**Terepasztal**

Szoftvertechnológia házi feladat

Jurin Maxim

O18PS2

# A feladat leírása

A Bakter játék egy terepasztal szimulációja. A terepasztal négyzet alakú mezőkre van osztva. Egy mezőben egy vasúti pályaelem lehet, amelynek a következő fajtái vannak: jobb és bal oldalt, alsó és felső oldalt összekötő elemek, kanyarodó elemek (például jobb oldalt alsó oldallal összekötő elem), váltó, amely egy oldalt két másikkal köt össze (pl. bal oldalt jobb oldallal és felső oldallal, de ilyenkor a jobb és a felső oldal között nincs kapcsolat), végül létezik kereszteződés is. A váltókat váltani lehet, ami azt jelenti, hogy egyszerre mindig csak az egyik átmenete működik. Az egyszerű elemeken, amelyek csak két oldalt kötnek össze, szemafor is lehet. Speciális elem a fejállomás, aminek csak egy kijárata van, a vonatok itt "keletkeznek". A pályán vonatok (szerelvények) közlekednek. A vonat pontosan egy mozdonyból és néhány vagonból áll, szélsőséges esetben nincs vagon a mozdonyra kapcsolva (ún. gépmenet). Mind a mozdony, mind a vagon egy mező hosszúságú. A mozdony, ha zárt szemaforhoz ér, megáll, és csak akkor indul tovább, ha a szemafort kinyitják. A játék során a vonatok a fejállomásról véletlenszerűen indulnak, és a pályán haladnak. A játékos a váltókat és a szemaforokat tudja kezelni. Pontot időegységenként az éppen haladó mozdonyok után lehet kapni. Cél, hogy a játékos minél több pontot gyűjtsön. Ha két szerelvény összeütközik, a játéknak vége.

# Funkcionális követelmények

## Elsődleges követelmények

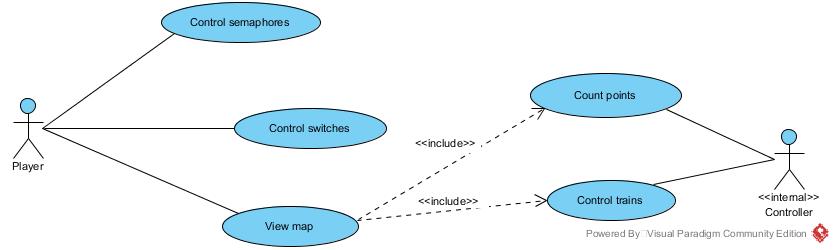
|  |  |  |
| --- | --- | --- |
| **Azonosító** | **Leírás** | **Use-case** |
| R01 | A terepasztal négyzet alakú mezőkre van osztva. | View map |
| R02 | Egy mezőben egy vasúti pályaelem lehet. | View map |
| R03 | A pályaelemek fajtái a következők: egyenes összekötő elemek, kanyarodó elemek, váltó elemek és kereszteződés. | View map |
| R04 | A váltóknak mindig csak az egyik átmenete működik. | View map |
| R05 | Az egyszerű elemeken, amelyek csak két oldalt kötnek össze, szemafor is lehet. | View map |
| R06 | A szemafor nyitott vagy zárt állású lehet. |  |
| R07 | A fejállomásnak egy kijárata van, és itt keletkeznek a vonatok. | View map, Control trains |
| R08 | A pályán vonatok (szerelvények) közlekednek. | Control trains |
| R09 | A vonat pontosan egy mozdonyból és néhány vagonból áll, szélsőséges esetben nincs vagon a mozdonyra kapcsolva (ún. gépmenet). | Control trains |
| R10 | Mind a mozdony, mind a vagon egy mező hosszúságú. | View map |
| R11 | A mozdony, ha zárt szemaforhoz ér, megáll, és csak akkor indul tovább, ha a szemafort kinyitják. | Control trains |
| R12 | A vonatok a fejállomásról véletlenszerűen indulnak. | Control trains |
| R13 | A vonatok a pályán haladnak. | View map, Control trains |
| R14 | A játékos tudja kezelni a váltókat. | Control switches |
| R15 | A játékos tudja kezelni a szemaforokat. | Control semaphores |
| R16 | Pontot időegységenként az éppen haladó mozdonyok után lehet kapni. | Count points |
| R17 | Cél, hogy a játékos minél több pontot gyűjtsön. | Count points |
| R18 | Ha két szerelvény összeütközik, a játéknak vége. | Control trains |

## További követelmények

|  |  |  |
| --- | --- | --- |
| **Azonosító** | **Leírás** | **Use-case** |
| R19 | A vonatok a kereszteződésben egyenesen haladnak tovább. | Control trains |
| R20 | Egy fejállomás van a játékban. | View map |

# Use-case-ek

## Use-case diagram



## Use-case leírások

|  |  |
| --- | --- |
| **Cím** | **Control semaphores** |
| **Leírás** | A játékos kezeli a szemaforokat |
| **Aktorok** | Player |
| **Főforgatókönyv** | 1. A játékos megváltoztatja a szemaforok állását |

|  |  |
| --- | --- |
| **Cím** | **Control switches** |
| **Leírás** | A játékos kezeli a váltókat |
| **Aktorok** | Player |
| **Főforgatókönyv** | 1. A játékos megváltoztatja a váltók állását |

|  |  |
| --- | --- |
| **Cím** | **View map** |
| **Leírás** | A játékos megtekinti a pályát |
| **Aktorok** | Player |
| **Főforgatókönyv** | 1. A rendszer kirajzolja a labirintus aktuális állapotát  2. A játékos megtekinti a labirintus aktuális állapotát |

|  |  |
| --- | --- |
| **Cím** | **Control trains** |
| **Leírás** | A rendszer irányítja a vonatokat |
| **Aktorok** | Controller |
| **Főforgatókönyv** | 1. Szerelvények keletkeznek az állomáson  2. A szerelvény időegységenként egy mezőt mozdul előre |
| **Alternatív forgatókönyv** | 2.A.1. Ha a szemafor zárva van, akkor a vonat megáll |
| **Alternatív forgatókönyv** | 2.B.1 Ha a vonat ütközik, a játéknak vége |

|  |  |
| --- | --- |
| **Cím** | **Count points** |
| **Leírás** | A rendszer számolja a játékos pontjait. |
| **Aktorok** | Controller |
| **Főforgatókönyv** | 1. A játékos időegységenként pontot kap minden mozgásban levő vonatért. |
| **Alternatív forgatókönyv** | 1.A.1. Ha a jatéknak vége, a számláló megáll. |

# Strukturális leírás

## Az osztályok leírása

### Game

**Felelősségek**

A game játékot indítja és fejezi be.

**Attribútumok**

**Metódusok**

|  |  |
| --- | --- |
| +startGame | Elindítja a játékot. |
| +endGame | Befejezi a játékot. |

### Map

**Felelősségek**

Tárolja a játék részeit, továbbá ezen keresztül lehet irányítani a szemaforokat és a váltókat.

**Attribútumok**

|  |  |
| --- | --- |
| -points: int | Tárolja a pontok számát. |
| -controllable[int]: Controllable | Tárolja a játékos által kontrollálható elemeket. |
| -trainsOnTrack: Locomotive[0..\*] | Tárolja a mozdonyokat. |
| -elements: RailElements[0..\*] | Tárolja a pályát alkotó elemeket. |

**Metódusok**

|  |  |
| --- | --- |
| +step() | Növeli a pontok számát a mozgó vonatoktól függően. |
| +AddTrain(l: Locomotive) | Hozzáad egy új vonatot az eltárolt vonatokhoz. |
| +switch() | A Controllable interface egy elemét hívja meg. |
| +increasePoints() | Megnöveli a pontok számát. |

### Direction (enumeráció)

**Felelősségek**

A játékbeli irányokat tárolja.

**Attribútumok**

|  |  |
| --- | --- |
| +up | Felfele. |
| +down | Lefele. |
| +left | Balra. |
| +right | Jobbra. |

### RailElement

**Felelősségek**

Ez az osztály reprezentálja a pálya elemeit.

**Attribútumok**

|  |  |
| --- | --- |
| -currTrain | Tárolja az éppen pályán levő szerelvényt. |
| -neighbor[Direction] | Irányonként tárolja a szomszédos mezőket. |

**Metódusok**

|  |  |
| --- | --- |
| +Accept(t: TrainElement) | Eltárolja a vonatelemet a mezőben. |
| +Remove(t: TrainElement) | Eltávolítja a vonatelemet a mezőről. |
| +getNextDir(): Direction | Megadja a következő irányt, amerre haladnia kell a vonatnak. |
| +getNeighbor(d: Direction): RailElement | Megadja az irány alapján a szomszédját a mezőnek. |

### Switch

**Felelősségek**

Ez az elem egy váltó, amelynek 3 kijárata van.

**Attribútumok**

|  |  |
| --- | --- |
| -openDirection: Direction | Ez a nyitott kijárata a váltónak. |
| -closedDirection: Direction | Ez az éppen nem használt kijárata a váltónak. |

**Metódusok**

|  |  |
| --- | --- |
| +switch() | Ez a metódus átállítja a nyitott kijáratot a nem használtra. |
| +getNextDir() : Direction | Ez a metódus meghatározza, hogy merre megy tovább az éppen rajta álló vonat. |

### Simple

**Felelősségek**

Ez egy egyszerű elem, ami kettő irányt köt össze.

**Attribútumok**

|  |  |
| --- | --- |
| -semaphore: Semaphore | Ez eltárolja a szemafort, hogyha az van. |

**Metódusok**

|  |  |
| --- | --- |
| +getNextDir() : Direction | Ez a metódus meghatározza, hogy merre megy tovább az éppen rajta álló vonat. |

### Cross

**Felelősségek**

Cross

**Attribútumok**

**Metódusok**

|  |  |
| --- | --- |
| +getNextDir() : Direction | Ez a metódus meghatározza, hogy merre megy tovább az éppen rajta álló vonat. |

### Station

**Felelősségek**

Ez a fejállomás, ahol keletkeznek a vonatok.

**Attribútumok**

|  |  |
| --- | --- |
| -map: Map | Tárolja a pályát. |

**Metódusok**

|  |  |
| --- | --- |
| +step() | Ez a metódus véletlenszerű időpontokaban létrehoz egy vonatot. |
| +getNextDir() : Direction | Ez a metódus meghatározza, hogy merre megy tovább az éppen rajta álló vonat. |

### Semaphore

**Felelősségek**

Megállítja a vonatot, hogyha blokkolt állapotban van.

**Attribútumok**

|  |  |
| --- | --- |
| -status: boolean | Ez az állapota a szemafornak. Ha true, akkor át tud menni a vonat. |

**Metódusok**

|  |  |
| --- | --- |
| +switch() | Ez a metódus megváltoztatja a statust. |
| +getStatus(): boolean | Ez visszaadja a statusát a szemafornak. |

### Controllable

**Felelősségek**

Ez az interface a kontrollálható objektumokat képviseli.

**Metódusok**

|  |  |
| --- | --- |
| +switch() | Ez a metódus jelképezi az interface által egybefoglalt műveleteket. |

### Locomotive

**Felelősségek**

A mozdony van a vonat elején, ami után megy a többi vagon is.

**Attribútumok**

**Metódusok**

|  |  |
| --- | --- |
| +step() | Ez a metódus a következő mezőre lépteti a mozdonyt. |
| +stop() | Ez megállítja a mozdonyt. |
| +move() | Ez elindítja a mozdonyt. |

### Wagon

**Felelősségek**

Ez egy kocsit jelképez a vonaton, ami a mozdonyra vagy egy másik kocsira van szerelve.

**Attribútumok**

**Metódusok**

|  |  |
| --- | --- |
| +step() | Ez a metódus az előtte levő vonatelem helyére lépteti a vagont. |

### TrainElement

**Felelősségek**

Ez az osztály a vonat elemeit képviseli, amikből felépül egy szerelvény.

**Attribútumok**

|  |  |
| --- | --- |
| -moving : boolean | Ez tárolja el azt, hogy a vonat mozog-e éppen. |
| -before: TrainElement | Ez tárolja ez a vonatelem előtti elemet. |
| -after: TrainElement | Ez tárolja el a vonatelem után következő elemet. |
| -currDirection: Direction | Ez tárolja el a vonatelem aktuális irányát. |
| -currRail: RailElement | Ez tárolja el a vonataelem helyét, azaz a hozzátartozó sínelemet. |

**Metódusok**

|  |  |
| --- | --- |
| +setDirection(d : Direction) | Ez állítja be a vonatelem írányát. |
| +setCurrRail(r : RailElement) | Ez állítja be a vonatelem aktuális helyét. |
| +getCurrRail() : RailElement | Ez adja vissza a vonatelem aktuális helyét. |
| +addAfter(t : TrainElement) | Ez hozzáad egy vonatelemet egy vonat végéhez. |
| +getMoving() : boolean | Ez visszaadja a moving attribútum értékét. |
| +getBefore() : TrainElement | Ez visszaadja a vonatelem előtt következő vonatelemet. |

### Steppable

**Felelősségek**

Ez az interface a léptehető dolgokat képviseli.

**Metódusok**

|  |  |
| --- | --- |
| +step() | Ez a metódus jelképezi az interface által egybefoglalt műveleteket. |

### Timer

**Felelősségek**

Ez a timer időként szolgál és léptet.

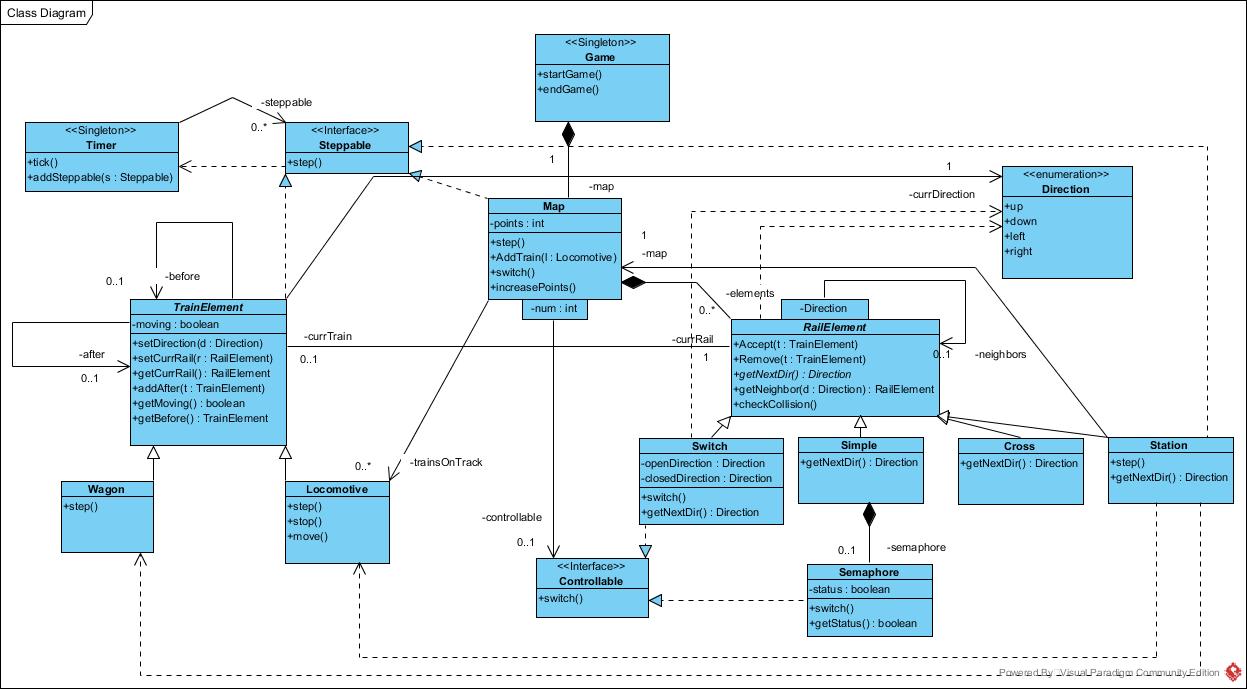
**Attribútumok**

|  |  |
| --- | --- |
| -steppable: Steppable[0..\*] | Ez tárolja a lépthető elemeket. |

**Metódusok**

|  |  |
| --- | --- |
| +Tick | Lépteti a lépthetető elemeket. |
| +AddSteppable(s: Steppable) | Hozzáad egy léptethető elemet. |

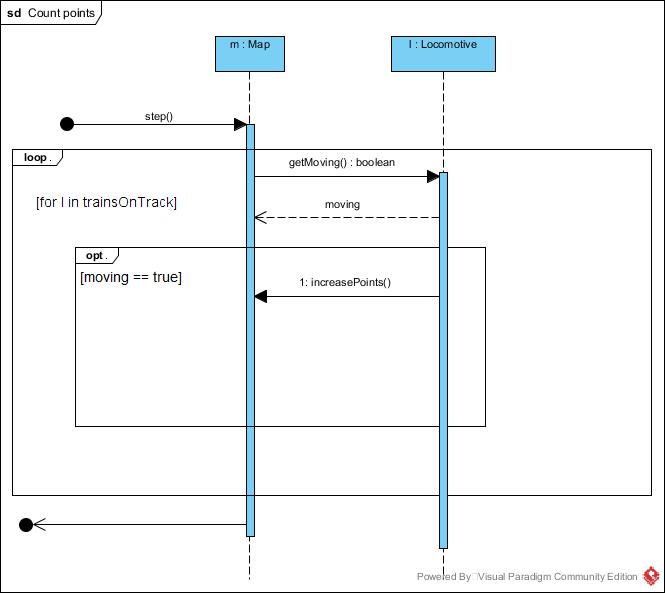
## Osztálydiagram



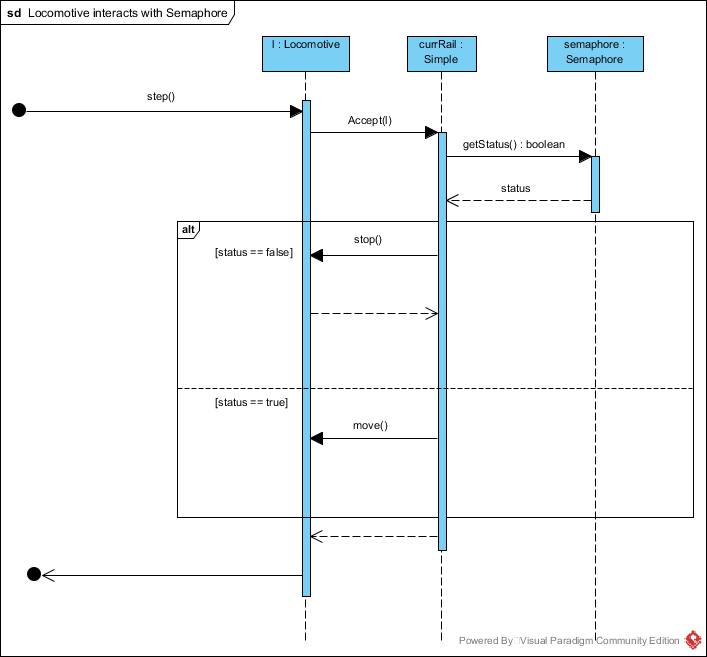
# Viselkedés leírása

## Szekvencia diagramok

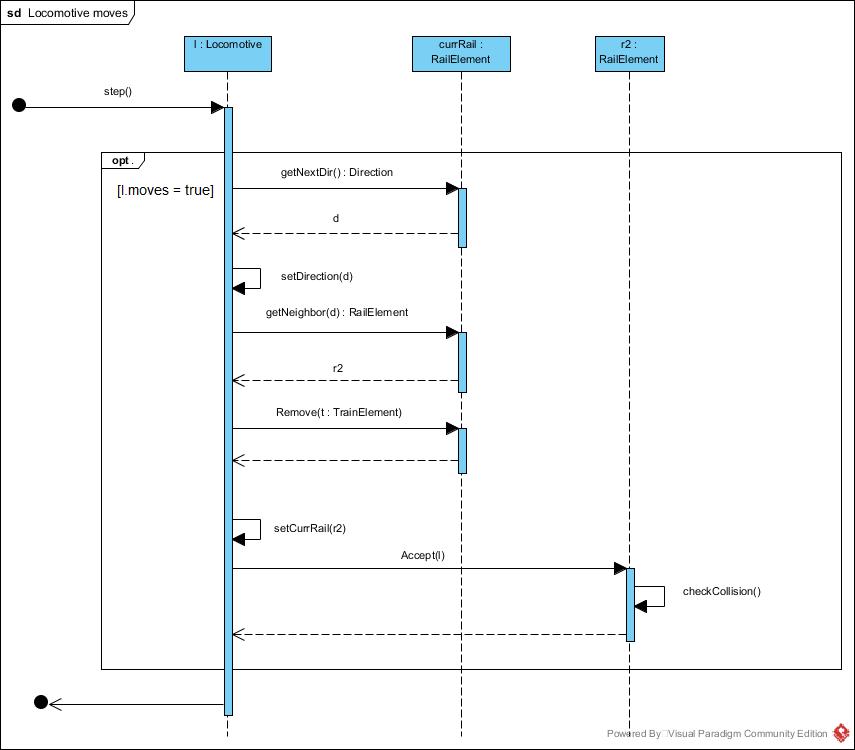
### Count Points



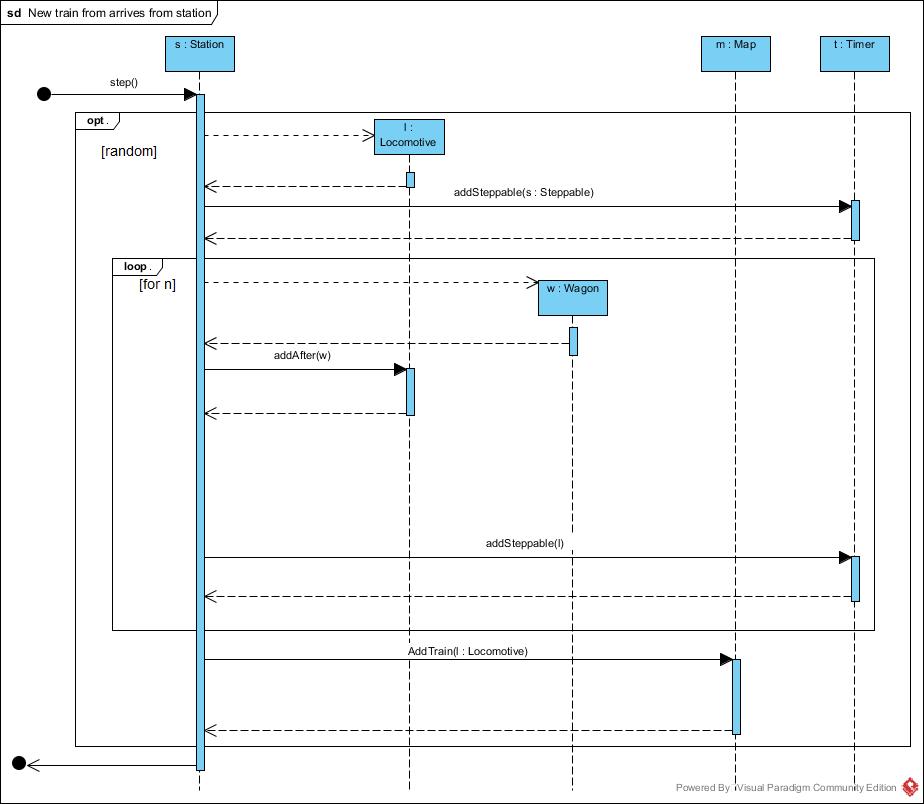
### Locomotive interacts with Semaphore



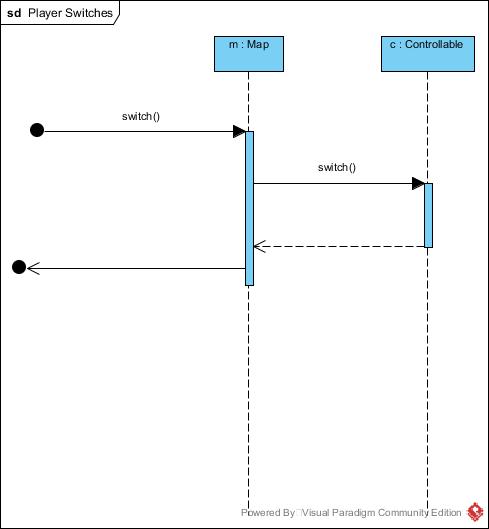
### Locomotive moves



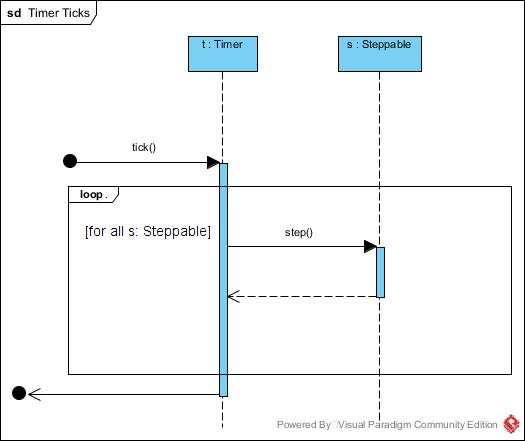
### New train from arrives from station



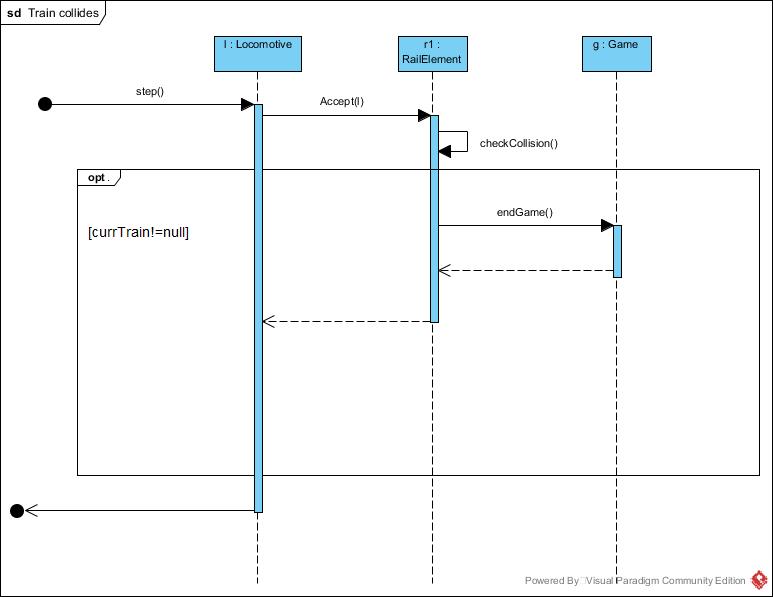
### Player Switches



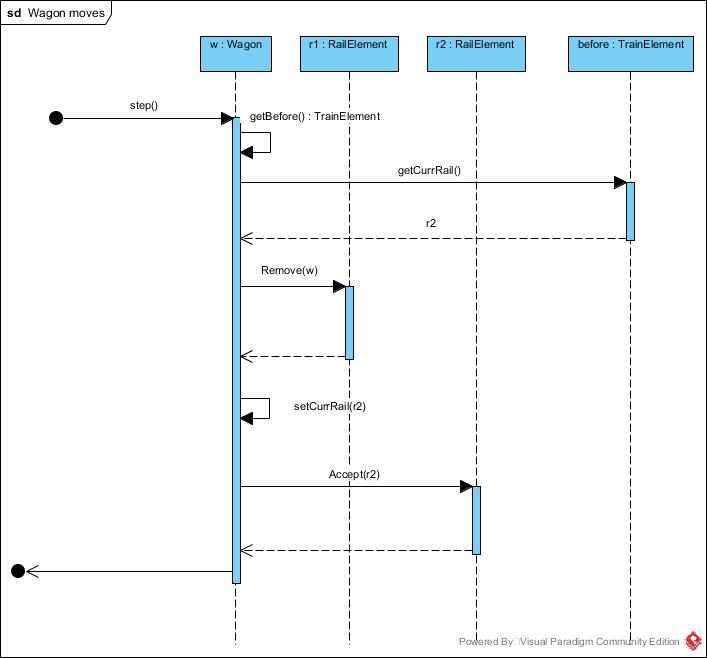
### Timer Ticks



### Train collides

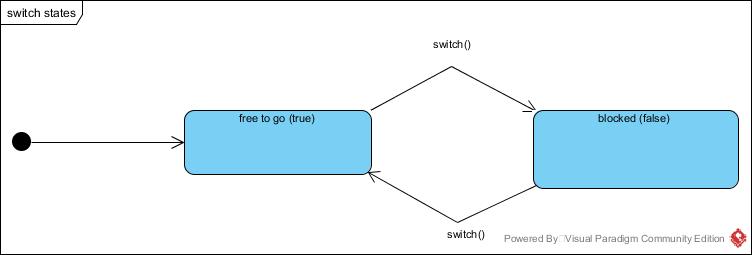


### Wagon moves

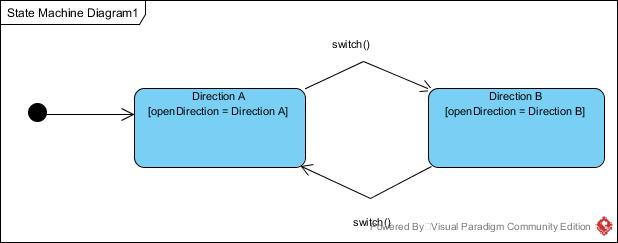


## Állapotgép diagramok

### Switch changed



### Semafor changed



# Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Elvégzett munka** | **Hivatkozások** |
| 2017.12.08. | 2 óra | Követelmények listájának elkészítése, usecase-ek megrajzolása és leírása | 2,3 |
| 2017.12.09 | 6 óra | További tervezés, osztályok elkészítése | 4 |
| 2017.12.11 | 4 óra | Osztálydiagram megrajzolása | 4 |
| 2017.12.12 | 4 óra | Szekvencia diagramok rajzolása, javítások az osztálydiagramban | 4,5 |
| 20.17.12.12 | 2 óra | Dokumentum átnézése, esetleges hibák javítása | egész |

**Összes elvégzett munka:** 18 óra

**Modellező eszköz:** Visual Paradgim

**Egyéb eszközök:** Word