

0. Változások a modellben

0.1 Változások leírása

0.1.1 Kereszteződő sínek

Eddig a pályaelemek szomszédai egyértelműen meghatározták a vonat haladási irányát, de a változás miatt szükséges lesz egyértelműbbé tenni, hogy egy `previousElement`-re milyen másik elemet kell visszakapnia a vonatnak a `getNext` függvény hatására. Ezért a pályaelemek párokban tárolják az egyes bemenetekre a kimeneteket.

0.1.2 Utasok felszállása

Amikor a vonat állomásra ér, nem csak a leszálló, hanem a felszálló utasokat is figyelni kell. Mivel minden vagonról tudjuk, hogy milyen a színe, illetve, hogy üres-e, már csak azt kell tudnunk, hogy az adott állomáson tartózkodnak-e utasok, illetve lehetőséget kell biztosítani arra, hogy a kocsit feltöltsük.

0.1.3 Szeneskocsi

A helyzetet egyszerűen tervezzük kezelni, a szeneskocsi egy átlagos kocsi lesz, aminek a színe fekete. Így csak azt kell megoldani, hogy amikor az első nem üres kocsit keressük az utasok leszállítása céljából, akkor ne vegyük figyelembe a szeneskocsit.

0.2 Változások a diagramokon

0.2.1 Osztálydiagram

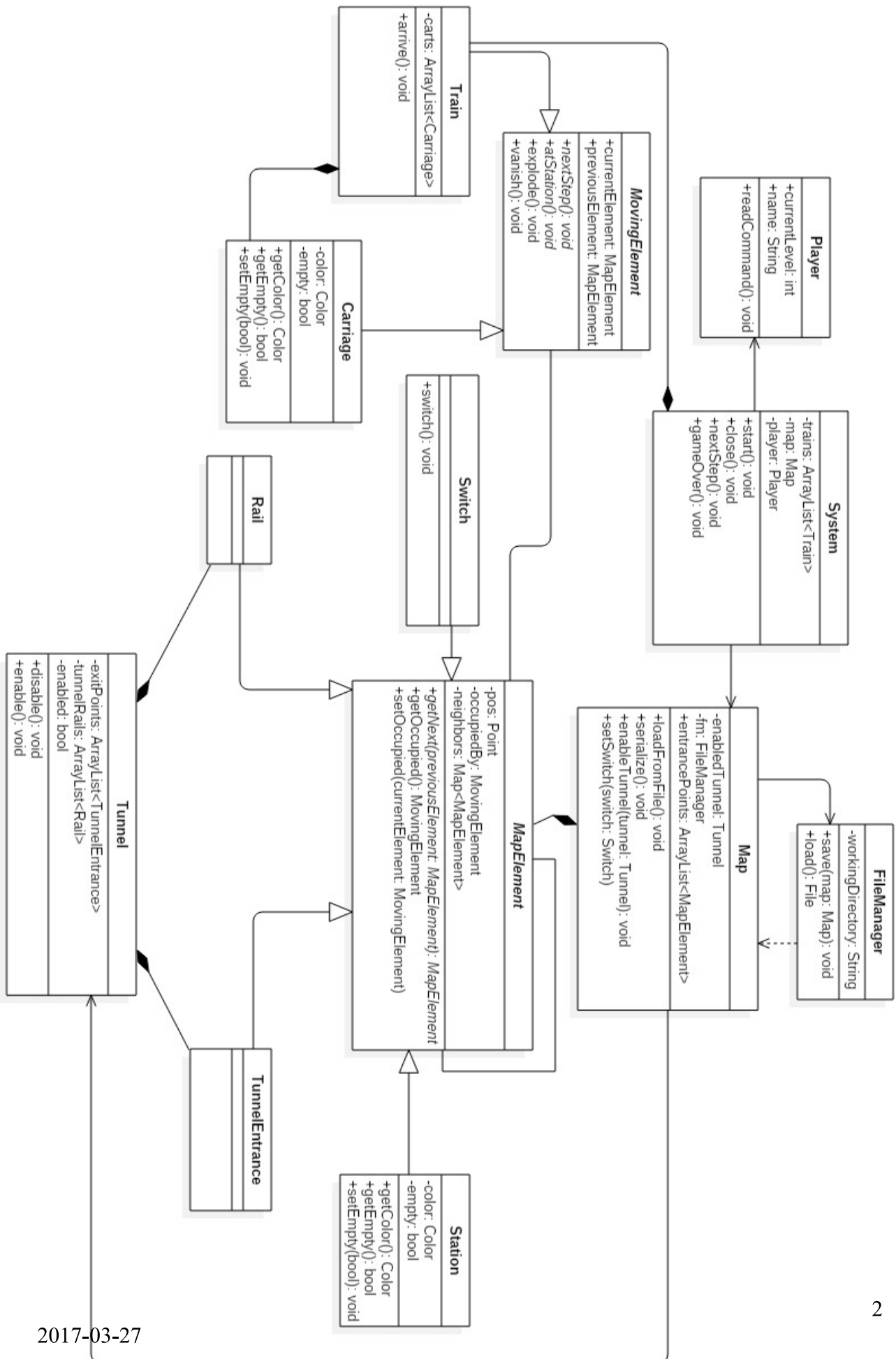
Üresség vizsgálata a `Station` osztálynál:

- `bool empty` - azt jelzi, hogy üres-e az állomás
- `getEmpty(): bool` - visszaadja, hogy üres-e az állomás
- `setEmpty(bool): void` - beállítja, hogy üres-e az állomás

Üresség vizsgálata a `Carriage` osztálynál:

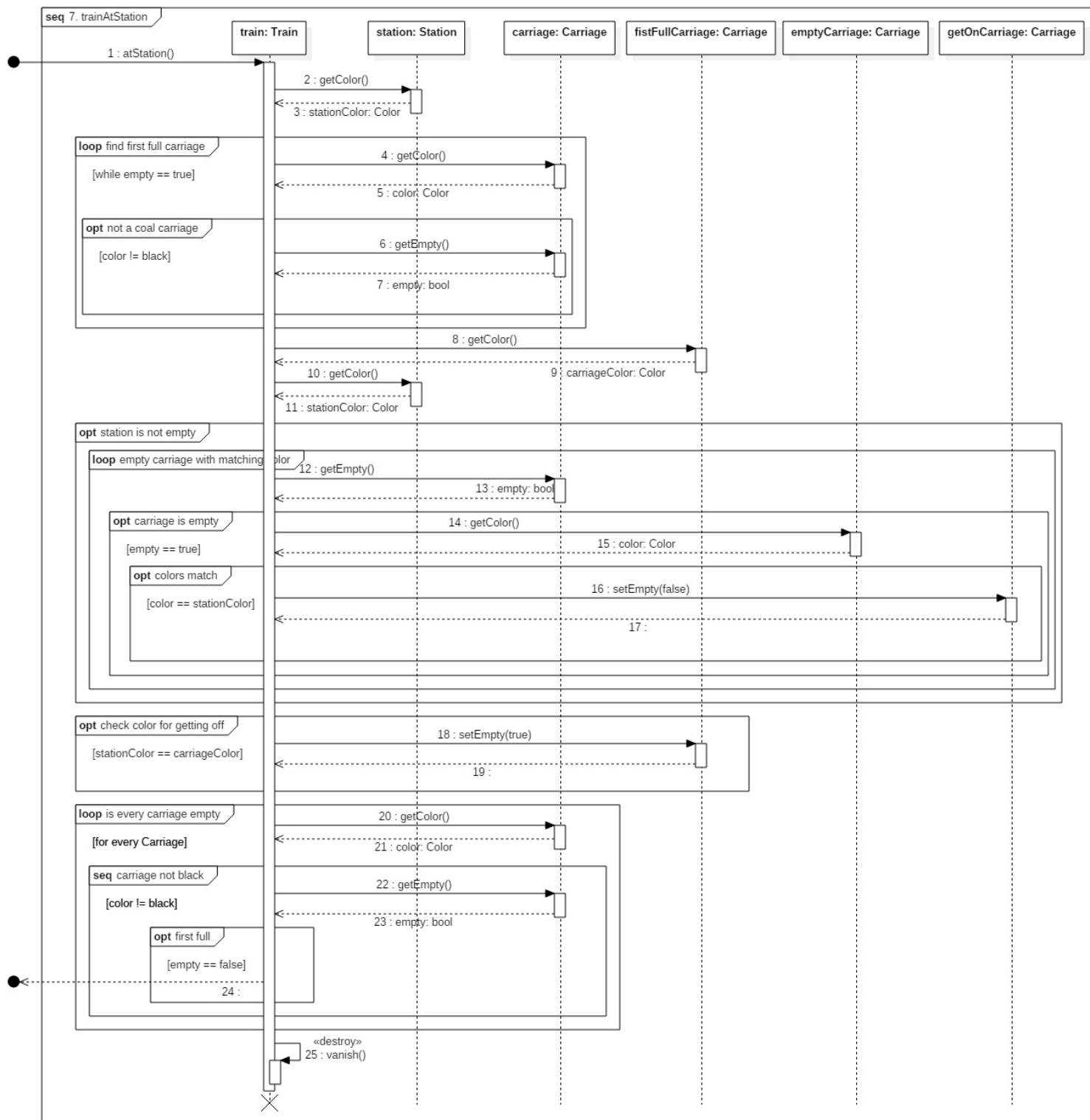
- `bool empty` - azt jelzi, hogy üres-e a kocsi
- `getEmpty(): bool` - visszaadja, hogy üres-e a kocsi
- `setEmpty(bool): void` - beállítja, hogy üres-e a kocsi

Ahhoz, hogy a kereszteződő sínek megfelelően működjenek, a `MapElement`-ek szomszédait ezentúl nem egy listában, hanem párokban tároljuk (`Map` típus). Így minden pozícióra eldönthető, hogy mi kell legyen a következő `MapElement`, amire a vonat lép.



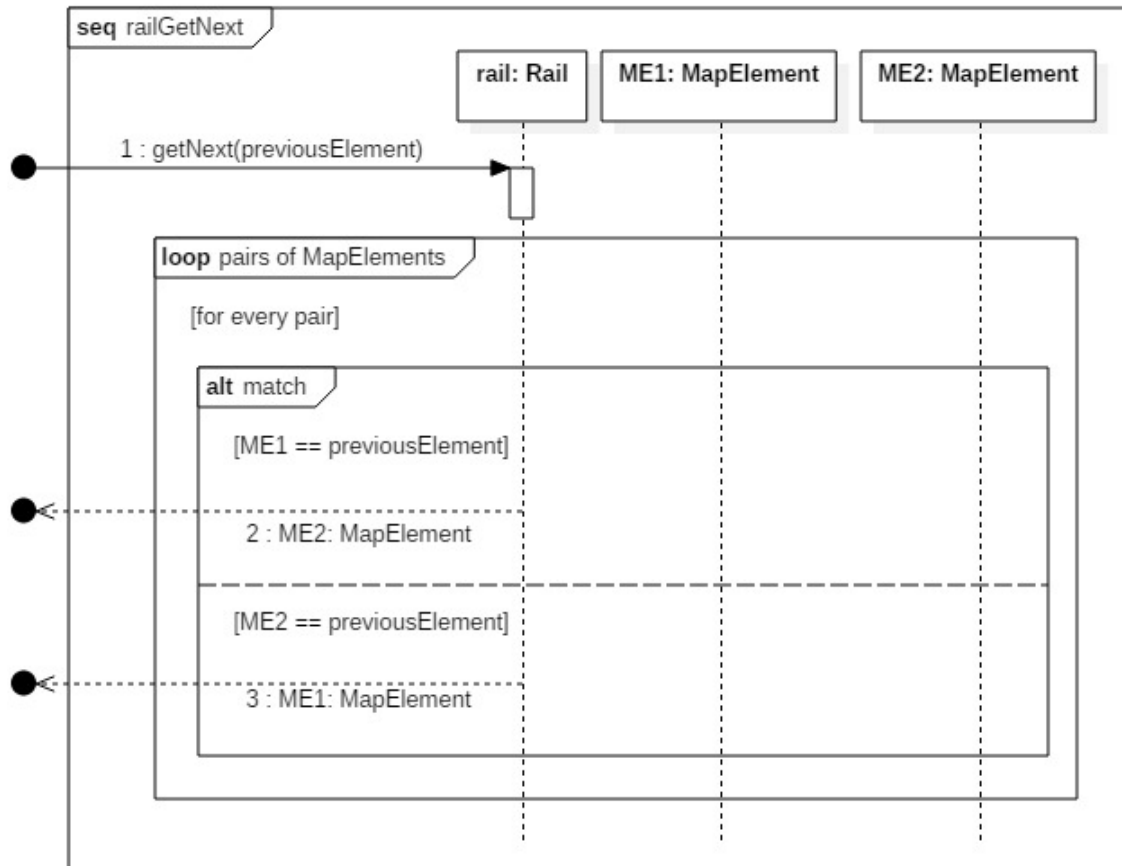
Az állomásra érkezéskor az alábbi lépések történnek:

1. Első nem üres kocsi megkeresése a leszállóknak
2. Ha nem üres az állomás, akkor az utasok felszállnak a megfelelő kocsira (ha tudnak)
3. A korábban kiszámolt első nem üres kocsi kiürítése
4. Annak eldöntése, hogy üres-e az egész vonat (minden nem szenes kocsi üres-e)



0.2.3 railGetNext szekvencia

Mivel az új modellben párokban tároljuk a lehetséges következő elemeket, elég végigmenni ezeken a párokon, és ha egyezik egy pár egyik tagja az előző elemmel, akkor a másik tag lesz a következő elem.



7. Prototípus koncepciója

7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A program alapvetően a szabványos be- és kimeneten keresztül kommunikál a felhasználóval, a következőkben leírt nyelven. A tesztek lefuttatásához a standard input és standard output átirányítására van szükség: az egyes tesztesetekhez külön-külön bemeneti fájl található, melyek tartalmazzák a teszt lefuttatásához szükséges parancsokat. A program a pályát minden esetben fájlból tölti be, ezek szerkezetének leírása a következő szakaszban található.

7.1.2 Bemeneti nyelv

init

Leírás: A pálya fájlból való betöltése és felépítése.

Opciók: -

save

Leírás: A pálya állapotának fájlba mentése.

Opciók: -

step

Leírás: A pályán lévő elemek léptetése, hatására minden vonat, kocsí a következő állapotába lép.

Opciók: -

setSwitch

Leírás: A felhasználó által megadott switch állásának megváltoztatása.

Opciók: A parancs után, a switch nevét, majd egy irányt megadva állíthatjuk be a kívánt váltót. Lehetséges irányok: N (north), S (south), E (east), W (west).

enableTunnel

Leírás: A felhasználó által megadott alagút engedélyezése.

Opciók: A felhasználó a parancs után, az alagút nevének megadásával tudja engedélyezni a kívánt alagutat.

disableTunnel

Leírás: A felhasználó által megadott alagút lezárása.

Opciók: A felhasználó a parancs után, az alagút nevének megadásával tudja engedélyezni a kívánt alagutat.

listTrains

Leírás: A pályán található vonatok kilistázása.

Opciók: -

listSwitches

Leírás: A pályán található váltók kilistázása.

Opciók: -

listStations

Leírás: A pályán található állomások kilistázása.

Opciók: -

listTunnels

Leírás: A pályán található alagutak kilistázása.

Opciók: -

spawnTrain

Leírás: Egy vonat elindítása a pálya széléről.

Opciók: A felhasználó eldöntheti, hogy a pálya melyik széléről szeretné indítani a vonatot. Például, ha a pályán két indulási pont van akkor 0-1 paraméterekkel választhat közülük.

7.1.3 Kimeneti nyelv

Az egyes elemeket a nevük egyértelműen meghatározza.

A kimenetek parancsok szerint:

- **init**
Az inicializálás sikeres/sikertelen volt.
- **spawnTrain**
A vonat megjelenítése sikeresen/sikertelenül lezajlott.
- **save**
A mentés sikeres/sikertelen volt.
- **step**
<lépés száma>. lépés.
- **setSwitch**
<switch neve> switch sikeresen/sikertelenül át lett állítva.
- **enableTunnel**
<tunnel neve> tunnel sikeresen/sikertelenül aktiválva lett.
- **disableTunnel**
<tunnel neve> tunnel sikeresen/sikertelenül deaktiválva lett.
- **listTrains**
<train neve> <pozíciója>
<hozzá tartozó carriage neve> <pozíciója> <színe> <üres-e>
...
- **listSwitches**
<switch neve> <pozíciója> <van-e rajta valami> <első bemenet> <második bemenet>
- **listStations**
<station neve> <pozíciója> <színe> <üres-e>
- **listTunnels**
<tunnel neve> <egyik bejárat pozíciója> <másik bejárat pozíciója> <van-e rajta valami>

7.2 Összes részletes use-case

Use-case neve	init
Rövid leírás	A pálya betöltése.
Aktorok	Tesztelő
Forgatókönyv	A parancs hatására egy fájlból betöltődik a pálya.

Use-case neve	save
Rövid leírás	A pálya elmentése.
Aktorok	Tesztelő
Forgatókönyv	A parancs hatására a pálya aktuális állása egy fájlba íródik ki.

Use-case neve	step
Rövid leírás	A pálya állapotának léptetése.
Aktorok	Tesztelő, System
Forgatókönyv	A parancs hatására a pályán lévő összes mozgatható elem a következő állapotába lép.

Use-case neve	setSwitch
Rövid leírás	Váltó állítása.
Aktorok	Tesztelő
Forgatókönyv	A parancs megfelelő paraméterezése esetén (váltó és irány megadásával) az adott váltó a felhasználó által megadott irányba áll, ha azt a pálya felépítése megengedi.

Use-case neve	enableTunnel
Rövid leírás	Alagút engedélyezése.
Aktorok	Tesztelő
Forgatókönyv	A parancs megfelelő paraméterezése esetén (az engedélyezni kívánt alagút megadásával) a választott alagút használhatóvá válik a vonatok számára. Mivel csak egy alagút lehet aktív egyszerre, ezért ha a választotton kívül egy másik alagút is aktív, akkor az automatikusan lezáródik.

Use-case neve	disableTunnel
Rövid leírás	Alagút lezárása.
Aktorok	Tesztelő
Forgatókönyv	A parancs megfelelő paraméterezése esetén (a lezárni kívánt alagút megadásával), a választott alagút lezáródik.

Use-case neve	listTrains
Rövid leírás	Vonatok kilistázása.
Aktorok	Tesztelő
Forgatókönyv	A parancs hatására megjelenik a pályán jelenleg

	tartózkodó vonatok és a hozzájuk tartozó kocsik listája, tulajdonságaikkal együtt.
--	--

Use-case neve	listSwitches
Rövid leírás	Váltók kilistázása.
Aktorok	Tesztelő
Forgatókönyv	A parancs hatására megjelenik a pályán található váltók listája, azok tulajdonságaival együtt.

Use-case neve	listTunnels
Rövid leírás	Alagutak kilistázása.
Aktorok	Tesztelő
Forgatókönyv	A parancs hatására megjelenik a pályán található alagutak listája, azok tulajdonságaival együtt.

Use-case neve	spawnTrain
Rövid leírás	Vonat indítása a pálya széléről.
Aktorok	Tesztelő
Forgatókönyv	A parancs és az indítás helyének megadásával egy vonat indul el az adott pozícióról.

7.3 Tesztelési terv

Teszt-eset neve	Vonat mozgatása sínen
Rövid leírás	Egy vonat a haladási irányának megfelelően átlép egy másik sín elemre.
Teszt célja	A vonat sínek közötti helyes lépésének tesztelése.

Teszt-eset neve	Vonat mozgatása alagútban
Rövid leírás	A vonatnak a létrehozott alagút bejáratánál be kell térnie az alagútba és itt végigmennie az alagutat tartalmazó sín elemeken.
Teszt célja	A vonat képes az alagútban a helyes haladásra.

Teszt-eset neve	Vonat mozgatása váltón
Rövid leírás	A vonat egy előre beállított váltóra lép, majd a beállított iránynak megfelelően lép tovább onnan.
Teszt célja	A vonat képes a váltóra lépni és onnan a helyes irányba továbbhaladni.

Teszt-eset neve	Váltó állítása
Rövid leírás	A váltó átállítása a különböző lehetséges haladási irányokba.
Teszt célja	A váltó helyes működésének tesztelése.

Teszt-eset neve	Játék vége ütközésnél
Rövid leírás	A játék véget ér ütközés esetén.
Teszt célja	Ha két vonat ütközik, akkor a játéknak vége. Ezen követelmény tesztelésének megvalósítása.

Teszt-eset neve	Játék vége rossz váltónál
Rövid leírás	A játék rossz váltónál is véget érhet.
Teszt célja	Mivel a vonat tkp. kisiklik, így a játék véget ér.

Teszt-eset neve	Alagút engedélyezése
Rövid leírás	Alagút építése a felhasználó igénye szerint.
Teszt célja	A pályán kijelölt helyekre alagutat lehet építeni. Ha egy alagút felépül, akkor egy másik megszűnik.

Teszt-eset neve	Szín meghatározás
Rövid leírás	Meg lehet-e határozni a MovingElement színét?
Teszt célja	Ezen teszt célja, hogy megnézzük, hogy helyesen adja-e vissza a ME a neki beállított szint, hogy ezt később felhasználhassuk.

Teszt-eset neve	Kocsi üresség
Rövid leírás	Annak a vizsgálata, hogy a kocsi üresség lekérdező függvénye jól működik-e.
Teszt célja	A kocsik ürességét kell ellenőrizni az állomásokon. Az üresség visszaadásáért felelős függvény helyes működése így létfontosságú.

Teszt-eset neve	Vonat megszüntetés
Rövid leírás	Mikor egy vonat kiürül, azaz az összes utas leszáll róla, és nincsenek az állomáson várakozó utasok, a vonatnak el kell tűnnie a pályáról.
Teszt célja	A vonat eltűnik a pályáról.

Teszt-eset neve	Utas leszállás
Rövid leírás	Utasok leszállása az állomáson.
Teszt célja	Ha a korábban leírt kritériumok teljesülnek, akkor az utasok leszállnak az állomásokon.

Teszt-eset neve	Utas felszállás
Rövid leírás	Utasok felszállása.
Teszt célja	Utasok felszállása egy üres kocsihoz az állomásról.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

7.4.1 batchTest

Szándékunkban áll használni egy batch fájlt, ami lefuttatja a programot minden teszt esetre egyszer, a megfelelő inputokra, majd a teszt eseteknek megfelelő output fájlokba írja a teszt esetek kimenetét.

7.4.2 Cél-ellenőrző

Előre definiáljuk, hogy mik az általunk elvárt kimeneti eredmények, majd ezeket összehasonlítjuk az előzőekben kapott output fájl(ok)ban található eredményekkel. Ha valahol eltérés van, akkor valamilyen szintű jelzést ad erről számunkra a tesztelést vezénylő egység. Esetleg a végén kiírja, hogy mennyi egyezés illetve eltérés volt a tesztelés folyamán. Ha az eltérések száma 0, akkor hibátlanul működik a programunk.

7.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2017. 03. 22.	1 óra	Dobó Fenes Papp Salamon Vizi	<p>Gyűlés, a változások átbeszélésre, megoldási javaslatok felvetése a probléma kezelésére. Feladatok kiosztása a hétre. Döntés:</p> <ul style="list-style-type: none"> ● Változások <ul style="list-style-type: none"> ○ neighbors: Map ○ szeneskocsi: fekete szín ○ felszállás: empty attribútum az állomásnak ● Feladatok <ul style="list-style-type: none"> ○ Dobó <ul style="list-style-type: none"> ■ 7.1.2 Bemeneti nyelv ■ 7.2 Összes részletes use-case ○ Fenes <ul style="list-style-type: none"> ■ 7.3 Teszteli terv ■ 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása ○ Papp <ul style="list-style-type: none"> ■ 7.1.1 Az interfész általános leírása ■ 7.3. Tesztelési terv (fele) ○ Salamon <ul style="list-style-type: none"> ■ Változások eszközölése ■ Kimeneti nyelv ○ Vizi <ul style="list-style-type: none"> ■ 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása ■ 7.2, 7.3 alpontoknál leírtak átnézése és a hiányosságok kiegészítése

2017. 03. 24. 13:00	3,5 óra	Salamon	A feladat változásaiból eredő változások eszközölése (0. pont).
2017.03.24 15:30	2 óra	Dobó	Bemeneti nyelv és a hozzájuk tartozó use-case leírások megfogalmazása.
2017.03.24 16:00	2 óra	Papp	Interfész általános leírása és dokumentáció inicializálása. Salamon munkájának ellenőrzése.
2017. 03. 24. 17:00	0,5 óra	Salamon	A kimeneti nyelv megírása (7.1.3).
2017.03.24 10:00	2 óra	Dobó	Hibák javítása, új parancsok hozzáadása a listához, majd a use-case leírások kiegészítése az előbbi változtatások szerint.
2017.03.24. 20:00	2 óra	Fenes	A teszt-esetek kigondolása, leírás készítése, dokumentumba írása
2017.03.25 10:00	2 óra	Papp	Tesztelési terv nevű alpont befejezése, a proto típus tervének átgondolása, hogy milyen változtatások szükségesek még.
2017.03.25 11:00	2 óra	Vizi	A Tesztelési terv(7.3) illetve az Összes részletes use-case(7.2) kiegészítése, előforduló hibák javítása.
2017.03.25 18:00	1,5 óra	Vizi	A tesztelést segítő, támogató segéd- és fordítóprogramok specifikálása.
2017.03.26 18:00	0,5 óra	Vizi	Dokumentáció rendbeszedése, nyomtatása.