

## 0. Változások

### 0.1 Kimeneti nyelv

A tesztelési nyelv kimenete több helyen módosult. Az új kimeneti nyelv az alábbi:

- **init**  
Az inicializálás sikeres/sikertelen volt.  
[Pálya mátrixos formája]
- **spawnTrain**  
A vonat megjelenítése sikeresen/sikertelenül lezajlott.
- **save**  
A mentés sikeres/sikertelen volt.
- **step**  
<lépés száma>. lépés.  
[Pálya mátrixos formája]
- **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>  
...  
Ha nincs vonat létrehozva, az alábbi kimenetet kapjuk: "Nincs vonat".
- **listSwitches**  
<switch neve> <pozíciója> <mi van rajta> <első bemenet> <második bemenet>  
Ha nincs váltó a pályán, az alábbi kimenetet kapjuk: "Nincs váltó a pályán".
- **listStations**  
<station neve> <pozíciója> <mi van rajta> <színe> <üres-e>  
Ha nincs állomás a pályán, az alábbi kimenetet kapjuk: "Nincs állomás a pályán".
- **listTunnels**  
<tunnel neve> <egyik bejárat pozíciója> <másik bejárat pozíciója> <engedélyezve van-e> <mi van rajta>  
Ha nincs alagút a pályán, az alábbi kimenetet kapjuk: "Nincs alagút a pályán".

Azoknál a részknél, ahol a [Pálya mátrixos formája] jelölés szerepel, ott a konzolon - a 8.2 szakasznak megfelelő - táblázatos formában megjelenik a pálya jelenlegi állása.

Azokon a helyeken, ahol <mi van rajta> szöveg kerül kiírásra, és a MapElement-en nem tartózkodik MovingElement, "nincs rajta elem" felirat jelenik meg.

Ha a játék a játékos hibájából ér véget, akkor az "A vonat felrobbant, a játéknak vége." felirat jelenik meg.

## 0.2 Bemeneti nyelv

A bemeneti nyelv annyiban módosult, hogy az init parancs paraméterként várja a betöltendő pálya nevét, valamint hozzáadásra került egy:

- **setEmptyness**  
setEmptyness carriage1 true  
setEmptyness <carriage neve> <érték>
- spawnTrain véletlenszerűen generál egy vonatot és kocsikat hozzá.
- spawnTrain <1.kocsi színe> <2.kocsi színe>...<i.kocsi színe>

nevű parancs.

## 0.3 Osztálydiagram

Megjelent a szeneskocsit szimbolizáló CoalCarriage osztály, illetve a Carriage kiegészült egy konstruktorral.

A MapElement neighbors attribútuma Pair-ekből álló ArrayList lett, mivel az könnyebben kezelhető, mint a Map típus.

A Switch-nél megjelent egy új attribútum, ami azt adja meg, hogy jelenleg melyik Pair aktív (azaz hogy milyen previousElement-re milyen MapElement-tel kell visszatérnie).

## 0.4 Pályafájlok felépítése

Az előző heti munkánkból kamaradt a pályákat tartalmazó fájlok felépítése. Ezt az alábbi rész definiálja.

Minden pálya külön fájlban van eltárolva.

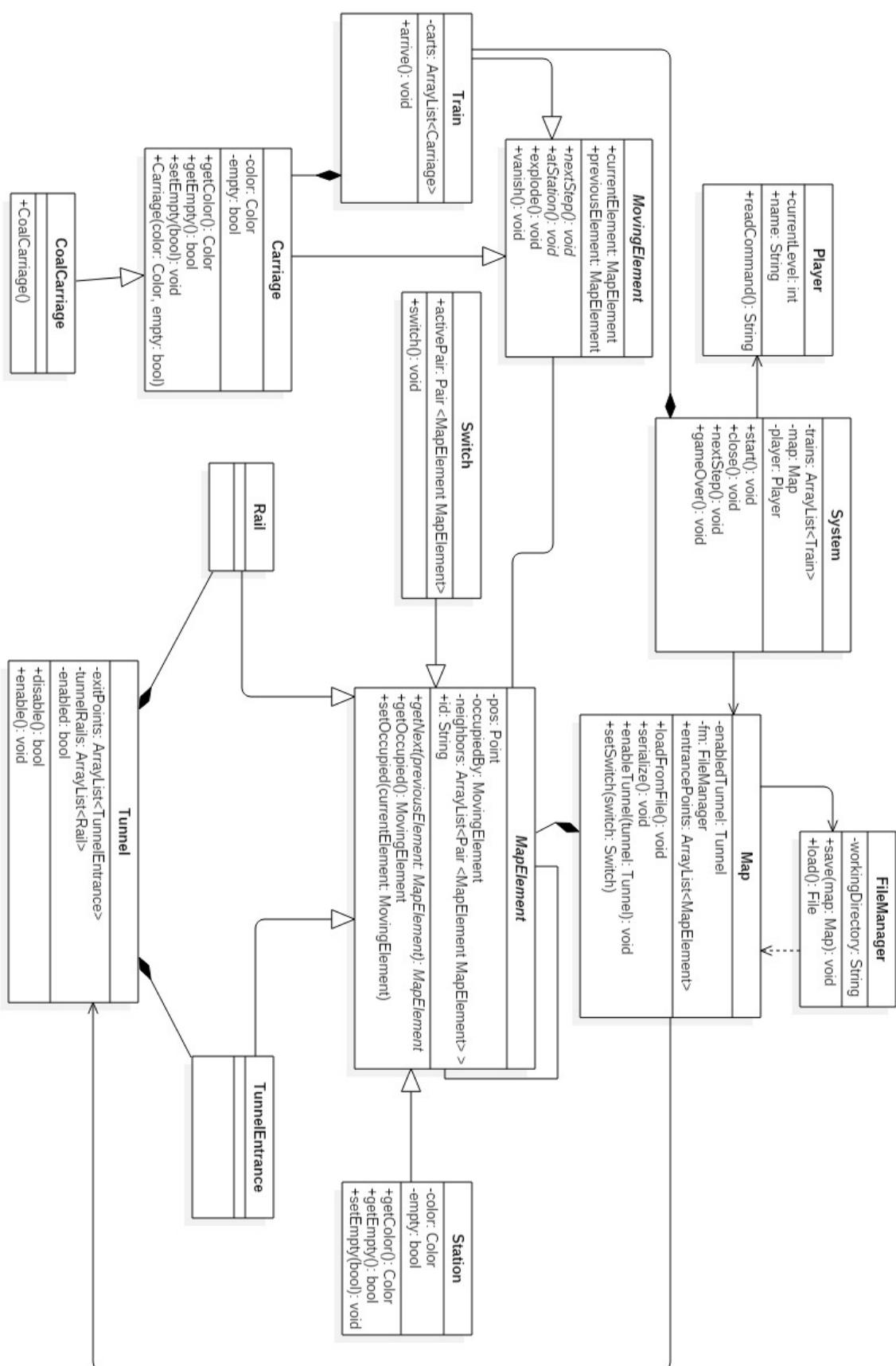
Az állomány szerkezetét talán egy példa szemlélteti a legjobban:

```
#Ez a fajl a peldapalyat tartalmazza.
max_x: "4"
max_y: "4"
x: "1" y: "1" <rail>
x: "1" y: "2" <>
x: "1" y: "3" <>
x: "1" y: "4" <rail>
x: "2" y: "1" <rail>
x: "2" y: "2" <station,blue>
x: "2" y: "3" <rail>
x: "2" y: "4" <switch>
x: "3" y: "1" <tunnelentrance>
x: "3" y: "2" <tunnelrail>
x: "3" y: "3" <tunnelrail>
x: "3" y: "4" <tunnelentrance>
x: "4" y: "1" <rail>
x: "4" y: "2" <rail>
x: "4" y: "3" <rail>
x: "4" y: "4" <rail>
```

A fenti fájl az alábbi pályát definiálja (lásd még: 8.2 szakasz):

y\x	1	2	3	4
1	r	r	te	r
2		s	tr	r
3		r	tr	r
4	r	sw	te	r

## **0.5 Módosult osztálydiagram**



## 8. Részletes tervez

### 8.1 Osztályok és metódusok tervez

#### 8.1.1 Map

- **Felelősség**

A pályaelemek tárolásáért illetve az ezekkel kapcsolatos műveletek elvégzéséért felelős.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**

- **-FileManager fm:** a FileManager osztály példánya.
- **+ArrayList<MapElement> entrancePoints:** lista, ami MapElement típusú elemeket tartalmaz
- **-Tunnel enabledTunnel:** a jelenleg megépített egyetlen alagút.

- **Metódusok**

- **+void loadFromFile():** Pályaelemek beolvasása a megadott fájlból.
- **+void serialize():** A System által utasított függvény a pálya jelenlegi állásának elmentésére. Meghívja a FileManager fájlba író save() függvényét.
- **+void enableTunnel (tunnel: TunnelEntrance):** A jelenleg engedélyezett alagutat lehet vele állítani, felülírja a jelenlegit, ezáltal biztosítva, hogy egyszerre csak egy van engedélyezve.
- **+setSwitch(switch: Switch):** Meghívja a paraméterül kapott Switch switch() függvényét.

#### 8.1.2 MovingElement

- **Felelősség**

A pálya mozgó elemeinek interfésze. Tudnia kell az időbeli előrehaladást kezelnie, erre szolgál a nextStep() függvény. Ez az osztály felelős továbbá a MovingElementek megszüntetéséért és felrobbanásáért.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**

- **+MapElement currentElement:** Az az elem, amin jelenleg tartózkodik.

- **+MapElement previousElement:** Az az elem, amin ezelőtt tartózkodott.
- **Metódusok**
  - **+void nextStep():** A léptetést kezelő metódus. Kiszámoltatja a következő pozíciót, a következő irányt.
  - **+void vanish():** Ha egy vonat kiürül, akkor eltűnik. Ezt valósítja meg. Megszünteti a jelenlegi vonat objektumot és a hozzá tartozó kocsikat.
  - **+void explode():** Ha egy vonat ütközik egy másikkal, vagy egy rosszul beállított switchnél nem tud továbbmenni (tehát kisiklana), akkor a mozdony és a kocsik felrobbannak. Ezt valósítja meg ez a metódus.

### 8.1.3 FileManager

- **Felelősség**

A játékban a fájlkezelést végzi. Kezeli a fájlok tárolásának helyét, jellegét, a Map szerializációs függvényeivel szoros összefüggésben áll, hiszen a szerializált adatot ez az osztály írja ki fájlba.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **String workingDirectory:** a lementett fájlok helyét tároló karakterfüzér.

- **Metódusok**

- **void save(Map):** a paraméterként kapott állást lementi fájlba.

- **File load():** a lementett állást visszatölteni, hogy a Map majd tudjon vele dolgozni.

### 8.1.4 Train

- **Felelősség**

A vonatok modellezését megvalósító osztály.

- **Ősosztályok**

MovingElement

- **Interfészek**

Nincs.

- **Attribútumok**

- **-ArrayList<Carriage> carts:** A vonathoz csatlakoztatott kocsik vannak benne. eltárolva.

- **Metódusok**

- **+void arrive():** A vonat állomásra történő megérkezését végzi. Azon követelmények és szabályok beteljesülését vizsgálja, amelyeket a funkcionális követelmények a vonatra és állomásra szabnak.

### 8.1.5 System

- **Felelősség**

A játék különböző komponenseit összefogó elem, mely elvégzi az indítást, bezárást, lépések ütemezését és a játékmenetet összefogja.

- **Ősosztályok**

Nincsenek.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **-ArrayList<Train> trains:** A játékban szereplő vonatok.
- **-Map map:** A jelenleg betöltött pálya.
- **-Player player:** A játékosról tárolt információkat tartalmazza.

- **Metódusok**

- **+void start():** A játék inicializálásáért felelős, létrehozza a pályát és feltölti elemekkel. Első lépésként, ahogy említettem létrehoz egy Map objektumot és eltárolja azt a map nevű attribútumában. Ennek során meghívódik a map loadFromFile() függvénye, bár ennek meghívása nem a start() feladata, csak annak hatása. Miután a map függvénye visszatér, a start() elvégezte a dolgát.
- **+void close():** A játékállás elmentését végzi, meghívja a pálya sorosító függvényét. Meghívja az attribútumként eltárolt Map objektum serialize() függvényét, ami elvégzi a fájlba történő sorosítást. A fájlkezelés során keletkezett kivételeket is a close() kezeli, a hibaüzenet felhasználóbarát megjelenítésével.
- **+void nextStep():** Az ütközés ellenőrzésének meghívását végzi, valamint a mozgó elemek pályán történő mozgatását, úgymond az ütemet biztosítja a rendszerben. Meghívja az eltárolt trains ArrayList minden elemére a nextStep függvényt, ezáltal egy ütemet biztosítva a játéknak, majd ezt követően beolvassa a felhasználótól a parancsot és a kívánt utasítást végrehajtja (pl.: setSwitch esetén váltó állítása a MapElement és Map osztályon keresztül a megfelelő id megadásával vagy alagút építése során meghívja a Map enableTunnel függvényét a tárolóból történő kikeresés után, ami szintén az id alapján történik).
- **+void gameOver():** Ha meghívta valaki, akkor véget ér a játék, ezt közöljük a felhasználóval és elindítjuk a fentiekben tárgyalt close() függvényt.

### 8.1.6 Player

- **Felelősség**

A játékos programbeli megnevezése. Ő kérdezi meg a parancsokat a rendszer (System) számára, így irányítva a játék működését. Lehetősége van a váltók állítgatására az ütközések elkerülése érdekében, valamint alagút elhelyezésére az erre kijelölt helyeken a megfelelő parancsok továbbításával a System osztályon keresztül..

- **Ősosztályok**

Nincsenek.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **+int currentLevel:** A játékos jelenlegi szintje. Ez hatással lesz a vonatok mozgási sebességére.
- **+String name:** A játékos neve. Tulajdonképpen lényegi jelentősége nincsen, csak ha szeretnénk megszólítani a felhasználót hasznos lehet.

- **Metódusok**

- **String readCommand():** Beolvassa a standard inputról a parancsot, ha volt. Ha ez érvényes, akkor visszaadja azt a Systemnek, egyéb esetben, vagyis érvénytelen vagy üres parancs megadása során egy üres Stringgel tér vissza. Egy parancs érvényes, ha az a bemeneti nyelven lett megfogalmazva.

### 8.1.7 Carriage

- **Felelősség**

A szerelvényhez tartozó kocsi. Van színe, ami az utasok le- és felszállításához kell, illetve tárolja, hogy üres-e. A Train osztály tartalmazza a hozzá tartozó kocsikat.

- **Ősosztályok**

MovingElement

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **-Color color:** A kocsi színe. Szükség van rá az utasok leszállításánál, felszállításánál.
- **-Boolean empty:** Azt mondja meg, hogy üres-e a kocsi.

- **Metódusok**

- **+Carriage(color: Color, empty: Boolean):** A kocsi konstruktora. Az objektumot a paraméterként kapott értékekkel inicializálja.
- **+Color getColor():** Visszaadja a kocsi jelenlegi színét.
- **+Boolean getEmpty():** Visszaadja, hogy jelenleg üres-e a kocsi.

- **+void setEmpty(empty: Boolean):** A paraméterként kapott értéktől függően vagy “kiüríti”, vagy “feltölti” a kocsit.

### 8.1.8 CoalCarriage

- **Felelősség**

Speciális kocsi, ami szenet szállít, de utast nem. Mindig telinek tekintendő. Konstruktorában megkapja a tuljadonságait. A setter függvényei nincsenek implementálva, mivel a szeneskocsi színét (fekete) és telítettségét (teli) a játék során nincs mód változtatni.

- **Ősosztályok**

Carriage

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **-Color color:** A kocsi színe. Mindig fekete.
- **-Boolean empty:** Azt mondja meg, hogy üres-e a kocsi. Értéke minden hamis.

- **Metódusok**

- **+CoalCarriage():** a szeneskocsi konstruktora. Inicializálja az objektumot a megfelelő adatokkal: a szín fekete, a kocsi tele van.
- **+Color getColor():** Visszaadja a kocsi jelenlegi színét, ami jelen esetben a fekete.
- **+Boolean getEmpty():** Visszaadja, hogy jelenleg üres-e a kocsi. False értékkal tér vissza.
- **+void setEmpty(empty: Boolean):** Nincs implementálva.

### 8.1.9 Tunnel

- **Felelősség**

A pályán található alagutakat reprezentáló osztály. Két összetevője a bejáratai és a benne található sínek. Tárolja, hogy jelenleg engedélyezve van-e. Adott esetben le lehet tiltani, és engedélyezni is lehet.

- **Ősosztályok**

Nincsenek.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **-ArrayList<TunnelEntrance> exitPoints:** Az alagút bejáratit ponjainak tárolója.
- **-ArrayList<Rail> tunnelRails:** Az alagútban futó sínek tárolója.
- **-Boolean enabled:** Azt adja meg, hogy az alagút jelenleg engedélyezve van-e.

- **Metódusok**

- **+void disable():** Megnézi, hogy van-e MovingElement a jelenleg engedélyezett

alagútban található síneken, illetve az alagút bejáratain.

Erre azért van szükség, mert ha az alagút nem üres, akkor nem lehet letiltani.

Ha le lehet tiltani az alagutat, akkor ezt megeszi (false-ra állítja az enabled bool-t) és true értékkel tér vissza. Sikertelen állítás esetén a visszatérési érték false.

- **+void enable():** Átállítja az enabled attribútum értékét igazra.

### 8.1.10 MapElement

- **Felelősség**

A pályán található elemeket reprezentáló osztály. Ez az űosztálya az összes pályaelemnek, biztosítja a mozgatás alapját. A felelőssége továbbá, hogy nyílvántartja, hogy az adott pályaelemen melyik, MovingElement tartózkodik és, hogy a pályaelemek milyen szomszédai vannak.

- **Ősosztályok**

Nincsenek.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **-Point pos:** A pályaelem pozíciója.
- **-MovingElement occupiedBy:** A pályaelemen lévő MovingElement.
- **-Map<MapElement> neighbors:** Az adott pályaelem szomszédainak a tárolója.
- **+String id:** Az adott pályaelem azonosítója.

- **Metódusok**

- **+MapElement getNext(previousElement: MapElement):** Absztrakt metódus, amit a leszármazottak funkcionálisuknak megfelelően valósítanak meg. A metódus visszatér azzal a pályaelemmel, amire a MovingElementnek következőnek lépnie kell. A MovingElement nextStep() függvényében hívódik ez a metódus, paramétereként azt a pályaelement kapja, ahonnan az adott MovingElement érkezett, így tudja megmondani, hogy mi lesz a következő elem, azáltal, hogy a szomszédainak a listájából kiválasztja a megfelelő MapElementet.
- **+MovingElement getOccupied():** A metódus visszaadja az occupiedBy attribútum értékét, azaz a pályaelementen található MovingElementet.
- **+void setOccupied(currentElement: MovingElement):** A metódussal az adott pályaelemre beállíthatunk egy MovingElementet a függvény paramétere segítségével. Ekkor a függvény az occupiedBy attribútum értékét változtatja meg.

### 8.1.11 Station

- **Felelősség**

A feladat kiírásból származó állomás modellezését megvalósító osztály.

- **Ősosztályok**

MapElement.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **-Color color:** az állomás színe
- **-Boolean empty:** jelzi, hogy üres-e az állomás.

- **Metódusok**

- **+getEmpty():** Visszaadja, hogy üres-e az állomás.
- **+setEmpty(bool):** Beállítja, hogy üres-e az állomás. Amennyiben üres, true-ra, ha nem üres false-ra állítja.

### 8.1.12 TunnelEntrance

- **Felelősség**

Az alagút ki és bejáratát modellez. Hasonló a switchez, egy alagútnak két bejárata van.

- **Ősosztályok**

MapElement.

- **Interfészek**

Nincsenek.

- **Attribútumok**

Nincsenek.

- **Metódusok**

- **+MapElement getNext(previousElement: MapElement):** Az ősosztályában deklarált metódus megvalósítása. Itt egyszerűen a paraméter segítségével kiválasztja a szomszédai közül, hogy melyik MapElementre kell a MovingElementnek továbbhaladnia.

### 8.1.13 Rail

- **Felelősség**

A síneket reprezentáló osztály. Felelőssége, hogy a rajta haladó MovingElementeknek, meg tudja határozni a következő helyzetét.

- **Ősosztályok**

MapElement

- **Interfészek**

Nincsenek.

- **Attribútumok**

Nincsenek.

- **Metódusok**

- **+MapElement getNext(previousElement: MapElement):** Az ōsosztályában deklarált metódus megvalósítása. Itt egyszerűen a paraméter segítségével kiválasztja a szomszédai közül, hogy melyik MapElementre kell a MovingElementnek továbbhaladnia.

### 8.1.14 Switch

- **Felelősség**

A váltókat reprezentáló osztály. Különbsége a síntől az, hogy egy bemenete van, de különböző állásai alapján több kimenete lehet. Felelőssége, hogy meg tudja határozni, hogy a váltó állása alapján, melyik lesz a MovingElement számára a következő elem.

- **Ósosztályok**

MapElement

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **-String state:** A váltó jelenlegi állása. (N (north), E (east), S (south), W (west))

- **Metódusok**

- **+MapElement getNext(previousElement: MapElement):** Az ōsosztályában deklarált metódus megvalósítása. A váltó itt az állása alapján kiválasztja a szomszédai közül, hogy melyik MapElementre kell a MovingElementnek továbbhaladnia.
- **+switch():** A metódus az állapotok közötti váltást végzi, ezt a Map setSwitch függvénye hívja meg. Az állapotok közötti váltás megadott sorrendben történik. (pl. N->E->S->W->N...)

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

A teszesetek egy fájlból beolvasott pálya inicializálásával kezdődnek. A tesztpályák 4x4-es méretűek, bennük a pozíció x és y koordinákkal jelenik meg (pl. az x=2, y=3 pozícióban található elem helye: 2,3). A táblázatban szerepelnek a pályán található elemek betűjelei, amik az alábbiak:

- r: Rail
- te: TunnelEntrance
- tr: TunnelRail
- sw: Switch
- s: Station
- T: Train
- C: Carriage
- üres: abban a pozícióban nem található elem

Azokon a helyeken, ahol egyszerre van egy pályaelem, és egy mozgó elem, egyenes elválasztót használunk ( | ). Például ha egy vonat egy TunnelEntrance elemen tartózkodik, akkor a megfelelő cellában az alábbi jelölés látható: te | T.

A pályára érkező vonatok mindig az 1,1 pozícióba érkeznek, és jobbra haladnak.

A pálya az alábbi táblázatnak felel meg, amiben most elemek helyett a pozíciók jelölése látható:

y\x	1	2	3	4
1	1,1	2,1	...	
2	1,2	...		
3	...			
4				

### 8.2.1 Alagút engedélyezése

y\x	1	2	3	4
1	r	te		
2		tr		te
3		te		tr
4		r		te

- **Leírás**

A felhasználó építhet a pályán meghatározott helyekre alagutakat. Egyszerre csak egy lehet felépítve.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ellenőrzött funkcionalitás az alagút engedélyezéséhez kapcsolódik, vagyis ha egyet megépítünk és már van egy másik építve, akkor az lerombolódik. Hiba az érvénytelen bemeneten túl az előbbi esetben léphet fel, ha a lerombolni kívánt alagútban vonat van. Ekkor nem történik meg az új alagút megépítése.

- **Bemenet**

init enableTunnel

listTunnels

enableTunnel tunnel1

- **Elvárt kimenet**

Az inicializálás sikeres volt.

```
<tunnel1> <2,1> <2,3> <false>
<tunnel2> <4,2> <4,4> <false>
```

<tunnel1> tunnel sikeresen aktiválva lett.

### 8.2.2 Szín meghatározás

y\x	1	2	3	4
1	r c	r c	r T	r
2	r	r	s	r
3				
4				

- **Leírás**

Megtekinthetjük a kocsik, állomások színét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük a kocsik és állomások színének lekérdezését. Hiba akkor léphet fel, ha ezek nem lettek megfelelően inicializálva.

- **Bemenet**

init szin

spawnTrain

listTrains

listStations

- **Elvárt kimenet**

Az inicializálás sikeres volt.

```
<train1> <3,2>
<carriage1> <2,2> <blue> <true>
<carriage2> <1,2> <yellow> <false>

<station1> <3,2> <red> <false>
```

### 8.2.3 Kocsi üresség

y\x	1	2	3	4
1	c	c	c	t
2				
3				
4				

- **Leírás**

A kocsik ürességét kell ellenőrizni az állomásokon. Az üresség visszaadásáért felelős függvény helyes működése így létfontosságú. Ezt teszteljük.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hiba léphet fel érvénytelen bemenet vagy azonosító megadása esetén. Az ellenőrzött funkcionalitás nem más mint a kocsik ürességének állítása és lekérdezése.

- **Bemenet**

init kocsiUresseg

spawnTrain

setEmptyness carriage1 true

setEmptyness carriage2 false

setEmptyness carriage3 true

listTrains

- **Elvárt kimenet**

Az inicializálás sikeres volt.

```
<train1><4,2>
<carriage1><3,2> <blue> <true>
<carriage2><2,2> <red> <false>
<carriage3><1,2> <green> <true>
```

### 8.2.4 Utasok leszállása (a vonat nem tűnik el)

- **Leírás**

Menet közben a vonat egy állomáshoz érkezik, ahol az utasok a szerelvény valamelyik kocsijáról leszállnak (mivel egyezik az adott nem üres kocsi színe az állomás színével). A vonaton marad még szerelvény, szóval nem tűnik el.

Az ellenőrzés menete: az állomásra érkezés előtt és után is kilistázzuk a vonat kocsijait, így könnyen észrevehető a változás.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszeset az utasok leszállításának logikáját ellenőrzi, ami a Train osztály atStation metódusában van megvalósítva.

Hiba lehet, ha több kocsiról szállnak le utasok, ha egyáltalán nem szállnak le utasok, vagy ha a leszállás után a vonat eltűnik.

- **Bemenet**

```
init utasokLe1
spawnTrain <red> <green>
listTrains
listStations
step
listTrains
```

- **Elvárt kimenet**

Az inicializálás sikeres volt.

y\x	1	2	3	4
1	s	r		
2				
3				
4				

A vonat megjelenítése sikeresen lezajlott.

```
<train1><nincs a pályán>
<carriage1><nincs a pályán> <red> <nem üres>
<carriage2><nincs a pályán> <green> <nem üres>

<station1><1,1> <nincs rajta elem> <zöld> <üres>
```

1. lépés.

y\x	1	2	3	4
1	s   T	r		
2				
3				
4				

<train1> <1,1>  
<carriage1> <nincs a pályán> <piros> <nem üres>  
<carriage2> <nincs a pályán> <zöld> <üres>

### 8.2.5 Utasok leszállása (a vonat eltűnik)

- Leírás

Menet közben a vonat egy állomáshoz érkezik, ahol az utasok a szerelvény valamelyik kocsijáról leszállnak (mivel egyezik az adott nem üres kocsi színe az állomás színével). A vonaton nem marad több teljes szerelvény, így az eltűnik.

Az ellenőrzés menete: az állomásra érkezés előtt és után is kilistázzuk a vonatok listáját. Az állomás elhagyása után a listának üresnek kell lennie (hiszen eltűnt a vonat).

- Ellenőrzött funkcionalitás, várható hibahelyek

A teszeset az utasok leszállításának logikáját ellenőrzi, ami a Train osztály atStation metódusában van megvalósítva.

Hiba lehet, ha több kocsiról szállnak le utasok, ha egyáltalán nem szállnak le utasok, vagy ha a leszállás után a vonat nem tűnik el.

- Bemenet

```
init utasokLe2
spawnTrain <red> <green>
listTrains
listStations
step
listTrains
```

- Elvárt kimenet

Az inicializálás sikeres volt.

y\x	1	2	3	4
1	s	r		
2				
3				
4				

A vonat megjelenítése sikeresen lezajlott.

```
<train1> <nincs a pályán>
<carriage1> <nincs a pályán> <red> <nem üres>
<carriage2> <nincs a pályán> <green> <üres>

<station1> <1,1> <nincs rajta elem> <piros> <üres>
```

1. lépés.

y\x	1	2	3	4
1	s	r		
2				
3				
4				

Nincs vonat a pályán.

## 8.2.6 Utasok nem szállnak le

- Leírás

Menet közben a vonat egy állomáshoz érkezik, ahol az utasok a szerelvény egyik kocsijáról sem szállnak le (mivel nincs egyezés teli koci és az állomás színe között). A vonaton így biztos marad teli szerelvény, szóval nem tűnik el.

Az ellenőrzés menete: az állomásra érkezés előtt és után is kilistázzuk a vonat kocsijait, így könnyen észrevehető, hogy nem változott a kocsik felépítése.

- Ellenőrzött funkcionalitás, várható hibahelyek

A teszeset az utasok leszállításának logikáját ellenőrzi, ami a Train osztály atStation metódusában van megvalósítva.

Hiba lehet, ha leszállnak utasok, vagy ha az állomás elhagyása után a vonat eltűnik.

- Bemenet

```
init utasokLe3
spawnTrain <red> <green>
listTrains
listStations
step
listTrains
```

- Elvárt kimenet

Az inicializálás sikeres volt.

y\x	1	2	3	4
1	s	r		
2				
3				

4				
---	--	--	--	--

A vonat megjelenítése sikeresen lezajlott.

```
<train1> <nincs a pályán>
<carriage1> <nincs a pályán> <red> <nem üres>
<carriage2> <nincs a pályán> <green> <nem üres>

<station1> <1,1> <nincs rajta elem> <kék> <üres>
```

1. lépés.

y\x	1	2	3	4
1	s   T	r		
2				
3				
4				

```
<train1> <1,1>
<carriage1> <nincs a pályán> <piros> <nem üres>
<carriage2> <nincs a pályán> <zöld> <nem üres>
```

## 8.2.7 Utasok felszállása

- **Leírás**

Menet közben a vonat egy állomáshoz érkezik, ahol az állomáson utasok várakoznak, akik fel szeretnének szállni a vonatra. Csak abban az esetben tudnak felszállni a vonatra, ha a kocsik között található üres kocsi. Amennyiben több üres kocsi is van, a várakozó utasok a vonat elejétől számított legelső üres kocsiba szállnak fel (feltételezzük, hogy minden felférnek egy kocsiba). Mivel egrészt csak üres kocsikkal rendelkező vonat nem érkezhet az állomásra, mivel az a vonat amelyiknek minden kocsija üres már megszűnik, így biztosan van legalább egy darab nem üres kocsija. Így az állomáson áthaladás után egyel több nem üres kocsija lesz, szóval a vonat nem tűnik el.

Az ellenőrzés menete: Az állomásra érkezés előtt és után is kilistázzuk a vonat kocsijait, így könnyen észrevehető a változás azáltal, hogy megnézzük, hogy az az első kocsi ami az állomás előtt üres volt, az állomás után már nem üres állapotba került.

- **Ellenőrzött funkcionálitás, várható hibahelyek**

A teszeset az utasok felszállásának logikáját ellenőrzi, ami a Train osztály atStation metódusában van megvalósítva.

Hiba lehet ha az utasok olyan kocsiba szállnak fel, ami nem üres, vagy ha a felszállás után a vonatunk mégis eltűnik.

● **Bemenet**

```
initUtasokFelszallasa
spawnTrain
listTrains
step
listTrains
```

● **Elvárt kimenet**

1. lépés.

Az inicializálás sikeres volt.

y\x	1	2	3	4
1	s	r	r	
2				
3				
4				

<train1> <nincs a pályán>

<carriage1> <nincs a pályán> <kék> <üres>

<carriage2> <nincs a pályán> <piros> <nem üres>

<station1> <1,1> <nincs rajta elem> <kék> <nem üres>

2. lépés.

A vonat megjelenítése sikeresen lezajlott.

y\x	1	2	3	4
1	s T	r	r	
2				
3				
4				

<train1> <1,1>

<carriage1> <nincs a pályán> <kék> <üres>

<carriage2> <nincs a pályán> <piros> <nem üres>

<station1> <1,1> <train1> <kék> <nem üres>

3. lépés.

Az első kocsi megjelenítése sikeresen lezajlott.

y\x	1	2	3	4
1	s C	r T	r	
2				
3				
4				

<train1> <2,1>

<carriage1> <1,1> <kék> <üres>

<carriage2> <nincs a pályán> <piros> <nem üres>

<station1> <1,1> <carriage1> <kék> <nem üres>

4. lépés.

A második kocsi megjelenítése is sikerült, valamint az első kocsi üres állapotból nem üres állapotba váltott, valamint az állomás nem üres állapotból üres állapotba ment át.

y\x	1	2	3	4
1	s C	r C	r T	
2				
3				
4				

<train1> <3,1>

<carriage1> <2,1> <kék> <nem üres>

<carriage2> <1,1> <piros> <nem üres>

<station1> <1,1> <carriage2> <kék> <üres>

## 8.2.8 Vonat megszűntetése

- Leírás

A vonat megszűntetése abban az esetben hajtódik végre, vagyis következik be, amikor a vonat

végighalad egy állomáson, ahol az utolsó nem üres kocsijából is leszállnak az utasok. Ugyanis ütközések nél a vonatok nem megszűnnék, hanem felrobbannak, ezáltal a játék végét okozva.

Az ellenőrzés menete: Az állomásra érkezés előtt és után is kilistázzuk a vonatok listáját. Így meg tudjuk figyelni, hogy melyek azok a vonatok, amelyek már csak egy darab nem üres kocsival rendelkeznek. Majd az állomáson való áthaladás során megfigyeljük, hogy az utolsó nem üres kocsi színe meg egyezik-e az állomás színével. Amennyiben igen, az utasok leszállhatnak a kocsiból. Ezáltal a vonat minden kocsija kiürül, így a vonatnak már nem kell megjelennie az állomás utáni listázás folyamán.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszeset alapvetően a vonatok megszüntetését (eltünését) ellenőrzi, amit a Train osztály vanish metódusa valósít meg.

Hiba lehet, ha az utolsó nem üres kocsi színe megegyezik az állomás színével, de az utasok mégsem szállnak le a vonatról, vagy ha az utasok leszállnak, azonban az állomás után a vonat továbbra is látható a listában, vagyis továbbhalad csupa üres kocsit húzva maga után.

- **Bemenet**

```
init
spawnTrain
listTrains
step
listTrains
```

- **Elvárt kimenet**

1. lépés.

Az inicializálás sikeres volt.

y\x	1	2	3	4
1	s	r	r	
2				
3				
4				

<train1> <nincs a pályán>

<carriage1> <nincs a pályán> <kék> <üres>

<carriage2> <nincs a pályán> <piros> <nem üres>

<station1> <1,1> <nincs rajta elem> <kék> <üres>

2. lépés.

A vonat megjelenítése sikeresen lezajlott.

y\x	1	2	3	4

1	s T	r	r	
2				
3				
4				

<train1> <1,1>  
 <carriage1> <nincs a pályán> <kék> <üres>  
 <carriage2> <nincs a pályán> <piros> <nem üres>  
  
 <station1> <1,1> <train1> <piros> <üres>

3. lépés.  
 Az első kocsi megjelenítése is sikeresen lezajlott.

y\x	1	2	3	4
1	s C	r T	r	
2				
3				
4				

<train1> <2,1>  
 <carriage1> <1,1> <kék> <üres>  
 <carriage2> <nincs a pályán> <piros> <nem üres>  
  
 <station1> <1,1> <carriage1> <piros> <üres>

4. lépés.  
 A második kocsi megjelenítése is sikeres volt.

y\x	1	2	3	4
1	s C	r C	r T	
2				
3				
4				

<train1> <3,1>

```
<carriage1> <2,1> <kék> <üres>
<carriage2> <1,1> <piros> <nem üres>

<station1> <1,1> <carriage2> <piros> <üres>
```

5. lépés.

Mivel az állomás és a második kocsi színe megegyezik, és az a vonat elejétől számított első kocsi, amelyik azonos színű és nem üres, ezért az állomáson való áthaladás során a kocsiból leszállnak az emberek. Így mivel ez volt az utolsó nem üres kocsija a vonatnak, a vonatunk köddé válik (vagyis eltűnik a pályáról).

y\x	1	2	3	4
1	s	r	r	
2				
3				
4				

```
<train1> <nincs a pályán>
<carriage1> <nincs a pályán> <kék> <üres>
<carriage2> <nincs a pályán> <piros> <üres>

<station1> <1,1> <nincs rajta elem> <piros> <üres>
```

### 8.2.9 Vonat mozgatása sínen

- **Leírás**

A vonat létfontosságú tulajdonsága, hogy képes a pályán a mozgásra. Ennek a tesztnak a célja, hogy belássuk, hogy a vonat képes a sínen mozogni. A tesztpálya csak 2 darab összekapcsolt sínből áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt ellenőrzi, hogy a vonat képes-e a sínen továbblépni.

Hiba lehet, ha a vonat nem lép a következő sínrre, pedig az lenne az elvárt.

- **Bemenet**

```
init vonat_mozg_sinen
spawnTrain
listTrains
step
listTrains
step
listTrains
```

- **Elvárt kimenet**

Az inicializálás sikeres volt.

y\x	1	2	3	4
1	r	r		
2				
3				
4				

A vonat megjelenítése sikeresen lezajlott.

<train1> <nincs a pályán>  
<carriage1> <nincs a pályán> <piros> <nem üres>

1. lépés.

y\x	1	2	3	4
1	r T	r		
2				
3				
4				

<train1> <1,1>  
<carriage1> <nincs a pályán> <piros> <nem üres>

2. lépés.

y\x	1	2	3	4
1	r C	r T		
2				
3				
4				

<train1> <2,1>  
<carriage1> <1,1> <piros> <nem üres>

### 8.2.10 Vonat mozgatása alagútban

- Leírás

A vonat létfontosságú tulajdonsága, hogy képes a pályán a mozgásra. Ennek a tesztnak a célja, hogy belássuk, hogy a vonat képes az alagútban mozogni. A pálya úgy épül fel hogy van egy sín amin a vonat elindul, ezután jön egy alagút(2 bejárat és egy sín), minden elem összekapcsolva a következővel. A teszt során az alagút engedélyezett állapotban van, az alagút engedélyezésére van külön teszt.

- Ellenőrzött funkcionális, várható hibahelyek

A teszt azt ellenőrzi, hogy a vonat képes-e az alagútban a továbbhaladásra.

Hiba lehet, ha a vonat nem lép a következő sínre, bejáratra, pedig az lenne az elvárt vagy ha rosszul lép.

- Bemenet

```
init vonat_mozg_alagutban
spawnTrain
listTrains
step
listTrains
step
listTrains
step
listTrains
```

- Elvárt kimenet

Az inicializálás sikeres volt.

y\x	1	2	3	4
1	r	te	tr	te
2				
3				
4				

A vonat megjelenítése sikeresen lezajlott.

```
<train1> <nincs a pályán>
<carriage1> <nincs a pályán> <piros> <nem üres>
```

1. lépés.

y\x	1	2	3	4
1	r T	te	tr	te

2				
3				
4				

<train1> <1,1>  
 <carriage1> <nincs a pályán> <piros> <nem üres>

2. lépés.

y\x	1	2	3	4
1	r C	te T	tr	te
2				
3				
4				

<train1> <2,1>  
 <carriage1> <1,1> <piros> <nem üres>

3. lépés

y\x	1	2	3	4
1	r	te C	tr T	te
2				
3				
4				

<train1> <3,1>  
 <carriage1> <2,1> <piros> <nem üres>

## 8.2.11 Vonat mozgatása váltón

- Leírás

A vonat létfontosságú tulajdonsága, hogy képes a pályán a mozgásra. Ennek a tesztnek a célja, hogy belássuk, hogy a vonat képes a váltón mozogni. A pálya úgy épül fel hogy van egy sín amin a vonat elindul, ezután jön egy váltó(előre déli irányba állítva), majd még egy sín.

- Ellenőrzött funkcionalitás, várható hibahelyek

A teszt azt ellenőrzi, hogy a vonat képes-e a váltón a továbbhaladásra.

Hiba lehet, ha a vonat nem lép a következő sínrre, váltóra, pedig az lenne az elvárt vagy ha rosszul lép.

- **Bemenet**

```
init vonat_mozg_valton
```

```
spawnTrain
```

```
listTrains
```

```
step
```

```
listTrains
```

```
step
```

```
listTrains
```

```
step
```

```
listTrains
```

- **Elvárt kimenet**

Az inicializálás sikeres volt.

y\x	1	2	3	4
1	r	sw		
2		r		
3				
4				

A vonat megjelenítése sikeresen lezajlott.

```
<train1> <nincs a pályán>
```

```
<carriage1> <nincs a pályán> <piros> <nem üres>
```

1. lépés.

y\x	1	2	3	4
1	r T	sw		
2		r		
3				
4				

```
<train1> <1,1>
```

```
<carriage1> <nincs a pályán> <piros> <nem üres>
```

2. lépés.

y\x	1	2	3	4
1	r C	sw T		
2		r		
3				
4				

<train1> <2,1>  
<carriage1> <1,1> <piros> <nem üres>

3. lépés.

y\x	1	2	3	4
1	r	sw C		
2		r T		
3				
4				

<train1> <2,2>  
<carriage1> <2,1> <piros> <nem üres>

### 8.2.12 Váltó állítása

y\x	1	2	3	4
1	r	sw	r	
2		r		
3				
4				

- Leírás

Ez a teszeset azt vizsgálja, hogy a felhasználó képes-e az általa kiválasztott váltót a számára

megfelelő pozícióba állítani, hogy a vonat továbbhaladhasson a megfelelő irányba. A pályán egy switch és négy sín található, a váltót a felhasználó E (east) állásból, S (south) állásba állítja.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt ellenőrzi, hogy a felhasználó tudja-e a váltókat állítani. Hibalehetőség lehet, ha nem a felhasználó által megadott váltó állítódik át, vagy a váltó állása egyáltalán nem változik, esetleg rossz irányba mutat az átváltás után.

- **Bemenet**

```
init valtoAll
listSwitches
setSwitch switch1 S
listSwitches
```

- **Elvárt kimenet**

Az inicializálás sikeres volt.

```
<switch1> <2,1> <nincs rajta elem> <1,1> <3,1>
```

A switch sikeresen át lett állítva.

```
<switch1> <2,1> <nincs rajta elem> <1,1> <2,2>
```

### 8.2.13

### Játék vége ütközésnél

y\x	1	2	3	4
1	r	r	r	r
2				
3				
4				

- **Leírás**

Ez a teszeset azt vizsgálja, hogy két vonat ütközésénél a játék véget ér-e. A pályán négy sín található, majd a pálya két széléről elindítunk 2 vonatot egy lépés késleltetéssel. Hárrom lépés után találkoznak és összeütköznek.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt ellenőrzi, hogy a vonatok ütközésénél vége van-e a játéknak. Hiba lehet, ha két vonat összeütközik, de a játék nem ér véget.

- **Bemenet**

```
init jatekVege1
spawnTrain 0
step
spawnTrain 1
step
listTrains
step
listTrains
```

- **Elvárt kimenet**

Az inicializálás sikeres volt.

A vonat megjelenítése sikeresen lezajlott.

1. lépés.

$y \setminus x$	1	2	3	4
1	r   T	r	r	r
2				
3				
4				

A vonat megjelenítése sikeresen lezajlott.

2. lépés.

$y \setminus x$	1	2	3	4
1	r	r   T	r	r   T
2				
3				
4				

<train1> <2,1>

<train2> <4,1>

3. lépés.

$y \setminus x$	1	2	3	4
1	r	r	r   TT	r
2				
3				
4				

<train1> <3,1>

<train2> <3,1>

A vonat felrobbant, a játéknak vége.

### 8.2.14 Játék vége rossz váltónál

y\x	1	2	3	4
1	r	sw	r	
2				
3				
4				

- **Leírás**

Ez a teszeset a rossz váltóállást vizsgálja. A váltó E irányát a felhasználó S-re állítja, ahol nem található elem. A vonat így erre lép és felrobban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt ellenőrzi, hogy véget ér-e a játék, ha a vonat egy olyan váltóra érkezik, ami nem biztosít továbbhaladást. Hiba lehet, ha a váltó átállítás után is a helyes irányba mutat, vagy helytelen irányban esetén a vonat nem robban fel.

- **Bemenet**

```
init jatekVege2
spawnTrain
step
listSwitches
setSwitch switch1 S
step
listSwitches
step
```

- **Elvárt kimenet**

Az inicializálás sikeres volt.

A vonat megjelenítése sikeresen lezajlott.

1. lépés.

y\x	1	2	3	4
1	r   T	sw	r	
2				
3				
4				

<switch1> <2,1> <nincs rajta elem> <1,1> <3,1>

2. lépés.

y\x	1	2	3	4
1	r	sw   T	r	
2				
3				
4				

<switch1> <2,1> <train1> <1,1> <null>

3. lépés.

y\x	1	2	3	4
1	r	sw	r	
2		T		
3				
4				

A vonat felrobbant, a játéknak vége.

### 8.3 A tesztelést támogató programok tervei

Készíteni fogunk egy tesztelő osztályt, melyet parancssori argumentumként lehet előhívni. Meg kell majd adni az ellenőrizni kívánt teszteset nevét, ami megegyezik a hozzá tartozó pálya nevével. Az osztály ezután elvégzi az in.txt fájlban megadott parancsok futtatását és leellenőrzi, hogy eredményük megegyezik-e az out.txt fájlban található értékkel.

## 8.4 Napló

Kezdet	Időtartam	Résznevők	Leírás
2017. 03. 29. 19:00	1 óra	Dobó Fenes Papp Salamon Vizi	<p>Gyűlés.</p> <p>A konzultáción elhangzottak átbeszélése, eheti feladat értelmezése, feladatak kiosztása.</p> <p>Döntés:</p> <ul style="list-style-type: none"> <li>• Dobó <ul style="list-style-type: none"> <li>○ MapElement osztályleírás</li> <li>○ Rail osztályleírás</li> <li>○ Switch osztályleírás</li> <li>○ Váltó állítása teszeset</li> <li>○ Játék vége ütközésnél teszeset</li> <li>○ Játék vége rossz váltónál teszeset</li> </ul> </li> <li>• Fenes <ul style="list-style-type: none"> <li>○ Map osztályleírás</li> <li>○ Train osztályleírás</li> <li>○ MovingElement osztályleírás</li> <li>○ Vonat mozgatása sínen teszeset</li> <li>○ Vonat mozgatása alagútban teszeset</li> <li>○ Vonat mozgatása váltón teszeset</li> </ul> </li> <li>• Papp <ul style="list-style-type: none"> <li>○ System osztályleírás</li> <li>○ Player osztályleírás</li> <li>○ Alagút engedélyezése teszeset</li> <li>○ Szín meghatározás teszeset</li> <li>○ Kocsi üresség teszeset</li> </ul> </li> <li>• Salamon <ul style="list-style-type: none"> <li>○ Konzultáción beszélt változtatások eszközölgése (CoalCarriage)</li> <li>○ Kimeneti nyelv bővítése</li> <li>○ Carriage osztályleírás</li> <li>○ CoalCarriage osztályleírás</li> <li>○ Tunnel osztályleírás</li> <li>○ Utasok leszállása (a vonat eltűnik) teszeset</li> <li>○ Utasok leszállása (a vonat nem tűnik el) teszeset</li> <li>○ Utasok nem szállnak le teszeset</li> </ul> </li> <li>• Vizi <ul style="list-style-type: none"> <li>○ Pályafájlok felépítésének megtervezése</li> <li>○ Station osztályleírás</li> <li>○ FileManager osztályleírás</li> <li>○ TunnelEntrance osztályleírás</li> <li>○ Utas felszállás teszeset</li> <li>○ Vonat megszüntetés teszeset</li> </ul> </li> </ul>
2017. 03. 31.	2 óra	Salamon	Osztálydiagram kijavítása a konzultáción

20:00			megbeszéltek szerint. Kimeneti nyelv bővítése. Carriage, CoalCarriage, Tunnel osztályok leírásának elkészítése.
2017. 03. 31. 20:00	2 óra	Papp	Elkészített dolgok: <ul style="list-style-type: none"> <li>○ System osztályleírás</li> <li>○ Player osztályleírás</li> </ul>
2017.03.31. 20:30	1,5 óra	Dobó	MapElement, Rail, Switch osztályok leírásának elkészítése.
2017. 03. 31. 21:00	1,5 óra	Fenes	Map, Train, MovingElement osztályok leírásának elkészítése.
2017. 03. 31. 22:30	1,5 óra	Salamon	Utasok leszállása (a vonat eltűnik), Utasok leszállása (a vonat nem tűnik el), Utasok nem szállnak le tesztesetek megírása.
2017. 04.01. 09:00	1,5 óra	Fenes	Vonat mozgatása sínen. vonat mozgatása alagútban, vonat mozgatása váltón tesztesetek elkészítése.
2017.04.01. 10:00	1,5 óra	Dobó	Váltó állítása, játék vége ütközésnél és játék vége rossz váltónál tesztesetek megírása.
2017.04.01 12:00	2 óra	Vizi	Station osztályleírás, FileManager osztályleírás, TunnelEntrance osztályleírás megírása.
2017. 04. 01. 14:00	0,5 óra	Salamon	Kimeneti nyelv javítása. Az ebből következő javítások eszközölése a teszteseteken.
2017. 04. 01. 14:30	1 óra	Dobó	Tesztesetek kijavítása a megváltoztatott ki és bemeneti nyelv alapján.
2017. 04. 01. 20:00	1 óra	Vizi	Pályafájlok felépítésének megtervezése.
2017. 04. 02 12:00	1,5 óra	Vizi	Utas felszállás teszteset, Vonat megszüntetés teszteset elkészítése.
2017. 04. 02. 20:00	2 óra	Papp	Elkészített dolgok: <ul style="list-style-type: none"> <li>○ Alagút engedélyezése teszteset</li> <li>○ Szín meghatározás teszteset</li> <li>○ Kocsi üresség teszteset</li> </ul>