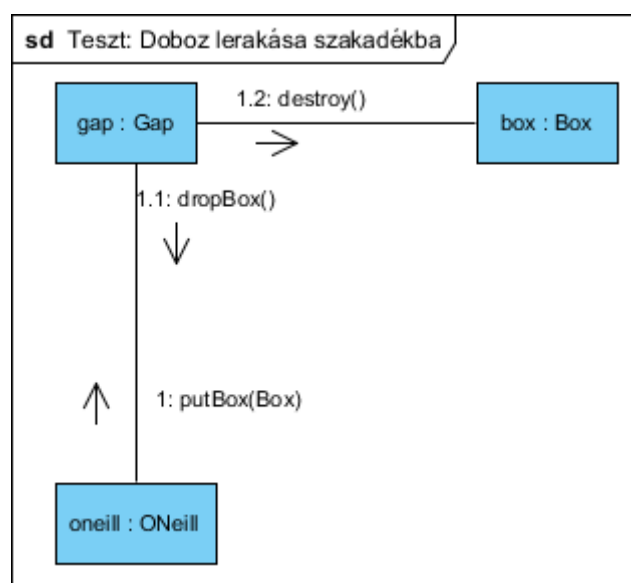


5.29. ábra Doboz lerakása falba

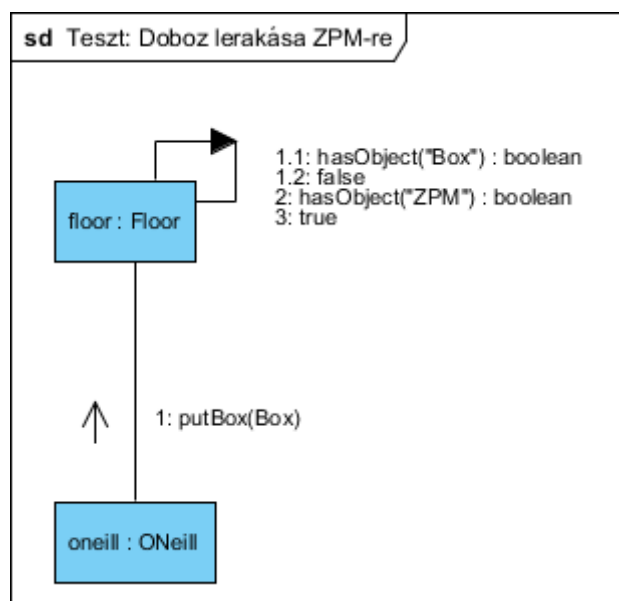




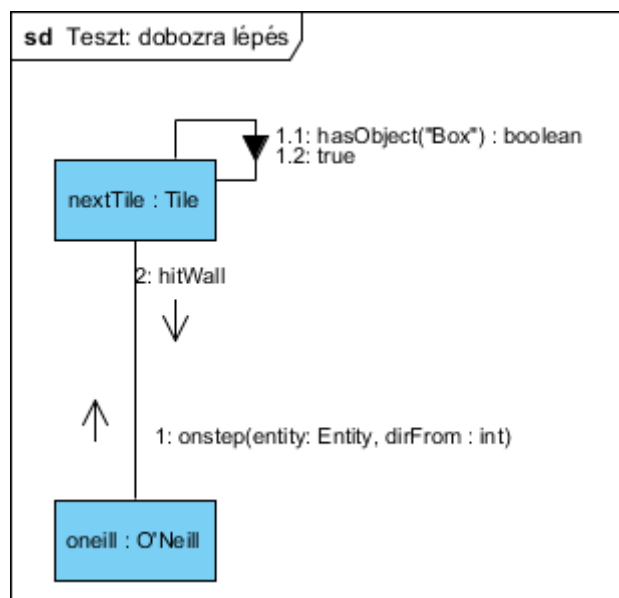
5.30. ábra Doboz lerakása mérlegre



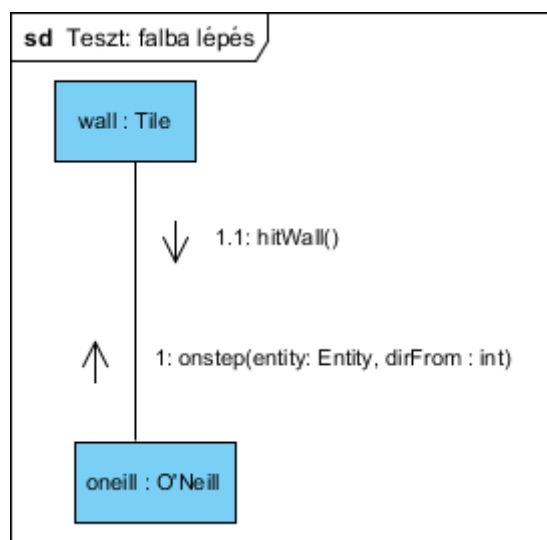
5.31. ábra Doboz lerakása szakadékba



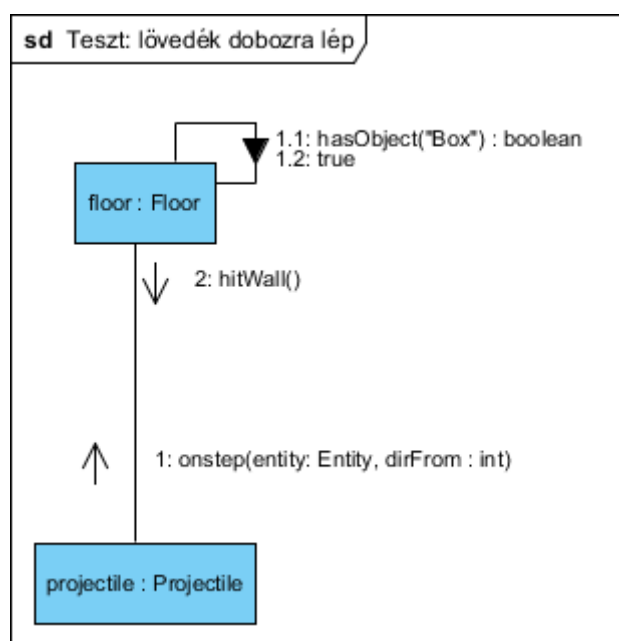
5.32. ábra Doboz lerakása ZPM-re



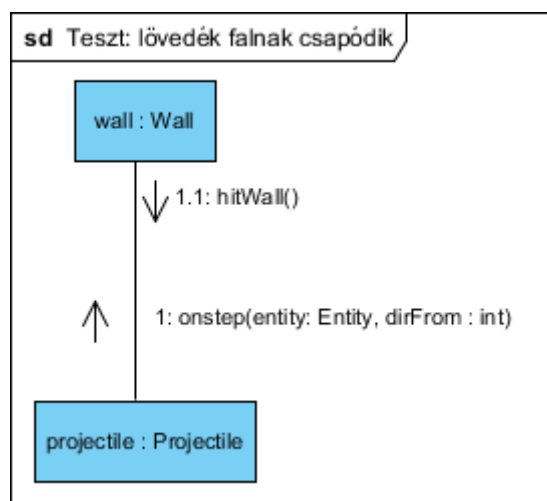
5.33. ábra Dobozra lépés



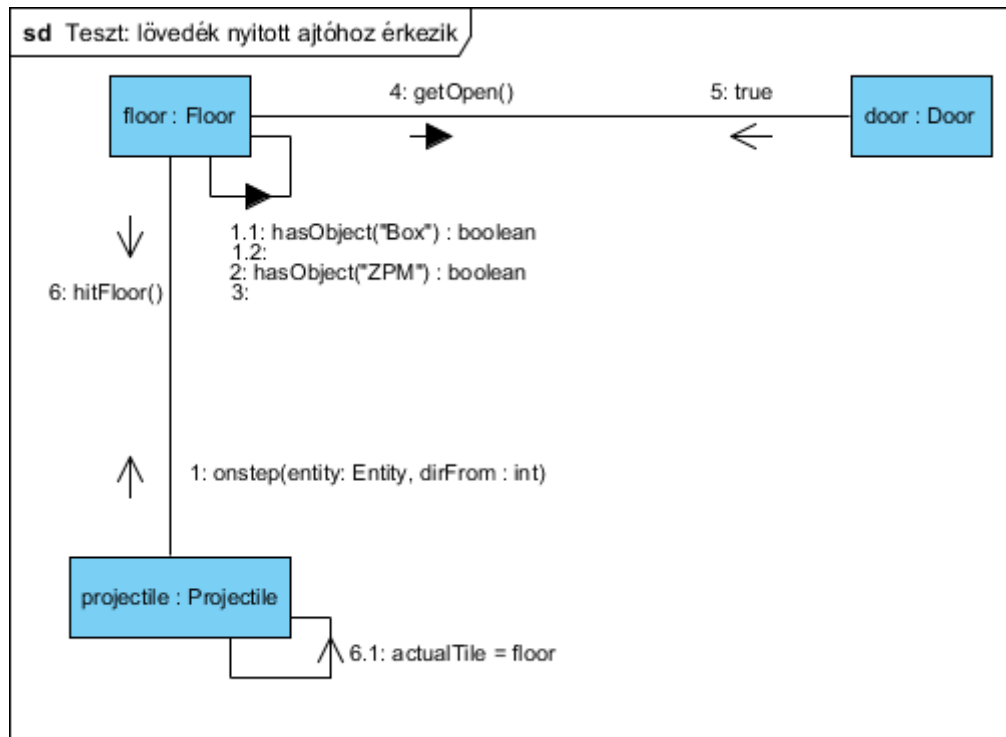
5.34. ábra Falba lépés



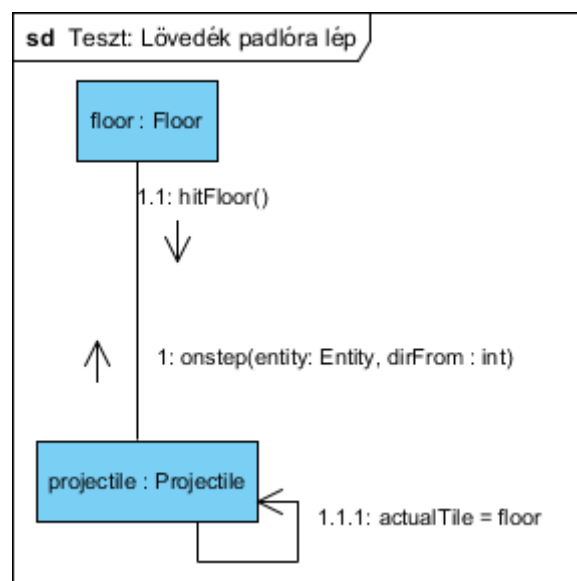
5.35. ábra Lövedék dobozra lép



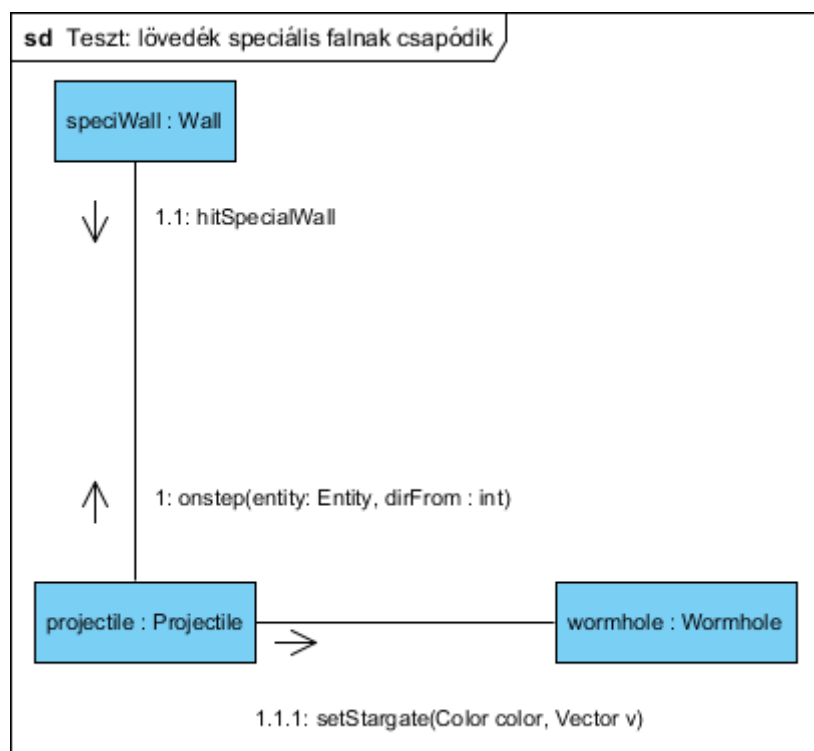
5.36. ábra Lövedék falnak csapódik



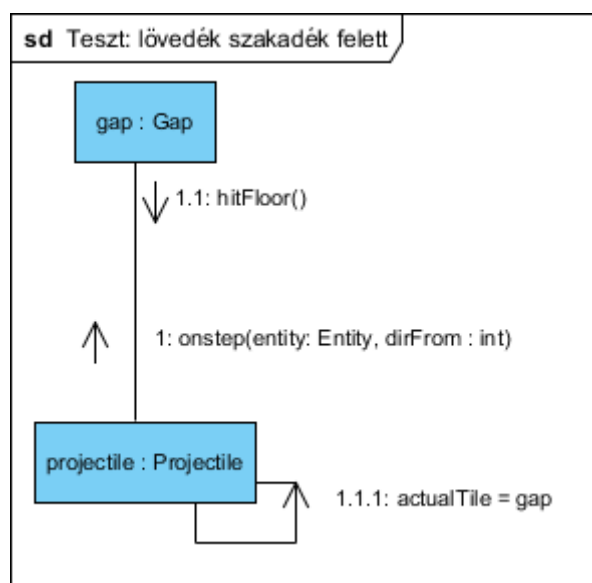
5.37. ábra Lövedék ajtóhoz érkezik



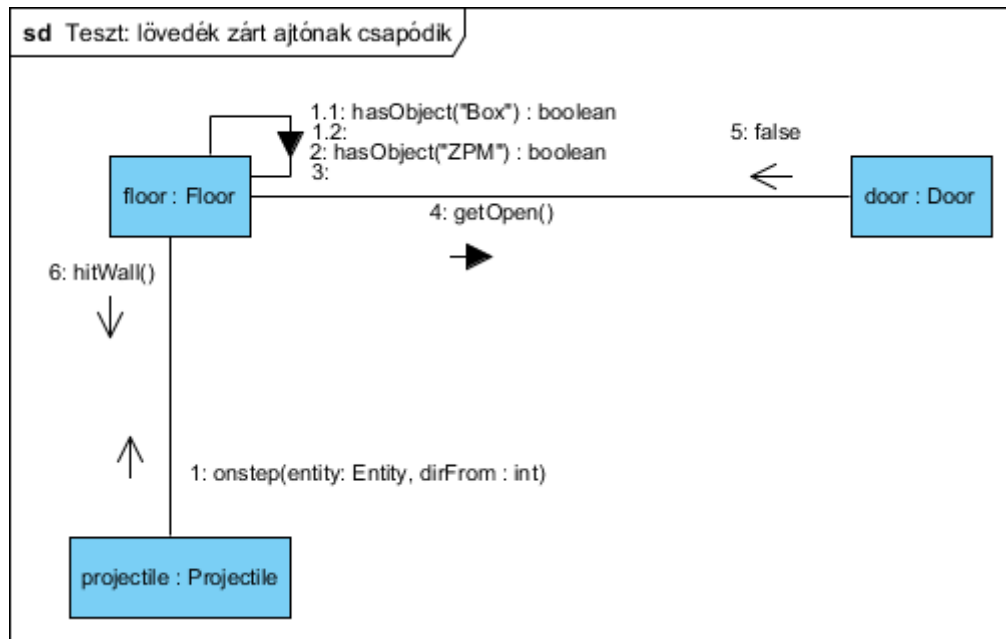
5.38. ábra Lövedék padló fölött mozog



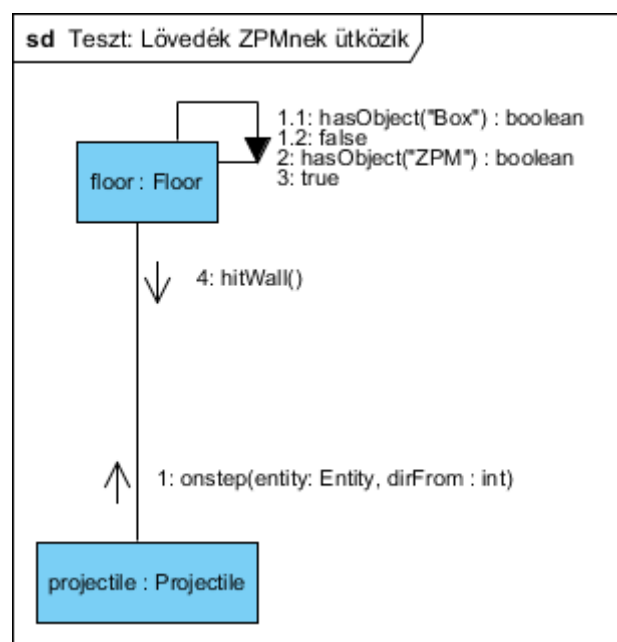
5.39. ábra Lövedék speciális falnak csapódik



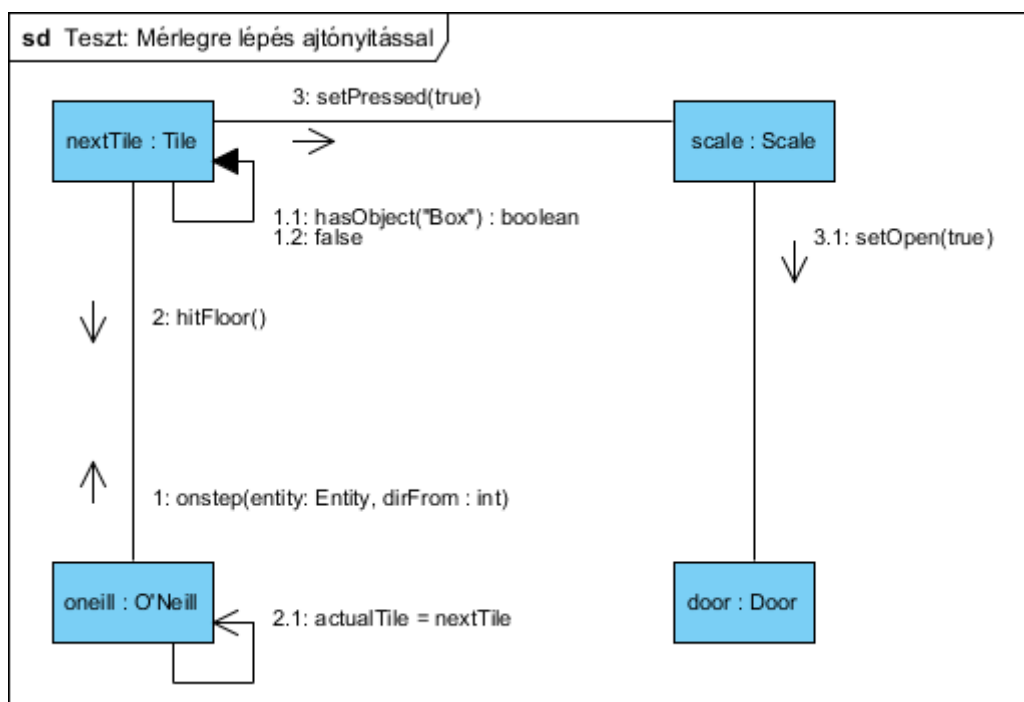
5.40. ábra Lövedék szakadék felett áthalad



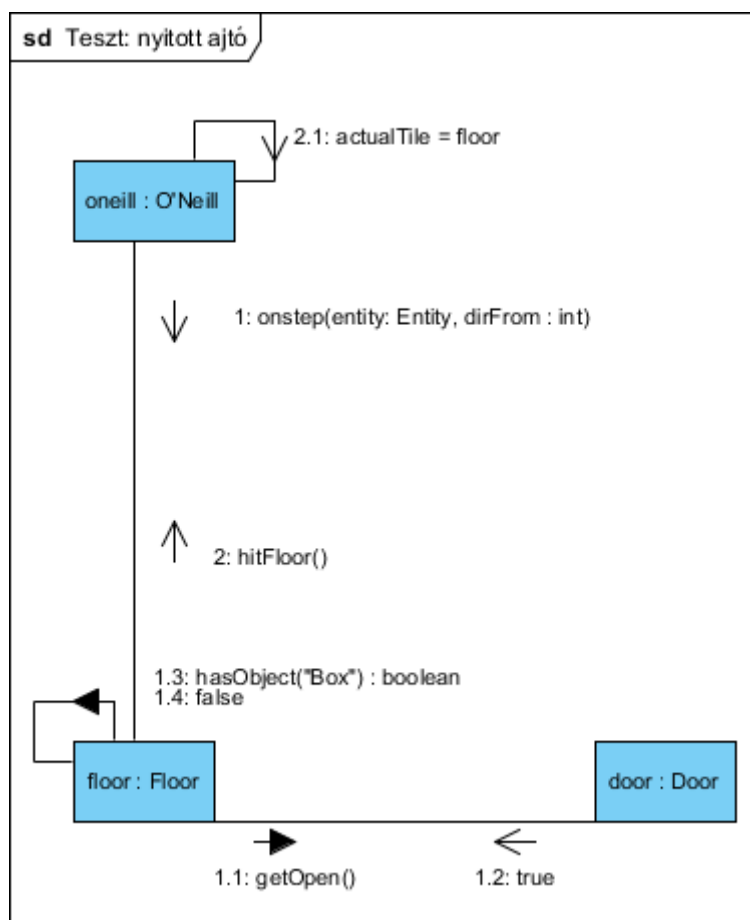
5.41. ábra Lövedék zárt ajtónak csapódik



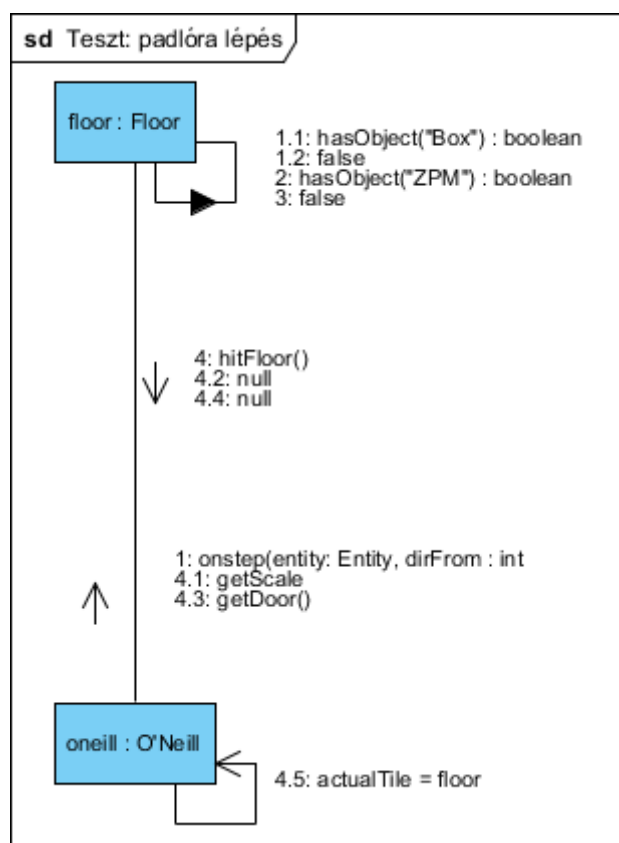
5.42. ábra Lövedék ZPM-nek ütközik



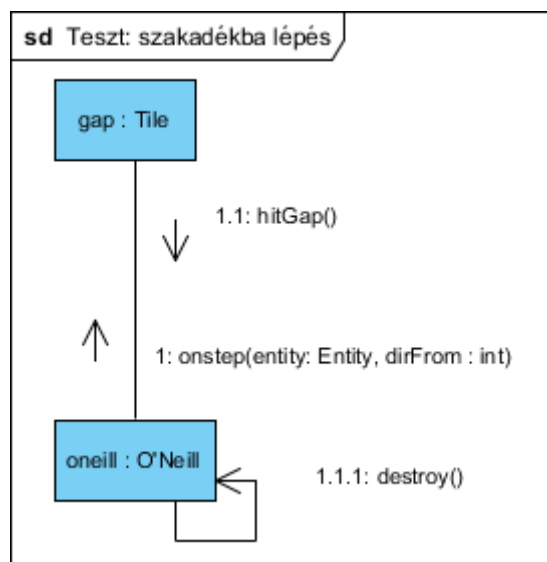
5.43. ábra Mérlegre lépés ajtónyitással



5.44. ábra Nyitott ajtóba lépés

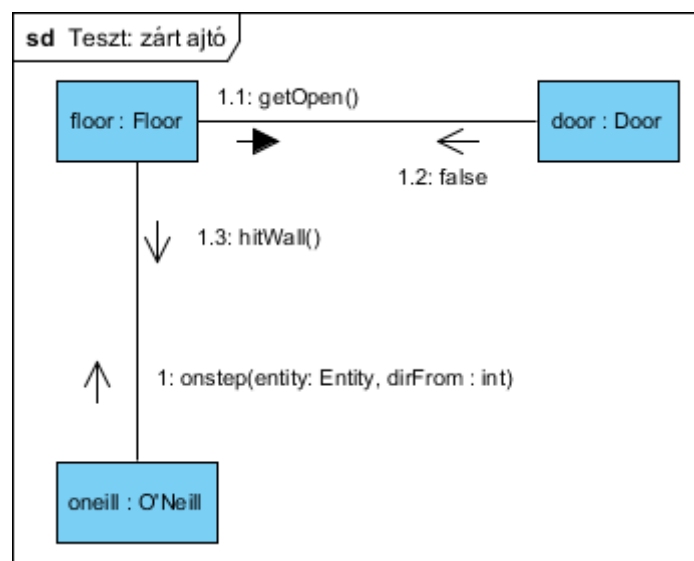


5.45. ábra Padlóra lépés

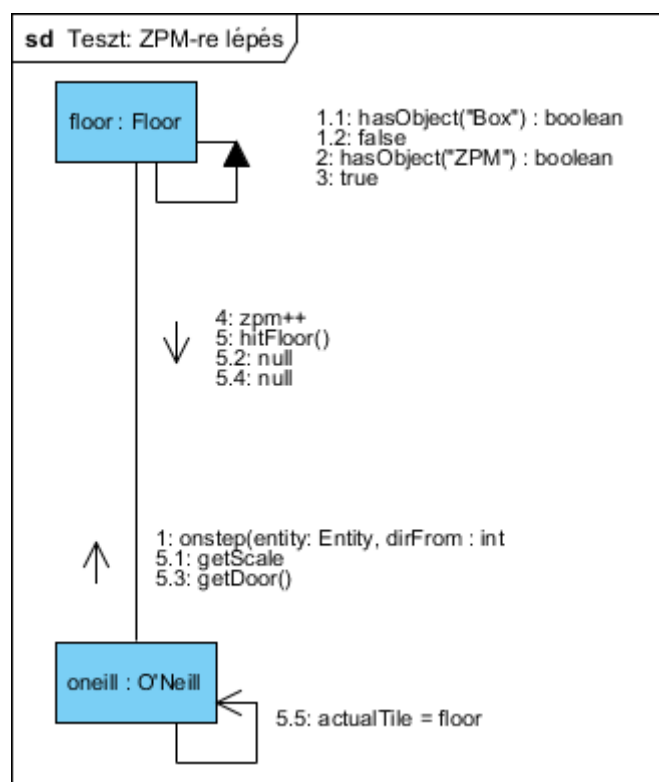


5.46. ábra Szakadékba lépés

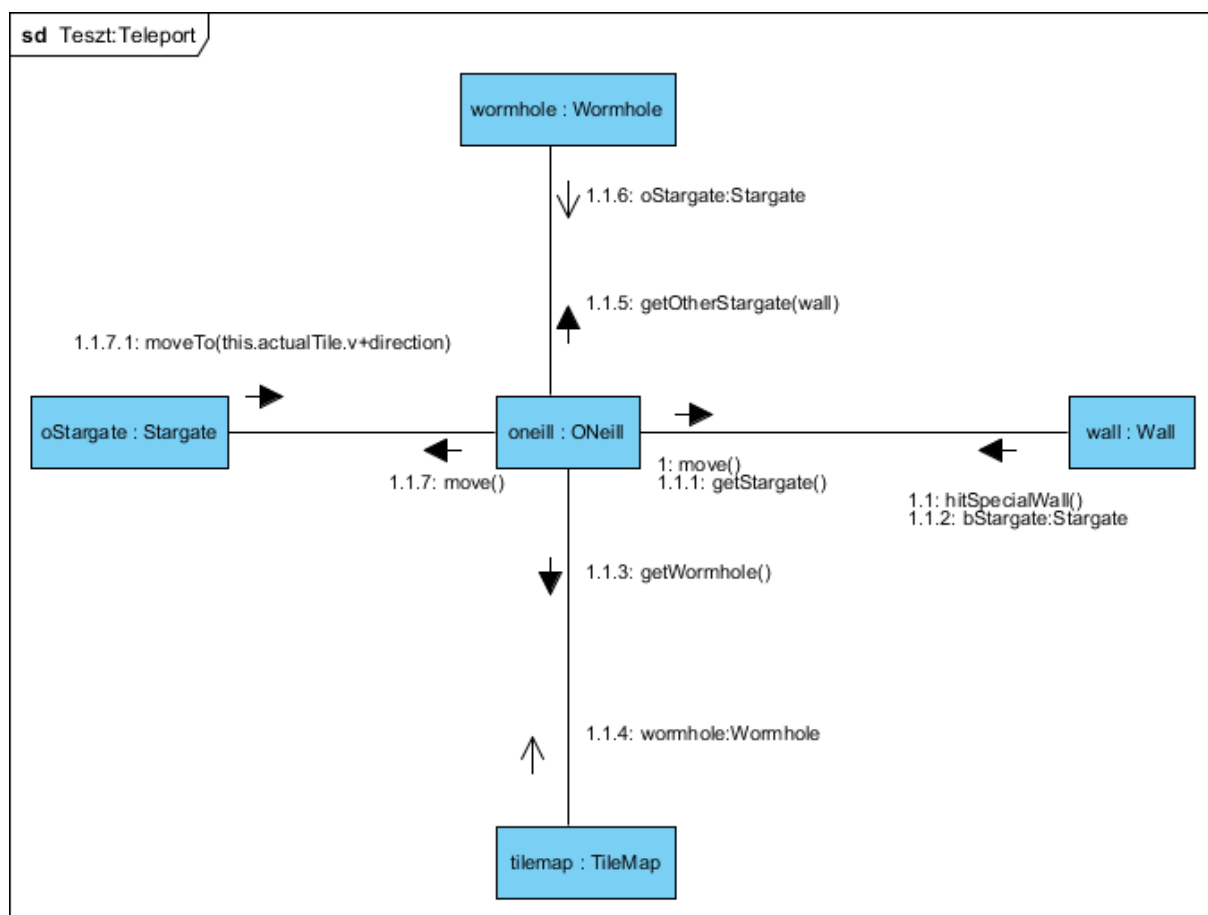




5.47. ábra Zárt ajtóba lépés



5.48. ábra ZPM-re lépés



5.49. ábra Teleportálás

## 5.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2016.03.20. 13:30	1,5 óra	Bujdosó Bukta Karácsony Vincze	Értekezlet. Döntés: Bukta elkészíti a use-case-ek leírását; Bujdosó, Bukta és Vincze felosztja egymás közt a szekvencia diagramokat és elkészíti őket.
2016.03.20. 14:00	1 óra	Bukta	Tevékenység: Use-case leírások
2016.03.20. 16:00	6 óra	Bujdosó	Tevékenység: Szekvenciadiagramok
2016.03.20. 17:30	2 óra	Vincze	Tevékenység: Szekvenciadiagramok
2016.03.20. 20:30	2 óra	Vincze	Tevékenység: Szekvenciadiagramok Előző beadás javítások
2016.03.20. 22:00	2 óra	Karácsony	Tevékenység: Use-case diagram elkészítése; Hiányzó use-case leírások pótlása; Kezelői felület menüstruktúrájának, dialógus képernyőjének leírása. Formázás megkezdése.
2016.03.20. 23:00	2 óra	Kelemen	Tevékenység: Kommunikációs diagramok elkészítése
2016.03.21. 07:15	1 óra	Karácsony	Tevékenység: Előző beadás javítása; Jelen dokumentum véglegesítése, ábrák átemelése.

## 6. Szkeleton beadás

### 6.1 Fordítási és futtatási útmutató

#### 6.1.1 Fájllista

Fájl neve	Méret (byte)	Keletkezés ideje	Tartalom
Box.java	66	2016.03.19	Box osztály
Door.java	629	2016.03.28	Door osztály
Entity.java	1034	2016.03.29	Entity osztály
Floor.java	1090	2016.03.29	Floor osztály
Game.java	2726	2016.03.28	Game osztály. Main()
GamePanel.java	1459	2016.03.18	GamePanel osztály
GameState.java	402	2016.03.18	GameState osztály
GameStateManager.java	1046	2016.03.28	GameStateManager osztály
Gap.java	686	2016.03.29	Gap osztály
LevelState.java	745	2016.03.18	LevelState osztály
MapObject.java	238	2016.03.18	MapObject osztály
MenuState.java	998	2016.03.18	MenuState osztály
O'Neill.java	2671	2016.03.29	O'Neill osztály
Projectile.java	1848	2016.03.28	Projectile osztály
Scale.java	606	2016.03.28	Scale osztály
Stargate.java	783	2016.03.29	Stargate osztály
Tests.java	8623	2016.03.29	Teszt osztály
Tile.java	1971	2016.03.29	Tile osztály
TileMap.java	698	2016.03.28	TileMap osztály
Vector.java	427	2016.03.28	Vector segédosztály
Wall.java	945	2016.03.29	Wall osztály
Wormhole.java	1099	2016.03.29	Wormhole osztály
ZPM.java	60	2016.03.18	ZPM osztály

#### 6.1.2 Fordítás

1. A mellékelt zipet kicsomagolni egy könyvtárban
2. Az adott könyvtárban kiadni az alábbi parancsot:

```
javac -d bin src/*.java
```

#### 6.1.3 Futtatás

Fordítás után adja ki az alábbi parancsokat:

- a) `cd bin`
- b) `java Stargate.Game`

## 6.2 Értékelés

Tag neve	Munka százalékban
Bujdosó Dániel	28%
Bukta Bence	16%
Horváth Balázs	15%
Karácsony Zsolt	12%
Kelemen András	10%
Vincze Zsolt	19%

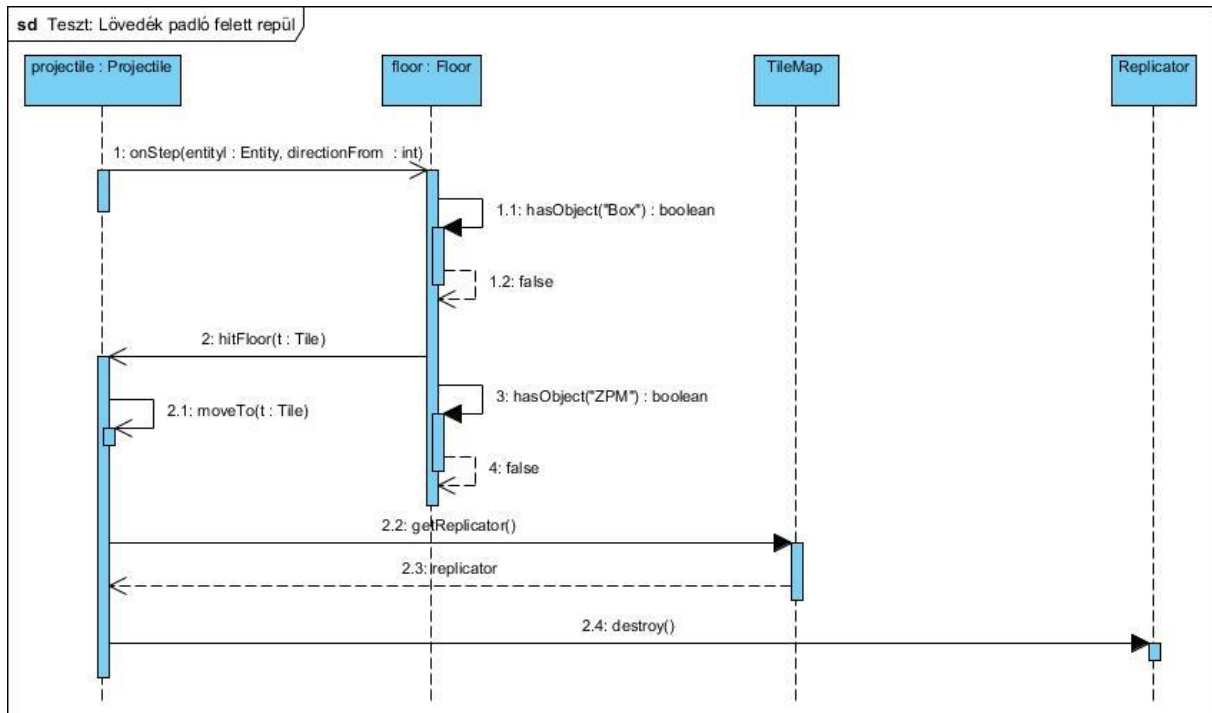
## 6.3 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2016.03.28. 16:00	8 óra	Bujdosó	Tevékenység: Szkeleton kód megírása
2016.03.29. 07:00	0,5 óra	Karácsony	Tevékenység: Dokumentáció frissítése, aktualizálása. Értékelés megírása.

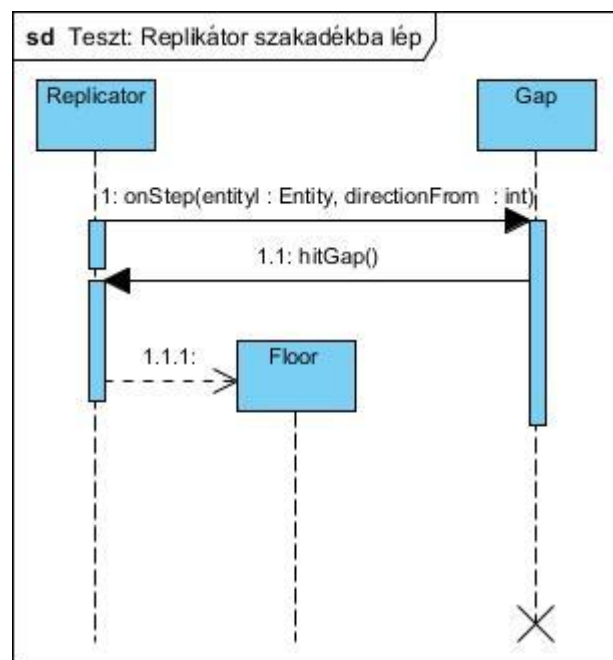
## 7.0 Osztálydiagram

### 7.1. ábra Megváltozott osztálydiagram

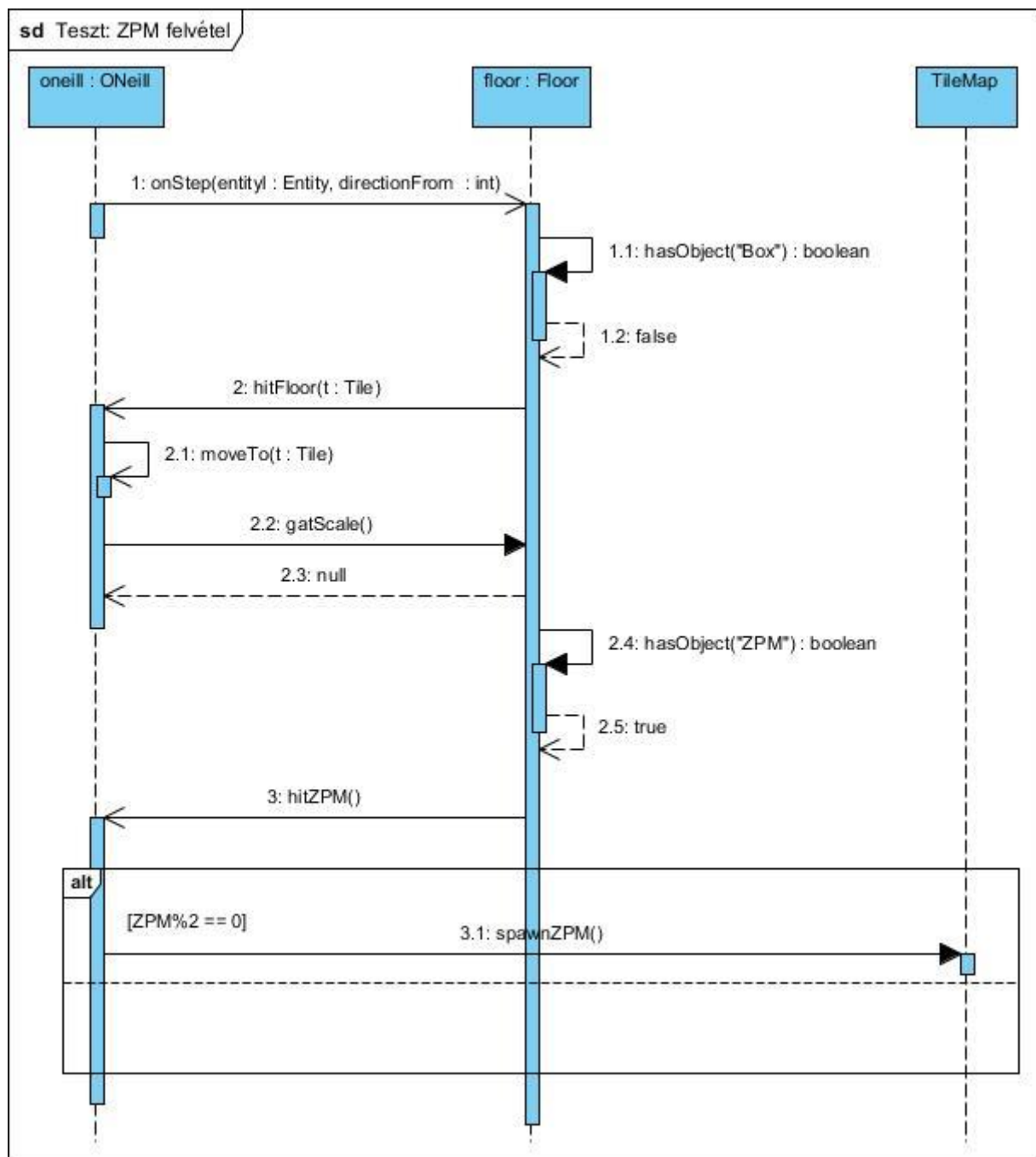
## Szekvencia diagramok



7.2. ábra Lövedék padló felett repülése

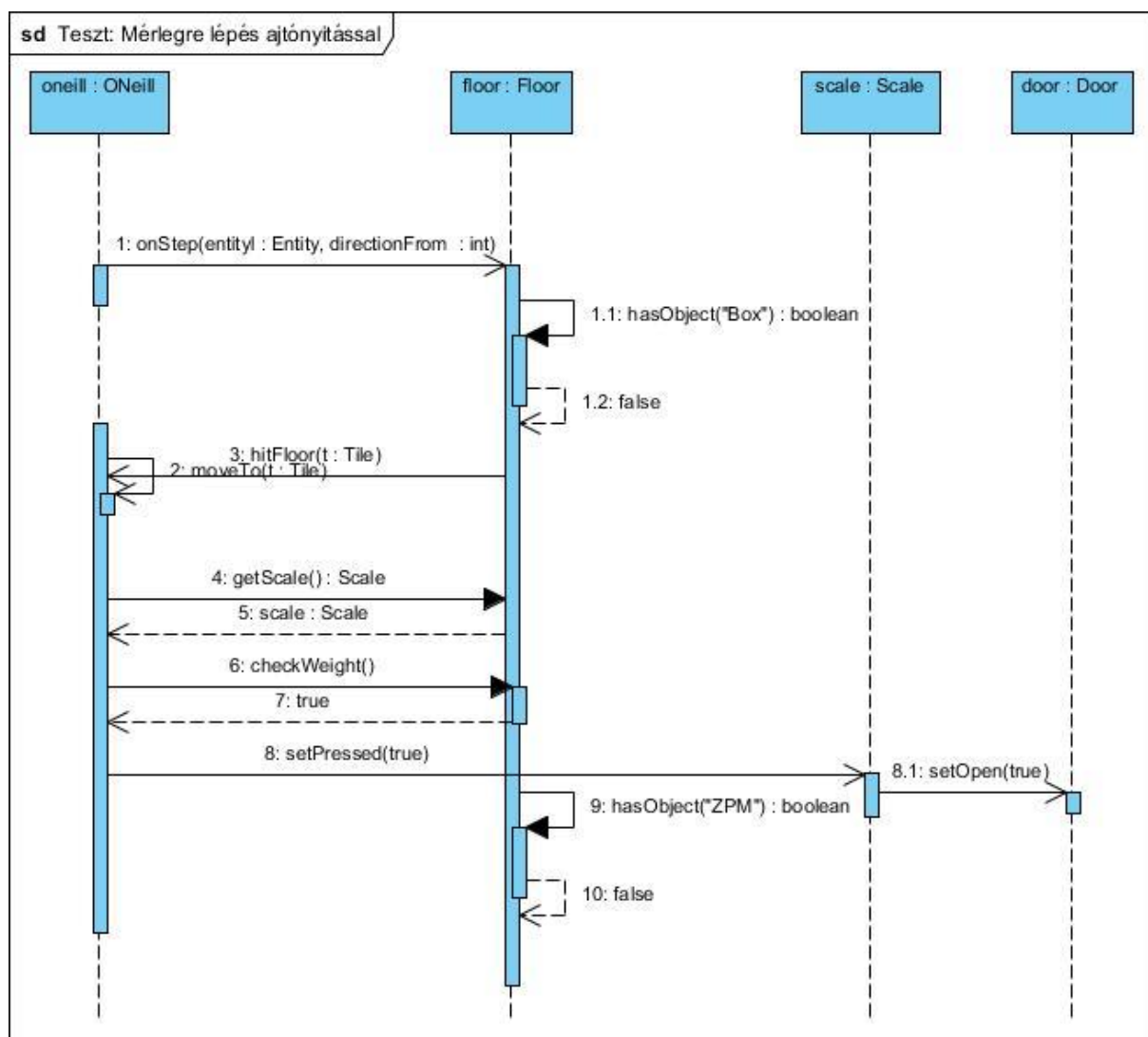


7.3. ábra Replikátor szakadékba lép

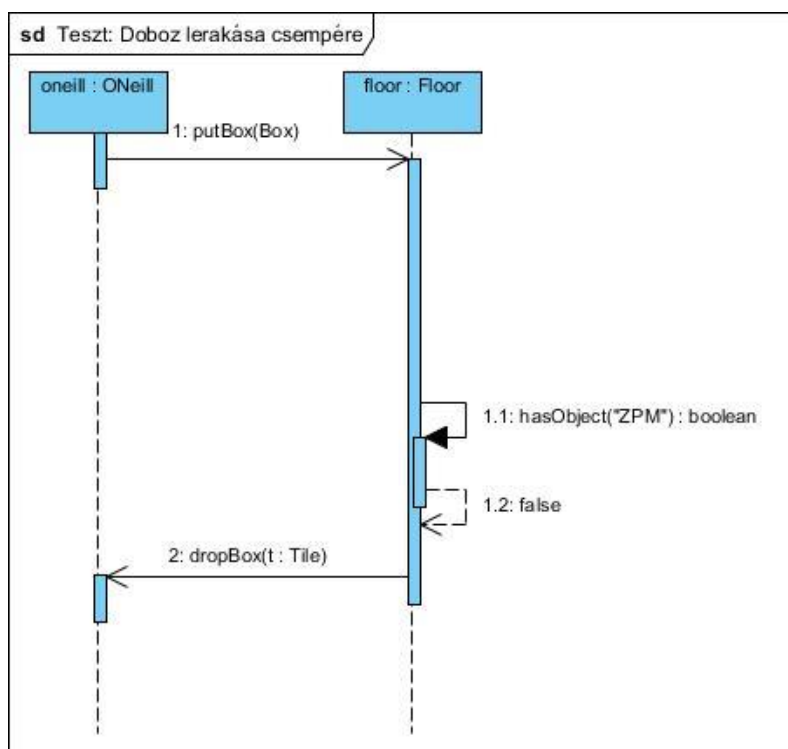


7.4. ábra ZPM felvétel

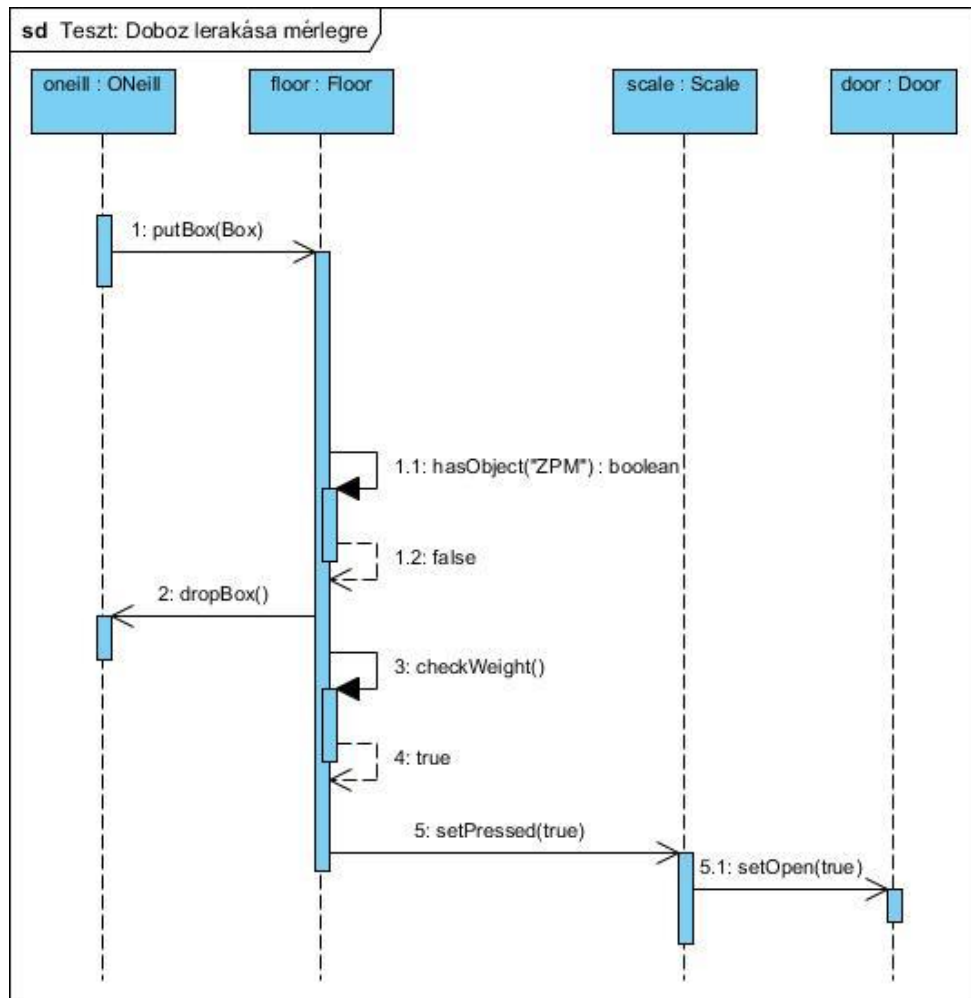




7.5. ábra Mérlegre lépés ajtónyitással



7.6. ábra Doboz lerakása csempére



7.7. ábra Doboz lerakása mérlegre

## 7.1 Prototípus interface-definíciója

### 7.1.1 Az interfész általános leírása

Az interfész csak a szabványos bemenetről fogad parancsokat valamint a szabványos kimenetre írja ki az esetleges kimenetet. Ezáltal terminálból is használható, az alábbiakban specifikált módon van lehetőség a ki- és bemeneti fájlok kezelésére is. Ez lehetőséget ad az automatikus tesztelés megteremtésére előre megírt standard tesztfájlokkal. A tesztesetek a prototípusnak adott parancsokból, valamint az elvárt kimenetből állnak. A teszteset akkor számít sikeresnek, ha az elvárt és a valóban produkált kimenet megegyezik.

### 7.1.2 Bemeneti nyelv

#### 7.1.2.1 Kapcsolók

- -in: Bemeneti fájl neve
- -rnd: Randomizálás ki/bekapcsolása
- -out: Kimeneti fájl neve

#### 7.1.2.2 Bemeneti parancsok

- Generate <x> <y>: Létrehoz egy x\*y méretű üres pályát

- `AddFloorTile <x> <y>`: Az adott koordinátára létrehoz egy megfelelő föld csempét
- `AddWallTile <x> <y><isSpecial>`: Az adott koordinátára létrehoz egy megfelelő fal csempét
- `AddGapTile <x> <y>`: Az adott koordinátára létrehoz egy megfelelő szakadék csempét
- `AddJaffa <x> <y> <d>`: Az adott koordinátára, az adott irányba nézve, elhelyez egy Jaffa-t.
- `AddONeill <x> <y> <d>`: Az adott koordinátára, az adott irányba nézve, elhelyez egy ONeill-t
- `AddReplicator <x> <y> <d>`: Az adott koordinátára, az adott irányba nézve, elhelyez egy Replicator-t.
- `AddScaleDoor <x1> <y1> <x2> <y2>`:  $x_1y_1$  koordinátára helyez egy mérleget, és  $x_2y_2$  koordinátára helyez egy hozzá tartozó ajtó
- `AddZPM <x> <y>`: Az adott koordinátára helyez egy ZPM-et.
- `AddBox <x> <y>`: Az adott koordinátára helyez egy dobozt.
- `Move <Entity> <d>`: Ha az adott entitást (Jaffa, ONeill, Replicator) az adott irányba néz, akkor mozog ebbe az irányba, egyébként csak elfordul erre.
- `PickBox <Entity>`: A megadott entitás (Jaffa, ONeill), megpróbál dobozt felvenni maga elől
- `DropBox <Entity>`: A megadott entitás (Jaffa, ONeill), megpróbál egy nála levő dobozt letenni maga elé
- `Shoot <color> <Entity>`: A megadott entitás (Jaffa, ONeill), kilövi az megadott színű lövedékét
- `AddGate <color> <x> <y> <d>`: A megadott koordinátán lévő speciális falra, a megadott irányba, létre jön egy megadott színű csillagkapu.

### 7.1.2.3 Kimeneti parancsok

- `PrintMap`: kirajzolja a labirintust
- `ListEntities`: felsorolja milyen entitások vannak a pályán, és azok koordinátáit is kiírja
- `ListWormHoles`: felsorolja a féregjáratokat és azok végpontjait.
- `ListDoorScaleState`: Felsorolja a mérlegeket a hozzájuk kapcsolt ajtókkal, és ezek állapotait.
- `ListBoxes`: Felsorolja a dobozokat és azok helyét és súlyát.
- `ListZPM`: Felsorolja a ZPM-ek helyét.

## 7.1.3 Kimeneti nyelv

### 7.1.3.1 Bemeneti parancsok kimeneti nyelve

- `Generate <x> <y>`: Kiírja a generálás eredményét.  
A lehetséges kimenetek:
  1. Sikeres generalas!
  2. Generalas hiba
- `AddFloorTile <x> <y>`: Kiírja, hogy hozzá adott egy padló csempét a koordinátájával  
A lehetséges kimenetek:

1. Padlo csempe hozzaadva. X: <x> Y: <y>
  2. Hiba a padlo csempe hozzaadasa kozben! X: <x> Y: <y>
- AddWallTile <x> <y> <isSpecial> - kiírja, hogy hozzá adott egy fal csempét a koordinátájával és típusával  
A lehetséges kimenetek:
    1. Fal csempe hozzaadva. X: <x> Y: <y> Specialis: <isSpecial>
    2. Hiba a fal csempe hozzaadasa kozben! X: <x> Y: <y> Specialis: <isSpecial>
  - AddGapTile <x> <y>: Kiírja, hogy hozzá adott egy szakadék csempét a koordinátájával.  
A lehetséges kimenetek:
    1. Szakadék csempe hozzaadva. X: <x> Y: <y>
    2. Hiba a szakadék csempe hozzaadasa kozben! X: <x> Y: <y>
  - AddJaffa <x> <y> <d>: Kiírja, hogy hozzá adott egy Jaffa etitást a koordinátájával és irányával.  
A lehetséges kimenetek:
    1. Jaffa entitas hozzaadva. X: <x> Y: <y> D: <d>
    2. Hiba a Jaffa entitas hozzaadasa kozben! X: <x> Y: <y> D: <d>
  - AddONeill <x> <y> <d>: Kiírja, hogy hozzá adott egy ONeill etitást a koordinátájával és irányával.  
A lehetséges kimenetek:
    1. ONeill entitas hozzaadva. X: <x> Y: <y> D: <d>
    2. Hiba a ONeill entitas hozzaadasa kozben! X: <x> Y: <y> D: <d>
  - AddReplicator <x> <y> <d>: Kiírja, hogy hozzá adott egy Replikátor etitást a koordinátájával és irányával.  
A lehetséges kimenetek:
    1. Replikator entitas hozzaadva. X: <x> Y: <y> D: <d>
    2. Hiba a Replikator entitas hozzaadasa kozben! X: <x> Y: <y> D: <d>
  - AddScaleDoor <x1> <y1> <x2> <y2>: Kiírja, hogy hozzá adott egy mérleg ajtó párost a koordinátájukkal.  
A lehetséges kimenetek:
    1. Merleg (X: <x1> Y:<y1>) es ajto (X: <x2> Y:<y2>) paros hozzaadva.
    2. Hiba a merleg (X: <x1> Y:<y1>) es ajto (X: <x2> Y:<y2>) paros hozzaadasa kozben!
  - AddZPM <x> <y>: Kiírja, hogy hozzá adott egy ZPM-et a koordinátájával.  
A lehetséges kimenetek:
    1. A ZPM hozzaadva. X: <x> Y: <y>
    2. Hiba a ZPM hozzaadasa kozben! X: <x> Y: <y>
  - AddBox <x> <y>: Kiírja, hogy hozzá adott egy dobozt a koordinátájával.  
A lehetséges kimenetek:
    1. Szakadék csempe hozzaadva. X: <x> Y: <y>
    2. Hiba a doboz hozzaadasa kozben! X: <x> Y: <y>
  - Move <Entity> <d>: Az adott entitás, mozgásával kapcsolatos információt ír ki.  
A lehetséges kimenetek:
    1. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol <d> irányba mozog, egy padlo csempere.
    2. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol <d> irányba mozog, egy szakadekba es meghal.

3. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol <d> irányba mozog, egy szakadekba es padlova alakul.
  4. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol nem tud <d> irányba mozogni, mert a mozgásat fal blokkolja.
  5. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol nem tud <d> irányba mozogni, mert a mozgásat doboz blokkolja.
  6. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol nem tud <d> irányba mozogni, mert a mozgásat fal blokkolja.
  7. <Entity> entitas <d> irányba fordul.
- PickBox <Entity>: Az adott entitás, doboz felvételével kapcsolatos információt ír ki.  
A lehetséges kimenetek:
    1. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyen allva, <d> iránybol, felvett egy dobozt.
    2. <Entity> entitas előtt, X: <Entity.v.x> Y: <Entity.v.y> helyen allva, <d> irányban, nincs doboz.
  - DropBox <Entity>: Az adott entitás, doboz letételével kapcsolatos információt ír ki.  
A lehetséges kimenetek:
    1. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol <d> irányba, letette a nala levo dobozt.
    2. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol <d> irányba, nem tudja letenni a nala levo dobozt.
    3. <Entity> entitasnal nincs doboz.
  - Shoot <color> <Entity>: Az adott entitás, megadott színű lövedékének kilövésével és becsapódásig történő mozgásával kapcsolatos információt ír ki.  
A lehetséges kimenetek:
    1. <Entity> entitas, X: <Entity.v.x> Y: <Entity.v.y> helyrol <Entity.v.d> irányba kilott, <color> szinu lovedeket.
    2. A lovedek X: <Projectile.actualTile.x> Y: <Projectile.actualTile.y> helyen levo, padlo csempere mozdult.
    3. A lovedek X: <Projectile.actualTile.x> Y: <Projectile.actualTile.y> helyen levo, szakadek csempere mozdult.
    4. A lovedek X: <Projectile.actualTile.x> Y: <Projectile.actualTile.y> helyen levo, falba utkozott es megsemmisult.
    5. A lovedek X: <Projectile.actualTile.x> Y: <Projectile.actualTile.y> helyen levo, specialis falba urkozott es csillagkaput hozott létre.
    6. A lovedek X: <Projectile.actualTile.x> Y: <Projectile.actualTile.y> helyen levo, dobozba utkozott es megsemmisult.
    7. A lovedek X: <Projectile.actualTile.x> Y: <Projectile.actualTile.y> helyen levo, replikatorba utkozott es megsemmisult.
  - AddGate <color> <x> <y> <d>: Adott koordinátára, az adott irányba történő adott színű csillagkapu felvételével kapcsolatos információt ír ki.  
A lehetséges kimenetek:
    1. <color> szinu csillagkapu jott létre, X: <x> Y: <y> helyen, <d> irányban.
    2. Nem jott létre <color> szinu csillagkapu, X: <x> Y: <y> helyen, <d> irányban.

### 7.1.3.2 Kimeneti parancsok kimeneti nyelve

- **PrintMap:** Kirajzolja a térképet

```
|W|W|W|W|W|W|W|
|W|F|F|S|D|F|W|
|W|W|W|W|W|W|W|
```

- **ListEntities:** Kilistázza az entitásokat, a fontosabb tulajdonságaikkal

A lehetséges kimenet:

Entitas lista:

-----

Nev: ONiell

Pozicio: X: <Entity.v.x> Y: <Entity.v.y> D:  
<Entity.v.d>

Textura azonosito: <Entity.textureID>

Sebesseg: <Entity.speed>

Halott: <Entity.isDead>

Van nala doboz: <ONeill.hasBox>

-----

Nev: ONiell

Pozicio: X: <Entity.v.x> Y: <Entity.v.y> D:  
<Entity.v.d>

Textura azonosito: <Entity.textureID>

Sebesseg: <Entity.speed>

Szin: <Projectile.color>

-----

- **ListWormHoles:** Kilistázza az fereglyukakat

A lehetséges kimenet:

Fereglyuk lista:

-----

Kek:

Pozicio: X:

<TileMap.Wormhole.stargateBlue.actualTile.v.x> Y:

<TileMap.Wormhole.stargateBlue.actualTile.v.y> D:

<TileMap.Wormhole.stargateBlue.actualTile.v.d>

Sarga: X:

<TileMap.Wormhole.stargateOrange.actualTile.v.x> Y:

<TileMap.Wormhole.stargateOrange.actualTile.v.y> D:

<TileMap.Wormhole.stargateOrange.actualTile.v.d>

Piros: X:

<TileMap.Wormhole.stargateRed.actualTile.v.x> Y:

<TileMap.Wormhole.stargateRed.actualTile.v.y> D:

<TileMap.Wormhole.stargateRed.actualTile.v.d>

Zold: X:

<TileMap.Wormhole.stargateGreen.actualTile.v.x> Y:

<TileMap.Wormhole.stargateGreen.actualTile.v.y> D:

<TileMap.Wormhole.stargateGreen.actualTile.v.d>

-----

- **ListDoorScaleState:** Kilistázza a mérleg, ajtó párosokat, a tulajdonságaikkal.

A lehetséges kimenet:

Merleg - ajtó lista:

-----

Merleg:

Pozicio: X: <TileScale.v.x> Y: <TileScale.v.y>

Alap textura azonosito: <Scale.textureIdUp>

Lenyomott textura azonosito: <Scale.textureIdDown>

Szukseges suly: <Scale.neededWeight>

Le van nyomva: <Scale.pressed>

Ajto:

Pozicio: X: <TileDoor.v.x> Y: <TileDoor.v.y>

Ajto allapot: <Scale.Door.getOpen()>

Zart ajto textura azonosito:

<Scale.Door.textureIdClose>

Nyitott ajto textura azonosito:

<Scale.Door.textureIdOpen>

-----

- ListBoxes: Kilistázza a dobozokat, a tulajdonságaikkal.

A lehetséges kimenet:

Doboz lista:

-----

Pozicio: X: <TileBox.v.x> Y: <TileBox.v.y>

-----

- ListZPM: Kilistázza a ZPM-ek helyét.

A lehetséges kimenet:

-----

ZPM poziciok:

Pozicio: X: <TileZPM.v.x> Y: <TileZPM.v.y>

Pozicio: X: <TileZPM.v.x> Y: <TileZPM.v.y>

-----

## 7.2 Összes részletes use-case

Use-case neve	-in
Rövid leírás	Bemeneti fájl nevének megadása.
Aktorok	Tesztelő
Forgatókönyv	Ezen kapcsoló segítségével adhatjuk meg a bemeneti fájl nevét.

Use-case neve	-rnd
Rövid leírás	Randimizálás ki- és bekapcsolása.
Aktorok	Tesztelő
Forgatókönyv	A Replicator mozgásának és a ZPM-ek megjelenésének véletlenszerűségének kapcsolása.

Use-case neve	-out
Rövid leírás	Kimeneti fájl nevének megadása.
Aktorok	Tesztelő
Forgatókönyv	Ezen kapcsoló segítségével adhatjuk meg a kimeneti fájl nevét.



<b>Use-case neve</b>	AddFloorTile
<b>Rövid leírás</b>	Hozzáadunk egy csempét a pályához.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A megadott koordinátájú helyre berakja a Tile

<b>Use-case neve</b>	AddWallTile
<b>Rövid leírás</b>	Hozzáadunk egy fal csempét a pályához.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A megadott koordinátájú helyre berakja a megfelelő (vagy speciális vagy nem speciális) fal Tile-t.

<b>Use-case neve</b>	AddGapTile
<b>Rövid leírás</b>	Hozzáadunk egy szakadék csempét a pályához.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A megadott koordinátájú helyre berakja a megfelelő szakadék Tile-t.

<b>Use-case neve</b>	AddJaffa
<b>Rövid leírás</b>	Hozzáadunk egy Jaffa-t.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Az adott koordinátára, az adott irányba nézve, elhelyez egy Jaffa-t.

<b>Use-case neve</b>	AddONeill
<b>Rövid leírás</b>	Hozzáadunk egy ONeill-t.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Az adott koordinátára, az adott irányba nézve, elhelyez egy ONeill-t.

<b>Use-case neve</b>	AddReplicator
<b>Rövid leírás</b>	Hozzáadunk egy Replicator-t.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Az adott koordinátára, az adott irányba nézve, elhelyez egy Replicator-t.

<b>Use-case neve</b>	AddZPM
<b>Rövid leírás</b>	Hozzáadunk egy ZPM-et.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A megadott koordinátára helyez egy ZPM-et.

<b>Use-case neve</b>	AddBox
<b>Rövid leírás</b>	Hozzáadunk egy dobozt.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Adott koordinátára helyez egy dobozt.

<b>Use-case neve</b>	Move
<b>Rövid leírás</b>	Mozgás vagy forgás az adott irányba.

<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Ha az adott entitást (Jaffa, ONeill, Replicator) az adott irányba néz, akkor mozog ebbe az irányba, egyébként csak elfordul erre.

<b>Use-case neve</b>	PickBox
<b>Rövid leírás</b>	Doboz felvétele.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A megadott entitás (Jaffa, ONeill), megpróbál dobozt felvenni maga elől.

<b>Use-case neve</b>	DropBox
<b>Rövid leírás</b>	Doboz lerakása.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A megadott entitás (Jaffa, ONeill), megpróbál egy nála levő dobozt letenni maga elé.

<b>Use-case neve</b>	AddGate
<b>Rövid leírás</b>	Csillagkapu létrehozása.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A megadott koordinátán lévő speciális falra, a megadott irányba, létre jön egy megadott színű csillagkapu.

<b>Use-case neve</b>	PrintMap
<b>Rövid leírás</b>	A pálya kirajzolása.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Kirajzolja a pályaként funkcionáló labirintust.

<b>Use-case neve</b>	ListEntities
<b>Rövid leírás</b>	A pályán aktuálisan jelen lévő entitások listázása.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Felsorolja, hogy éppen milyen entitások vannak a pályán, és azok aktuális koordinátáit is kiírja.

<b>Use-case neve</b>	ListWormHoles
<b>Rövid leírás</b>	Férekjáratok kilistázása.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Felsorolja az éppen jelen lévő féregjáratokat és a hozzájuk tartozó végpontok koordinátáit.

<b>Use-case neve</b>	ListDoorScaleState
<b>Rövid leírás</b>	Mérlegek és hozzájuk tartozó ajtók kilistázása.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Felsorolja a mérlegeket a hozzájuk kapcsolt ajtókkal, és ezek állapotait.

<b>Use-case neve</b>	ListBoxes
<b>Rövid leírás</b>	Dobozok kilistázása.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Kilistázza a pályán lévő dobozokat és azok helyét.

### 7.3 Tesztelési terv

<b>Teszt-eset neve</b>	O'Neill mozgása
<b>Rövid leírás</b>	O'Neill mozog, fordul és különböző csempékre lép, vagy próbál lépni.
<b>Teszt célja</b>	O'Neill mozgásának megvizsgálása. Megnézzük hogy melyik csempén hogy viselkedik.

<b>Teszt-eset neve</b>	Jaffa mozgása
<b>Rövid leírás</b>	Jaffa mozog, fordul és különböző csempékre lép, vagy próbál lépni.
<b>Teszt célja</b>	Jaffa mozgásának megvizsgálása. Megnézzük hogy melyik csempén hogy viselkedik.

<b>Teszt-eset neve</b>	Replicator mozgása
<b>Rövid leírás</b>	Replicator mozog, fordul és különböző csempékre lép, vagy próbál lépni.
<b>Teszt célja</b>	Replicator mozgásának megvizsgálása. Megnézzük hogy melyik csempén hogy viselkedik.

<b>Teszt-eset neve</b>	Pálya létrehozása
<b>Rövid leírás</b>	Létrehozunk a pályát, majd lerakunk rá csempéket, entitásokat valamint objektumokat. Majd ezután megnézzük, hogy megfelelő pálya jött-e létre.
<b>Teszt célja</b>	A pálya létrehozásának vizsgálata, megfelelően működik, e a generálásunk.

<b>Teszt-eset neve</b>	Doboz felvétele és lerakása
<b>Rövid leírás</b>	Felveszünk egy dobozt majd lerakjuk máshova.
<b>Teszt célja</b>	Megvizsgáljuk hogy képesek vagyunk felvenni dobozt, majd lerakni máshova (mozgatni) a dobozt.

<b>Teszt-eset neve</b>	Féreg-járat létrehozása
<b>Rövid leírás</b>	Létrehozunk két csillag-kaput, és ezáltal egy féreg-járatot.
<b>Teszt célja</b>	Megvizsgáljuk, hogy létrejön-e megfelelően a féregjárat.

<b>Teszt-eset neve</b>	Teleportálás
<b>Rövid leírás</b>	Belépünk a csillag-kapuba, majd a másik oldalon kijövünk.
<b>Teszt célja</b>	Megvizsgáljuk hogy át tudunk-e menni a csillagkapun. Ezt minden entitással elvégezhetjük.

<b>Teszt-eset neve</b>	ZPM-et felvesz
<b>Rövid leírás</b>	Felveszünk egy ZPM-et.
<b>Teszt célja</b>	Megnézzük hogy feltudunk-e venni ZPM-et.

<b>Teszt-eset neve</b>	Mérlegre lépés
<b>Rövid leírás</b>	Rálépünk a mérlegre, majd lelépünk róla.
<b>Teszt célja</b>	Megnézzük hogy rálépéskor változott-e a mérleg illetve a hozzá tartozó ajtó állapota.

<b>Teszt-eset neve</b>	Doboz mérlegre helyezése, majd felvétele
<b>Rövid leírás</b>	Dobozt helyezünk a mérlegre, majd felveszük.
<b>Teszt célja</b>	Megnézzük hogy mi történik a mérleggel és a hozzá kapcsolódó ajtóval, ha rárakunk egy dobozt.

#### 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A teszteléshez nem kívánunk különösebb segédeszközt felhasználni. A bemeneti és kimeneti állományokat manuálisan fogjuk előállítani. A tesztelés menetének könnyítése érdekében tervezzük egy egyszerűbb tesztelő program írását, amely csak annyit tud, hogy előállítja a program kimenetét az összes megadott bemenetre és összeveti az előre elvárt állományokkal.

## 7.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2016.04.02. 15:00	1,5 óra	Bujdosó Bukta Horváth Kelemen	Értekezlet. Döntés: Horváth szerkeszti a tesztelési tervet, Bukta elkészíti a bemeneti és kimeneti nyelv leírását, Kelemen megírja részletes use-case-eket, Bujdosó frissíti az osztálydiagramot és a szekvenciákat.
2016.04.02. 16:30	0,5 óra	Horváth	Tevékenység: Tesztelési terv szerkesztése.
2016.04.02. 16:30	0,5 óra	Kelemen	Tevékenység: Use-case-ek kifejtésének megírása.
2016.04.02. 16:30	3 óra	Bukta	Tevékenység: Bemeneti nyelv leírások. Kimeneti nyelv fejezet.
2016.04.03. 16:00	0,5 óra	Bujdosó	Tevékenység: Osztálydiagram frissítése
2016.04.03. 22:00	1.5 óra	Bukta	Tevékenység: Kimeneti nyelv véglegesítése
2016.04.04. 00:00	1,5 óra	Bujdosó	Tevékenység: Szekvencia diagram frissítése
2016.04.04. 6:15	1,5 óra	Karácsony	Tevékenység: Dokumentum véglegesítése, specifikáció változás átvezetése külön dokumentumba.

## 8. Részletes tervek

### 8.1 Osztályok és metódusok tervei

#### 8.1.1 Game

- **Felelősség**  
A játék belépési pontja, létrehozza az ablakot illetve kezeli az esetleges kapcsolókat a program indításakor.
- **Össztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**
  - + Main(String[] args)

#### 8.1.2 GamePanel

- **Felelősség**  
A GamePanel osztály indítja a játékot futtató szálát, valamint tartalmazza játékciklust (gameloop). Ez az osztály tölti ki a Game osztályban létrehozott ablakot, valamint fogadja a bemenetet a billentyűzetről és továbbadja azokat.
- **Össztályok**  
java.lang.Object -> java.awt.Component -> java.awt.Container -> javax.swing.JComponent
- **Interfészek**  
Runnable, KeyListener
- **Attribútumok**
  - **+WIDTH : int**: A játéklablak szélessége. Default értéke 320.
  - **+HEIGHT : int**: A játéklablak magassága. Default értéke 240.
  - **-thread : Thread**: A játékot futtató szálát ebben az objektumban tároljuk.
  - **-running : boolean**: Logikai érték, mely azt mutatja, hogy a játék fut vagy sem.
  - **-FPS : int**: A játék FPS értékét tartalmazó egész szám. Default értéke 60.
  - **-targetTime : long**: Az újrarajzoltatás frekvenciáját tároló érték. Default értéke 1000ms/FPS
  - **-image : BufferedImage**: A játék képét tartalmazó mutató.
  - **-g : Graphics2D**: A kirajzolásért felelős objektum példánya.

- **-gsm : GameStateManager:** A játékmenet kezelését végző GameStateManagerobjektum egy példánya.
- **Metódusok**
  - **+GamePanel():** Az osztály konstruktora.
  - **+addNotify() : void:** Kapcsolatot létesít a natív kimenet és a program között.
  - **+run() : void:** A játékot futtató szál elindításáért felelős illetve a játékcikust (gameloop) tartalmazó metódus.
  - **+keyTyped(): void:** Billentyűzeten érkező karaktersorozatot figyelő metódus
  - **+keyPressed(KeyEvent key) : void:** Egy billentyű leütését figyelő metódus. A leütött billentyű kódját továbbítja a GameStateManager felé
  - **+keyReleased(KeyEvent key) : void:** Egy billentyű felengedését figyelő metódus. A felengedett billentyű kódját továbbítja a GameStateManager felé

### 8.1.3 GameState

- **Felelősség**

A játékállapotok abstract őssztálya, tartalmazza a megvalósítandó metódusokat valamint egy GameStateManager-t.
- **Őssztályok**

Nem értelmezhető.
- **Interfészek**

Nem értelmezhető.
- **Attribútumok**
  - **#gsm:** GameStateManager: A játék állapotát kezelő GameStateManager objektum egy példánya.
- **Metódusok**
  - **-init() : void:** Játékállapot inicializálására szolgáló metódus.
  - **+draw(Graphics2D g) : void:** A játékállapot és annak elemeit kirajzoló metódus.
  - **+update() : void:** A játékállapot frissítését végző metódus.
  - **+keyPressed() : void:** Billentyű lenyomását figyelő metódus.
  - **+keyReleased() : void:** Billentyű felengedését figyelő metódus.

### 8.1.4 GameStateManager

- **Felelősség**

A játék különböző állapotai közötti váltásért felelős osztály. Ez az osztály kapja meg a GamePanel osztálytól a leütött billentyűket, majd továbbadja őket az éppen aktuális játékállapotnak.
- **Őssztályok**

Nem értelmezhető.

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

- **-gameStates : ArrayList<GameState>**: A játék során előforduló állapotokat tároló lista.
- **-currentState : int**: Az aktuális játékállapotot jelző változó
- **+MENUSTATE : int**: A játék menüjének állapotát reprezentáló változó. Default értéke 0
- **+LEVELSTATE : int**: A választott pálya állapotát reprezentáló változó. Default értéke 1.

- **Metódusok**

- **+GameStateManager()**: Az osztály konstruktora.
- **+setState(int state): void**: A játék állapotának beállítása ezen a metóduson keresztül történik
- **+update() : void**: A játékállapotot frissítő metódus. Mindig az aktuális játékállapot update() metódusát hívja.
- **+draw(Graphics2D g) : void**: A játékállapotot kirajzoló metódus. Mindig az aktuális játékállapot draw(Graphics2D g) metódusát hívja meg.
- **+keyPressed(int k) : void**: Billentyű lenyomását figyelő metódus. Továbbítja az aktuális játékállapot felé.
- **+keyReleased(int k) : void**: Billentyű felengedését figyelő metódus. Továbbítja az aktuális játékállapot felé

### 8.1.5 MenuState

- **Felelősség**

A játék főmenüjéért felelős osztály, ezen osztály felelőssége hogy a képernyőre a megfelelő menüpontok legyenek kirajzolva, majd a megfelelő billentyűk leütése után elindítja a játékot vagy kilép belőle.

- **Ősosztályok**

GameState

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

- **-currentChoice : int**: A felhasználó által aktuálisan kiválasztott menüpont értéke. Default értéke 0.
- **-options : String[]**: A menü elérhető állapotait tartalmazza. Lehetséges értékei "Start" és "Quit"
- **-titleColor : Color**: A főcím színét tartalmazó változó.
- **-titleFont : Font**: A főcím betűtípusát tartalmazó változó.
- **-font : Font**: A menüben előforduló betűtípust tartalmazó változó.



- **Metódusok**

- **+MenuState(GameStateManager gsm):** Az osztály konstruktora, amiben átadjuk a játékállapotokat kezelő GameStateManager példányt.
- **+init() void:** A menü inicializálását végző metódus, ezen állapot esetén nincs feladata
- **+update() void:** A menü frissítéséért felelős metódus, ezen állapot esetén nincs feladata.
- **+draw(Graphics2D g) void:** A menüt a megadott színnel és betűtípussal kirajzoló metódus.
- **+keyPressed(int k) : void:** Billentyű lenyomását figyelő metódus. Billentyűtől függően változtatja az aktuálisan kiválasztott menüpontot, vagy kiválasztja azt.
- **+keyReleased(int k) : void:** Billentyű felengedését figyelő metódus.

### 8.1.6 LevelState

- **Felelősség**

A játékmenetért felelős játékállapot (GameState) osztály. Ő felel a labirintusért illetve a labirintusban szereplő elemekért, mint például O'Neill ezredes.

- **Ősosztályok**

GameState

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

- **-tileMap : TileMap:** A pálya alapját adó csempéket tartalmazó objektum.

- **Metódusok**

- **+LevelState (GameStateManager gsm) : void:** Az osztály konstruktora, amiben átadjuk a játékállapotokat kezelő GameStateManager osztály példányát.
- **+init() : void:** A pálya inicializálását végző metódus. Ez a metódus olvassa be a pályát tartalmazó JSON fájlból a pályaelemeket és ez alapján tölti fel a tileMap változót, valamint elhelyezi a pályán a kezdő entitásokat (pl. O'Neill).
- **+update() : void:** A pályaállapot frissítését végző metódus, ami majd az entitások folyamatos mozgatásáért felel.
- **+draw(Graphics2D g) : void:** A labirintust és az azon lévő elemeket kirajzoló metódus.
- **+keyPressed(int k) : void:** Billentyű lenyomását figyelő metódus. A billentyezettől függően cselekszik (mozog, lö dobozt felvesz stb.)
- **+keyReleased(int k) : void:** Billentyű felengedését figyelő metódus.

### 8.1.7 TileMap

- **Felelősség**

Ez az osztály tárolja magában a labirintust, illetve az ahhoz szükséges grafikai elemeket. A labirintus valamint a hozzá tartozó csempegrafika beolvasás is az ő feladata.

- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-textureMap : Map<int, BufferedImage>**: Ez az attribútum tárolja a pályát alkotó csempék képeit és azok sorszámár kulcs-érték párokban.
  - **-tiles : Tile[][]**: A csempék ebben a tömbben vannak eltárolva.
  - **-wormhole : Wormhole**: A pályán található féregjáratokat tároló és kezelő változó.
  - **-replicator**: A pályán tartózkodó replikátor referenciája.
- **Metódusok**
  - **+getTextureMap(int textureID) : BufferedImage**: Ez a függvény visszaadja az adott ID-hoz tartozó csempe képét.
  - **+loadTextureMap(String filePath) : void**: A pályán található csempék képét ez a függvény olvassa be egy fájlból.
  - **+loadTileMap(String filePath) : void**: A pályát(labirintust) egy fájlból beolvasó metódus.
  - **+getTile(int x, int y) : Tile**: Az adott (x,y) koordinátán található csempét visszaadó metódus.
  - **+getWormhole : Wormhole**: Visszaadja a pályán található féregjáratokat kezelő objektumot.
  - **+spawnZPM():void**: Egy véletlenszerű, de szabad mezőn létrehoz egy új ZPM-et.
  - **+getReplicator():Tile**: visszaadja a replikátor helyzetét jelző csempét.

### 8.1.8 Tile

- **Felelősség**  
A játékkeret felépítő labirintus alap építőegysége.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-x : int**: A csempe x koordinátáját tároló változó.
  - **-y : int**: A csempe y koordinátáját tároló változó.
  - **+textureID : int**: A csempéhez tartozó grafika azonosítóját tartalmazó változó.
  - **+mapObjects : List<MapObject>**: A csempén található MapObject-ek egy listában tárolva.
  - **-door : Door**: A csempéhez tartozó ajtót tárolja.
  - **-scale : Scale**: A csempéhez tartozó mérleget tárolja.

- **Metódusok**

- **+onStep(Entity e, int directionfrom) : void**: Ez a metódus hívódik meg ha egy entitás (Entity) “rálép” az adott csempére. Működése az adott csempe típusától függ.
- **+putBox(Player p, Box box):void**: Ez a metódus hívódik meg ha egy játékos (Player) egy dobozt próbál az adott csempére helyezni. Működése az adott csempe típusától függ.
- **+draw(Graphics2D g) : void**: Ez a metódus lekérdezi a grafikákat tartalmazó Map-tól a hozzá tartozó grafikát majd kirajzolja azt illetve a vele egy koordinátán szereplő MapObjectek draw metódusait is meghívja.
- **+getNextTile(int direction) : Tile**: A megadott irányban lévő következő csempét adja vissza.
- **+getDoor(): Door**: A csempén található ajtót adja vissza.
- **+getScale(): Scale**: A csempén található mérleget adja vissza.
- **+getX(): int**: A csempe x koordinátáját adja vissza.
- **+getY(): int**: A csempe y koordinátáját adja vissza.
- **+hasObject(String type) : boolean**: Visszaadja hogy van e az adott csempén az adott típusú MapObject-ből.
- **+getObject(String type) : MapObject**: Visszaadja az első “type” típusú objektumot a csempén.
- **+countObect(String type) : int**: Megszámolja hogy az adott típusú objektumból hány darab található a csempén.

### 8.1.9 Floor

- **Felelősség**

A labirintus egyik alapeleme. Ezen a csempén a játékos szabadon mozoghat, illetve vehet fel vagy tehet le dobozokat. A padlón szabadon sétálni s átlőni is lehetséges.

- **Ősosztályok**

Tile

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

Nem értelmezhető.

- **Metódusok**

- **+Floor(int x, int y)**: Az osztály konstruktora ahol megadjuk a csempe pozícióját.
- **+onStep(Entity e, int directionfrom) : void**: Ez a metódus hívódik meg, ha egy entitás (Entity) rálépne az adott csempére. Ha a csempén nincs mozgást akadályozó elem (pl doboz) akkor meghívja a rálépő entitás hitFloor metódusát.
- **+putBox(Player p, Box box) : void**: ez a metódus hívódik meg akkor, ha egy játékos dobozt kíván elhelyezni a csempén. Ha a csempén nincs doboz lerakását akadályozó elem (ajtó, ZPM) akkor meghívja az adott játékos osztály

dropBox metódusát valamint ha van mérleg a mezőn, elegendő súly esetén lenyomja azt.

- **+draw(Graphics2D g) : void**: Kirajzolja a padló játékos által is látott textúráját.
- **+killZPM(): void**: Megsemmisíti a mezőn található ZPM-et.
- **+checkWeight() : void**: Ellenőrzi hogy elég súly van-e a mezőn ahhoz, hogy a hozzá tartozó mérleget lenyomja.

### 8.1.10 Gap

- **Felelősség**

Egy olyan speciális csempe, amire ha egy játékos rálép, meghal. Ha replikátor lép rá akkor feltöltődik és padló (Floor) lesz belőle. Ha egy tárgyat helyeznek bele a tárgy megsemmisül.

- **Ősosztályok**

Tile

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

Nem értelmezhető

- **Metódusok**

- **+Gap(int x, int y)**: Az osztály konstruktora ahol megadjuk a csempe pozícióját.
- **+onStep(Entity e, int directionfrom): void**: Ez a metódus hívódik meg, ha egy entitás megpróbál rálépni erre a mezőre, meghívja az adott entitás hitGap metódusát.
- **+putBox(Player p, Box box): void**: Ez a metódus hívódik meg abban az esetben, ha egy játékos egy dobozt kíván elhelyezni a csempén. A doboz eltűnik a játékos kezéből, de itt nem tárolódik el tehát megsemmisül.
- **+draw(Graphics2D g) : void**: Kirajzolja a padló játékos által is látott textúráját.

### 8.1.11 Wall

- **Felelősség**

Speciális csempe, feladata hogy elválasztó elemként működjön a játékban, korlátozva a játékost a mozgásban valamint bizonyos falakon lehet csillagkaput is nyitni. A falon sem átsétálni se átlőni nem lehet.

- **Ősosztályok**

Tile

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**
  - **+isSpecial: boolean:** Ez az attribútum jelöli, ha egy fal speciális, azaz lehet rajta csillagkaput nyitni.
- **Metódusok**
  - **Wall(int x, int y, boolean isSpecial):** Az osztály konstruktora ahol megadjuk a csempe pozícióját, valamint hogy speciális-e.
  - **+onStep(Entity e, int directionFrom):void:** Ez a metódus hívódik meg abban az esetben, ha egy entitás rálépne az adott mezőre. Ha a fal speciális, akkor az entitás hitSpecialWall metódusa hívódik meg, elenkező esetben a hitWall metódus.
  - **+putBox(Player p, Box box) :void:** ez a metódus hívódik meg ha valaki falra szeretne dobozt helyezni. Mivel falra nem lehet dobozt helyezni így ez a függvény jelenleg nem csinál semmit.
  - **+draw(Graphics2D g) : void:** Kirajzolja a padló játékos által is látott textúráját.

### 8.1.12 MapObject

- **Felelősség**

Ezen osztály alá a játékosal interakcióba lépésre képes elemek osztályai tartoznak
- **Össztályok**

Nem értelmezhető.
- **Interfészek**

Nem értelmezhető.
- **Attribútumok**
  - **+textureID : int : textureID:** Az adott textúra-elem sorszámát tartalmazza.
- **Metódusok**
  - **+draw(Graphics2D g): void : texture:** Ez a metódus rajzolja ki a játékos által látott textúrát az adott elemhez.

### 8.1.13 Box

- **Felelősség**

A játékos által fel és letehető doboz objektumot leíró osztály. A dobozt a játékos felveheti, majd azt egy másik alkalmas csempére ráteheti. A dobozon átsétálni és átlőni sem lehet.
- **Össztályok**

MapObject
- **Interfészek**

Nem értelmezhető.

- **Attribútumok**
  - **+textureID : int : textureID**: Az adott textúra-elem sorszámát tartalmazza.
- **Metódusok**

Nem értelmezhető.

#### 8.1.14 ZPM

- **Felelősség**

A játék célját jelentő Zero Point Module. A játékos felszedheti majd amint már egy ZPM sem maradt a pályán a játékos teljesítette a feladatát.
- **Ősosztályok**

MapObject
- **Interfészek**

Nem értelmezhető.
- **Attribútumok**
  - **+textureID : int : textureID**: Az adott textúra-elem sorszámát tartalmazza.
- **Metódusok**

Nem értelmezhető.

#### 8.1.15 Door

- **Felelősség**

Az ajtót leró osztály, feladata, hogy nyilvántartja, hogy az adott ajtó nyitva van-e, és ez által átjárható-e.
- **Ősosztályok**

Nem értelmezhető.
- **Interfészek**

Nem értelmezhető.
- **Attribútumok**
  - **- open : boolean**: Nyilvántartja, hogy az ajtó éppen nyitott vagy csukott állapotban van.
  - **-textureIDOpen: int**: A nyitott ajtóhoz tartozó textúra.
  - **-textureIDClosed: int**: A zárt ajtóhoz tartozó textúra.
- **Metódusok**
  - **+setOpen(boolean b) : void**: Beállítja az ajtó nyitva tartását.
  - **+getOpen(): boolean**: Visszaadja, hogy az ajtó nyitva vagy zárva van.
  - **+draw(Graphics2D g): void**: Kirajzolja az ajtóhoz tartozó képet

### 8.1.16 Scale

- **Felelősség**  
A mérleget leíró osztály, feladata hogy kinyissa a hozzá tartozó ajtót, ha súlyt tesznek rá.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-pressed: boolean:** Mérleg állapotát nyilvántartó változó.
  - **-textureIDUP: int:** A nem lenyomott mérleghez tartozó textúra.
  - **-textureIDDown: int:** A lenyomott mérleghez tartozó textúra.
  - **-door: Door:** Az adott mérleghez tartozó ajtó referenciája.
  - **-neededWeight: int:** a mérleg lenyomásához szükséges súly.
- **Metódusok**
  - **+setPressed(boolean b):** Beállítja, hogy a mérleg éppen lenyomott vagy felengedett állapotban van.
  - **+draw(Graphics2D g): void:** Kirajzolja a mérleghez tartozó képet.

### 8.1.17 Wormhole

- **Felelősség**  
A pályán található csillagkapuk és féregjáratok működéséért felelős osztály. Ő tárolja a kapuk helyét és ügyel arra, hogy egy fajta kapuból csak 1 darab létezzen.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-stargateBlue : Stargate:** A kék csillagkapu referenciája.
  - **-stargateYellow : Stargate:** A sárga csillagkapu referenciája.
  - **-stargateRed : Stargate:** A piros csillagkapu referenciája.
  - **-stargateGreen : Stargate:** A zöld csillagkapu referenciája.
  - **-openBlueYellow: boolean:** A kék-sárga féregjárat állapotát tárolja.
  - **-openRedGreen: boolean:** A piros-zöld féregjárat állapotát tárolja.
- **Metódusok**
  - **+draw(Graphics2D g): void:** Kirajzolja a nyitott csillagkapukat.
  - **+setStargate(String color, Tile t, int dir):void:** az adott csempe adott oldalára nyit egy megadott színű csillagkaput.

- **+getStargate(String color): Stargate:** Visszaadja az adott színű csillagkaput.
- **+getOtherStargate(Tile t): Stargate:** A megadott csempén lévő csillagkapu párját adja vissza.

### 8.1.18 Stargate

- **Felelősség**  
A csillagkapuk adatainak tárolását végző osztály, tárolja a kapuk helyzetet valamint irányát.
- **Össztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-actualTile : Tile:** A csillagkapu adott helyzetét jelző változó-
  - **-textureID:** A csillagkapuhoz tartozó textúra.
  - **-direction : int:** A csillagkapu irányát tároló változó.
- **Metódusok**
  - **Stargate(Tile t, int dir, int textureID):** Az osztály konstruktora, beállítja a helyzetet jelző csempét, az irányt és a hozzá tartozó textúrát.
  - **+draw(Graphics2D g): void:** Kirajzolja a csillagkaput.

### 8.1.19 Entity

- **Felelősség**  
A játékos által „irányított” és animált objektumok őssztálya.
- **Össztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-actualTile : Tile:** Az aktuális pozíciót jelző változó.
  - **-textureID: int:** Az entitáshoz tartozó grafika sorszámát tároló változó.
  - **-tileMap: TileMap:** Az aktuális labirintust tároló TileMap referenciája
  - **-direction: int:** az entitás irányát tároló változó.
  - **+absx: int:** Az entitás abszolút koordinátáját tároló változó, amit a folyamatos illetve animált mozgáshoz fogunk felhasználni.
  - **+absy:int:** Az entitás abszolút koordinátáját tároló változó, amit a folyamatos illetve animált mozgáshoz fogunk felhasználni.
  - **+speed: int:** Az entitás mozgásának sebességét tároló változó, amit a folyamatos mozgáshoz és animáláshoz fogunk felhasználni.



- **Metódusok**

- **+Move(int dir): void:** Az entitást az adott irányba mozgó metódus. Az adott irányban következő Tile onStep metódusát hívja meg.
- **+draw(Graphics2D g): void:** Kirajzolja az entitást.
- **+destroy(): void:** Megsemmisíti az adott entitást.
- **+moveTo(Tile t): void:** Az entitást áthelyezi a paraméterként kapott csempére.
- **+hitWall(Tile t): void:** Akkor hívódik meg ez a metódus, ha az entitás falnak ütközik.
- **+hitgap(Tile t): void:** Akkor hívódik meg ez a metódus, ha az entitás szakadékra lép.
- **+hitFloor(Tile t): void:** Akkor hívódik meg ez a metódus, ha az entitás padlóra lép.
- **+hitSpecialWall(Tile t): void:** Akkor hívódik meg ez a metódus, ha az entitás speciális falnak ütközik.

### 8.1.20 Replicator

- **Felelősség**

A replikátort leíró osztály. Feladata, hogy véletlenszerűen mozgatja önmagát, és megfelelően lép interakcióba a különböző pályaelemekkel.

- **Ősosztályok**

Entity

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

Nem értelmezhető.

- **Metódusok**

- **randomStep():** Bizonyos időközönként léptet egyet magán egy véletlenszerű irányba.
- **+hitgap(Tile t): void:** Akkor hívódik, ha replikátor szakadékba esik, ekkor a replikátor megsemmisíti önmagát és a szakadékot(Gap) padlóvá(Floor) változtatja.
- **+hitFloor(Tile t):void:** Ha nincs a mozgást akadályozó elem a mezőn, akkor oda lépteti a replikátort.
- **+hitSpecialWall(Tile t):void:** Megvizsgálja, hogy a csempén van e csillagkapu, ha van, és nyitva van, a főregjárt átlép rajta, ha nincs, akkor marad az adott csempén.

### 8.1.21 Projectile

- **Felelősség**

A játékos által kilőtt lövedéket leíró osztály, feladata hogy biztosítsa a lövedék specifikációban leírt működését pl. a kapunyitást.

- **Ősosztályok**

Entity

- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-player : Player**: A lövedék gazdájára mutató referencia.
  - **-color: String**: A lövedék színét tároló változó.
- **Metódusok**
  - **+Projectile(String color, Player p)**: Az osztály konstruktora, ahol megadjuk a lövedék színét és a kilövő játékos referenciáját.
  - **+hitGap(Tile t):void**: Akkor hívódik meg, ha a lövedék szakadékra lépne, a lövedék átrepül a szakadékon így ez a függvény a szakadékra léptet.
  - **+hitFloor(Tile t): void**: Akkor hívódik meg, ha a lövedék padlóra lépne, ha nincsen a mezőn a lövedék mozgását akadályozó elem (doboz, ZPM zárt ajtó) akkor rálépteti a lövedéket, ellenkező esetben megsemmisíti azt.
  - **hitSpecialWall(Tile t): void**: Akkor hívódik, ha a lövedék speciális falra lépne. A fal megfelelő oldalán egy a lövedék színével megegyező színű csillagkapu nyílik majd a lövedék megsemmisül.
  - **destroy(): void**: Megsemmisíti a lövedéket.

### 8.1.22 Player

- **Felelősség**  
A 2 játékos osztály ősosztálya, ez az osztály felel a legtöbb játékos karakter (O'Neill, Jaffa) interakcióért.
- **Ősosztályok**  
Entity
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-ZPM: int**: Az adott játékos által összegyűjtött ZPM-ek száma.
  - **+hasBox: Box**: A játékos kezében lévő doboz referenciája, ha van ilyen.
  - **+isdead:boolean**: Jelzi, hogy az adott játékos halott-e.
  - **+projectilePrimary: Projectile**: Az elsődleges lövedéket tároló változó.
  - **+projectileSecondary: Projectile**: A másodlagos lövedéket tároló változó.
- **Metódusok**
  - **+hitgap(Tile t): void**: Akkor hívódik meg, ha a játékos szakadékba lépne, ezzel megölve a játékost, beállítva az isdead változót true értékre.
  - **+hitFloor:void**: Akkor hívódik meg, ha a játékos padlóra szeretne lépni, ha a padlón nincs doboz vagy zárt ajtó, akkor rálépteti a karaktert, ha van ZPM a mezőn azt felveszi, illetve ha van mérleg akkor elegendő súly esetén lenyomja azt. Minden második felvett ZPM után meghívja a tileMap spawnZPM() metódusát.

- **+hitSpecialWall(Tile t): void:** Akkor hívódik meg, ha a játékos speciális falra szeretne lépni. Ha a falon van csillagkapu, akkor a játékost átlépteti a fal túlsó oldalára, ellenkező esetben nem történik semmi.
- **+pickBox():void:** Akkor hívódik, ha a játékos a vele szemben lévő mezőről megpróbál felvenni egy dobozt. Ha nincs nála doboz és a mezőn van doboz, akkor felveszi és beleteszi a játékos hasBox változójába a mezőről pedig kitörli a referenciát. Ellenkező esetben nem történik semmi.
- **+dropBox():void:** Elveszi a játékos dobozát. A hasBox változót null-ra állítja.
- **+shootProjectile(String type):void:** Akkor hívódik, ha a játékos lő egyet. Létrehoz egy megfelelő színű lövedéket a játékos lábánál majd útjára engedi azt.

### 8.1.23 O'Neill

- **Felelősség**  
Az O'Neill-t irányító játékos avatárja, ő az elsődleges játékos. Ő kezeli a sárga és a kék lövedékeket.
- **Ősosztályok**  
Entity -> Player
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető
- **Metódusok**
  - **ONeill(Tile t):** Az osztály konstruktora, létrehoz egy új példányt a paraméterként megadott csempére.
  - **+shootProjectile(type String):void:** Létrehoz egy megfelelő színű lövedéket O'Neill lábánál. Elsődleges tűz esetén kéket másodlagos esetén sárgát.

### 8.1.24 Jaffa

- **Felelősség**  
A Jaffa-t irányító játékos avatárja, ő a másodlagos játékos. Ő kezeli a piros és zöld lövedékeket.
- **Ősosztályok**  
Entity -> Player
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.

- **Metódusok**

- **ONeill(Tile t):** Az osztály konstruktora, létrehoz egy új példányt a paraméterként megadott csempére.
- **+shootProjectile(type String):void:** Létrehoz egy megfelelő színű lövedéket a Jaffa lábánál. Elsődleges tűz esetén pirosat másodlagos esetén zöldet.

## 8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

### 8.2.1 Teszteset1: A játékos padló csempére lép

- **Leírás**

A játékos egy padló csempére lép amin nincs semmi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos mozgását ellenőrzi.

Ha nem változik a játékos helyzete, hibás a program.

- **Bemenet**

```
Generate 2 1
AddFloorTile 0 0
AddFloorTile 1 0
AddONeill 0 0 1
Move oneill 1
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzáadva. X: 0 Y: 0
Floor csempe hozzáadva. X: 1 Y: 0
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban
ONeill entitas megprobal ralepni Floor csempere
ONeill entitas sikeresen ralepett a Floor csempere X: 1
Y: 0 koordinatan
```

### 8.2.2 Teszteset2: A játékos szakadékba lép

- **Leírás**

Szakadékba lépés hatására a játékos meghal.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos halálát ellenőrzi.

Ha nem hal meg a játékos, hibás a program.

- **Bemenet**

```
Generate 2 1
AddFloorTile 0 0
AddGapTile 1 0
AddONeill 0 0 1
Move oneill 1
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzáadva. X: 0 Y: 0
Gap csempe hozzáadva. X: 1 Y: 0
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban
ONeill entitas megprobal ralepni Gap csempere
ONeill entitas szakadekba lep és meghal
```

### 8.2.3 Teszteset3: A játékos falba próbál lépni

- **Leírás**  
A játékos megpróbál falba lépni, de ez nem sikerül.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A fal nem átjárható, ezért, a játékos az eredeti helyén marad.  
A játékos pozíciójának megváltozása esetén, hibás a program.
- **Bemenet**  
Generate 2 1  
AddFloorTile 0 0  
AddWallTile 1 0 0  
AddONeill 0 0 1  
Move oneill 1
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Wall csempe hozzáadva. X: 1 Y: 0  
Specialis: false  
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitás megpróbál rálépni Wall csempére  
ONeill entitas doboz/fal/zart ajtó miatt nem tud lépni

### 8.2.4 Teszteset4: A játékos mérlegre lép

- **Leírás**  
A játékos mérlegre lépéssel kinyit egy ajtót.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A helyes ajtónyitást ellenőrzi. A játékosnak önmagában elegendő a súlya, a mérleg lenyomásához.  
Ha nem nyomódik le a mérleg, vagy nem nyílik ki az ajtó, akkor hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddFloorTile 2 0  
AddScaleDoor 1 0 2 0  
AddONeill 0 0 1  
Move oneill 1
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
Floor csempe hozzáadva. X: 2 Y: 0  
Scale hozzáadva. X: 1 Y: 0  
Door hozzáadva. X: 2 Y: 0  
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas megprobal ralepni Floor csempere  
ONeill entitas sikeresen ralepett a Floor csempere X: 1 Y: 0 koordinatan  
ONeill entitas lenyomta a merleget a X: 1 Y: 0 helyen  
Az ajto kinyilt

### 8.2.5 Teszteset5: A játékos zárt ajtónak nekimegy

- **Leírás**  
A játékos megpróbál zárt ajtóba lépni, de ez nem sikerül.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A zárt ajtó nem átjárható, ezért, a játékos az eredeti helyén marad.  
A játékos pozíciójának megváltozása esetén, hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddFloorTile 2 0  
AddScaleDoor 2 0 1 0  
AddONeill 0 0 1  
Move oneill 1
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
Floor csempe hozzáadva. X: 2 Y: 0  
Scale hozzáadva. X: 2 Y: 0  
Door hozzáadva. X: 1 Y: 0  
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas megprobal ralepni Floor csempere  
ONeill entitas doboz/fal/zart ajtó miatt nem tud lépni

### 8.2.6 Teszteset6: A játékos átmegy egy nyitott ajtón

- **Leírás**  
Egy Jaffa játékos rááll egy mérlegre, aminek hatására kinyílik a hozzá tartozó ajtó.  
O'Neill játékos átmegy a nyitott ajtón.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A nyitott ajtó átjárható, ezért, O'Neill át tud mozogni rajta.  
Ha nem változik O'Neill helyzete, hibás a program.
- **Bemenet**  
Generate 5 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddFloorTile 2 0  
AddFloorTile 3 0  
AddFloorTile 4 0  
AddScaleDoor 3 0 1 0  
AddONeill 0 0 1  
AddJaffa 4 0 3  
Move jaffa 3  
Move oneill 1  
Move oneill 1
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0

```

Floor csempe hozzaadva. X: 2 Y: 0
Floor csempe hozzaadva. X: 3 Y: 0
Floor csempe hozzaadva. X: 4 Y: 0
Scale hozzaadva. X: 3 Y: 0
Door hozzaadva. X: 1 Y: 0
ONeill entitas hozzáadva az X:0 Y:0 helyre 1 irányban
Jaffa entitas hozzáadva az X:4 Y:0 helyre 3 irányban
Jaffa entitas megprobal ralepni Floor csempere
Jaffa entitas sikeresen ralepett a Floor csempere X: 3 Y:
0 koordinatan
Jaffa entitas lenyomta a merleget a X: 3 Y: 0 helyen
Az ajto kinyilt
ONeill entitas megprobal ralepni Floor csempere
ONeill entitas sikeresen ralepett a Floor csempere X: 1
Y: 0 koordinatan
ONeill entitas megprobal ralepni Floor csempere
ONeill entitas sikeresen ralepett a Floor csempere X: 2
Y: 0 koordinatan

```

### 8.2.7 Teszteset7: A játékos dobozra próbál lépni

- **Leírás**  
A játékos megpróbál dobozra lépni, de ez nem sikerül.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A doboz, nem átjárható, ezért, a játékos az eredeti helyén marad.  
A játékos pozíciójának megváltozása esetén, hibás a program.
- **Bemenet**  
Generate 2 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddBox 1 0  
AddONeill 0 0 1  
Move oneill 1
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzaadva. X: 0 Y: 0  
Floor csempe hozzaadva. X: 1 Y: 0  
Box hozzaadva. X: 1 Y: 0  
ONeill entitas hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas megprobal ralepni Floor csempere  
ONeill entitas doboz/fal/zart ajtó miatt nem tud lépni

### 8.2.8 Teszteset8: A játékos ZPM-et vesz fel.

- **Leírás**  
A játékos egy olyan padló csempére lép, amin ZPM van. A helyzete megváltozik a ZPM eltűnik és növekszik a megszerzett ZPM-ek száma.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A ZPM-et tartalmazó csempére rá lehet mozogni.  
A ZPM, eltűnik közben.  
A játékos ZPM számlálója növekszik.  
Ha a fenti funkciók nem teljesülnek, hibás a program.

- **Bemenet**  
Generate 2 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddZPM 1 0  
AddONeill 0 0 1  
Move oneill 1
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
ZPM hozzáadva. X: 1 Y: 0  
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas megprobal ralepni Floor csempere  
ONeill entitas sikeresen ralepett a Floor csempere X: 1  
Y: 0 koordinatan  
ONeill entitas ZPM-re lepett a Floor csempen és felvette  
azt  
ZPMek szama: 1

### 8.2.9 Teszteset9: A játékos teleportál

- **Leírás**  
Egy speciális fal két átellenes oldalán létezik egy csillagkapu, amik előtt egy padló csempe található.  
A játékos az egyik olyan oldalon áll, ahol van kapu és felé mozdul, aminek hatására átteleportál (mozdul) a másik kapu kijáratánál lévő csempére.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A csillagkapuk átjárhatóságát ellenőrzi.  
Ha nem mozdul, vagy nem a megfelelő mezőre mozdul a játékos, hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddFloorTile 2 0  
AddWallTile 1 0 true  
AddGate blue 1 0 4  
AddGate orange 1 0 1  
AddONeill 0 0 1  
Move oneill 1  
ListEntities
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
Floor csempe hozzáadva. X: 2 Y: 0  
Wall csempe hozzáadva. X: 1 Y: 0  
Specialis: true  
blue színű csillagkapu jött létre, X: 1 Y: 0 helyen, 4 irányban.



```

orange szinu csillagkapu jott letre, X: 1 Y: 0 helyen, 1
iranyban.
ONeill entitas hozzaadva az X:0 Y:0 helyre 1 irányban
ONeill entitas megpróbál rálépni Wall csempére
ONeill entitas megprobal ralepni Floor csempere
ONeill entitas sikeresen ralepett a Floor csempere X: 2
Y: 0 koordinatan
ONeill X: 2 Y: 0

```

### 8.2.10 Teszteset10: A játékos dobozt vesz fel

- **Leírás**  
A játékos felveszi az előtte lévő dobozt.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A játékos doboz felvétel akciójára eltűnik előle és a játékoshoz kerül.  
Ha a doboz nem tűnik el, hibás a program.
- **Bemenet**  
Generate 2 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddBox 1 0  
AddONeill 0 0 1  
PickBox oneill
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzaadva. X: 0 Y: 0  
Floor csempe hozzaadva. X: 1 Y: 0  
Box hozzaadva. X: 1 Y: 0  
ONeill entitas hozzaadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas megprobal dobozt felvenni az X: 1 Y: 0  
csemperol  
ONeill entitas felvett egy dobozt a X:1 Y:0 helyrol

### 8.2.11 Teszteset11: A játékos dobozt tesz le

- **Leírás**  
A játékos felveszi az előtte lévő dobozt, egyet előre megy és leteszi azt.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A doboz letetését ellenőrizzük vele, a felvétel csak szükséges ennek végrehajtásához.  
Ha a dobozt nem sikerül letenni a cél területre, hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddFloorTile 2 0  
AddBox 1 0  
AddONeill 0 0 1  
PickBox oneill  
Move oneill 1  
DropBox oneill
- **Elvárt kimenet**

```

Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
Floor csempe hozzaadva. X: 2 Y: 0
Box hozzaadva. X: 1 Y: 0
ONeill entitas hozzaadva az X:0 Y:0 helyre 1 irányban
ONeill entitas megprobal dobozt felvenni az X: 1 Y: 0
csemperol
ONeill entitas felvett egy dobozt a X:1 Y:0 helyrol
ONeill entitas megprobal ralepni Floor csempere
ONeill entitas sikeresen ralepett a Floor csempere X: 1
Y: 0 koordinatan
ONeill entitas megprobal dobozt letenni az X: 2 Y: 0
csempere
ONeill entitas sikeresen letette a dobozt a Floor
csempere az X:2 Y:0 helyre

```

### 8.2.12 Teszteset12: A játékos szakadékba lép

- **Leírás**  
A játékos felvesz egy dobozt, ráteszi egy mérlegre, aminek a hatására, az nem nyomódik még le, majd ezt megismétli egy másik dobozzal és ekkor már lenyomódik, aminek a hatására az ajtó is kinyílik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A mérleg súly korlátjának és ajtó nyitásának ellenőrzése.  
Ha egy doboztól lenyomódik a mérleg, vagy kettőtől nem, hibás a program.  
Ha lenyomott mérleg hatására nem nyílik ki az ajtó, hibás a program.
- **Bemenet**  
Generate 3 2  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddFloorTile 2 0  
AddFloorTile 0 1  
AddFloorTile 1 1  
AddFloorTile 2 1  
AddONeill 1 0 1  
AddScaleDoor 0 0 0 1  
AddBox 1 1  
AddBox 0 2  
PickBox oneill  
Move oneill 3  
DropBox oneill  
Move oneill 0  
PickBox oneill  
Move oneill 3  
DropBox oneill
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzaadva. X: 0 Y: 0  
Floor csempe hozzaadva. X: 1 Y: 0  
Floor csempe hozzaadva. X: 2 Y: 0

```

Floor csempe hozzaadva. X: 0 Y: 1
Floor csempe hozzaadva. X: 1 Y: 1
Floor csempe hozzaadva. X: 2 Y: 1
ONeill entitas hozzaadva az X:1 Y:0 helyre 1 irányban
Scale hozzaadva. X: 0 Y: 1Door hozzaadva. X: 0 Y:
1java.lang.NullPointerException
Box hozzaadva. X: 1 Y: 1
Box hozzaadva. X: 2 Y: 0
ONeill entitas megprobal dobozt felvenni az X: 2 Y: 0
csemperol
ONeill entitas felvett egy dobozt a X:2 Y:0 helyrol
ONeill entitas forgott 3 irányba
ONeill entitas megprobal dobozt letenni az X:0 Y: 0
csempere
ONeill entitas sikeresen letette a dobozt a Floor
csempere az X:0 Y:0 helyre
ONeill entitas forgott 2 irányba
ONeill entitas megprobal dobozt felvenni az X: 1 Y: 1
csemperol
ONeill entitas felvett egy dobozt a X:1 Y:1 helyrol
ONeill entitas forgott 3 irányba
ONeill entitas megprobal dobozt letenni az X:0 Y: 0
csempere
ONeill entitas sikeresen letette a dobozt a Floor
csempere az X:0 Y:0 helyre

```

### 8.2.13 Teszteset13: A játékos dobozt tesz egy szakadékba

- **Leírás**  
A játékos dobozt tesz egy szakadékba, aminek a hatására, a doboz megsemmisül.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ha a doboz nem semmisül meg, hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddGapTile 2 0 0  
AddBox 0 0  
AddONeill 1 0 3  
PickBox oneill  
Move oneill 1  
DropBox oneill
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzaadva. X: 0 Y: 0  
Floor csempe hozzaadva. X: 1 Y: 0  
Gap csempe hozzaadva. X: 2 Y: 0  
Box hozzaadva. X: 0 Y: 0  
ONeill entitas hozzaadva az X:1 Y:0 helyre 3 irányban  
ONeill entitas megprobal dobozt felvenni az X: 0 Y: 0  
csemperol

```

ONEill entitas felvett egy dobozt a X:0 Y:0 helyrol
ONEill entitas forgott 1 irányba
ONEill entitas megprobal dobozt letenni az X: 2 Y: 0
csempere
ONEill entitas dobozt tett a Gap csempere az X:2 Y:0
helyen es a doboz megsemmisül

```

#### 8.2.14 Teszteset14: A játékos dobozt próbál tenni fal csempére

- **Leírás**  
A játékos dobozt próbál tenni fal csempére, de ezt nem lehet, így nála marad a doboz.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ha a dobozt le tudja tenni, hibás a program.
- **Bemenet**  

```

Generate 3 1
AddFloorTile 0 0
AddFloorTile 1 0
AddWallTile 2 0 0
AddBox 0 0
AddONEill 1 0 3
PickBox oneill
Move oneill 1
DropBox oneill

```
- **Elvárt kimenet**  

```

Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
Wall csempe hozzaadva. X: 2 Y: 0
Specialis: false
Box hozzaadva. X: 0 Y: 0
ONEill entitas hozzaadva az X:1 Y:0 helyre 3 irányban
ONEill entitas megprobal dobozt felvenni az X: 0 Y: 0
csemperol
ONEill entitas felvett egy dobozt a X:0 Y:0 helyrol
ONEill entitas forgott 1 irányba
ONEill entitas megprobal dobozt letenni az X: 2 Y: 0
csempere
ONEill entitas nem tud dobozt letenni a Wall csempre X:2
Y:0 helyen

```

#### 8.2.15 Teszteset15: A játékos dobozt próbál tenni ajtóra

- **Leírás**  
Előkészítés:
  - Az ajtó nyitást egy Jaffa játékos biztosítja, azzal, hogy rálép az egyik ajtóhoz tartozó mérlegre
  - A dobozt az ONEill játékos felveszi maga elől, majd az első ajtó felé fordul.
 A játékos dobozt próbál tenni, zárt majd nyitott ajtóra, amit nem lehet végrehajtani, a doboz mindkét esetben nála marad.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ha a dobozt le tudja tenni, hibás a program.

- **Bemenet**

```
Generate 5 1
AddFloorTile 0 0
AddFloorTile 1 0
AddFloorTile 2 0
AddFloorTile 3 0
AddFloorTile 4 0
AddScaleDoor 3 0 2 0
AddBox 0 0
AddONeill 1 0 3
AddJaffa 4 0 3
PickBox oneill
Move oneill 1
DropBox oneill
Move jaffa 3
DropBox oneill
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
Floor csempe hozzaadva. X: 2 Y: 0
Floor csempe hozzaadva. X: 3 Y: 0
Floor csempe hozzaadva. X: 4 Y: 0
Scale hozzaadva. X: 3 Y: 0Door hozzaadva. X: 2 Y: 0Box
hozzaadva. X: 0 Y: 0
ONeill entitas hozzáadva az X:1 Y:0 helyre 3 irányban
Jaffa entitas hozzáadva az X:4 Y:0 helyre 3 irányban
ONeill entitas megprobal dobozt felvenni az X: 0 Y: 0
csemperol
ONeill entitas felvett egy dobozt a X:0 Y:0 helyrol
ONeill entitas forgott 1 irányba
ONeill entitas megprobal dobozt letenni az X: 2 Y: 0
csempere
ONeill entitas nem tud dobozt letenni X:2 Y:0 helyre ZPM
vagy ajto miatt
Jaffa entitas megprobal ralepni Floor csempere
Jaffa entitas sikeresen ralepett a Floor csempere X: 3 Y:
0 koordinatan
Jaffa entitas lenyomta a merleget a X: 3 Y: 0 helyen
Az ajto kinyilt
ONeill entitas megprobal dobozt letenni az X: 2 Y: 0
csempere
ONeill entitas nem tud dobozt letenni X:2 Y:0 helyre ZPM
vagy ajto miatt
```

## 8.2.16 Teszteset16: A játékos dobozt próbál letenni, míg nincs nála

- **Leírás**

A játékos egy padló csempére megpróbál dobozt letenni, úgy, hogy nincs nála.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ha létre jön így is a doboz, hibás a program.

- **Bemenet**

```
Generate 2 1
AddFloorTile 0 0
AddFloorTile 1 0
AddONeill 0 0 1
DropBox oneill
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
ONeill entitas hozzaadva az X:0 Y:0 helyre 1 irányban
ONeill entitas nem tud dobozt letenni mert nincs nala
doboz
```

### 8.2.17 Teszteset17: A játékos dobozt tesz le egy ZPM-re

- **Leírás**

Előkészítés: a dobozt az O'Neill játékos felveszi maga elől, majd az első ZPM felé fordul.

A játékos dobozt próbál letenni egy előtte lévő ZPM-re, ami nem sikerül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ha le tudja tenni a dobozt, hibás a program.

- **Bemenet**

```
Generate 3 1
AddFloorTile 0 0
AddFloorTile 1 0
AddFloorTile 2 0
AddBox 0 0
AddZPM 2 0
AddONeill 1 0 3
PickBox oneill
Move oneill 1
DropBox oneill
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
Floor csempe hozzaadva. X: 2 Y: 0
Box hozzaadva. X: 0 Y: 0
ZPM hozzaadva. X: 2 Y: 0
ONeill entitas hozzaadva az X:1 Y:0 helyre 3 irányban
ONeill entitas megprobal dobozt felvenni az X: 0 Y: 0
csemperol
ONeill entitas felvett egy dobozt a X:0 Y:0 helyrol
ONeill entitas forgott 1 irányba
ONeill entitas megprobal dobozt letenni az X: 2 Y: 0
csempere
ONeill entitas nem tud dobozt letenni X:2 Y:0 helyre ZPM
vagy ajto miatt
```

**8.2.18 Teszteset18: A játékos lövedéket lő ki**

- **Leírás**  
A játékos lövedéket lő ki, aminek a hatására létrejön egy lövedék.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ha nem jön létre a lövedék, hibás a program.
- **Bemenet**  
Generate 2 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddONeill 0 0 1  
Shoot primary oneill
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas kilott egy primary tipusu lovedeket

**8.2.19 Teszteset19: A lövedék egy szakadék fölött mozog**

- **Leírás**  
Előkészítés: a játékos kilő egy lövedéket, egy szakadék felé.  
A lövedék átmozog a szakadék fölött.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A lövedéknek át kel tudnia haladni a szakadék fölött, ha nem, hibás a program.
- **Bemenet**  
Generate 4 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddGapTile 2 0  
AddFloorTile 3 0  
AddONeill 0 0 1  
Shoot primary oneill  
MoveProjectile oneill  
MoveProjectile oneill
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
Gap csempe hozzáadva. X: 2 Y: 0  
Floor csempe hozzáadva. X: 3 Y: 0  
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas kilott egy primary tipusu lovedeket  
Projectile entitas megprobal ralepni Floor csempere  
Projectile entitas ralepett Floor csempere az X: 1 Y: 0 helyen  
Projectile entitas megprobal ralepni Gap csempere  
Projectile entitas Gap csempere lepett az X: 2 Y: 0 helyen

**8.2.20 Teszteset20: A lövedék falnak ütközik**

- **Leírás**  
Előkészítés: a játékos kilő egy lövedéket, egy fal csempe felé.  
A lövedék megsemmisül.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ha nem semmisül meg, vagy csillagkaput hoz létre, hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddWallTile 2 0 false  
AddONeill 0 0 1  
Shoot primary oneill  
MoveProjectile oneill
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
Wall csempe hozzáadva. X: 2 Y: 0  
Specialis: false  
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas kilott egy primary tipusu lovedeket  
Projectile entitas megprobal ralepni Floor csempere  
Projectile entitas ralepett Floor csempere az X: 1 Y: 0 helyen

**8.2.21 Teszteset21: A lövedék doboznak ütközik**

- **Leírás**  
Előkészítés: a játékos kilő egy lövedéket, egy fal csempe felé.  
A lövedék megsemmisül.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ha nem semmisül meg, hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddFloorTile 2 0  
AddBox 2 0  
AddONeill 0 0 1  
Shoot primary oneill  
MoveProjectile oneill
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
Floor csempe hozzáadva. X: 2 Y: 0  
Box hozzáadva. X: 2 Y: 0  
ONeill entitás hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas kilott egy primary tipusu lovedeket



```

Projectile entitas megprobal ralepni Floor csempere
Projectile entitas ralepett Floor csempere az X: 1 Y: 0
helyen
Projectile entitas megprobal ralepni Floor csempere
Projectile entitas nekiutkozott valaminek az X: 2 Y: 0
helyen
A lovedek megsemmisult

```

### 8.2.22 Teszteset22: A lövedék speciális falba ütközik

- **Leírás**  
Előkészítés: a játékos kilő egy lövedéket, egy speciális fal csempe felé.  
A lövedék speciális falba ütközik és csillagkaput hoz létre.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ha nem jön létre a csillagkapu, hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddWallTile 2 0 false  
AddONeill 0 0 1  
Shoot primary oneill  
MoveProjectile oneill
- **Elvárt kimenet**  
Sikeres Generalas  
Floor csempe hozzáadva. X: 0 Y: 0  
Floor csempe hozzáadva. X: 1 Y: 0  
Wall csempe hozzáadva. X: 2 Y: 0  
Specialis: false  
ONeill entitas hozzáadva az X:0 Y:0 helyre 1 irányban  
ONeill entitas kilott egy primary tipusu lovedeket  
Projectile entitas megprobal ralepni Floor csempere  
Projectile entitas ralepett Floor csempere az X: 1 Y: 0  
helyen

### 8.2.23 Teszteset23: A lövedék replikátorba ütközik

- **Leírás**  
Előkészítés: a játékos kilő egy lövedéket, egy replikátor felé.  
A replikátor és a lövedék megsemmisül.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ha nem semmisülnek meg, hibás a program.
- **Bemenet**  
Generate 3 1  
AddFloorTile 0 0  
AddFloorTile 1 0  
AddFloorTile 2 0  
AddReplicator 2 0 4  
AddONeill 0 0 1  
Shoot primary oneill  
MoveProjectile oneill

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
Floor csempe hozzaadva. X: 2 Y: 0
Replicator entitas hozzaadva. X: 2 Y: 0
ONEill entitas hozzaadva az X:0 Y:0 helyre 1 irányban
ONEill entitas kilott egy primary tipusu lovedeket
Projectile entitas megprobal ralepni Floor csempere
Projectile entitas ralepett Floor csempere az X: 1 Y: 0
helyen
Projectile entitas megprobal ralepni Floor csempere
Projectile entitas ralepett Floor csempere az X: 2 Y: 0
helyen
Projectile entitas replikatornak utkozott es
megsemmisitette azt a X: 2 Y: 0 helyen
```

### 8.2.24 Teszteset24: A lövedék zárt ajtóba ütközik

- **Leírás**

Előkészítés: a játékos kilő egy lövedéket, egy zárt ajtó felé.  
A lövedék megsemmisül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ha a lövedék nem semmisül meg, hibás a program.

- **Bemenet**

```
Generate 4 1
AddFloorTile 0 0
AddFloorTile 1 0
AddFloorTile 2 0
AddFloorTile 3 0
AddScaleDoor 3 0 2 0
AddONEill 0 0 1
Shoot primary oneill
MoveProjectile oneill
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
Floor csempe hozzaadva. X: 2 Y: 0
Floor csempe hozzaadva. X: 3 Y: 0
Scale hozzaadva. X: 3 Y: 0
Door hozzaadva. X: 2 Y: 0
ONEill entitas hozzaadva az X:0 Y:0 helyre 1 irányban
ONEill entitas kilott egy primary tipusu lovedeket
Projectile entitas megprobal ralepni Floor csempere
Projectile entitas ralepett Floor csempere az X: 1 Y: 0
helyen
```

### 8.2.25 Teszteset25: A lövedék nyitott ajtón át halad

- **Leírás**

Előkészítés:

- Az ajtó nyitást a játékos biztosítja, azzal, hogy rálép az ajtóhoz tartozó mérlegre
- A játékos kilő egy lövedéket, a nyitott ajtó felé.

A lövedék áthalad a nyitott ajtón.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ha a lövedék nem tud áthaladni a nyitott ajtón, hibás a program.

- **Bemenet**

```
Generate 5 1
AddFloorTile 0 0
AddFloorTile 1 0
AddFloorTile 2 0
AddFloorTile 3 0
AddFloorTile 4 0
AddScaleDoor 1 0 3 0
AddONeill 0 0 1
Move oneill 1
Shoot primary oneill
MoveProjectile oneill
MoveProjectile oneill
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
Floor csempe hozzaadva. X: 2 Y: 0
Floor csempe hozzaadva. X: 3 Y: 0
Floor csempe hozzaadva. X: 4 Y: 0
Scale hozzaadva. X: 1 Y: 0Door hozzaadva. X: 3 Y: 0ONeill
entitas hozzaadva az X:0 Y:0 helyre 1 irányban
ONeill entitas megprobal ralepni Floor csempere
ONeill entitas sikeresen ralepett a Floor csempere X: 1
Y: 0 koordinatan
ONeill entitas lenyomta a merleget a X: 1 Y: 0 helyen
Az ajto kinyilt
ONeill entitas kilott egy primary tipusu lovedeket
Projectile entitas megprobal ralepni Floor csempere
Projectile entitas ralepett Floor csempere az X: 2 Y: 0
helyen
Projectile entitas megprobal ralepni Floor csempere
Projectile entitas ralepett Floor csempere az X: 3 Y: 0
helyen
Projectile entitas megprobal ralepni Floor csempere
Projectile entitas ralepett Floor csempere az X: 4 Y: 0
helyen
```

## 8.2.26 Teszteset26: A lövedék ZPM-nek ütközik

- **Leírás**

Előkészítés: a játékos kilő egy lövedéket, egy ZPM felé.

A lövedék megsemmisül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ha a lövedék nem semmisül meg, hibás a program.

- **Bemenet**

```
Generate 3 1
AddFloorTile 0 0
AddFloorTile 1 0
AddFloorTile 2 0
AddZPM 2 0
AddONeill 0 0 1
Shoot primary oneill
MoveProjectile oneill
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Floor csempe hozzaadva. X: 1 Y: 0
Floor csempe hozzaadva. X: 2 Y: 0
ZPM hozzaadva. X: 2 Y: 0
ONeill entitas hozzaadva az X:0 Y:0 helyre 1 irányban
ONeill entitas kilott egy primary tipusu lovedeket
Projectile entitas megprobal ralepni Floor csempere
Projectile entitas ralepett Floor csempere az X: 1 Y: 0
helyen
```

### 8.2.27 Teszteset27: A replikátor szakadékba mozog

- **Leírás**

A replikátor szakadékba mozog és átváltozik padló csempévé.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ha nem változik padló csempévé, rossz a program.

- **Bemenet**

```
Generate 3 1
AddFloorTile 0 0
AddGapTile 1 0
AddReplicator 0 0 1
Move replicator 1
```

- **Elvárt kimenet**

```
Sikeres Generalas
Floor csempe hozzaadva. X: 0 Y: 0
Gap csempe hozzaadva. X: 1 Y: 0
Replicator entitas hozzaadva. X: 0 Y: 0
Replicator entitas megprobal ralepni Gap csempere
Replicator entitas szakekba lepett es feltoltotte azt az
X: 1 Y: 0 helyen
Floor csempe hozzaadva. X: 1 Y: 0
```

## 8.3 A tesztelést támogató programok tervei

A kimenetek txt formában fognak előállni. A Windows parancssor FC fájl összehasonlító programjával, egy script fájlal fogjuk összevetni a mellékelt elvárt kimenetek fájlokkal.

## 8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2016.04.09. 22:00	0,5 óra	Bujdosó Bukta	Értekezlet. Döntés: Bukta elkészíti a teszteseteket és a tesztelést támogató programok fejezetet. Pontosítva PrintMap kimenete.
2016.04.09. 22:30	3 óra	Bukta	Tevékenység: tesztesetek leírása.
2016.04.10. 21:15	2 óra	Vincze	Tevékenység: Tesztesetek be és kimeneteinek leírása
2016.04.10. 16:00	4 óra	Bukta	Tevékenység: tesztesetek leírása.
2016.04.10. 21:00	2 óra	Bujdosó	Tevékenység: Osztályleírások elkészítése
2016.04.10. 23:00	2 óra	Bujdosó	Tevékenység: osztályleírások véglegesítése valamint osztálydiagram utánigazítása
2016.04.11. 07:00	2,5 óra	Karácsony	Tevékenység: Dokumentum véglegesítése, javított osztálydiagram nyomtatása.

## 10. Prototípus beadása

### 10.1 Fordítási és futtatási útmutató

#### 10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Entity.java	2214 byte	2016.04.27	Entity osztály
Jaffa.java	711 byte	2016.04.27	Jaffa osztály
ONeill.java	755 byte	2016.04.27	ONeill osztály
Player.java	6014 byte	2016.04.27	Player osztály
Projectile.java	2310 byte	2016.04.27	Projectile osztály
Replicator.java	1995 byte	2016.04.27	Replicator osztály
Game.java	809 byte	2016.04.27	Game osztály
GamePanel.java	1494 byte	2016.04.27	GamePanel osztály
GameState.java	404 byte	2016.04.27	GameState osztály
GameStateManager.java	1138 byte	2016.04.27	GameStateManager osztály
LevelState.java	774 byte	2016.04.27	LevelState osztály
MenuState.java	1000 byte	2016.04.27	MenuState osztály
Box.java	65 byte	2016.04.27	Boksz osztály
Door.java	647 byte	2016.04.27	Door osztály
MapObject.java	237 byte	2016.04.27	MapObject osztály
Scale.java	766 byte	2016.04.27	Scale osztály
ZPM.java	59 byte	2016.04.27	ZPM osztály
Stargate.java	783 byte	2016.04.27	Stargate osztály
Wormhole.java	1246 byte	2016.04.27	Wormhole osztály
ProtoTestManager.java	12043 byte	2016.04.27	ProtoTestManager osztály
ProtoTests.java	10407 byte	2016.04.27	ProtoTests osztály
TestBed.java	206 byte	2016.04.27	TestBed osztály
testOutPutter.java	1475 byte	2016.04.27	testOutPutter osztály
Tests.java	12528 byte	2016.04.27	Tests osztály
TileMap.java	1836 byte	2016.04.27	TileMap osztály
Floor.java	1716 byte	2016.04.27	Floor osztály
Gap.java	931 byte	2016.04.27	Gap osztály
Tile.java	2507 byte	2016.04.27	Tile osztály
Wall.java	1282 byte	2016.04.27	Wall osztály

#### 10.1.2 Fordítás

- A mellékelt ZIP-et ki kell csomagolnunk tetszőleges helyre.
- A mappába navigálva kiadni a következő parancsot:

```
javac -d bin src/Tiles/*.java src/TileMap/*.java
src/Testing/*.java src/Stargate/*.java
src/Objects/*.java src/Gamestates/*.java
src/Game/*.java src/Entities/*.java
```

#### 10.1.3 Futtatás

- Lépünk be az előző pontban kapott bin mappába
- Indítsuk el a játékot a **java Game.Game** paranccsal. A lehetséges paraméterek az alábbiak:

- **test**: teszt üzemmódban fut a program
- **true**: fájlból fájlba dolgozik a program, ez esetben meg kell neki adni a bemeneti fájlt
- **false**: manuális tesztelés

Tehát egy lehetséges futtatás: **test true bin/ProtoTestIn/1-in.txt**

## 10.2 Tesztek jegyzőkönyvei

### 10.2.1 Teszteset1

Tesztelő neve	Karácsony Zsolt
Teszt időpontja	2016.04.26, 23:59

### 10.2.2 Teszteset2

Tesztelő neve	Karácsony Zsolt
Teszt időpontja	2016.04.27, 0:02

### 10.2.3 Teszteset3

Tesztelő neve	Horváth Balázs
Teszt időpontja	2016.04.27, 0:21

### 10.2.4 Teszteset4

Tesztelő neve	Karácsony Zsolt
Teszt időpontja	2016.04.27, 0:04

### 10.2.5 Teszteset5

Tesztelő neve	Horváth Balázs
Teszt időpontja	2016.04.27, 1:53

Tesztelő neve	Horváth Balázs
Teszt időpontja	2016.04.27, 1:50
Teszt eredménye	Az ajtó és a mérleg létrehozásánál nem írta ki a koordinátát.
Lehetséges hibaok	A kiíratásban nem íratjuk ki.
Változtatások	Kiíratunk a koordinátát is.

### 10.2.6 Teszteset6

Tesztelő neve	Karácsony Zsolt
Teszt időpontja	2016.04.27, 0:07

### 10.2.7 Teszteset7

Tesztelő neve	Horváth Balázs
Teszt időpontja	2016.04.27, 0:25

### 10.2.8 Teszteset8

Tesztelő neve	Karácsony Zsolt
Teszt időpontja	2016.04.27, 0:20

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:10
<b>Teszt eredménye</b>	ZPM-ek száma 1 helyett 0.
<b>Lehetséges hibaok</b>	A teszt azelőtt kérdezte le a ZPM-ek számát, hogy meg lett volna növelve.
<b>Változtatások</b>	Vonatkozó metódus javítása.

### 10.2.9 Teszteset19

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 0:27

### 10.2.10 Teszteset10

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:23

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:20
<b>Teszt eredménye</b>	Összecsúsztik a kiíratott szöveg.
<b>Lehetséges hibaok</b>	Kiíratásnál hiányzik egy newline.
<b>Változtatások</b>	Kiíratás javítása.

### 10.2.11 Teszteset11

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:53

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:28
<b>Teszt eredménye</b>	java.lang.ArrayIndexOutOfBoundsException: 2, valamint nem lép a doboz helyére felvételkor.
<b>Lehetséges hibaok</b>	Valami túllép a 3 csempe lerakásánál.
<b>Változtatások</b>	Hibás bemeneti fájl volt megadva, ezt javítva megszűnt az indextúllépés, a doboz helyére lépés pedig egy typo volt, ezt javítva megszűnt minden probléma.

### 10.2.12 Teszteset12

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:25

### 10.2.13 Teszteset13

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:32

### 10.2.14 Teszteset14

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:30



**10.2.15 Teszteset15**

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:34

**10.2.16 Teszteset16**

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:32

**10.2.17 Teszteset17**

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:35

**10.2.18 Teszteset18**

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:35

**10.2.19 Teszteset19**

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:38

**10.2.20 Teszteset20**

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:40

**10.2.21 Teszteset21**

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 2:40

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:40
<b>Teszt eredménye</b>	A doboznak ütközést nem írja ki.
<b>Lehetséges hibaok</b>	Nem íratjuk ki.
<b>Változtatások</b>	A bemeneti fájlból hiányzott, egy lépés.

**10.2.22 Teszteset22**

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:42

**10.2.23 Teszteset23**

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 2:44

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:42
<b>Teszt eredménye</b>	A replikátor nem látszik hogy létrejön.
<b>Lehetséges hibaok</b>	Vagy nem íratjuk ki vagy nem jött létre.
<b>Változtatások</b>	A bemeneti fájlból hiányzott a létrehozás..

**10.2.24 Teszteset24**

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:50

**10.2.25 Teszteset25**

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 2:50

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:45
<b>Teszt eredménye</b>	A lövedék az ajtó kinyílása után sem jut el a 4. csempéig.
<b>Lehetséges hibaok</b>	kevesebbet lépünk.
<b>Változtatások</b>	A bemeneti fájlból hiányzott, egy lépés.

**10.2.26 Teszteset26**

<b>Tesztelő neve</b>	Karácsony Zsolt
<b>Teszt időpontja</b>	2016.04.27, 0:55

**10.2.27 Teszteset27**

<b>Tesztelő neve</b>	Horváth Balázs
<b>Teszt időpontja</b>	2016.04.27, 1:47

**10.3 Értékelés**

<b>Tag neve</b>	<b>Munka százalékban</b>
Bujdosó Dániel	29%
Bukta Bence	21%
Horváth Balázs	15%
Karácsony Zsolt	13%
Kelemen András	8%
Vincze Zsolt	14%

**10.4 Napló**

<b>Kezdet</b>	<b>Időtartam</b>	<b>Résztevők</b>	<b>Leírás</b>
2016.04.22. 14:00	6 óra	Bujdosó	Tevékenység: Metódusok implementálása
2016.04.24. 21:00	2 óra	Bukta	Tevékenység: ProtoTestManager osztály tervezés
2016.04.25. 12:00	5 óra	Bukta	Tevékenység: ProtoTestManager osztály implementálás és tesztelés
2016.04.25. 18:00	4 óra	Bujdosó	Tevékenység: Tesztesetek megírása
2016.04.25. 20:30	0,5 óra	Karácsony	Tevékenység: kimeneti fájlok összegyűjtése, teszteléstámogatás előkészítése
2016.04.25. 22:00	0,5 óra	Horváth	Tevékenység: bemeneti fájlok összegyűjtése
2016.04.26. 12:00	4 óra	Bukta	Tevékenység: összehasonlító script készítése és kód kimenet egységesítése
2016.04.26. 14:00	8 óra	Bujdosó	Tevékenység: tesztesetek kidolgozása
2016.04.26. 17:00	1,5 óra	Horváth	Tevékenység: A fájl-lista elkészítése
2016.04.26. 19:00	1,5 óra	Horváth	Tevékenység: Tesztelési környezet beállítás, dokumentumok előkészítése.
2016.04.26. 20:30	1 óra	Karácsony	Tevékenység: Teszt környezet beállítása, felkészülés
2016.04.26. 22:00	5 óra	Horváth	Tevékenység: Tesztelés, dokumentálás, fájl- lista pontosítása.

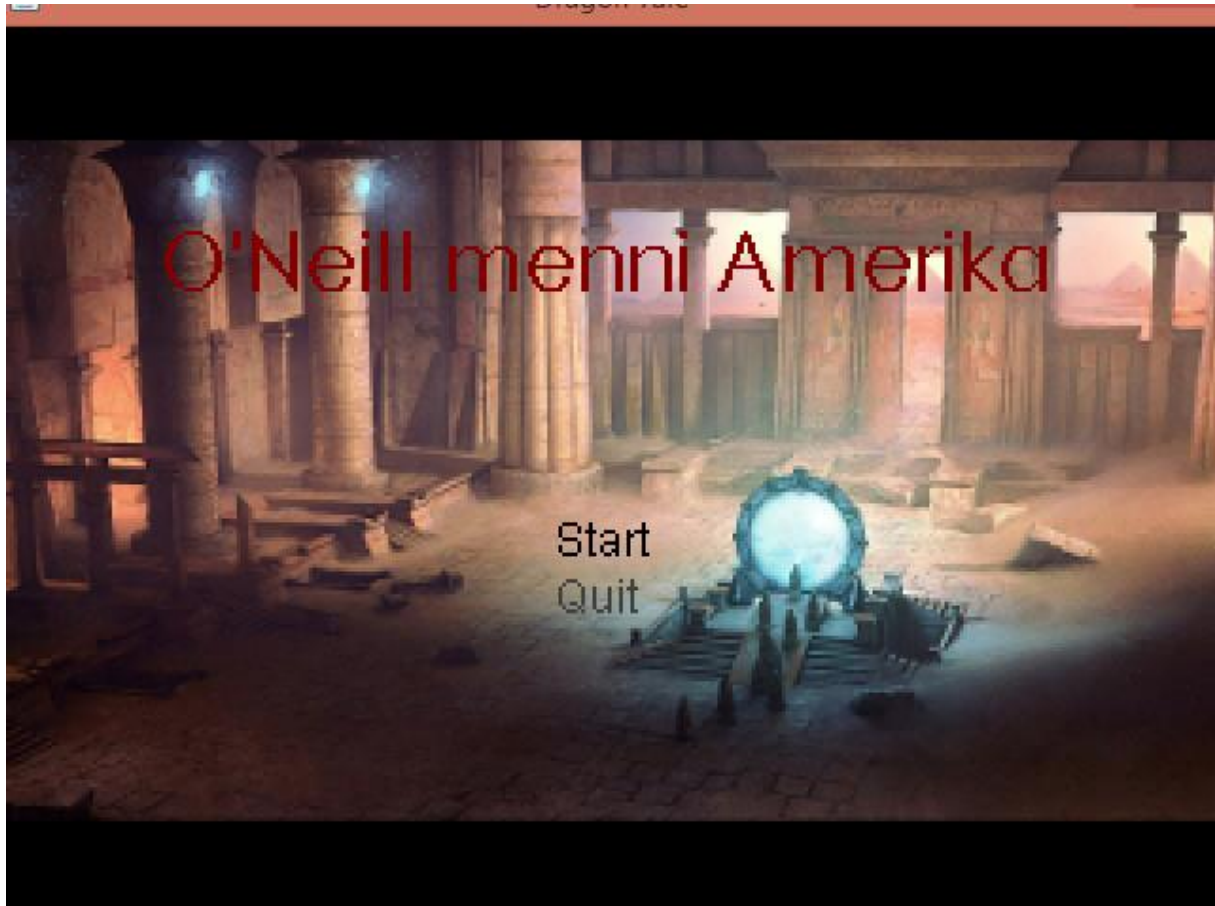
2016.04.26, 22:50	4 óra	Karácsony	Tevékenység: automatizált teszt eredményeinek kiértékelése, Részletes tervek frissítése a megváltozott kimeneti nyelv átvezetése. Jelen dokumentum megírása.
2016.04.26. 23:50	2 óra	Bukta	Tevékenység: tesztelés támogatás
2016.04.27. 06:20	1 óra	Karácsony	Tevékenység: Dokumentáció véglegesítése.

## 11. Grafikus felület specifikációja

### 11.1 A grafikus interfész

A játékban menü található, amiben a játékos új játékot tud indítani. A pályák 32\*32-es „csempékből” épülnek fel.

A játék jelen állapotában az O'Neill menni Amerika fedőnevet viseli.



11.1. ábra A játék indító képernyője

#### 11.1.1 Felhasznált képek

##### 11.1.1.1 Padló

Ezen közlekedik O'Neill, a Jaffa és a Replikátor is. A padló elemeken továbbá lehetnek ajtók, mérlegek, dobozok, ZPM-ek, lövedékek.



11.2. ábra Padló képe

### 11.1.1.2 Ajtó

Zárt és nyitott verziója van, attól függően, hogy a hozzá tartozó mérlegen van-e megfelelő súly.



11.3. ábra Zárt ajtó



11.4. ábra Nyitott ajtó

### 11.1.1.3 Lövedék

4 színben létezik, a padló fölött haladnak.



11.5. ábra Lövedékek

### 11.1.1.4 Doboz

Padlón, vagy padlón lévő mérlegen lehet doboz. Mérlegre téve súlyként viselkedik.



11.6. ábra Doboz képe

### 11.1.1.5 O'Neill, Jaffa

A játékban irányítható karakterek. Ugyan olyan tulajdonságokkal rendelkeznek, csak a kilőtt portálokban illetve saját grafikájuk színében különböznek.



11.7. ábra O'Neill és Jaffa képe

### 11.1.1.6 Replikátor

Hasonlóan Oneill-hez és a Jaffához a padlón közlekedik. Ellentétben az előbbiekkal, játékos által nem irányítható. Ha lövedék éri elpusztul és eltűnik. Ha szakadékba lép akkor eltűnik és a szakadék normál mezővé változik.



11.8. ábra Replikátor képe

### 11.1.1.7 Mérleg

A lenyomott és le nem nyomott mérleg két különböző színt kapott. Padló elemen helyezkedik el. Ha lenyomott állapotban van, a hozzá tartozó ajtó kinyílik.



11.9. ábra Aktív és inaktív mérleg képe

### 11.1.1.8 Csillagkapu

Négy színben létezik, ahogy a lövedékek is. Adott színű lövedék adott színű csillagkaput hoz létre speciális falba ütközés után.



11.10. ábra Csillagkapuk képei

### 11.1.1.9 Fal

Más elemek nem kerülhetnek rá, nem léphetnek rá.



11.11. ábra Fal képe

### 11.1.1.10 Speciális fal

Hasonló a falhoz, azzal a különbséggel, hogy ha ezt lövedék találja el, akkor megfelelő színű csillagkapu nyílik rajta.



11.12. ábra Speciális fal képe

### 11.1.1.11 Szakadék

Oneill és a Jaffa közvetlenül nem tud átmenni, de át tud lőni felettük. Ha az ezredes vagy a Jaffa szakadékba esik, akkor meghal. Ha egy tárgy szakadékba esik, akkor megsemmisül. Ha replikátor lép rá, akkor normál elemmé változik és a replikátor megsemmisül.



11.13. ábra Szakadék képe

### 11.1.1.12 ZPM

Az ezredes és a Jaffa gyűjti őket. Elszórtan helyezkednek el a pályán.



11.14. ábra ZPM képe

## 11.1.2 Minta összeállítás

Az alább található egy összeállítás, hogy hogy nézzenek ki az egyes elemek egymás mellé, illetve egymásra illesztve.



11.15. ábra Játék mintaösszeállítása

## 11.2 A grafikus rendszer architektúrája

### 11.2.1 A felület működési elve

### 11.2.2 A felület osztály-struktúrája

A grafikus felület már eddig is része volt a programnak.

A GamePanel osztályon belül a gameloop-ban van egy draw() függvényhívás, ami láncreakciószerűen végighívja a kirajzolandó objektumok saját draw() metódusát, ami végül megjeleníti őket a képernyőn.

A draw() metódusok lényege, hogy egy azonosító alapján kikeresik a hozzájuk tartozó grafikát egy Map entitásból és azt rajzolják ki az állapotuknak megfelelően (pl. megfelelő irányban.)

## 11.3 A grafikus objektumok felsorolása

Az osztálydiagram változatlan marad, csak az osztályok `draw()` metódusa kerül implementálásra.

### 11.3.1 GamePanel

- **Felelősség**  
A GamePanel osztály indítja a játékot futtató szálát, valamint tartalmazza játékciklust (gameloop). Ez az osztály tölti ki a Game osztályban létrehozott ablakot, valamint fogadja a bemenetet a billentyűzetről és továbbadja azokat.
- **Ősosztályok**  
java.lang.Object -> java.awt.Component -> java.awt.Container -> javax.swing.JComponent
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**
  - **+draw(Graphics2D g) : void**: Meghívja a GameStateManager draw() metódusát.

### 11.3.2 GameStateManager

- **Felelősség**  
A játék különböző állapotai közötti váltásért felelős osztály. Ez az osztály kapja meg a GamePanel osztálytól a leütött billentyűket, majd továbbadja őket az éppen aktuális játékállapotnak.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**
  - **+MenuState(GameStateManager gsm)**: Ebben a konstruktorban állítjuk be a menuelemek betűszínét és betűtípusát.
  - **+draw(Graphics2D g) : void**: Kirajzolja a menu opcióit.



### 11.3.3 LevelState

- **Felelősség**  
A játékmenetért felelős játékállapot (GameState) osztály. Ő felel a labirintusért illetve a labirintusban szereplő elemekért, mint például O'Neill ezredes.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**
  - **+LevelState(GameStateManager gsm):** Ebben a konstruktorban hozzuk létre a TileMapot a megfelelő paraméterekkel
  - **+draw(Graphics2D g) : void:** Kirajzolja, meghívja a játékelemek draw() metódusait.

### 11.3.4 TileMap

- **Felelősség**  
Ez az osztály tárolja magában a labirintust, illetve az ahhoz szükséges grafikai elemeket. A labirintus valamit a hozzá tartozó csempegrafika beolvasás is az ő feladata.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **-oneill : ONeill :** az 1-es játékos referenciáját tároló változó
  - **-jaffa : Jaffa :** a 2-es játékos referenciáját tároló változó
- **Metódusok**
  - **+draw(Graphics2D g) : void :** Végigmegy a Tiles tömbön és meghívja minden egyes csempe draw() metódusát.

### 11.3.5 Tile

- **Felelősség**  
A játékteret felépítő labirintus alap építőegysége.
- **Ősosztályok**  
Nem értelmezhető.

- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**
  - **#textureMap : HashMap**: A tile osztály kapott egy textureMap változót, ami a grafikát tartalmazó HashMap referenciáját tartalmazza.
- **Metódusok**
  - **+Tile(int x, int y, HashMap textureMap)** : Mivel a csempe nem látja a TilaMap osztályt, ezért a konstruktorban adjuk át a textureMap-ra mutató referenciát.
  - **+draw(Graphics2D g) : void** : Kirajzolja a megfelelő koordinátára a csempéhez tartozó grafikát egy azonosító alapján.

### 11.3.6 Wall

- **Felelősség**  
Speciális csempe, feladata hogy elválasztó elemként működjön a játékban, korlátozva a játékost a mozgásban valamint bizonyos falakon lehet csillagkaput is nyitni. A falon sem átsétálni se átlőni nem lehet.
- **Ősosztályok**  
Tile
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**
  - **+draw(Graphics2D g) : void** : Más-más grafikát rajzol ki attól függően, hogy a fal speciális e.

### 11.3.7 Entity

- **Felelősség**  
A játékos által „irányított” és animált objektumok ősosztálya.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.

- **Metódusok**

- **+draw(Graphics2D g) : void** : Kirajzolja a entitást jelképező grafikát, a entitás irányának megfelelő irányba.

### 11.3.8 Player

- **Felelősség**

A 2 játékos osztály ősosztálya, ez az osztály felel a legtöbb játékos karakter (O'Neill, Jaffa) interakcióért.

- **Ősosztályok**

Nem értelmezhető.

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

Nem értelmezhető.

- **Metódusok**

- **+draw(Graphics2D g) : void** : Az entitás kirajzolófüggvényén felül, meghívja az aktív lövedékek draw() metódusát is.

### 11.3.9 Wormhole

- **Felelősség**

A pályán található csillagkapuk és féregjáratok működéséért felelős osztály. Ő tárolja a kapuk helyét és ügyel arra, hogy egy fajta kapuból csak 1 darab létezzen.

- **Ősosztályok**

Nem értelmezhető.

- **Interfészek**

Nem értelmezhető.

- **Attribútumok**

Nem értelmezhető.

- **Metódusok**

- **+draw(Graphics2D g) : void** : Meghívja az éppen aktív csillagkapuk draw() metódusát

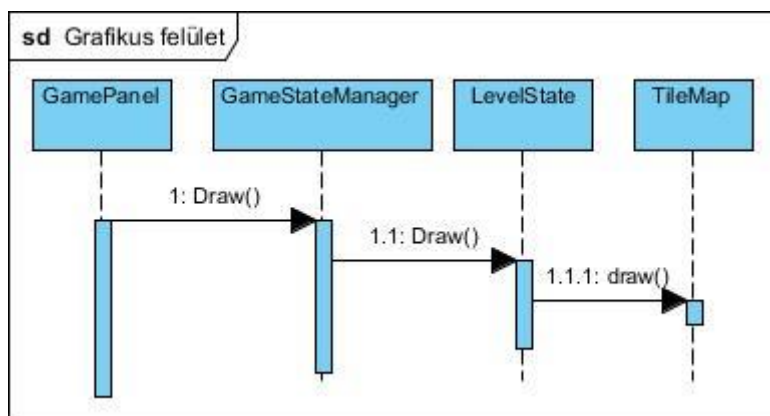
### 11.3.10 Stargate

- **Felelősség**  
A csillagkapuk adatainak tárolását végző osztály, tárolja a kapuk helyzetet valamint irányát.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**
  - **+draw(Graphics2D g) : void** : Az adott csempe fölé, kirajzolja a csillagkapuhoz tartozó grafikát.

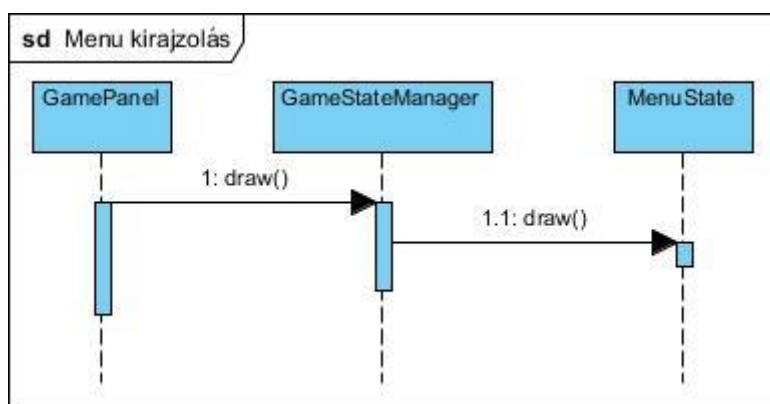
### 11.3.11 Replicator

- **Felelősség**  
A replikátort leíró osztály. Feladata, hogy véletlenszerűen mozgatja önmagát, és megfelelően lép interakcióba a különböző pályaelemekkel.
- **Ősosztályok**  
Nem értelmezhető.
- **Interfészek**  
Nem értelmezhető.
- **Attribútumok**  
Nem értelmezhető.
- **Metódusok**
  - **+draw(Graphics2D g) : void** : A replikátor irányától függően kirajzolja a replikátorhoz tartozó grafikát.

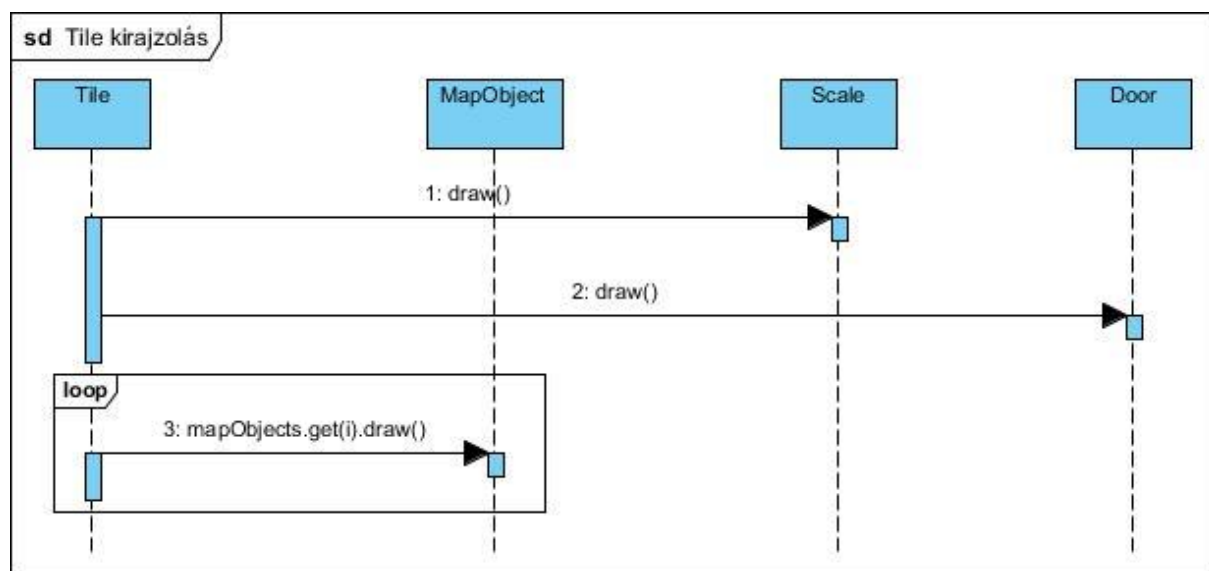
## 11.4 Kapcsolat az alkalmazói rendszerrel



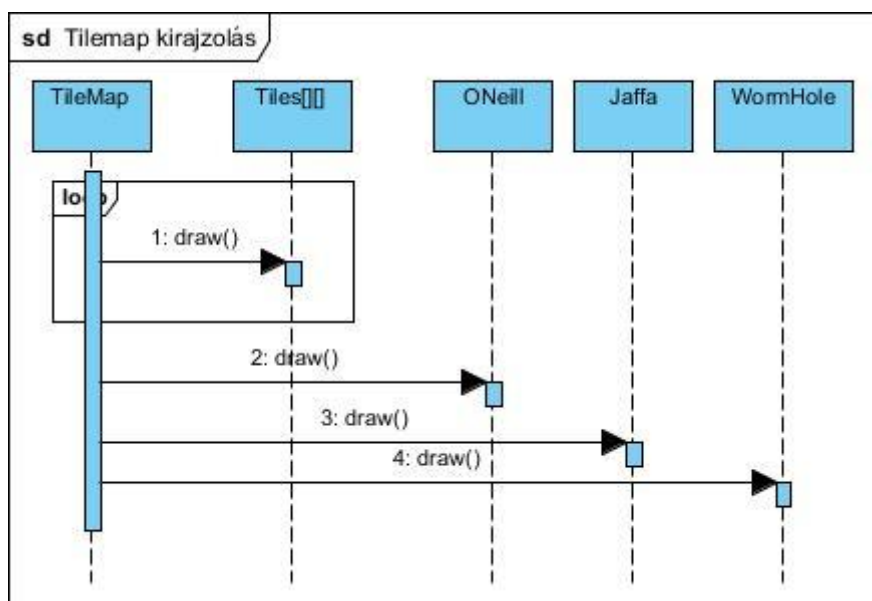
11.16. ábra Grafikus felület működése



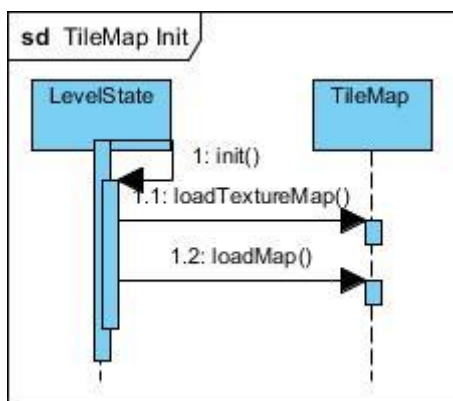
11.17. ábra Menü kirajzolása



11.18. ábra Csempék kirajzolása



11.19. ábra Tilemap kirajzolása



11.20. ábra Csempék inicializálása

**11.5 Napló**

<b>Kezdet</b>	<b>Időtartam</b>	<b>Résztevők</b>	<b>Leírás</b>
2016.04.30. 15:00	2 óra	Vincze	Tevékenység: Grafikus elemek készítése.
2016.04.30. 18:00	2 óra	Vincze	Tevékenység: Grafikus elemek készítése.
2016.04.30. 21:00	1 óra	Vincze	Tevékenység: Grafikus elemek készítése.
2016.04.30. 18:00	4 óra	Bujdosó	Tevékenység: Tervek elkészítése.
2016.05.01. 20:00	2 óra	Bujdosó	Tevékenység: Tervek elkészítése.
2016.05.01. 22:00	2 óra	Bujdosó	Tevékenység: Szekvenciadiagramok és dokumentáció.
2016.05.02. 07:00	1 óra	Karácsony	Tevékenység: Dokumentáció véglegesítése.

## 13. Grafikus változat beadása

### 13.1 Fordítási és futtatási útmutató

#### 13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
play.bat	24	2016.05.16 21:41	Futtató script
install.bat	172	2016.05.16 21:41	Telepítő script
bin\resources\menubckgrnd.jpg	78 852	2016.04.30 18:20	Grafikus felület forrás
bin\resources\ajto_csukva01.png	1 200	2016.05.01 19:58	Grafikus felület forrás
bin\resources\ajto_nyitva01.png	587	2016.05.01 19:58	Grafikus felület forrás
bin\resources\background.png	342	2016.05.02 17:49	Grafikus felület forrás
bin\resources\bgr.png	1 352	2016.05.15 21:53	Grafikus felület forrás
bin\resources\blue.png	798	2016.05.01 19:58	Grafikus felület forrás
bin\resources\box01.png	1 200	2016.05.01 19:58	Grafikus felület forrás
bin\resources\floor01.png	1 155	2016.05.01 19:58	Grafikus felület forrás
bin\resources\green.png	786	2016.05.01 19:58	Grafikus felület forrás
bin\resources\jaffa.png	630	2016.05.01 19:58	Grafikus felület forrás
bin\resources\jaffa4dir.png	2 180	2016.05.15 21:22	Grafikus felület forrás
bin\resources\minden.png	26 369	2016.05.01 19:58	Grafikus felület forrás
bin\resources\oneill.png	633	2016.05.01 19:58	Grafikus felület forrás
bin\resources\oneill4dir.png	2 467	2016.05.02 16:54	Grafikus felület forrás
bin\resources\red.png	815	2016.05.01 19:58	Grafikus felület forrás
bin\resources\replikator.png	938	2016.05.01 19:58	Grafikus felület forrás
bin\resources\replikator4dir.png	3 516	2016.05.02 18:44	Grafikus felület forrás
bin\resources\scale_down3.png	15 650	2016.05.01 19:58	Grafikus felület forrás



bin\resources\scale_up5.png	905	2016.05.01 19:58	Grafikus felület forrás
bin\resources\sg_blue01.png	952	2016.05.01 19:58	Grafikus felület forrás
bin\resources\sg_blue4dir.png	2 682	2016.05.02 17:49	Grafikus felület forrás
bin\resources\sg_green01.png	908	2016.05.01 19:58	Grafikus felület forrás
bin\resources\sg_green4dir.png	2 564	2016.05.02 17:49	Grafikus felület forrás
bin\resources\sg_red01.png	989	2016.05.01 19:58	Grafikus felület forrás
bin\resources\sg_red4dir.png	2 728	2016.05.02 17:49	Grafikus felület forrás
bin\resources\sg_yellow01.png	1 058	2016.05.01 19:58	Grafikus felület forrás
bin\resources\sg_yellow4dir.png	3 044	2016.05.02 17:49	Grafikus felület forrás
bin\resources\specialWallproba03.png	1 737	2016.05.01 19:58	Grafikus felület forrás
bin\resources\szakadek02.png	1 100	2016.05.01 19:58	Grafikus felület forrás
bin\resources\tileset.png	17 787	2016.05.01 19:58	Grafikus felület forrás
bin\resources\wall01.png	1 359	2016.05.01 19:58	Grafikus felület forrás
bin\resources\yellow.png	826	2016.05.01 19:58	Grafikus felület forrás
bin\resources\zpm01.png	1 367	2016.05.01 19:58	Grafikus felület forrás
bin\resources\maps\tmap2.json	9 163	2016.05.06 17:01	Pálya forrás
src\Entities\Entity.java	2 199	2016.05.14 19:38	Forráskód
src\Entities\Jaffa.java	1 535	2016.05.15 21:23	Forráskód
src\Entities\ONeill.java	1 665	2016.05.15 20:50	Forráskód
src\Entities\Player.java	7 602	2016.05.15 21:02	Forráskód
src\Entities\Projectile.java	3 422	2016.05.15 21:26	Forráskód
src\Entities\Replicator.java	3 040	2016.05.03 15:36	Forráskód
src\Game\Game.java	1 100	2016.05.01 21:31	Forráskód
src\Game\GamePanel.java	2 428	2016.05.10 13:39	Forráskód

src\GameStates\GameState.java	532	2016.05.03 15:40	Forráskód
src\GameStates\GameStateManager.java	1 514	2016.05.02 18:37	Forráskód
src\GameStates\LevelState.java	2 467	2016.05.15 20:53	Forráskód
src\GameStates\MenuState.java	2 080	2016.05.15 21:57	Forráskód
src\Objects\Box.java	283	2016.05.03 14:45	Forráskód
src\Objects\Door.java	817	2016.05.03 14:59	Forráskód
src\Objects\MapObject.java	247	2016.05.03 14:43	Forráskód
src\Objects\Scale.java	954	2016.05.03 21:06	Forráskód
src\Objects\ZPM.java	273	2016.05.03 14:45	Forráskód
src\Stargate\Stargate.java	1 854	2016.05.15 21:14	Forráskód
src\Stargate\Wormhole.java	1 542	2016.05.15 21:03	Forráskód
src\Testing\ProtoTestManager.java	13 437	2016.05.01 20:30	Forráskód
src\Testing\ProtoTests.java	11 117	2016.05.01 19:35	Forráskód
src\Testing\TestBed.java	206	2016.04.26 19:39	Forráskód
src\Testing\testOutPutter.java	1 475	2016.04.25 19:30	Forráskód
src\Testing\Tests.java	12 528	2016.04.25 15:15	Forráskód
src\TileMap\JsonObjectLayerObject.java	758	2016.05.06 16:31	Forráskód
src\TileMap\JsonTileLayerObject.java	678	2016.05.06 16:31	Forráskód
src\TileMap\MapInputter.java	4 452	2016.05.06 16:30	Forráskód
src\TileMap\TileMap.java	8 451	2016.05.15 21:49	Forráskód
src\Tiles\Floor.java	2 479	2016.05.05 16:17	Forráskód
src\Tiles\Gap.java	1 051	2016.05.03 15:34	Forráskód
src\Tiles\Tile.java	2 760	2016.05.03 15:21	Forráskód
src\Tiles\Wall.java	1 477	2016.05.01 21:26	Forráskód

### 13.1.2 Fordítás és telepítés

A fordításhoz legalább 1.8-as verziójú JDK szükséges.

A fordításhoz futtassuk le az `install.bat` fájlt.

Ha error-mentesen lefut, akkor sikeres volt a telepítés/fordítás.

### 13.1.3 Futtatás

A játékot a `play.bat` scripttel kell elindítani.

## 13.2 Értékelés

Tag neve	Munka százalékban
Bujdosó Dániel	31%
Bukta Bence	19%
Horváth Balázs	13%
Karácsony Zsolt	12%
Kelemen András	9%
Vincze Zsolt	16%

### 13.3 Napló

Kezdet	Időtartam	Résztevők	Leírás
2016.05.07	2,5 óra	Bujdosó	Tevékenység: Kódolás
2016.05.15	1 óra	Bujdosó	Tevékenység: grafikus elemek véglegesítése
2016.05.15	1 óra	Bujdosó	Tevékenység: kódolás
2016.05.16.	1 óra	Bukta	Tevékenység: Dokumentáció kitöltése
2016.05.17	1 óra	Karácsony	Tevékenység: Dokumentáció véglegesítése, munkaidő frissítése

## 14. Összefoglalás

### 14.1 Projektösszegzés

Fázis	Forrássor
Szkeleton	1518
Prototípus	2584
Grafikus	3108

Tag neve	Teljes idő (óra)	Munka százalékban
Bujdosó Dániel	91	31%
Bukta Bence	56	19%
Horváth Balázs	38	13%
Karácsony Zsolt	36,5	12%
Kelemen András	26	9%
Vincze Zsolt	46,5	16%
<b>Összesen</b>	<b>294</b>	<b>100%</b>

### 14.2 Kérdések, válaszok

#### 14.2.1 Bujdosó Dániel

- **Mit tanultak a projektből konkrétan és általában?**  
Azt, hogy hogyan is néz ki az, amikor egy projektet a nulláról, de elég időt hagyva a tervezésre valósítunk meg.  
Git verziókezelő használata.
- **Mi volt a legnehezebb és a legkönnyebb?**  
A projekt legeleje, a második harmadik beadás volt talán a legnehezebb és onnan folyamatosan egyre egyszerűbbek voltak a beadások. A legkönnyebb talán a grafikus változat lekódolása volt.
- **Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?**  
Igen.
- **Ha nem, akkor hol okozott ez nehézséget?**  
-
- **Milyen változtatási javaslatuk van?**  
Szerintem nem szerencsés véletlenszerű eseményeket beletenni a specifikációba, mert azoknak sokszor nem sok pozitív hatásuk van a játékmenetre. (Pl. replikátornak gyakorlatilag semmi hatása nincs, csak kóvályog, és egyszer beleesik egy lukba vagy lelövik, de ezen felül semmi.) Herkules rendszerben esetleg lehetne követni a pontszámokat és/vagy a jelenléteket.
- **Milyen feladatot ajánlanának a projektre? Szívesen fogadunk minden egyéb kritikát és javaslatot.**  
Egy egyszerűbb roguelike játék ahol a karakternek egy véletlenszerűen generált pályán kell végigmenni, tárgyakat gyűjteni. (Itt a véletlenszerűséget szabályozni

kell valahogy, hogy a pályák teljesíthetőek legyenek.) Egy nyersanyag gyűjtögetős, építkezős játék, ami hasonlít a Minecraft-hoz vagy Terrariához.

#### 14.2.2 Bukta Bence

- **Mit tanultak a projektből konkrétan és általában?**  
Konkrétan: a valóságban nem ilyen egy projekt.  
Általában: hogyan kell közösen egy projekten dolgozni.
- **Mi volt a legnehezebb és a legkönnyebb?**  
Legnehezebb: mindenkit bevonni az összes munkafolyamatba. Ez szinte lehetetlen és nem is működő modell.  
Legkönnyebb: listát készíteni a forráskódokról.
- **Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?**  
Igen
- **Ha nem, akkor hol okozott ez nehézséget?**  
-
- **Milyen változtatási javaslatuk van?**  
Fogadják el, a teljes mértékben, digitális formában történő, dokumentáció leadást.
- **Milyen feladatot ajánlanának a projektre? Szívesen fogadunk minden egyéb kritikát és javaslatot.**  
-

#### 14.2.3 Horváth Balázs

- **Mit tanultak a projektből konkrétan és általában?**  
Konkrétan: a szoftverfejlesztésnek csak kis töredéke a tényleges kódolás.  
Általában: csapatmunkát nehéz jól ütemezni időben.
- **Mi volt a legnehezebb és a legkönnyebb?**  
Legnehezebb: a határidők betartása.  
Legkönnyebb: ha az ember nem ért rá megoldható volt, hogy abban a fázisban nem vesz részt a munkában.
- **Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?**  
Nem. A tantárgy rengeteg időt felemésztett még így is hogy kicsit kevesebbet dolgoztam részesedés szerint. Kétkredites tárgyhoz képest túl sok munka. A Pontozás korrekt volt.
- **Ha nem, akkor hol okozott ez nehézséget?**  
Rengeteg sablonos, és "feleslegesen" ismétlődő feladat.
- **Milyen változtatási javaslatuk van?**  
Sokkal kevesebb, válogatottabb dokumentáció. Tisztában vagyok vele hogy ez a tantárgy lényege, de a projekt nagyságához mérten feleslegesen sok. Főleg az ismétlődő elemeket kellene szűrni.

- **Milyen feladatot ajánlanának a projektre? Szívesen fogadunk minden egyéb kritikát és javaslatot.**

Telefonközpont szimulálása, könnyű osztályokban gondolkodni, remek szekvenciák állíthatók fel, valamint kevesebb a grafikai hangsúly. Mindemellett a Távháló/Komháló2 tudást is lehetne alkalmazni.

Bár van fent előző évi "jó" házi feladatok, de szerintem jó lenne egy tanszék által kiadott, TÖKÉLETES projekt, ami minden feltételnek teljesen megfelel, standardokat teljesíti.

#### 14.2.4 Karácsony Zsolt

- **Mit tanultak a projektből konkrétan és általában?**

Konkrétan, hogy egy szoftver fejlesztése mennyire részletes folyamat, ha ilyen módszertannal végzi az ember, mint amit a tárgyban elvárnak. Általánosságban az ütemezés, feladatok meghatározásának fontosságát, a határidők tartását.

- **Mi volt a legnehezebb és a legkönnyebb?**

A legnehezebb a határidők tartása volt. A legkönnyebb a konzulensükkel való egyeztetés, aki rendkívül segítőkész volt, nagyon pozitív hozzáállással.

- **Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?**

Részben. Kétkredites tárgyhöz képest irreálisan sok erőbefektetést követel meg a tárgy.

- **Ha nem, akkor hol okozott ez nehézséget?**

Ismétlődő, szükségtelenül elbonyolított dokumentációk, a kiadott template-ek elavultsága miatt a dokumentálás nem volt gördülékeny.

- **Milyen változtatási javaslatuk van?**

2016 van, javaslom, hogy térjen át a tanszék a tisztán elektronikus beadásra. Ezzel egyidejűleg a Herculest érdemes lenne felokosítani arra, hogy a leadási határidőt rendszeresen kiírja az egyes feladatokhoz, hogy ne szabja meg 3 maximális feltöltésben a próbálkozásokat.

- **Milyen feladatot ajánlanának a projektre? Szívesen fogadunk minden egyéb kritikát és javaslatot.**

Feladatot nem tudok ajánlani, általános javaslat a template-ek frissítése mind stílusban, mind guideline szintjén.

#### 14.2.5 Kelemen András

- **Mit tanultak a projektből konkrétan és általában?**

Azt, hogy egy fejlesztési projekthez rengeteg dokumentáció szükséges és ezek konzisztenciája mennyire fontos.

- **Mi volt a legnehezebb és a legkönnyebb?**

A legnehezebb a projekten dolgozó emberek kommunikációjának összehangolása, főleg ha fizikailag távol vannak egymástól.

- **Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?**

Nagyságrendileg igen.

- **Ha nem, akkor hol okozott ez nehézséget?**  
A végtelen mennyiségű doksi (különböző diagramokkal) gyártása sok helyen túl sok időt igényelt.
- **Milyen változtatási javaslatuk van?**  
A fentebb említett elvárt dokumentumok mennyiségén lehetne csökkenteni. Talán átláthatóbb is lenne az egész projekt a végére.
- **Milyen feladatot ajánlanának a projektre? Szívesen fogadunk minden egyéb kritikát és javaslatot.**  
Alapvetően tetszett a feladat, az ilyen és ehhez hasonló sci-fi/játék témákkal meg lehet fogni a VIK-es hallgatókat.

#### 14.2.6 Vincze Zsolt

- **Mit tanultak a projektből konkrétan és általában?**  
Azt, hogy ha nem egy munkahelyen vagy kollégiumban van az ember azokkal, akikkel együtt akar dolgozni, akkor szinte lehetetlen összeszervezni a munkát. Ha az emberek nem tudnak sokszor hosszabb időre összeülni egyszerre egy helyen, akkor szinte lehetetlen együtt dolgozni.
- **Mi volt a legnehezebb és a legkönnyebb?**  
A legnehezebb, és egyben feleslegesen részletesen kidolgozandó rész az „Analízis modell kidolgozása” volt. Olyan részletekbe menően kellett kidolgozni a programot, hogy nem is tudtam elképzelni, hogy a félév hátralevő részében milyen feladatot fogunk még kapni azon kívül, hogy akkor ezt kódoljuk le. Az hogy ennyire az elején ennyire részletesen kellett specifikálni mindent, azzal csak azt érték el, hogy a félév hátralevő részében ezt folyamatosan javíthatnánk visszamenőlegesen. Az alapok kis mértékben változtak csak, de rengeteg részletet kellett megváltoztatni, ahogy fejlődött a program. Ha egy kicsit kevésbé pontosan kellett volna ezt a részt megcsinálni, az sokkal ésszerűbb, és életszerűbb lett volna.
- **Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?**  
Általában igen.
- **Ha nem, akkor hol okozott ez nehézséget?**  
Ha megnézzük a munkaórákat az egyes beadásokra, akkor az Analízis modell kidolgozására  $2x/3x$  annyi pont is mehetett volna... Vagyis pontosabban inkább olyan feladatnak kellett volna lennie, amire nem kell sokszoros időbefektetés a többi feladathoz képest.
- **Milyen változtatási javaslatuk van?**  
Érjék el, hogy ne 2 kredites legyen a tárgy vagy azt, hogy kevesebb munka legyen vele, mert ez így elfogadhatatlan, hogy egy 2 kredites tárggyal ennyit kelljen dolgozni.
- **Milyen feladatot ajánlanának a projektre? Szívesen fogadunk minden egyéb kritikát és javaslatot.**  
A játékkészítés tökéletes feladat ehhez a tárggyhoz, azt általában mindenki szeret csinálni.