

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

Ezen dokumentum célja, hogy összefoglalja a *Javallérok* csapat *Hókotrók* nevű szoftver implementációjának egyes részleteit, illetve követelményeit.

2.1.2 Szakterület

A szoftver célja a *Szoftver projekt laboratórium* tárgy oktatói / felelősei által kiírt *Hókotrók* nevű játék - a meghirdetett követelmények mellett - minél pontosabb megoldása. Az elsődleges célkitűzés magában foglal egy szórakoztató számítógépes játék megvalósítását.

2.1.3 Definíciók, rövidítések

Projlab.: Szoftver projekt laboratórium

2.1.4 Hivatkozások

Projlab. feladatkiírás: <https://www.iit.bme.hu/file/11582/feladat>

Projlab. beadandó dokumentummal kapcsolatos információk, követelmények:

<https://www.iit.bme.hu/file/13619/2-k%C3%B6vetelm%C3%A9ny-projekt-funcionalit%C3%A1s>

2.1.5 Összefoglalás

A szoftverhez tartozó követelmények, illetve ezen pontok részletesebb leírása kerül ismertetésre a dokumentum további részén. A szoftver ismertetése és tervei mind felhasználói, mind pedig fejlesztői szempontból is megvalósulnak.

2.2 Áttekintés

2.2.1 Általános áttekintés

A program alapvetően egy pályából, és egy vagy több játékosból tevődik össze, akik a pályán hókotróvezetőként vagy buszsofőrként vannak jelen.

A pálya egy komplikált úthálózatból áll, ahol az utak lehetnek többsávosak, de lehetnek hidak vagy alagutak is (egyes szakaszaik, vagy akár a teljes út). Az úton autók, illetve a játékosok által irányított járművek közlekednek. A játék elsősorban kompetatív, de van minimális lehetőség kooperációra. A két fajta játékos a maga módján pontokat gyűjthet, és akinek a legtöbb pontja van a játék végén, ő nyer.

2.2.2 Funkciók

Zúzmaravárosban minden évben megérkezik a tél, most mégis váratlanul nagy havazás kezdődik, és a hó azóta is megállás nélkül esik. A város utcáin egyre vastagabb hóréteg áll és még azokat a sávokat is újra befedi, amelyeket korábban már sikerült letisztítani. A közlekedés emiatt gyorsan kaotikussá válik, ezért a takarítóknak minél előbb járhatóvá kell tenniük az utakat.

A városban többek között autók is közlekednek, amelyeket nem a játékosok irányítanak. Az autósok a lakásuk és a munkahelyük között járnak, és mindig arra próbálnak menni, amerre a legkevesebbet kell kerülniük, de csak ott, ahol még járható az út. Ha egy sávban a hó túl magas, az autó elakad, és nem tud tovább menni. Ha viszont a hó még nem túl vastag, az

autók át tudnak rajta jutni. Azonban ha sok autó egymás után ugyanott halad át, a letaposott hó csúszós jégréteggé alakul. Ilyenkor a járművek könnyebben megcsúsznak, és ha két autó összeecsúszik, az érintett sáv használhatatlanná válik a baleset miatt. Az autók ütközés után még röcid ideig akadályt képeznek. majd meghatározott idő elteltével eltűnnek a helyszínről és a sáv ismét felszabadul.

Vannak egy- és többsávos utak, nagy csomópontok, ahol akár négyenél több út találkozik, és olyan kereszteződések is, ahol az egyik út hídon vagy alagúton vezet át a másik felett. Emiatt nem minden kereszteződés jelent valódi találkozást. Az utak rögzített helyzetéhez kötése tiltott. A takarítók hókotrókat irányítanak, hogy a sávokat minél gyorsabban letisztítsák. Egy hókotró egyszerre mindig csak egyetlen sávban tud dolgozni. Ha egy autó elakad, de a szomszédos sávot közben letakarítják, akkor át tud váltani oda, és folytathatja az útját. A hókotrók különféle fejeket használhatnak, de egyszerre csak egyet viselhetnek, és szükség esetén cserélgethetik őket.

Ötféle fej létezik. A söprő fej a havat oldalra tolja, közvetlenül a hókotró mellett, vagyis a szomszédos sávba, illetve az út szélére. A hányó fej szintén oldalra viszi a havat, de messzebbre szórja, így az nem kerül a közvetlen közelbe. A jégtörő fej a kialakult jégréteget feltöri, de az így keletkező törmelékek még ott maradnak az adott sávban, ezért utána további takarításra lehet szükség. A sószóró fej sót juttat az útra és a só idővel megolvasztja a havat és a jeget, és egy ideig azt is segít megakadályozni, hogy az újonnan hulló hó megmaradjon. A sárkány fej egy erős hőforrással egyszerre olvasztja el a havat és a jeget a hókotró előtt.

A takarítók jutalmat kapnak minden letisztított útszakaszért. Ebből új fejeket vehetnek, illetve beszerezhetik a működéshez szükséges anyagokat is, mint például a sószóróhoz só, a sárkány fejhez biokerozin. Ha a só vagy a biokerozin elfogy akkor a megfelelő fej hatástalanná válik, ezért ilyenkor érdemes másik fejre váltani, amíg új készletet nem sikerül vásárolni. A fejek egyre drágábbak a söprőtől a sárkány felé haladva, és a legnagyobb értékű vásárlás egy új hókotró beszerzése.

A második fajta játékos a buszvezető. A buszok nem csak két végállomás között járhatnak: a vonaluk több megállót is érinthet, ahol utasokat vehetnek fel és tehetnek le. A buszvezetők célja, hogy a játék ideje alatt minél több sikeres megálló-elérést és szállítást teljesítsenek. Az utasok után pont vagy jutalom járhat, és a buszon jegyvásárlás is történhet, ami bevételt termel a buszvezetőnek. A buszvezető a bevételből fejlesztheti a buszát, vagy akár új, jobb buszt vásárolhat: a jobb busz több utast tud kiszolgálni, és a szállításért több jutalmat kaphat. A buszoknak a megállók között időben is helyt kell állniuk: egy megadott időn belül el kell érniük a következő megállót. Ha egy busz ütközés miatt késik, az utasok elégedettsége csökken, és a leszállításért járó jutalom is kisebb lehet. Ütközéskor a busz egy időre mozgásképtelenné válhat

Buszvezetőkkel és takarítókkal egyszerre többen is játszhatnak.

A játék végén az eredmények összesítése megmutatja, hogy a buszvezetők hány fordulót teljesítettek, a takarítók pedig mennyi útszakaszt tettek járhatóvá és mennyi jutalmat szereztek.

2.2.3 Felhasználók

A felhasználóknak nincs szüksége különösebb előtudásra, sem képzettségre a szoftver használatához. A programot egyidejűleg több felhasználó is használhatja.

2.2.4 Korlátozások

A programtól elvárt, hogy az stabilan és helyesen működjön. A program nem léphet ki / zárhatja be magát a felhasználók akarata nélkül, illetve időben kell reagálnia a játékosok inputjaira.

2.2.5 Feltételezések, kapcsolatok

A tárgykövetelményeket, a konzultációkon megbeszélteket, és az előadáson elhangzottakat a projekt részeinek szakszerű megtervezéséhez használtuk.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
MAP001	A pálya úthálózatként kezelendő; csomópontban 4+ út is találkozhat.	Demo: létrehozott pályán 5+ ágú csomópont működik.	MUST	Feladatkiírás	View game	
MAP002	Az utak többsávosak is lehetnek, amik külön állapotot tárolnak. Egy út lehet havas, jeges, járható	Unit teszt: sávállapotok függetlenek	MUST	Feladatkiírás	View game	
MAP003	Hidak és alagutak létezése. Keresztezés létezhet úgy, hogy nem csomópont.	Demo: két út kereszteződhet, de nincsen átjárás	MUST	Feladatkiírás	View game	
MAP004	Koordinátákhoz kötés szabályellenes.	Review: nincs koordináta alapú azonosítás	MUST	Feladatkiírás	View game	
MAP005	Az úthálózat csomópontokból és utakat összekötő szakaszból áll. Többsávos út esetén párhuzamos sávok léteznek.	Demo: többsávos út és csomópont működés	MUST	Csapatdöntés	View game	

WTH001	A havazás a játék ideje alatt folyamatos	Demo: idő múlásával nő a hó	MUST	Feladatkiírás	View game	
WTH002	A letakarított útszakaszokat a folyamatos havazás újra belepí.	Demo: takarítás után újra havas lesznek az utak.	MUST	Feladatkiírás	View game	
WTH003	Egy sáv hóvastagsága 5 szintű. Az 1-4 szintek lassítanak, az 5. szint blokkol.	Unit teszt + demo mozgással	MUST	Csapatdöntés	Move car / Move player-vehicle	
CAR001	Az autók a túlságosan mély hóban elakadnak, alacsonyabb hórétegben tudnak haladni	Unit teszt: küszöbök alapján a mozgás engedélyezett/ tiltott	MUST	Feladatkiírás	Move player-vehicle	Küszöbérték?
CAR002	Az autók mindig a lehető legrövidebb úton próbálnak közlekedni	Demo: több útból mindig a legrövidebb járhatót választja, ha lezáródik újratervez	MUST	Feladatkiírás	Move player-vehicle	
CAR003	Ha elég sok autó letapossa a havat, az adott sáv jégpáncéllá válik	Demo: ismételt áthaladás állapotváltás jegesre	MUST	Feladatkiírás	Move player-vehicle	Elég sok nincs specielve
CAR004	Jégpáncélon az autók csúszkálnak	Demo: jeges út az esetén az autó csúszik	MUST	Feladatkiírás	Move player-vehicle	Csúszásmodell nincs specielve
CAR005	Ha két autó ütközik, az adott sáv járhatatlanná válik	Demo: ütközést követően az adott sáv lezár	MUST	Feladatkiírás	Move player-vehicle	meddig járhatatlan?

CAR006	Ha az autó elakad a sávban, de a szomszédos sávot letakarítják, akkor sávot válthat és továbbhaladhat	Demo: elakad, de a szomszédos sáv letakarítását követően sávot tud váltani	MUST	Feladatkiírás	Move player-vehicle	
BUS001	Buszok járnak és egy járatnak a két végállomás között minél többször kell megfordulnia	Demo: járat oda-vissza számlálás	MUST	Feladatkiírás	Reach station	
BUS002	Ha buszba autó vagy busz belecsúszik, a busz egy időre mozgásképtelen lesz	Demo: ütközést követően a busz egy ideig nem mozdul	MUST	Feladatkiírás	Move player-vehicle	az időtartam 10 másodperc
BUS003	A buszvonal több megállót is tartalmazhat, nem csak két végállomást.	Demo: 3 vagy több megállósvonal	MUST	Csapatdöntés	Reach station	
BUS004	A busz megállóiban utasokat vehet fel és tehet le; a leszállított utasok után pont/jutalom járhat.	Demo : jutalom nő	MUST	Csapatdöntés	Reach station	Utas képződés részlete következő fázisokban lesz tisztázva
BUS005	Jegyvásárlás: az utasok jegyet váltanak és a jegybevétel a	Demo: jegybevétel nő megállóknál	MUST	Csapatdöntés	Reach station	

	buszvezető erőforrása.					
BUS006	A buszvezető bevételel fejlesztési buszát vagy új buszt vehet.	Demo: vásárlás után módosuló paraméterek.	MUST	Csapatdöntés	View bus-shop	
BUS007	A busz minősége szorzót ad a szállításért kapott jutalomra.	Unit teszt: szorzók működnek	MUST	Csapatdöntés	Reach station	
BUS008	A busznak X időn belül el kell érnie a következő megállót. A késés csökkenti az utasok elégedettségét.	Demo: késés követően az utasok elégedettségi skálája leesik.	MUST	Csapatdöntés	Reach station	X idő és csökkenés értéke paraméter
BUS009	Késés után a leszállításért kapott jutalom X idő felett csökken.	Demo: késve kevesebb jutalmat kap a buszvezető játékos.	MUST	Csapatdöntés	Reach station	
COL001	Ütközési rendszer legyen: hókotó-hókotró, busz-busz, autó-minden. Tehát busz és hókotró nem tud ütközni.	Demo: mindhárom párosítás bekövetkezik	MUST	Csapatdöntés	Move player-vehicle	
COL002	Autó az ütközés után 10 másodperc múlva eltűnik a pályáról.	Demo: időmérés	MUST	Csapatdöntés	Move car	
COL003	Ütközés esetén az érintett sáv átmenetileg járhatatlan	Demo: kerülni kell / nem járható.	MUST	Feladatkiírás + csapatdöntés	Move car	

PLW001	A hókotró egyszerre csak egy sávot tud takarítani	Demo: több sávós út esetén csak az adott sáv tisztul	MUST	Feladatkiírás	Move player-vehicle	
PLW002	A takarítók irányítják a hókotrókat: mozgatás, fejcsere, vásárlás	Demo: játékos műveletek elérhetők.	MUST	Feladatkiírás	Move player-vehicle/ View head-shop	
HD001	Egy hókotróra egyszerre 1 kotrófej szerelhető	Demo: fej váltása hatással van a takarításra	MUST	Feladatkiírás	View headshop	
HD002	A söprő fej a havat oldalra tolja közvetlenül a nyomvonal mellé. Jégpáncélt nem szed	Demo: hó átkerül a szomszéd sáv szélére, jég nem tűnik el	MUST	Feladatkiírás	Move player-vehicle	nincs en szomszéd?
HD003	A hányó fej oldalra, de messzire szórja a havat. Jégpáncélt nem szed	Demo: a hó eltűnik, de a jég megmarad	MUST	Feladatkiírás	Move player-vehicle	messzire=éter?
HD004	A jégtörő fej a jeget feltöri, de nem takarítja el. A feltört jég a sávon marad.	Demo: jégpáncél használatát követően jég törmelék megjelenése az úton.	MUST	Feladatkiírás	Move player-vehicle	
HD005	Sószóró fej a só kiszórása után idővel elolvasztja a havat/jeget és megakadályozza a hó megmaradását.	Demo: sószóró fej használata után az utak tiszták és hóesés következtében is azok maradnak.	MUST	Feladatkiírás	Move player-vehicle	Hatás időtartama?
HD006	A sárkányfej a gázturbinával	Demo: előtte lévő szakasz	MUST	Feladatkiírás	Move player-vehicle	előtte pontosan

	elolvasztja a havat/jeget a hókotró előtt.	azonnal tiszta lesz.			/ View head-shop	mit takar ?
ECO001	Hókotrók minden egységnyi letisztított útszakaszért pénz jutalmat kapnak.	Demo: takarítás után nő az egyenleg	MUST	Feladatkiírás	Clear road section	szakasz-jutalom arány defelése
ECO002	A pénzből segédeszközök vásárolhatók: fejek, fogóanyagok, új hókotró.	Demo: shop tranzakció	MUST	Feladatkiírás	View head-shop	
ECO003	Fejek ára növekvő sorrendben: söprő < hányó < jégtörő < sószóró < sárkány.	Unit teszt: árlista monoton.	MUST	Feladatkiírás	View head-shop	Konkrét árak
ECO004	Sószóró működéséhez só, sárkányhoz biokerozin kell; ha kifogy, a fej hatástalan, célszerű másik fejre váltani.	Demo: fogyás után nincs hatás, váltás működik	MUST	Feladatkiírás	Move player-vehicle / View head-shop	Fogyás sebessége?
ECO005	Legdrágább segédeszköz az új hókotró.	Unit teszt: új hókotró ára a maximum.	MUST	Feladatkiírás	View head-shop	
ECO006	Kiemelt jutalom jár azokért az utakért, amelyek két busz megálló közötti aktív útvonalon vannak.	Demo: ugyanannyi tisztításért több jutalom a kiemelt szakaszon.	MUST	Csapatdöntés	Move player-vehicle	
PLY001	Két játékoszerep	Demo: több buszvezető és	MUST	Feladatkiírás	Choose player type	

	van: buszvezető és takarító. Mindkettőből lehet több játékos egy játékban	játékos belép és játszik.				
PLY002	Buszvezetők célja a játékidő alatt minél több forduló teljesítése.	Demo: pontozás a fordulók alapján.	MUST	Feladatkiírás	Reach station	
PLY003	Takarítók kezdetben 1 hókotróval indulnak, ami vagy hányó vagy jégtörő fejes. Később új fejek és új hókotrók vehetők.	Demo: indulási állapot és későbbi vásárlás.	MUST	Feladatkiírás	View head-shop	Kezd ő válas ztás hogy an legye n
UI001	A rendszernek meg kell jelenítenie a játék állapotát	Demo: View game képernyő.	SHOU LD	Csapatdöntés	View game	“pont tábla ” miatt erőse n ajánl ott
UI002	A sávok állapota vizuálisan jelölve legyen: havas / jeges / tiszta	Demo: állapotváltásk or látszik	MUST	Csapatdöntés	View game	
UI002	Ponttábla: buszvezetők fordulóinak és/vagy takarítók eredményéne k megjelenítése	Demo: megnyitható ponttábla.	SHOU LD	Csapatdöntés	View game	Taka rítók pontj ának pont osítá sa

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
-----------	--------	------------	-----------	--------	---------

RR001	A program futtatható legyen a cél oktatási környezetben	Telepítés + futtatás teszt.	MUST	Csapatdöntés	
RR002	Beépített tesztek legyenek hó/jég átmenetekre	CI	SHOULD	Csapatdöntés	

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
DR001	Legyen "egyparancsos" futtatás		SHOULD	Csapatdöntés	
DR002	A dokumentáció tartalmazzon rövid futtatási útmutatót.	Review	SHOULD	Csapatdöntés	

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
NFR001	A megjelenítés ne igényeljen koordinátakra épülő játékszabályt (MAP004)	Review	MUST	Feladatkiírás	
NFR002	Teljesítmény: közepes számú jármű mellett is folyamatos futás	Terheléses demo.	MAY	Csapatdöntés	

2.3.5 Hardveres és szoftveres követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
COM001	A hardvernek legalább 4GB rammal kell rendelkeznie.	Egyszerű ellenőrzés	MUST	Csapatdöntés és	
COM002	A számítógépen rajta kell lennie a JAVA legalább 11-es verziójának.	Egyszerű ellenőrzés	MUST	Csapatdöntés és	

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	View game
Rövid leírás	A játékos megnyitja a játékot.
Aktorok	Játékos
Forgatókönyv	1. A rendszer kirajzolja a játék jelenlegi állapotát.
Forgatókönyv	2. A játékos megnézi a játék kirajzolt állapotát.
Alternatív forgatókönyv	2. (A) A játékos megnézi az aktuális pont táblát.

Use-case neve	View head-shop
Rövid leírás	A hókotró játékos megnézi a fejek boltját.
Aktorok	Hókotró irányító
Forgatókönyv	1. A hókotró megnézi a fejeket.
Forgatókönyv	2. A hókotró vásárol egy új fejet.
Alternatív forgatókönyv	2. (A) A hókotró átvált egy már megvásárolt fejre.

Use-case neve	Reach station
Rövid leírás	A buszvezető elér a járatának egy megállójába
Aktorok	Buszvezető
Forgatókönyv	1. A buszvezető eljuttatja a buszát egy járatának megfelelő megállóba.
Forgatókönyv	2. A rendszer pontokat ír jóvá a játékos számára.

Use-case neve	Move player-vehicle
----------------------	---------------------

Rövid leírás	A játékos az irányított járművét mozgatja.
Aktorok	Játékos
Forgatókönyv	1. A játékos a járművet a szomszédos útra lépteti.
Alternatív forgatókönyv	1. (A) A jármű megcsúszik a jegen és mozgásképtelenné válik.
Alternatív forgatókönyv	1. (B) A jármű nem tud tovább haladni a hó miatt.
Alternatív forgatókönyv	1. (C) A jármű nekimegy egy másik járműnek és mozgásképtelenné válik.

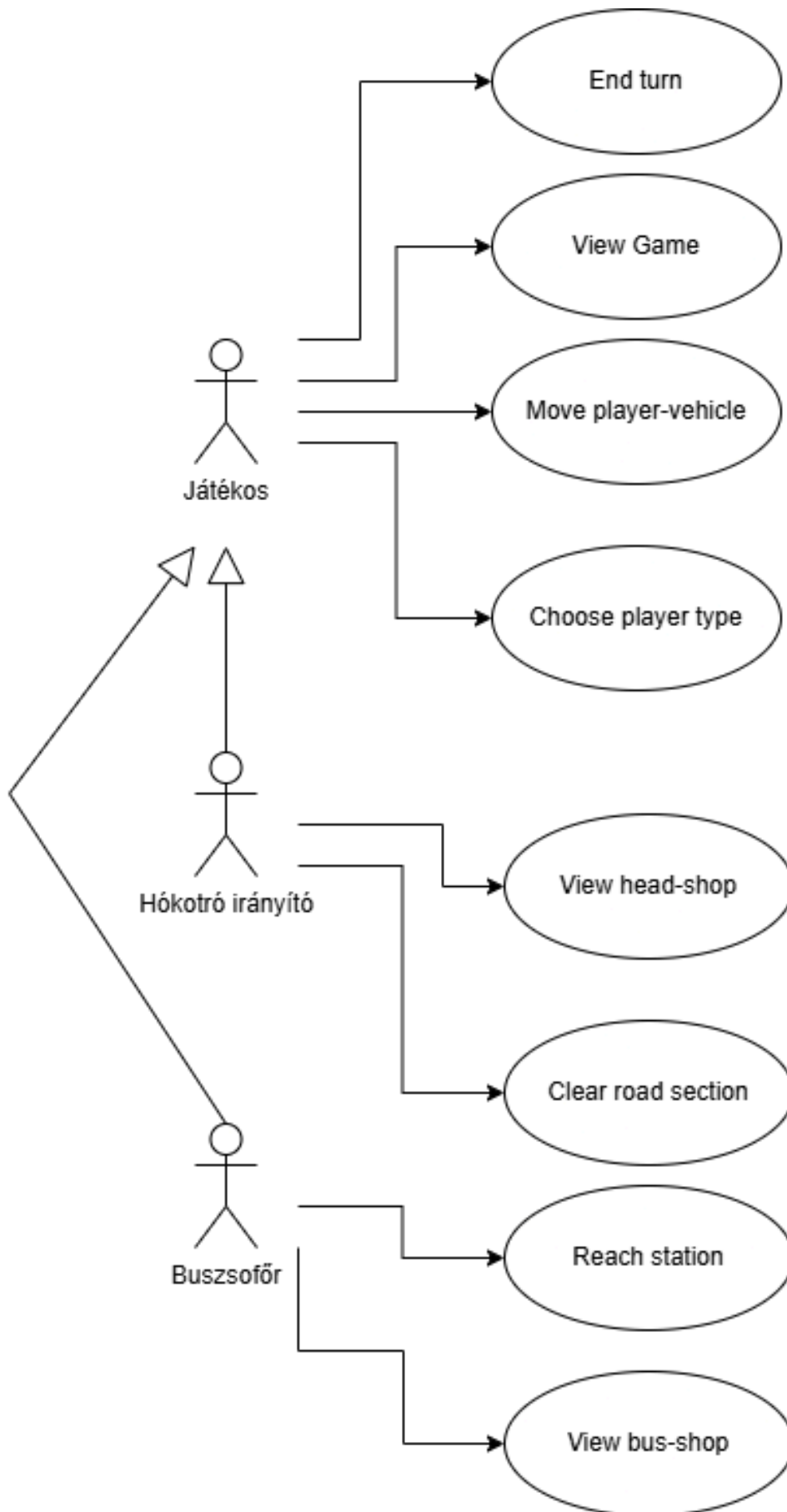
Use-case neve	Choose player type
Rövid leírás	A játékos kiválasztja hogy buszsofőr vagy takarító akar lenni
Aktorok	Játékos
Forgatókönyv	1. A játékos választ járműtípust.

Use-case neve	Clear road section
Rövid leírás	A hókotró takarító letisztít egy adott útszakaszt
Aktorok	Hókotró irányító
Forgatókönyv	1. A takarító letakarítja az adott útszakaszt.
Forgatókönyv	2. A játék pénzt utal a takarítónak.

Use-case neve	View bus-shop
Rövid leírás	Buszvezető megnézi a buszok boltját.
Aktorok	Buszvezető
Forgatókönyv	1. A buszvezető megnézi a buszokat.
Forgatókönyv	2. A buszvezető megvásárolja a jobb buszt.
Alternatív forgatókönyv	2. (A) A buszvezető nem vált egy jobb buszra.

Use-case neve	End turn
Rövid leírás	A játékos befejezi ezzel a körét, ezzel átadva a következőnek.
Aktorok	Játékos
Forgatókönyv	1. A játékos önszántából befejezi az ő körét.
Alternatív forgatókönyv	1. (A) A játékost valami blokkolta, hogy a körét kihasználja így nem önszántából ki hagyja azt.

2.4.2 Use-case diagram



2.5 Szótár

Alagút: Olyan útszakasz, amely egy másik út alatt halad át, ezáltal nem keletkezik kereszteződés.

Autó: A sofőrök által használt jármű lakásuk és munkahelyük közötti közlekedésre. Túl mély hóban elakad, jégpáncélon pedig csúszik. Ütközés után valamennyi kör után eltűnik, addig foglalja a sávot.

Biokerozin: Sárkány fej működéséhez szükséges üzemanyag, pénzért vásárolható.

Busz: Olyan jármű, amely két végállomás között közlekedik, köztük valamennyi megállót érintve. A buszsofőrök tudnak venni jobb buszokat, amikkel több pénzt tudnak keresni. Túl mély hóban elakad, jégpáncélon pedig csúszik.

Buszvezető: Olyan játékos, akinek célja minél több forduló megtétele buszokkal a végállomások között.

Elakadás: Olyan állapot, amikor egy jármű a túl mély hóréteg miatt nem képes tovább haladni az adott sávban.

Feltört jég: Ilyen állapotba helyezi a jégtörő fej a jeget.

Forduló: Egy busz által megtett útvonal az egyik végállomástól a másikig. Buszvezetők célja, hogy minél több legyen.

Hányó fej: A hókotróra szerelhető eszköz, amely a havat vagy a feltört jeget az úttól távolra szórja el. Hatástalan a le nem tört jégpáncéllal szemben.

Havazás: Ennek hatására a körök telésével az összes út folyamatosan havasabb lesz.

Híd: Olyan útszakasz, amely egy másik út felett halad át, ezáltal nem keletkezik kereszteződés.

Hó: Az utakra hulló csapadék, amely akadályozza a közlekedést. Vastagságától függően elakadást okozhat, elég sok jármű által letaposva pedig jégpáncéllá alakul.

Hókotró: A takarítók számára elérhető segédeszköz melyre különböző fejeket lehet szerelni, ezek segítségével pedig az utakon sávokat lehet takarítani.

Játékos: Lehet buszvezető, vagy takarító, 3-3 van a játékban mindegyikből.

Jégpáncél: Hórétegből alakul ki, valamennyi jármű áthaladása után. Csúszást okoz a rajta áthaladó járműveknek, ami balesethez vezethet.

Jégtörő fej: A hókotróra szerelhető eszköz, amely a jégpáncélt feltört jéggé alakítja, de ez továbbra is a sávon marad.

Kereszteződés: Olyan csomópont, ahol több, akár 4+ út találkozik.

Kotrófej: Olyan eszköz, amely a hókotróra szerelhető. Több fajta van belőle, a takarítók pénzért tudják ezeket megvásárolni.

Lakás: A sofőrök egyik lehetséges kiindulási vagy érkezési pontja a városban.

Menetrend: A buszsofőröknek valamennyi kör alatt kell eljutni a következő megállóhoz, ezt átlépve csökken a pénz jutalmuk.

Munkahely: A sofőrök egyik lehetséges kiindulási vagy érkezési pontja a városban.

Összecsúszás: Két jármű véletlenszerű érintkezése jégpáncéllal borított úton. Az esemény következtében az érintett sáv járhatatlanná válik. Az ütközött autók eltűnnek, hókotrók nem tudnak ütközni, buszok pedig nem tűnnek el ütközéskor.

Pénz: Takarítók kapják minden egység letisztított útszakaszért, többet kapva, ha 2 aktív busz végállomás között takarítanak. Kotrófejeket, üzemanyagokat, vagy akár új hókotrókat vásárolhatnak. Buszsofőrök sikeres forduló teljesítéséért kapják, és vehetnek belőle jobb buszokat.

Pont: Lásd pénz

Ponttábla: Az a felület, ahol látszik, hogy a játékosok közül ki hány pontot ért el.

Sárkány fej: A hókotróra szerelhető eszköz, amely egy gázturbina segítségével elolvasztja a havat és jeget is. Működéséhez biokerozin szükséges.

Sáv: Az útnak egy része, egy út egy vagy több ilyenből épül fel. A járművek ezeken haladnak, a hókotró pedig sávokat takarít le. A sáv láthatóan havas, jeges, vagy tiszta.

Só: Sószóró fej működéséhez szükséges üzemanyag, pénzért vásárolható.

Sószóró fej: A hókotróra szerelhető eszköz, amely sót szór a sávra. A só valamennyi kör után elolvasztja a havat és a jeget, és megakadályozza, hogy a lehulló hó megmaradjon az utakon. A só által kifejtett olvadási idő a hó és a jég vastagságától függ.

Söprő fej: A hókotróra szerelhető eszköz, amely a havat és a feltört jeget a közvetlenül szomszédos sávba vagy az út szélére tolja.

Takarító: Olyan játékos, akinek célja a hókotrók menedzselése, az utak tisztítása és a segédeszközök beszerzése.

Út: A város helyszíneit összekötő hálózat eleme, amely egy vagy több sávból áll.

Végállomás: A buszok útvonalának két végpontja, amelyek között a járatok közlekednek.

2.6 Projekt terv

2.6.1 Projekt végrehajtásának mérföldkövei

Feladat	Határidő	Felelős(ök)
Követelmény, projekt, funkcionalitás	2026. március 2.	██████████
Analízis modell (I. változat)	2026. március 9.	██████████
Analízis modell (II. változat)	2026. március 16.	██████████
Szkeleton tervezése	2026. március 23.	██████████████████
Szkeleton elkészítése	2026. március 30.	██████████████████
Prototípus koncepciója	2026. április 13.	██████████
Részletes tervek	2026. április 20.	██████████

Prototípus elkészítése	2026. május 4.	██████████, ██████████
Grafikus változat tervei	2026. május 11.	██████████
Grafikus változat elkészítése	2026. május 27.	██████████
Egyesített dokumentáció	2026. május 29.	██████████

2.6.2 Eszközök, technikák

A csapat tagjai közötti kommunikációra, illetve közös munkamegosztásra alapvetően 3 platformot jelöltünk ki.

- Általános kommunikációs csatorna: Discord, ahol szombatonként tartunk közös meetinget is.
- Dokumentumok megosztása: Google Drive, Codeberg
- Forráskód és egyéb fájlok megosztása: Codeberg

Az alábbi eszközöket használjuk:

- Dokumentációk megszerkesztése: Google Docs
- Modellezés: draw.io, WebSequenceDiagrams
- Fejlesztőkörnyezetek: Visual Studio Code, IntelliJ
- Forráskód megosztása, és verziókezelés: Git, Codeberg

Az alábbi technológiákat használjuk:

- Nyelv: Java 11+
- Automatizálás: Maven (fejlesztés alatt), Windows batch script (fejlesztés alatt, és productionben)
- Verziókezelés: Git

Napló

Kezdet	Időtartam	Résztevők	Leírás
2026.02.21. 10:00:00	1 óra	Funk	Tevékenység: Custom java framework létrehozása.
2026.02.21. 10:00:00	0,8 óra	Funk	Tevékenység: Dokumentáció sablon létrehozása.
2026.02.21. 10:00:00	1 óra	Heltai	Tevékenység: Codeberg repo létrehozása, beállítások kezelése, szabályok lefektetése.
2026.02.21. 10:00:00	0,3 óra	Mindenki	Előzetes értekezlet.
2026.02.25. 14:00:00	0,5 óra	Heltai	Tevékenység: Dokumentum sablonok előkészítése, és dokumentáció bevezetésének elkezdése.
2026.02.25. 15:00:00	1,25 óra	Funk	Tevékenység: 2.4 szakasz ("Lényeges use case-k") megírása; feladatra vonatkozóan utánanézés.
2026.02.25. 15:00:00	1,25 óra	Csordás	Tevékenység: 2.5 szakasz ("Szótár") elkészítése.
2026.02.26. 0:00:00	2,5 óra	Rózsa	Tevékenység: 2.3 szakasz ("Követelmények" táblázat) elkészítése; eddig írtak ellenőrzése; 2.2.2 ("Funkciók") alszakasz elkészítése.
2026.02.26. 14:20:00	0,2 óra	Rózsa	Tevékenység: Külön brainstorming dokumentum létrehozása, és ötletek hozzáadása.

2026.02.26. 16:40:00	0,6 óra	Heltai	Tevékenység: 2.6 szakasz ("Projektterv") első iterációjának megírása és formázása.
2026.02.27. 0:40:00	1,25 óra	Fekete	Tevékenység: 2.4 szakasz ("Lényeges use case-k") kiegészítése, valamint diagram elkészítése és utána nézés a feladatnak
2026.02.28. 14:00:00	1,5 óra	Mindenki	Értekezlet. Döntés: Projekt szakaszainak felelőseit átbeszéltük; sok dolgot a játékdizájnnal kapcsolatban megbeszéltünk és arra jutottunk, hogy turn-based lesz a játék.
2026.02.28. 15:30:00	0,4 óra	Fekete	Tevékenység: 2.4.2 UML use-case diagram megcsinálása.
2026.02.28. 15:30:00	0,4 óra	Funk	Tevékenység: 2.4.1. Use-casek pontosítása.
2026.02.28. 15:50:00	0,2 óra	Funk	Tevékenység: Tech support a google drive-al kapcsolatban
2026.02.28. 16:00:00	0,3 óra	Heltai	Tevékenység: 2.2 szakasz ("Áttekintés") kiegészítése.
2026.03.01. 9:30:00	0,2 óra	Heltai	Tevékenység: Dokumentáció átvizsgálása, formázási hibák kijavítása; munkanapló elvárt formátumba hozása.
2026.03.01. 14:00:00	0,2 óra	Funk	Tevékenység: 2.4.1. Új Use-casek felvétele.

2026.03.01. 14:00:00	0,4 óra	Csordás	Tevékenység: 2.5 szakasz ("Szótár") frissítése.
2026.03.01. 16:45:00	0,9 óra	Rózsa	Tevékenység: 2.2.2. és 2.3. Funkciók és Funkcionális követelmények frissítése.

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 Biokerosene

A Biokerosene a sárkány fej üzemanyaga. Feladata, hogy készletként kezelhető legyen, és a használat során csökkenjen.

Ha elfogy a biokerozin, a sárkány fej nem tud olvasztani.

4.1.2 Bridge

A Bridge egy olyan útszakasz, ami más utak felett halad át. Feladata, hogy biztosítsa az átjutást anélkül, hogy a keresztezett út valódi találkozást jelentene. A Bridge is kezeli a belépést és frissíti az állapotát.

4.1.3 Bus

A Bus a játékos által irányított irányított jármű. Feladata, hogy a menetrendnek megfelelően megállóról megállóra haladjon, és minél több sikeres fordulót illetve megálló-elérést teljesítsen.

A busz képes ütközni más járművekkel; ütközéskor egy időre mozgásképtelenné válhat. A buszhoz fejlesztések kapcsolódhatnak, amelyek az utasok elégedettségét és a jutalmakat befolyásolják.

4.1.4 BusStation

A BusStation egy megállóhelyet reprezentál. Feladata, hogy jelezze, mikor érkezik ide egy busz, és a megállóhoz tartozó eseményeket kezelje, mint például utasok fel- és leszállása, pontozás.

A megálló egyben csomópontként is viselkedik.

4.1.5 Car

A Car a nem játékos által irányított autókat reprezentálja. Feladata, hogy a pályán közlekedjen, és a számára elérhető útvonalak közül mindig a legrövidebb járható utat válassza a cél felé.

Az autó figyelembe veszi az út állapotát, így elakadhat, csúszhat, valamint ütközés esetén ideiglenes akadályt képezhet.

4.1.6 DragonHead

A DragonHead lánggal olvasztja el a havat és a jeget közvetlenül a hókotró saját sávjában. Feladata, hogy a legrosszabb helyzetekben is gyorsan járhatóvá tegye az utat.

A működéséhez biokerozin szükséges, ennek hiányában a fej hatástalan.

4.1.7 IceBreakerHead

Az IcebreakerHead a kialakult jégvéteget feltöri. Feladata, hogy a csúszós jégpáncélt feltörje, és így előkészítse a sávot a további takarításhoz.

A feltört jég önmagában még a sávon maradhat.

4.1.8 Intersection

Az Intersection egy csomópontot jelent, ahol több út találkozhat. Feladata, hogy megadja, merre lehet tovább haladni a csomópontból.

Az Intersection a rá érkező járművel is tud kapcsolatba lépni. Ez alatt értendő például az útválasztás.

4.1.9 Map

A Map a város úthálózatát és a pálya teljes állapotát írja le. Feladata, hogy a játék elején előállítsa a pályát, majd a játék alatt frissítse a környezeti hatásokat (például a folyamatos havazás miatti változásokat).

A Map kezel minden olyan elemet, ami a közlekedéshez tartozik: csomópontokat, útszakaszokat, állomásokat, illetve az ezek közötti kapcsolatokat.

4.1.10 NPCHomeStation

Az NPC-k lakása, vagy munkahelyét jelölheti, amik között navigálnak az autók. Funkcionálhatnak csomópontként is.

4.1.11 Player

A Player egy résztvevőt reprezentál, azaz buszvezetőt vagy takarítót. Feladata, hogy összekösse a játékost az általa irányított járművel.

A Player is nyilvántartja a megszerzett pontokat, és biztosítja a pontok és a jármű lekérdezését.

4.1.12 RoadState

A RoadState egy felsorolás, ami egy útszakasz aktuális állapotát jelöli. Segít egységesen megfogalmazni, hogy egy sáv tiszta, havas vagy épp jeges e.

Az állapot alapján dől el, hogy a járművek tudnak-e haladni, illetve milyen kockázatokkal kell majd szembenéznük.

4.1.13 RoadSection

A RoadSection egy konkrét, bejárható sávot jelent két csomópont között. Feladata, hogy összekösse a két végpontot, és tárolja, mennyire havas vagy jeges az adott sáv.

A RoadSection dönt arról, hogy egy jármű beléphet-e rá (például nincs-e baleset miatti lezárás), és körönként frissíti a saját állapotát.

4.1.14 Tunnel

A Tunnel a hídhoz hasonlóan útszakasz, csak épp más utak alatt halad. Feladata, hogy az átjutás itt se hozzon létre valódi kereszteződést a felszíni úttal. Szintúgy kezeli a belépést és az állapotának a frissítését.

4.1.15 TurnController

A TurnController felel az idő múlásáért. Ő lépteti a játékot körönként, és nyilvántartja, hogy éppen hányadik körnél tartunk.

A TurnController egy listát kezel azokról a szereplőkről, akiknek körönként frissülniük kell, és minden körben sorban értesíti őket.

4.1.16 Salt

A Salt a sószóró fej működéséhez szükséges anyag. Feladata, hogy készletként nyilvántartható legyen, és a sószórás használata során fogyjon.

Ha nincs elegendő só, a sószóró fej nem képes kifejteni az olvasztó és megelőző hatását.

4.1.17 SalterHead

A SalterHead sót juttat a sávra. Feladata, hogy idővel megolvasztja a havat és a jeget, és egy ideig megakadályozza, hogy az újonnan hulló hó megmaradjon az adott szakaszon.

A fej hatékonysága a rendelkezésre álló sókészlettől függ. Ha elfogy a só, a sószórás nem fejt ki hatást.

4.1.18 Schedule

A Schedule a buszok útvonalát és megállóit írja le. Feladata, hogy megadja, mely megállókat kell a busznak sorban érintenie és mikor.

4.1.19 SnowPlow

A SnowPlow a játékos által irányított hókotró. Feladata, hogy egy adott sávban haladva eltakarítsa a havat és/vagy a jeget, és így járhatóvá tegye az utakat.

A hókotró nyilvántartja, milyen fej van éppen felszerelve, milyen fejlesztést használ, illetve mennyi sót és biokerozint tartalekol a speciális fejek működéséhez.

4.1.20 SnowPlowShop

A SnowPlowShop egy bolt jellegű hely, ahol a hókotrók játékos vásárolhat fejlesztéseket a szerzett pontjait felhasználva. Feladata, hogy a játékos számára megnyissa a kínálatot, kilistázza a megvehető tárgyakat, és ellenőrizze, hogy a játékos megengedheti-e magának a kiválasztott vásárlást.

Sikeres vásárlás esetén a SnowPlowShop gondoskodik róla, hogy a megvett tárgy vagy fejlesztés elérhetővé váljon a játékos számára.

4.1.21 SplittingHead

A SplittingHead a havat és a feltört jeget áttolja egy párhuzamos sávba, vagy pedig egy szomszédos útszakaszra. Feladata, hogy a hó ne közvetlenül a szomszédos sávba kerüljön, így nem nehezíti a közlekedést szomszédos sávban.

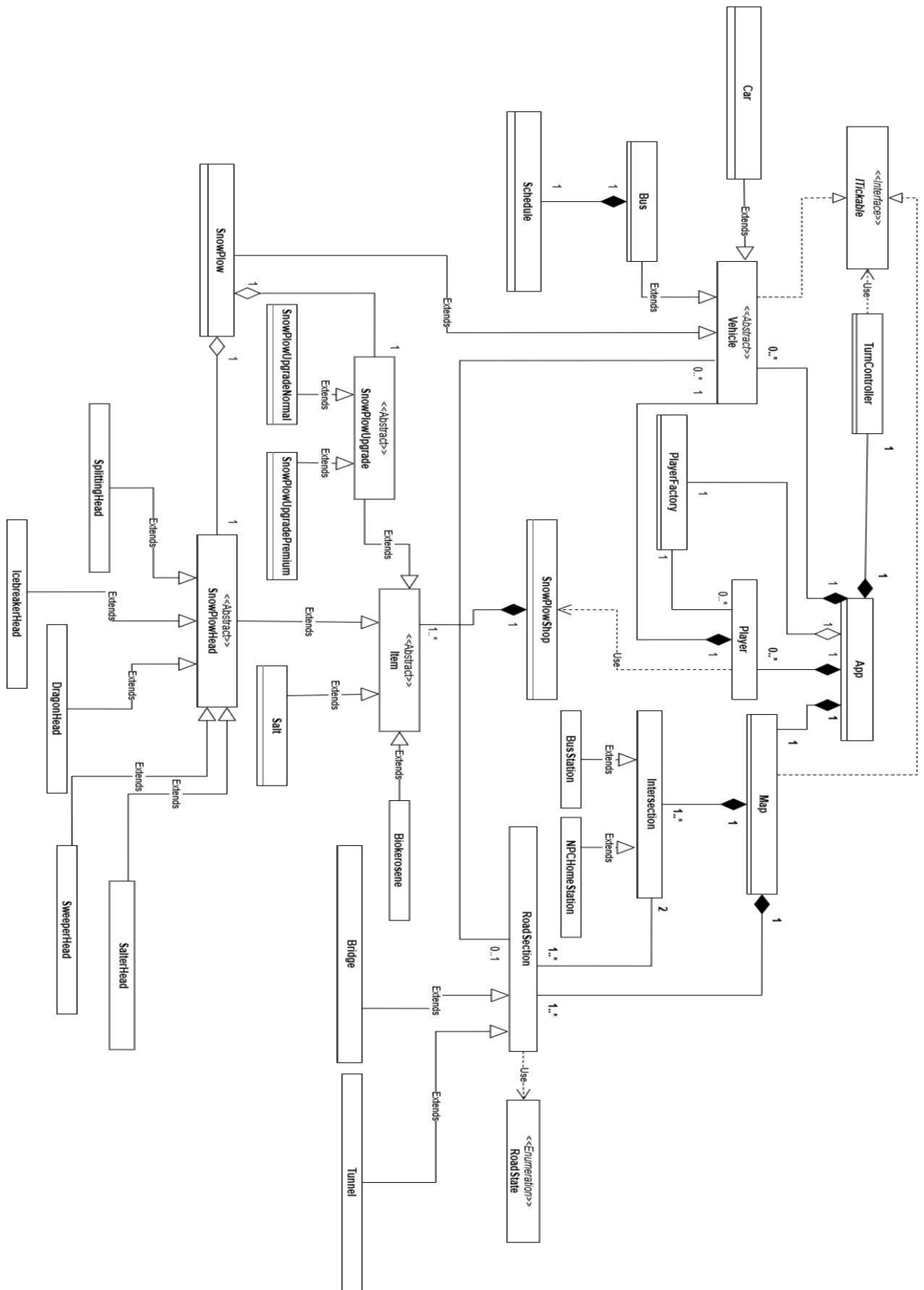
Jég esetén ez a fej is csak a már feltört jeget tudja eltávolítani.

4.1.22 SweeperHead

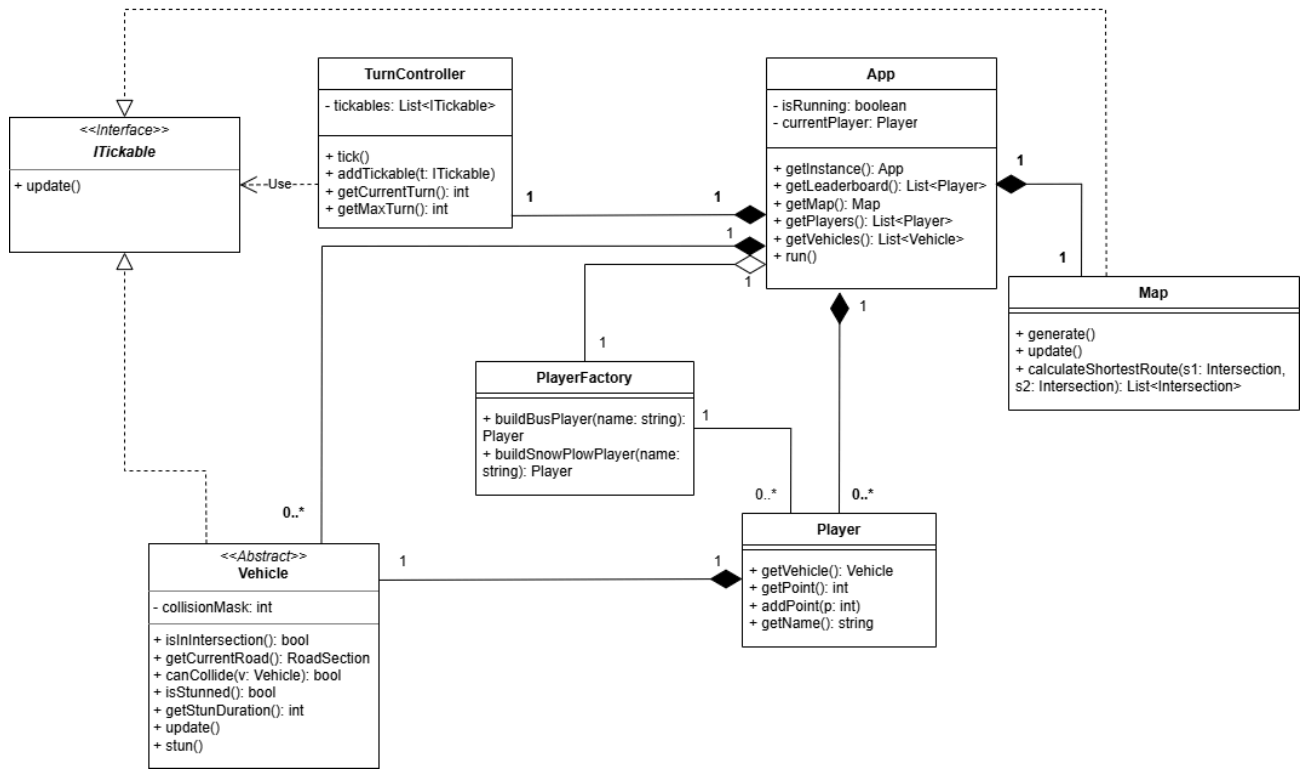
A SweeperHead a havat és a feltört jeget a hókotró melletti területre tolja. Feladata, hogy megoldást adjon a hóréteg eltávolítására, ugyanakkor a közvetlen szomszédos sávot terhelheti a félretolt hóval.

Jég esetén csak akkor hatékony, ha a jég már feltört állapotban van.

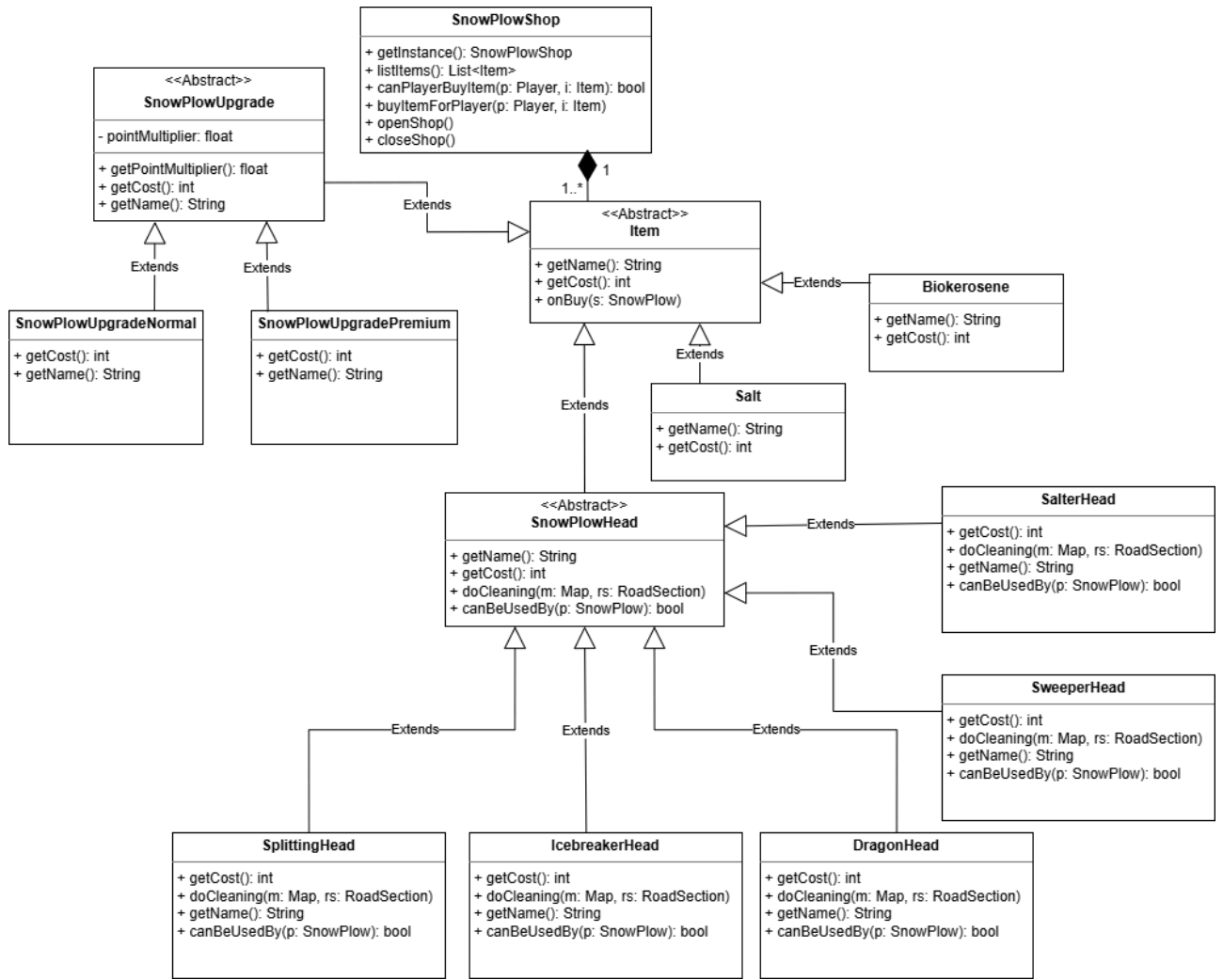
4.2 Statikus struktúra diagramok



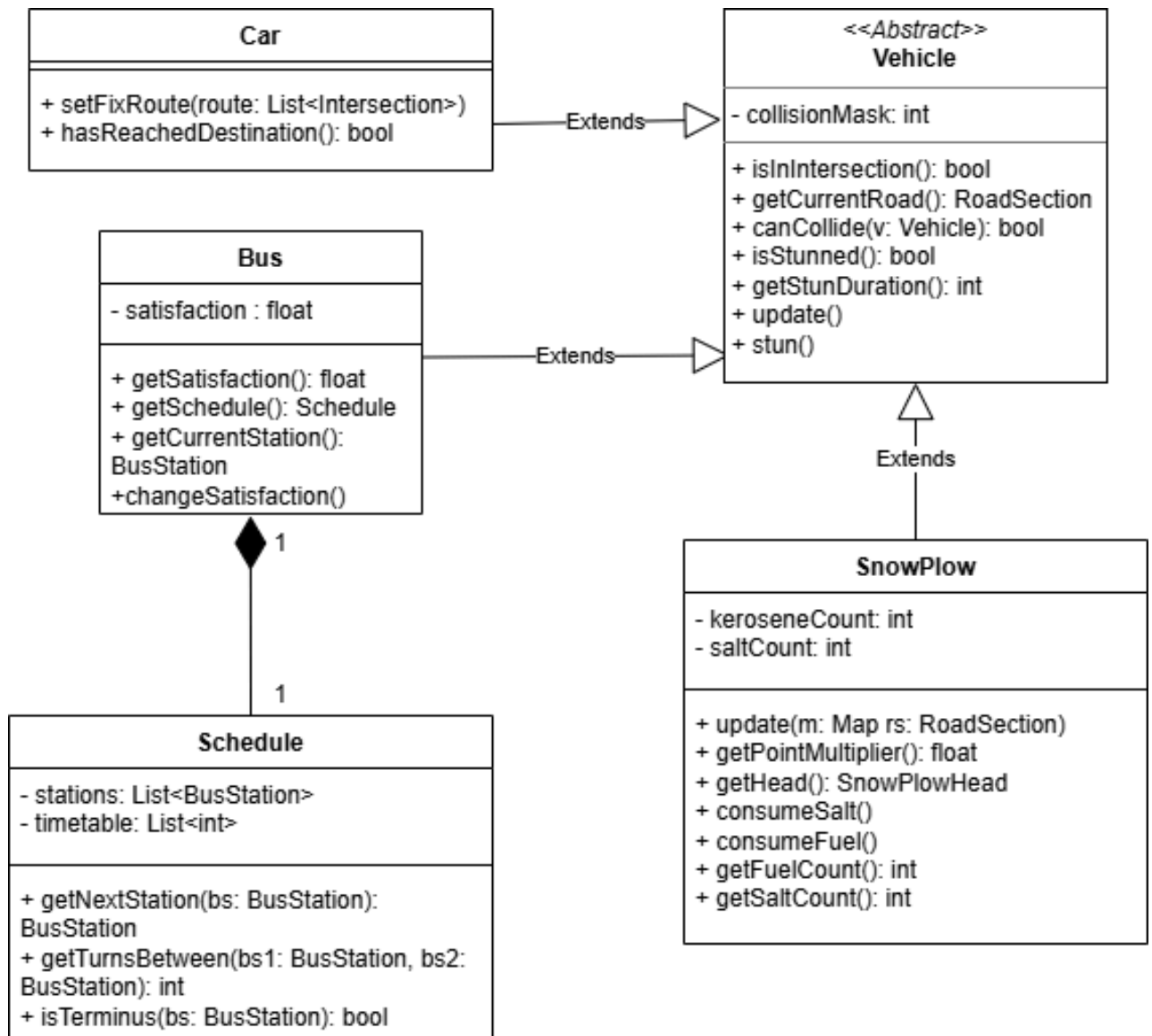
Teljes diagram (csak nevek)



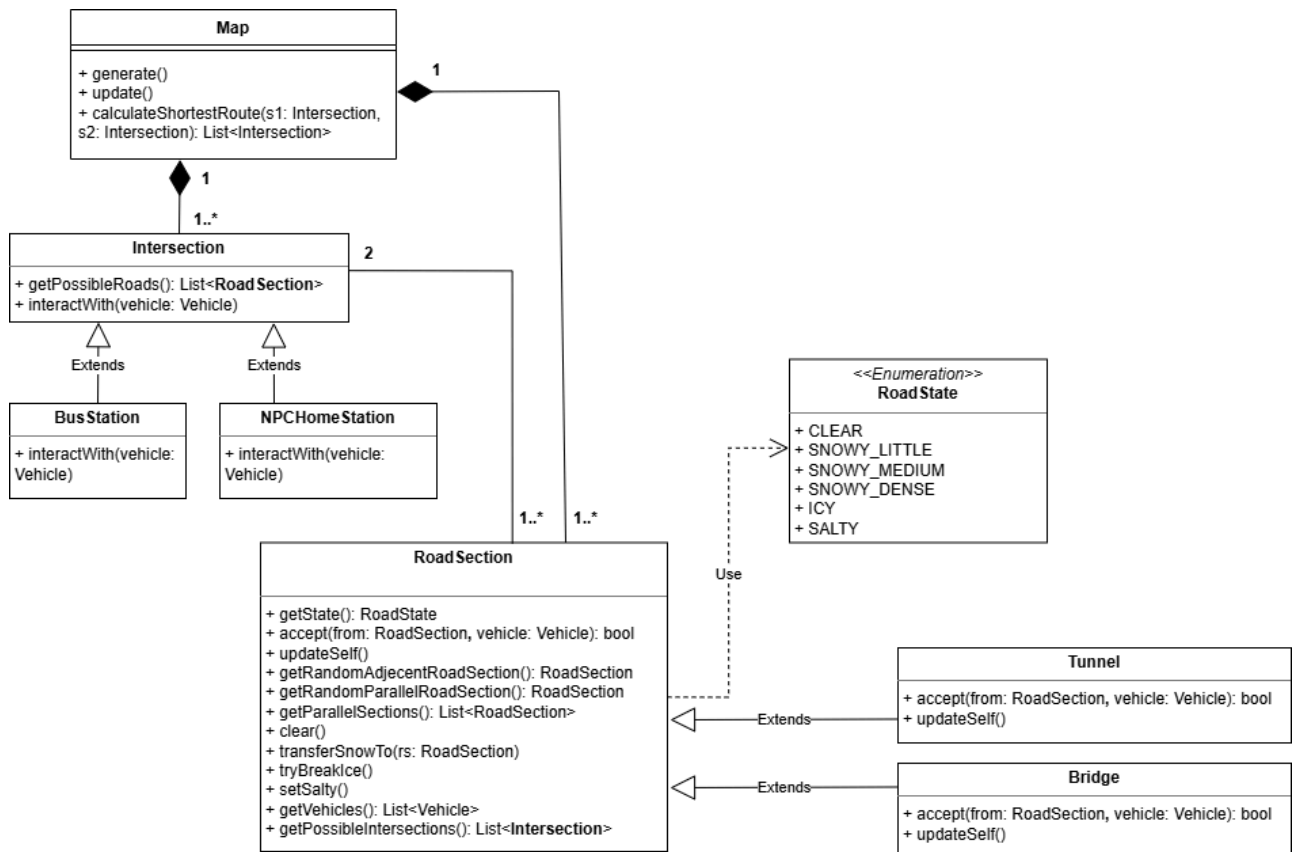
Core



Shop (bolt)



Vehicle (járművek)



Map (pálya)

4.3 Osztályok leírása

4.3.1 App

- **Felelősség**

Az *App* a játék futásának központi felügyelője. Feladata, hogy a játék indításakor létrehozza és elérhetővé tegye a legfontosabb részeket, mint a pálya, játékosok és járművek. Felelőssége, hogy a játék futása alatt ezek állapotát egyben tartsa.

- **Össztályok**

Nincs

- **Interfészek**

Nincs

- **Asszociációk**

- **kompozíció-Map:** Az *App* pontosan egy pályát tárol (1-1 kapcsolat).
- **aggregáció-PlayerFactory:** Az *App* a játékosok regisztrációja során egy *PlayerFactory* factory-t használ fel (1-1).
- **kompozíció-Player:** Az *App* egy vagy több játékost tárol (1-1..*).
- **kompozíció-TurnController:** Az *App* pontosan egy, a körökért (turns) felelős menedzser osztályt tárol (1-1).
- **kompozíció-Vehicle:** Az *App* egy vagy több járművet tárol (1-1..*).

- **Attribútumok**
 - **bool isRunning:** Fut-e éppen a játék.
 - **Player currentPlayer:** A jelenleg aktív játékos, akinek éppen a köre tart.
- **Metódusok**
 - **App getInstance():** Az *App* egy singleton osztály. Ezzel a metódussal lehet elérni a statikus példányt, illetve legelső meghívásakor hozza azt létre.
 - **Map getMap():** Pálya lekérdezése.
 - **List<Player> getLeaderboard():** A ponttábla; játékosok listája pontszámaik szerinti csökkenő sorrendben.
 - **List<Player> getPlayers():** Játékosok lekérdezése.
 - **List<Vehicle> getVehicles():** A pályán aktív járművek lekérdezése.
 - **void run():** A játék elindításáért és futtatásáért felelős metódus.

4.3.2 Biokerosene

- **Felelősség**

A DragonHead üzemanyaga, amely nélkül a fej használhatatlan.
- **Asszociációk (Itemen keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*on keresztül lehet vásárolni biokerozint, így a bolt tárol (legalább) egy közös *Biokerosene* példányt (1-1..*).
- **Interfészek**

Nincs
- **Ősosztályok**

Item
- **Attribútumok**

Nincs
- **Metódusok**
 - **int getCost():** A tárgy árát adja vissza.
 - **String getName():** A tárgy nevét adja vissza (“Biokerozin”).

4.3.3 Bridge

- **Felelősség**

A *Bridge* egy olyan útszakasz, amely a föld fölött haladva elkerüli a többi útszakasszal való találkozást, így nem alakít kereszteződést más, nem híd típusú útszakasszal.
- **Ősosztályok**

RoadSection
- **Interfészek**

Nincs

- **Asszociációk (*RoadSection*ön keresztül)**
 - **kompozíció-Map:** Egy *Map* tárol egy vagy akár több *Bridge*-et (1-1..*).
 - **asszociáció-Intersection:** Egy *Bridge*-t kettő *Intersection* határoz meg, ami a híd 2 végpontja. De egy kereszteződés több útszakasznak is lehet végpontja (2-1..*).
 - **asszociáció-Vehicle:** Egy *Bridge* nyilvántartja a rajta lévő *Vehicle*-öket (1-0..*).

- **Attribútumok**

Nincs

- **Metódusok**

- **bool accept(*RoadSection from*, *Vehicle vehicle*):** Visszaadja logikai értéként hogy az adott *from RoadSection* felől az adott *vehicle Vehicle* át tud-e menni (fel tud-e rá hajtani) ezen a hídon.
- **void updateSelf():** A saját belső állapotát frissítő metódus, pl.: hószint növelése.

4.3.4 Bus

- **Felelősség**

A *Bus* a buszsofőr játékosok járműve. Minden busznak van egy menetrendje, amit teljesítenie kell, és minden fordulóért pontot kap.

- **Ősosztályok**

Vehicle

- **Interfészek**

Nincs

- **Asszociációk**

- **kompozíció-Schedule:** Minden busznak van egy saját menetrendje (1-1).

- **Attribútumok**

float: Satisfaction

- **Metódusok**

- **float getSatisfaction():** Az elégedettségi szorzó lekérdezése. A pontszámításnál fontos.
- **Schedule getSchedule():** A menetrend lekérdezése.
- **BusStation getCurrentStation():** A jelenlegi buszmegálló lekérdezése.
- **Void changeSatisfaction():** Ezzel a metódussal állítjuk át a *satisfaction* azaz elégedettség értékét attól függően mennyire voltunk pontosak a menetrendhez képest

4.3.5 BusStation

- **Felelősség**

A *BusStation*-ök a buszsofőrök céljai. Ha eléri a kijelöltet (következőt), akkor pontot kapnak érte.

- **Ősosztályok**

Intersection

- **Interfészek**

Nincs

- **Asszociációk (*Intersection*ön keresztül)**

- **kompozíció-Map:** Egy *Map* tárol egy vagy több *BusStation*-t (1..*-1).
- **asszociáció-RoadSection:** Egy *BusStation* tartalmaz egy *RoadSection* listát, amiben szerepel minden onnan elérhető *RoadSection* (2-2..*).

- **Attribútumok**

Nincs

- **Metódusok**

- **void interactWith(Vehicle vehicle):** A buszsofőr ennek a függvénynek a segítségével interaktál a *BusStation*ökkel.

4.3.6 Car

- **Felelősség**

A *Car* az NPC autósokat jelöli. Minden autónak van egy fix útvonala, ami mindig a legrövidebb útvonal a lakása és munkahelye között.

- **Ősosztályok**

Vehicle

- **Interfészek**

Nincs

- **Asszociációk (*Vehicle*-ön keresztül)**

- **kompozíció-App:** A program akárhány autót tartalmazhat (0..*-1).
- **aggregáció-RoadSection:** Egy *RoadSection* eltárolja az összes rajta tartózkodó *Car*-t. Közben pedig egy jármű nem biztos, hogy rajta van bármelyik útszakaszon, mert lehet, hogy kereszteződésben van (0..*-0..1).

- **Attribútumok**

Nincs

- **Metódusok**

- **void setFixRoute(List<Intersection> route):** Adott *route* útvonal beállítása.
- **bool hasReachedDestination():** Elérte-e az úticélját.

4.3.7 DragonHead

- **Felelősség**

Egy fajta hókotró fej. Csak akkor használható, ha a játékos rendelkezik *Biokerosene*-al. Ekkor ennek segítségével az adott sávban képes megolvasztani a jeget és a havat, további elkotrásra nincs szükség.

- **Asszociációk (Itemen és keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).
 - **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fej tartozhat (1-1).
- **Interfészek**
Nincs
- **Ősosztályok**
SnowPlowHead → Item
- **Attribútumok**
Nincs
- **Metódusok**
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
 - **void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow* ezt a fejet használva.
 - **bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

4.3.8 IcebreakerHead

- **Felelősség**
Feltöri a jeget egy útszakaszcól, de csak akkor csinál valamit, ha az útszakasz tényleg jeges.
- **Asszociációk (Itemen és keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).
 - **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fej tartozhat (1-1).
- **Interfészek**
Nincs
- **Ősosztályok**
SnowPlowHead → Item
- **Attribútumok**
Nincs
- **Metódusok**
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
 - **void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow* ezt a fejet használva.
 - **bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

4.3.9 Item

- **Felelősség**

A boltban megvásárolható dolgok absztrakt őosztálya.

- **Őosztályok**

Nincs

- **Interfészek**

Nincs

- **Asszociációk**

- **kompozíció-Item:** Egy hókotró bolt több olyan tárgyat tartalmaz, amit a hókotró játékosok meg tudnak venni (1..*-1).

- **Attribútumok**

Nincs

- **Metódusok**

- **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
- **String getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
- **void onBuy(SnowPlow s):** Ez az absztrakt metódus csinál valamit a vevő játékos járművével.

4.3.10 Intersection

- **Felelősség**

Az *Intersection* egy olyan útelem ahol több *RoadSection* találkozik, vagy pedig valamilyen zsákutcát jelöl.

- **Őosztályok**

Nincs

- **Interfészek**

Nincs

- **Asszociációk**

- **kompozíció-Map:** Egy *Map* tárol egy vagy több kereszteződést (1..*-1).
- **asszociáció-RoadSection:** Egy kereszteződés tartalmaz egy *RoadSection* listát, amiben szerepel minden onnan elérhető *RoadSection* (2-2..*).

- **Attribútumok**

Nincs

- **Metódusok**

- **List<RoadSection> getPossibleRoads():** Megadja az *Intersection*ből elérhető lehetséges *RoadSection*öket.
- **void interactWith(Vehicle vehicle):** Ennek a függvénynek segítségével interaktálódik a paraméterben megadott *Vehicle* az adott *Intersection*nel.

4.3.11 ITickable

- **Felelősség**

Ezt az interfészt azok az osztályok implementálják, amely körről-körre valamilyen módon frissítik a belső állapotukat.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Asszociációk**

- **asszociáció-TurnController:** Az *ITickable*-t implementáló osztályokat egy *TurnController* tartja nyilván (1-0..*).

- **Attribútumok**

Nincs

- **Metódusok**

- **void update():** Körönként frissíti a szükséges állapotokat.

4.3.12 Map

- **Felelősség**

A *Map* osztály felelős a játék pályájának kezeléséért és tárolásáért. A játék elején le kell generálnia a pályát, a játék során pedig folyamatosan frissítenie kell a változó részeit.

- **Ősosztályok**

Nincs

- **Interfészek**

ITickable

- **Asszociációk**

- **kompozíció-App:** Az *App* osztály tárolja a játék pályáját (1-1).
- **kompozíció-RoadSection:** A *Map*ot részben *RoadSection*ök alkotják. Egy *Map*-ban egy vagy több *RoadSection* lehet (1-1..*).
- **kompozíció-InterSection:** A *Map*-ot részben *Intersection*ök alkotják. Egy *Map*-ban egy vagy több *Intersection* lehet (1-1..*).

- **Attribútumok**

Nincs

- **Metódusok**

- **void generate():** A játék elején fut le, generál egy random valid pályát *RoadSection*ökből és *Intersection*ökből.
- **void update():** Körönként hívja meg a *TurnController*: frissíti a környezeti hatásokat.
- **List<Intersection> calculateShortestRoute(s1 Intersection, s2 Intersection):** Kiszámítja a legrövidebb utat az adott csomópontok között.

4.3.13 NPCHomeStation

- **Felelőség**

Az *NPCHomeStation*ök között járnak az NPC *Car*-ok.

- **Őosztályok**

Intersection

- **Interfészek**

Nincs

- **Asszociációk** (*Intersection*ön keresztül)

- **kompozíció-Map:** Egy *Map* tárol egy vagy több *NPCHomeStation*-t (1..*-1).
- **asszociáció-RoadSection:** Egy keresztződés tartalmaz egy *RoadSection* listát, amiben szerepel minden onnan elérhető *RoadSection*, beleértve az *NPCHomeStation*öket (2-2..*).

- **Attribútumok**

Nincs

- **Metódusok**

- **void interactWith(Vehicle vehicle):** Interakciókezelés járművekkel.

4.3.14 Player

- **Felelőség**

A *Player* osztály felelős a játékosok járművének tárolásáért, továbbá rajta keresztül kezelhető egy játékos pontszáma.

- **Őosztályok**

Nincs

- **Interfészek**

Nincs

- **Asszociációk**

- **kompozíció-App:** Egy *App* tárol akárhány *Player*-t (0..*-1).

- **kompozíció-Vehicle:** Egy *Player* tárol egy *Vehicle*, egy *Vehicle* pedig egy *Player*-hez tartozik (1-1).
- **asszociáció-PlayerFactory:** A *PlayerFactory* a *Player* példányok létrehozásához kapcsolódik. A *PlayerFactory* a kapott név alapján előállítja a megfelelő játékos objektumot és azt visszaadja a hívó félnek (0..*-1).
- **asszociáció-SnowPlowShop:** A hókotrós *Player*-ök használják a *SnowPlowShop*ot, hogy *Item*eket vegyenek (1-1).

- **Attribútumok**

Nincs

- **Metódusok**

- **Vehicle getVehicle():** Visszaadja a játékos járművét.
- **int getPoint():** Visszaadja a játékos pontszámát.
- **void addPoint(int p):** Paraméterben kapott mennyiségű pontot ad a játékosnak.
- **string getName():** Visszaadja a játékos nevét.

4.3.15 PlayerFactory

- **Felelősség**

A *PlayerFactory* a játékosok létrehozásáért felel, és hozzájuk rendeli a kezdő járművet.

- **Őosztályok**

Nincs

- **Interfészek**

Nincs

- **Asszociációk**

- **aggregáció-App:** Az *App* a játékosok létrehozásakor egy *PlayerFactory*ot tartalmaz és használ (1-1).
- **asszociáció-Player:** *Player* példányt hoz létre és ad vissza (1-0..*).

- **Attribútumok**

Nincs

- **Metódusok**

- **Player buildBusPlayer(String name):** Létrehoz egy buszvezető játékost a megadott névvel, és visszaadja a létrejött *Player* példányt.
- **Player buildSnowPlowPlayer(String name):** Létrehoz egy takarító (hókotró) játékost a megadott névvel, és visszaadja a létrejött *Player* példányt.

4.3.16 RoadSection

- **Felelősség**

Egy *RoadSection* két kereszteződés közötti sáv. Összeköttetést biztosít a kereszteződések között, és tárolja az összeköttetés állapotát.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Asszociációk**

- **kompozíció-Map:** Egy *Map* tárol egy vagy több *RoadSection*-t (1..*-1).
- **aggregáció-Intersection:** Egy *RoadSection* tárol két *Intersection*-t, ami az út 2 végpontja. Egy *Intersection* pedig tárol egy *RoadSection* listát, amiben szerepel minden onnan elérhető *RoadSection* (2-1..*).
- **aggregáció-Vehicle:** Egy *RoadSection* tud a rajta lévő *Vehicle*-ökről és fordítva. De előfordulhat, hogy egy sincs rajta (0..1-0..*).

- **Attribútumok**

- **state:** A *RoadState* enumeráció valamelyik értéke. Ez az út jelenlegi állapota.

- **Metódusok**

- **RoadState getState():** Visszaadja az adott útszakasz állapotát.
- **bool accept(RoadSection from, Vehicle vehicle):** Visszaadja logikai értéként hogy az attribútumként kapott *RoadSection* felől az attribútumként kapott *Vehicle* át tud-e menni erre a *RoadSection*-re.
- **void updateSelf():** Körönként hívódik meg, frissíti az útszakasz a saját állapotát (pl. havasabb lesz).
- **RoadSection getRandomAdjacentRoadSection():** Visszaad egy random szomszédos sávot. (Aminek az egyik végpontja egyezik)
- **RoadSection getRandomParallelRoadSection():** Visszaad egy random párhuzamos sávot. (Az a párhuzamos sáv, aminek mind a 2 végpontja egyezik a jelenlegi sávval)
- **List<RoadSection> getParallelSections():** A párhuzamos sávok listáját adja vissza.
- **void clear():** Sáv tisztítását végzi.
- **void transferSnowTo():** Áttolja a havat a másik (szomszédos) sávba.
- **void tryBreakIce():** Az út megkísérel a jég levételét magáról. Csak abban az esetben sikeres, ha jeges az út.
- **void setSalty():** Sózott állapotra állítja a sávot.
- **List<Vehicle> getVehicles():** Lekéri az adott sávon tartózkodó járműveket

4.3.17 RoadState

- **Felelősség**

Egy enumeráció, amelynek értékei a lehetséges útállapotok.

- **Őosztályok**

Nincs

- **Interfészek**

Nincs

- **Értékek**

- **CLEAR:** Tiszta út.
- **SNOWY_LITTLE:** Enyhén havas út.
- **SNOWY_MEDIUM:** Közepesen havas út.
- **SNOWY_DENSE:** Sűrűn havas út.
- **SALTY:** Sóval beszórt út
- **ICY:** Jeges út.

4.3.18 Salt

- **Felelősség**

A *SalterHead* üzemanyaga. Nélküle nem lehet használni a fejet.

- **Interfészek**

Nincs

- **Őosztályok**

Item

- **Asszociációk (Itemen keresztül)**

- **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).

- **Attribútumok**

Nincs

- **Metódusok**

- **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
- **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.

4.3.19 SalterHead

- **Felelősség**

Ha van *Salt* az adott játékosnál, és ezt a fejet használja, akkor besózhatja az utat.

- **Asszociációk (Itemen és keresztül)**

- **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).
- **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fej tartozhat (1-1).

- **Interfészek**

Nincs

- **Ősosztályok**
SnowPlowHead → Item
- **Attribútumok**
Nincs
- **Metódusok**
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
 - **void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow*.
 - **bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

4.3.20 Schedule

- **Felelősség**
Buszok útvonalának és menetrendjének tárolására szolgál.
- **Ősosztályok**
Nincs
- **Interfészek**
Nincs
- **Asszociációk**
 - **kompozíció-Bus:** Egy *Bus*-hoz pontosan egy *Schedule* tartozik (1-1).
- **Attribútumok**
 - **List<BusStation> stations:** a megállókat tárolja el.
 - **List<int> timetable:** a megállók sorrendjét tárolja el.
- **Metódusok**
 - **BusStation getNextStation(BusStation bs):** Ez a metódus visszaadja a következő *BusStation*-t a sorrendből.
 - **int getTurnsBetween(BusStation bs1, BusStation bs2):** Ez a metódus megadja hogy hány kör alatt kell teljesíteni a kettő *BusStation* közötti utat ahhoz hogy a maximális elégedettséget ériük el.
 - **bool isTerminus(BusStation bs):** Ez a metódus megmondja hogy a jelenlegi *Schedule*-ban megadott *BusStation* az végállomás-e.

4.3.21 SnowPlow

- **Felelősség**
A *SnowPlow* a hókotró játékosok járműve. A hókotrókat fejleszteni lehet, és új fejeket lehet venni hozzájuk.

- **Őszosztályok**

Vehicle

- **Interfészek**

Nincs

- **Asszociációk**

- **aggregáció-SnowPlowHead:** A hókotrónak pontosan egy fajta feje van (1-1).
- **aggregáció-SnowPlowUpgrade:** A hókotrónak lehet pontosan egy fejlesztése (1-1).

- **Attribútumok**

- **int keroseneCount:** A hókotró által tárolt biokerozin-mennyiség.
- **int saltCount:** A hókotró által tárolt sómennyiség.

- **Metódusok**

- **void update (m Map, rs RoadSection):** Hókotró állapotának körönkénti frissítése.
- **float getPointMultiplier() :** Pontszorzó lekérése.
- **SnowPlowHead getHead():** Aktuálisan használt fej lekérése.
- **void consumeSalt():** Egységnyi só felhasználása.
- **void consumeFuel():** Egységnyi kerozin elfogyasztása.
- **int getFuelCount():** Felhasználható biokerozin mennyiség lekérése.
- **int getSaltCount():** Felhasználható só mennyiség lekérése.

4.3.22 SnowPlowHead

- **Felelősség**

A *SnowPlow*-ok fejeinek őszosztálya.

- **Asszociációk (Itemen és keresztül)**

- **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).
- **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fej tartozhat (1-1).

- **Interfészek**

Nincs

- **Őszosztályok**

Item

- **Attribútumok**

Nincs

- **Metódusok**

- **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
- **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
- **void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow*.

- **bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

4.3.23 SnowPlowShop

- **Felelősség**
Hókotrórs játékosok boltja.
- **Asszociációk**
 - **kompozíció-Item:** A *SnowPlowShop*ban több *Item*et lehet venni (1..*-1).
 - **asszociáció-Player:** A hókotrórs játékosok képesek megnyitni a boltot (1-1).
- **Interfészek**
Nincs
- **Össztályok**
Nincs
- **Attribútumok**
Nincs
- **Metódusok**
 - **List<Item> listItems():** Visszaadja a *Shop*ban található *Item*eket.
 - **bool canPlayerBuyItem(Player p, Item i):** Megmondja hogy az adott *p Player* meg tudja-e venni a jelenlegi pontjaiból az adott *i Item*-et.
 - **void buyItemForPlayer(Player p, Item i) :** A *p Player* számára megveszi az adott *i Item*et.
 - **SnowPlowShop getInstance():** Singleton példány elérése.
 - **void openShop():** Megnyitja a jelenlegi játékos számára a boltot.
 - **void closeShop():** Bolt bezárása.

4.3.24 SnowPlowUpgrade

- **Felelősség**
Össztály, amiből származnak a *SnowPlow*-hoz tartozó fejlesztések (nem a fejek!).
- **Asszociációk (Itemen és keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).
 - **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fejlesztés tartozhat (1-1).
- **Interfészek**
Nincs
- **Össztályok**
Item

- **Attribútumok**
 - **float pointMultiplier:** Pontszámításnál figyelembe vett pontszorzó.
- **Metódusok**
 - **float getPointMultiplier():** Ezzel a metódussal kérdezhető le a pontszorzó.
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.

4.3.25 SnowPlowUpgradeNormal

- **Felelősség**

A normál verziója a *SnowPlow* fejlesztéseinek.
- **Asszociációk (Itemen és keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).
 - **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fejlesztés tartozhat (1-1).
- **Interfészek**

Nincs
- **Ősosztályok**

SnowPlowUpgrade → Item
- **Attribútumok**
 - **float pointMultiplier:** Pontszámításnál figyelembe vett pontszorzó.
- **Metódusok**
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.

4.3.26 SnowPlowUpgradePremium

- **Felelősség**

A prémium verziója a *SnowPlow* fejlesztéseinek.
- **Asszociációk (Itemen és keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).
 - **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fejlesztés tartozhat (1-1).
- **Interfészek**

Nincs
- **Ősosztályok**

SnowPlowUpgrade → Item

- **Attribútumok**
 - **float pointMultiplier:** Pontszámításnál figyelembe vett pontszorzó.
- **Metódusok**
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.

4.3.27 SplittingHead

- **Felelősség**
Lényegében úgy funkcionál mint egy *SweeperHead*, azonban képes átlökni a havat és a jeget egy párhuzamos sávba, vagy szomszédos útszakaszra.
- **Asszociációk (Itemen és keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*ban lehet venni egyet vagy többet belőle (1..*-1).
 - **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fej tartozhat (1-1).
- **Interfészek**
Nincs
- **Ősosztályok**
SnowPlowHead → Item
- **Attribútumok**
Nincs
- **Metódusok**
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
 - **void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow*.
 - **bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

4.3.28 SweeperHead

- **Felelősség**
Eltolja a havat és jeget a szomszédos sávba. A nem-feltört állapotú jeget viszont nem képes feltörni.
- **Asszociációk (Itemen és keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*ban lehet venni egyet vagy többet belőle (1..*-1).
 - **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fej tartozhat (1-1).
- **Interfészek**

Nincs

- **Ősosztályok**
SnowPlowHead → Item
- **Attribútumok**
Nincs
- **Metódusok**
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
 - **void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow*.
 - **bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

4.3.29 Tunnel

- **Felelősség**

A *Tunnel* egy olyan útszakasz, amely a föld alatt haladva elkerüli a többi útszakasszal való találkozást, így nem alakulnak ki kereszteződések.

- **Ősosztályok**

RoadSection

- **Interfészek**

Nincs

- **Asszociációk (*RoadSection*ön keresztül)**

- **kompozíció-Map:** Egy *Map* tárol egy vagy akár több *Tunnelt* (1-1..*).
- **asszociáció-Intersection:** Egy *Tunnelt* kettő *Intersection* határoz meg, ami az alagút 2 végpontja. De egy kereszteződés több útszakasznak is lehet végpontja (2-1..*).
- **asszociáció-Vehicle:** Egy *Tunnel* nyilvántartja a rajta lévő *Vehicle*-öket (1-0..*).

- **Attribútumok**

Nincs

- **Metódusok**

- **bool accept(RoadSection from, Vehicle vehicle):** Visszaadja logikai értéként hogy az attribútumként kapott *RoadSection* felől az attribútumként kapott *Vehicle* át tud-e menni ezen a *Tunnel*-ön.
- **void updateSelf():** Körönként hívódik meg, frissíti az alagút saját állapotát, de nem lesz folyamatosan havasabb, mint a többi *RoadSection*.

4.3.30 TurnController

- **Felelősség**

Ez az osztály felel mindazon osztályok belső állapotának frissítéséért, amelyek implementálják az *ITickable* interfészt.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Asszociációk**

- **asszociáció-ITickable:** Az *ITickable*-t implementáló osztályokat egy *TurnController* tartja nyilván (1-0..*).

- **Attribútumok**

- **List<ITickable> tickables:** A frissítendő osztály példányok nyilvántartása.

- **Metódusok**

- **void tick():** Nyilvántartásba vett példányok frissítése.
- **void addTickable(ITickable t):** Új példány nyilvántartásba vétele.
- **int getCurrentTurn():** Visszaadja a jelenlegi kör (turn) számát.
- **int getMaxTurn():** Visszaadja, hogy hány körből áll a játék.

4.3.31 Vehicle

- **Felelősség**

A *Vehicle* a járművek közös absztrakt őse. Ez az osztály felel a járművek ütközésének kezeléséért, illetve megteremti a járművek interfészének alapját.

- **Ősosztályok**

Nincs

- **Interfészek**

ITickable

- **Asszociációk**

- **kompozíció-App:** A járművek listáját az *App* tartalmazza, a kapcsolat jellege kompozíció. Az *App* egyet sem, vagy akár több járművet is tárolhat (1-0..*).
- **asszociáció-RoadSection:** Egy útszakasz nyilvántartja a rajta lévő *Vehicle*-öket. Illetve maga a jármű is tud a jelenlegi útszakaszáról (1-0..*).
- **kompozíció-Player:** Minden *Player*-höz tartozik egy *Vehicle* (1-1).

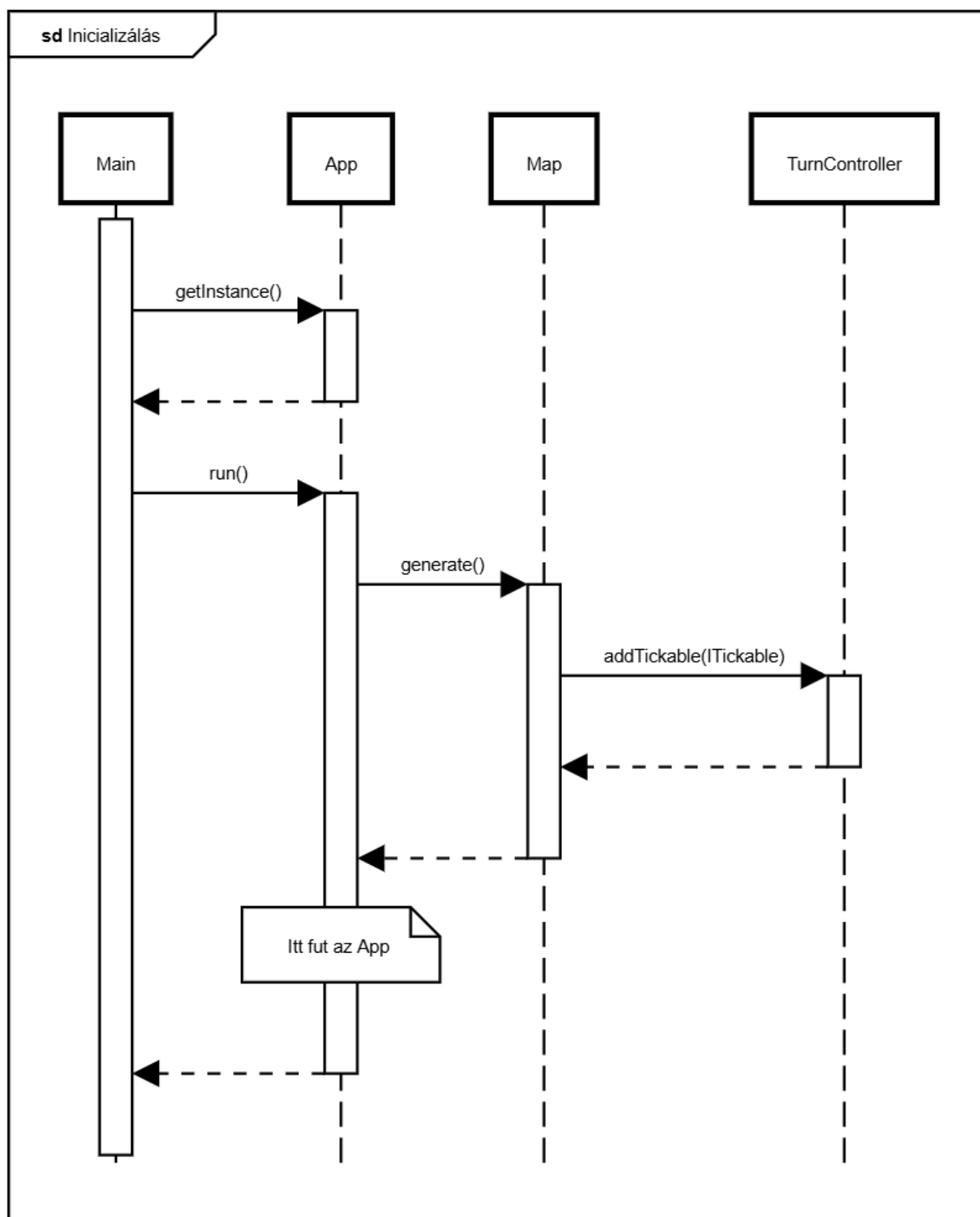
- **Attribútumok**

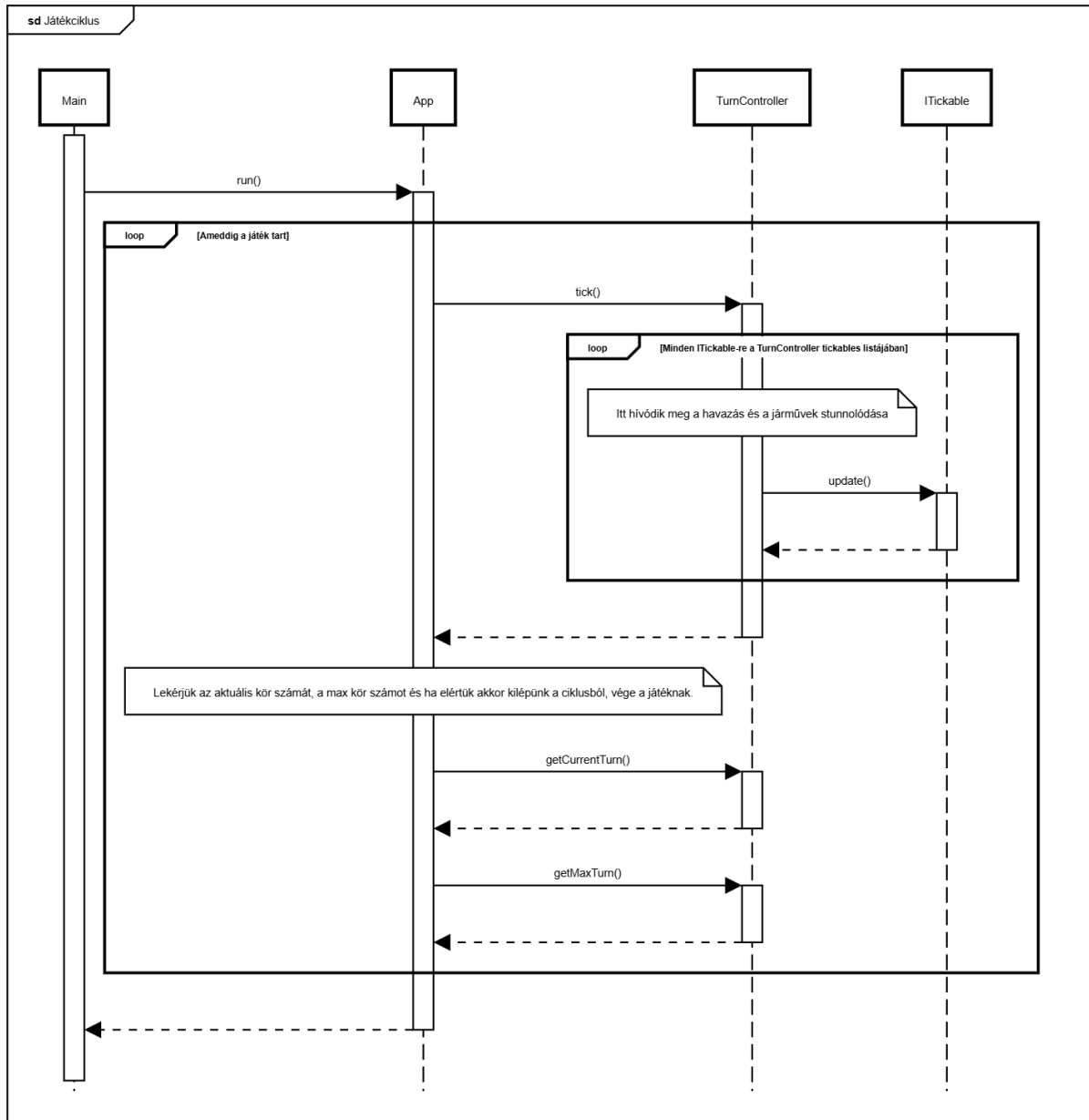
Nincs

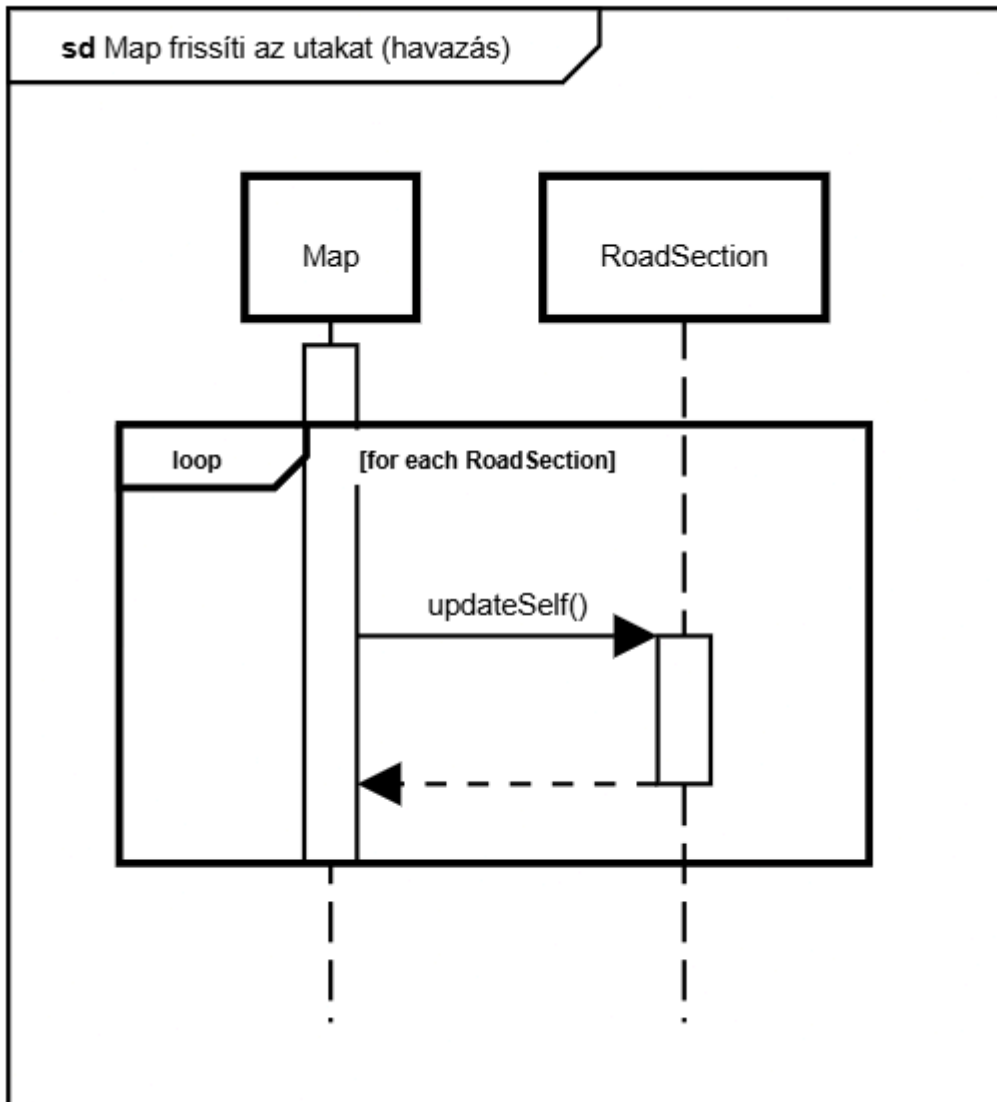
- **Metódusok**

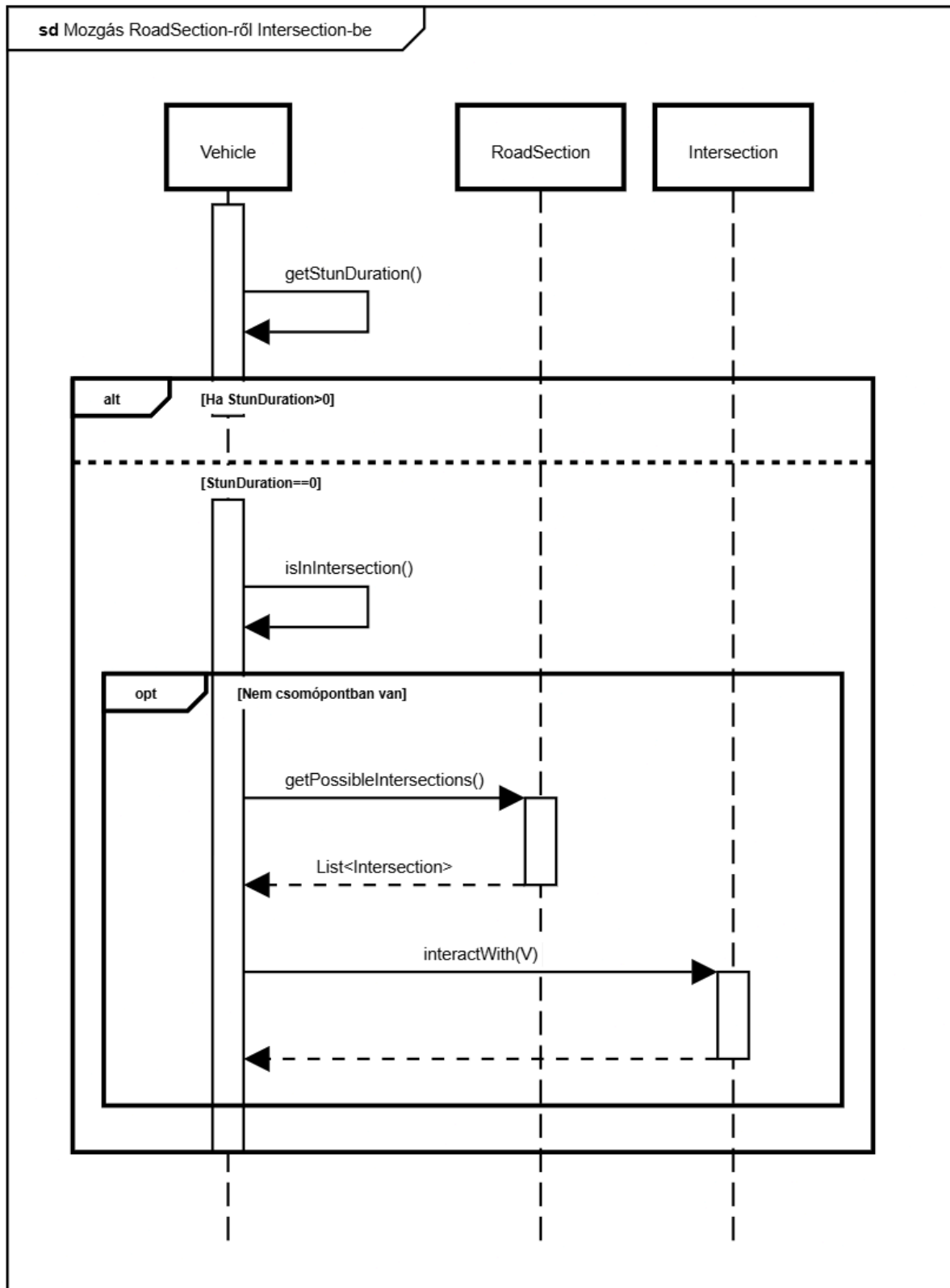
- **bool isInIntersection():** Éppen egy kereszteződésben van-e.
- **RoadSection getCurrentRoad():** Jelenleg melyik útszakaszon (azon belül sávon) tartózkodik - ez egy lépés (turn) közben releváns. Ha éppen nem lépés közben van, akkor null.
- **bool canCollide(Vehicle v):** A v jármű tud-e ezzel a járművel összecsúszni.
- **bool isStunned():** Egy összecsúszás következtében tud-e mozogni, vagy sem.
- **int getStunDuration():** Még hány körig nem mozoghat a legutolsó összecsúszás következtében.
- **void update():** A járművek körönkénti állapotfrissítése.
- **void stun():** Az ütközést követő, mozgásképtelen állapot beállítását végzi.

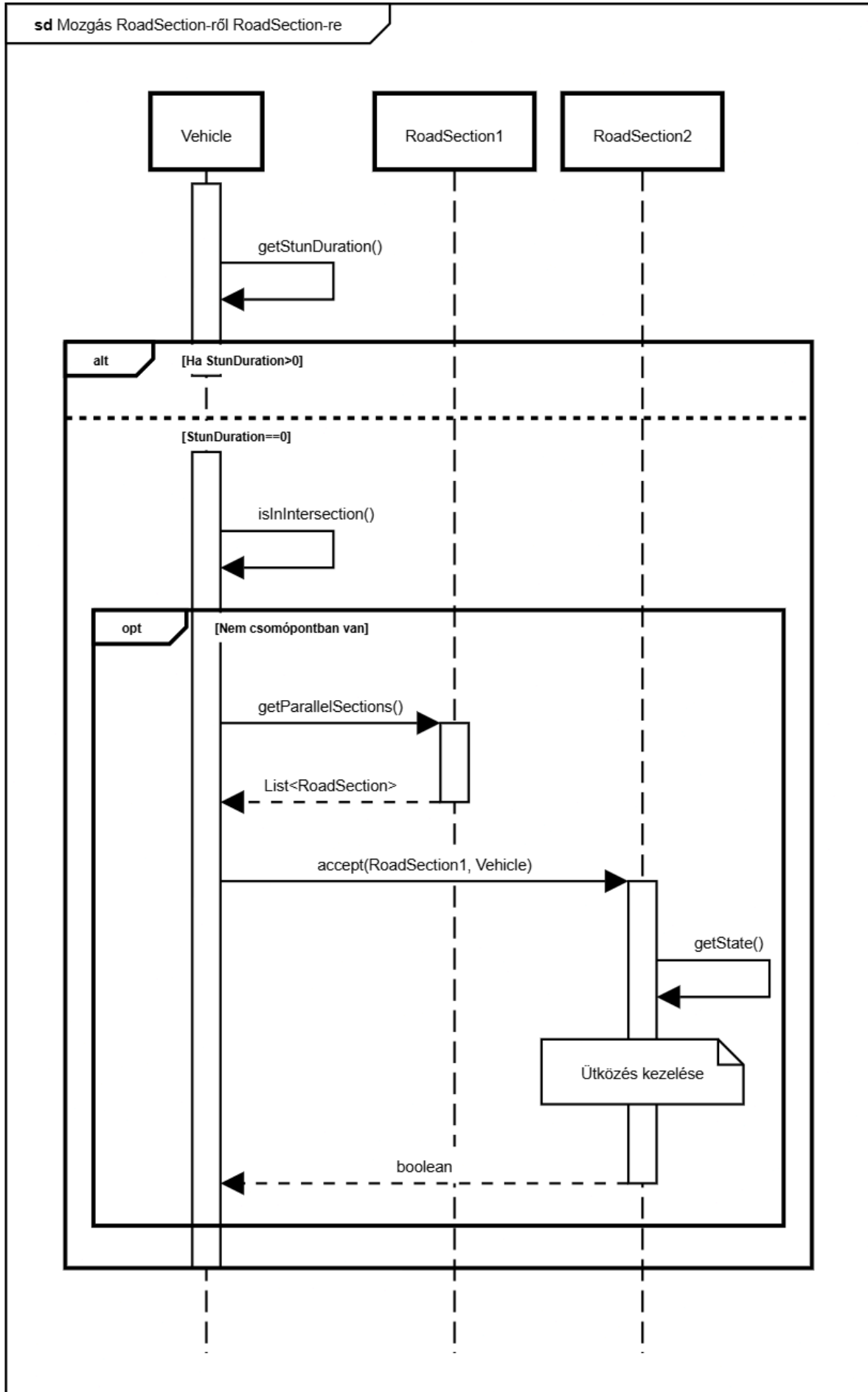
4.4 Szekvencia diagramok

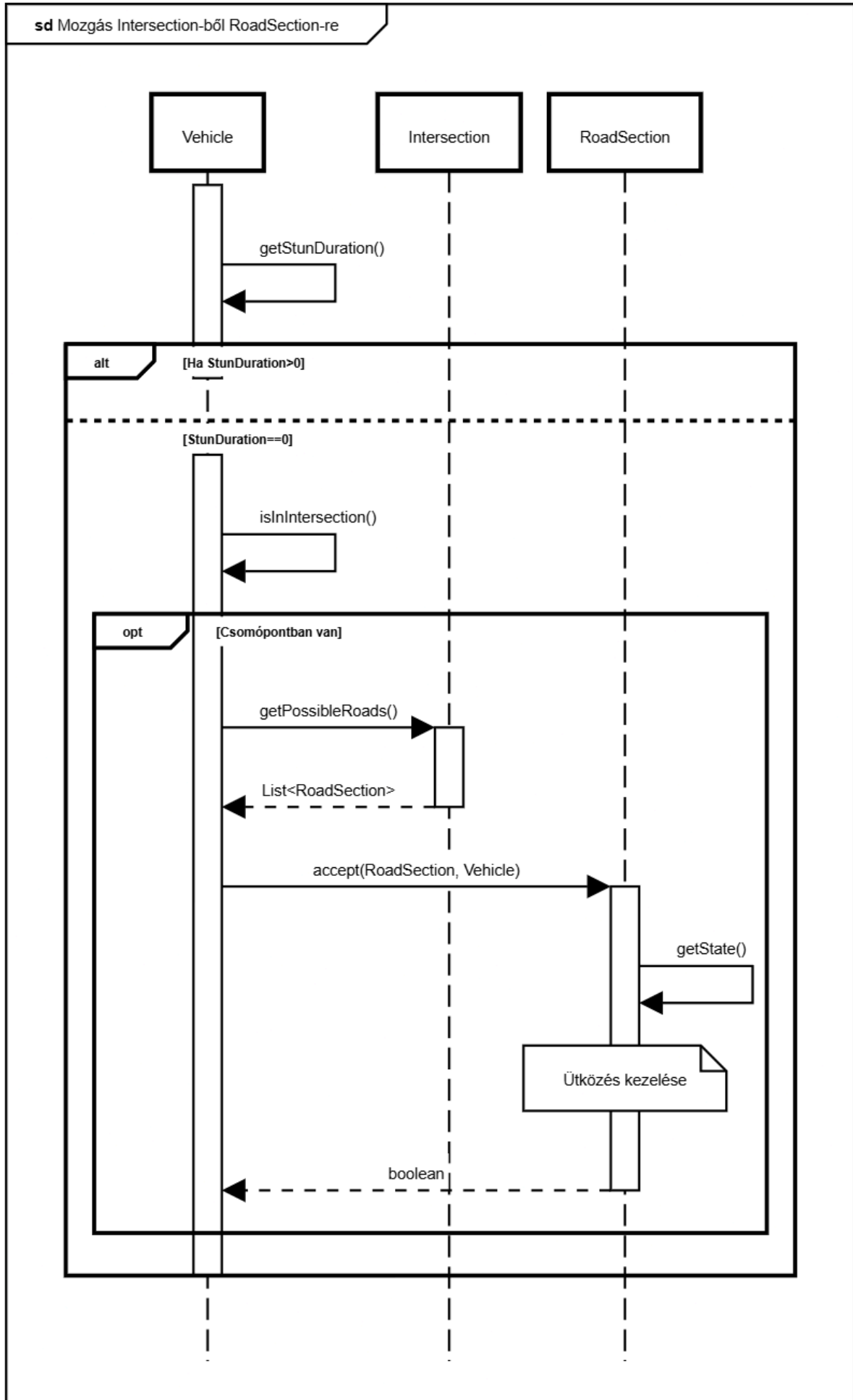


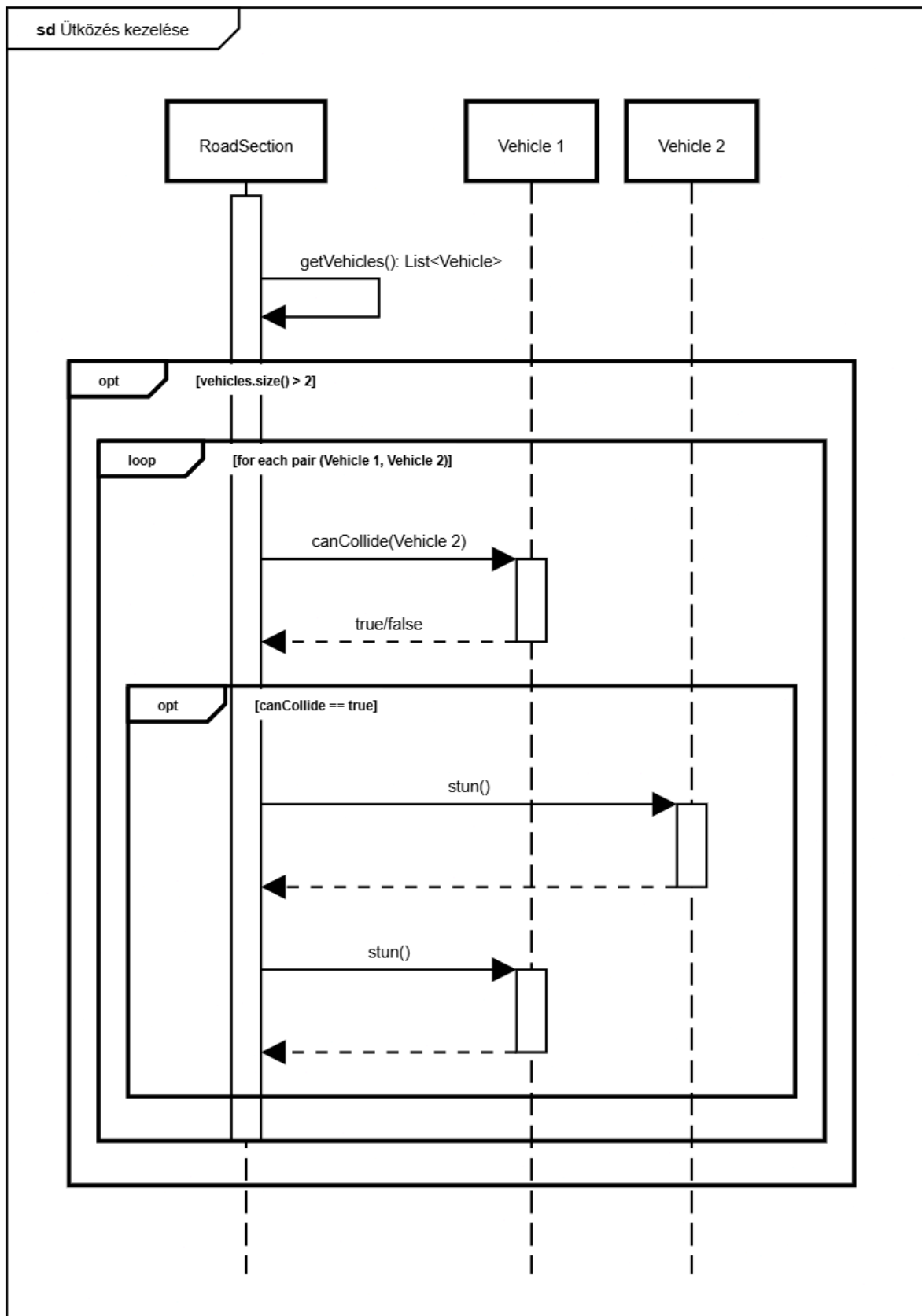


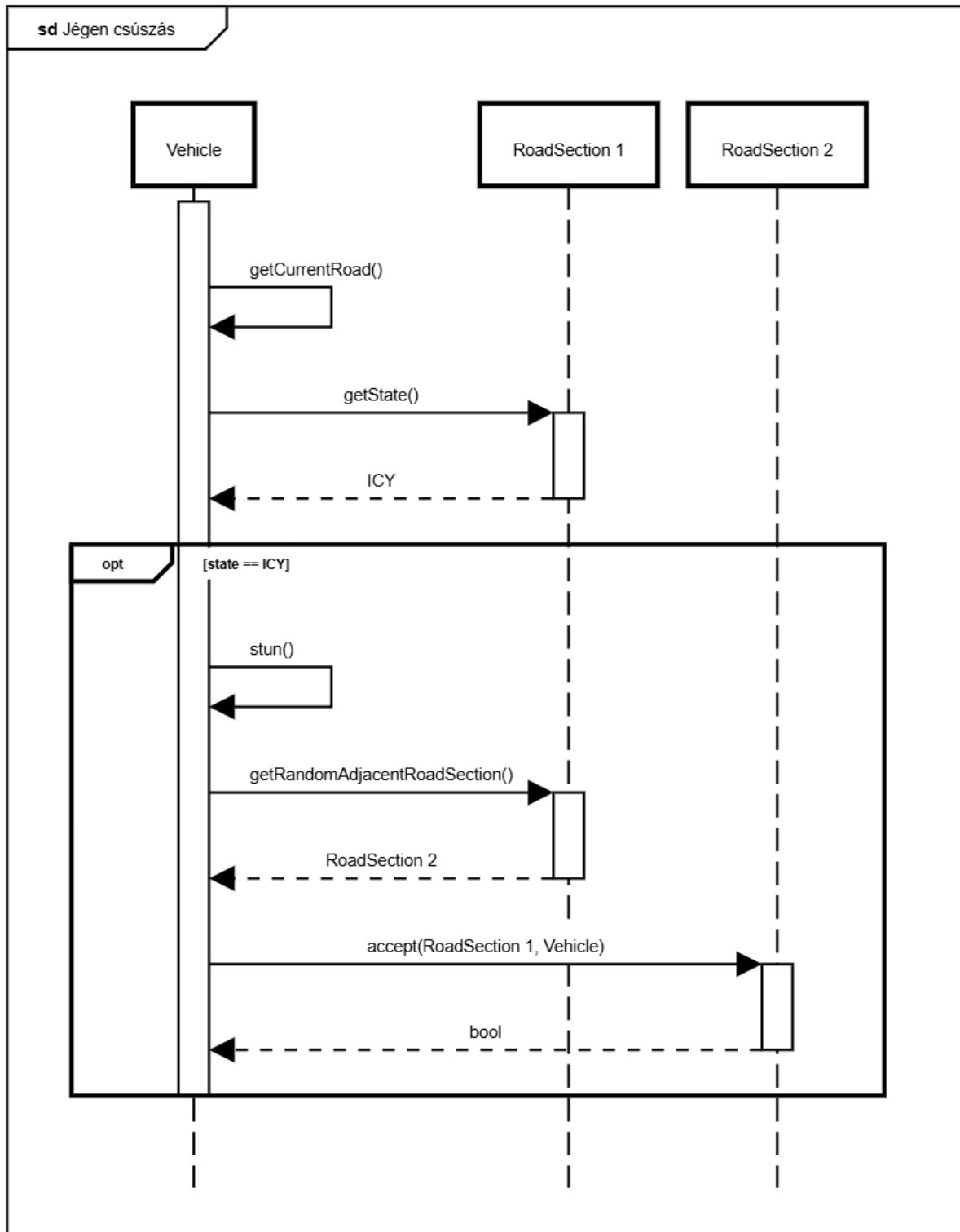


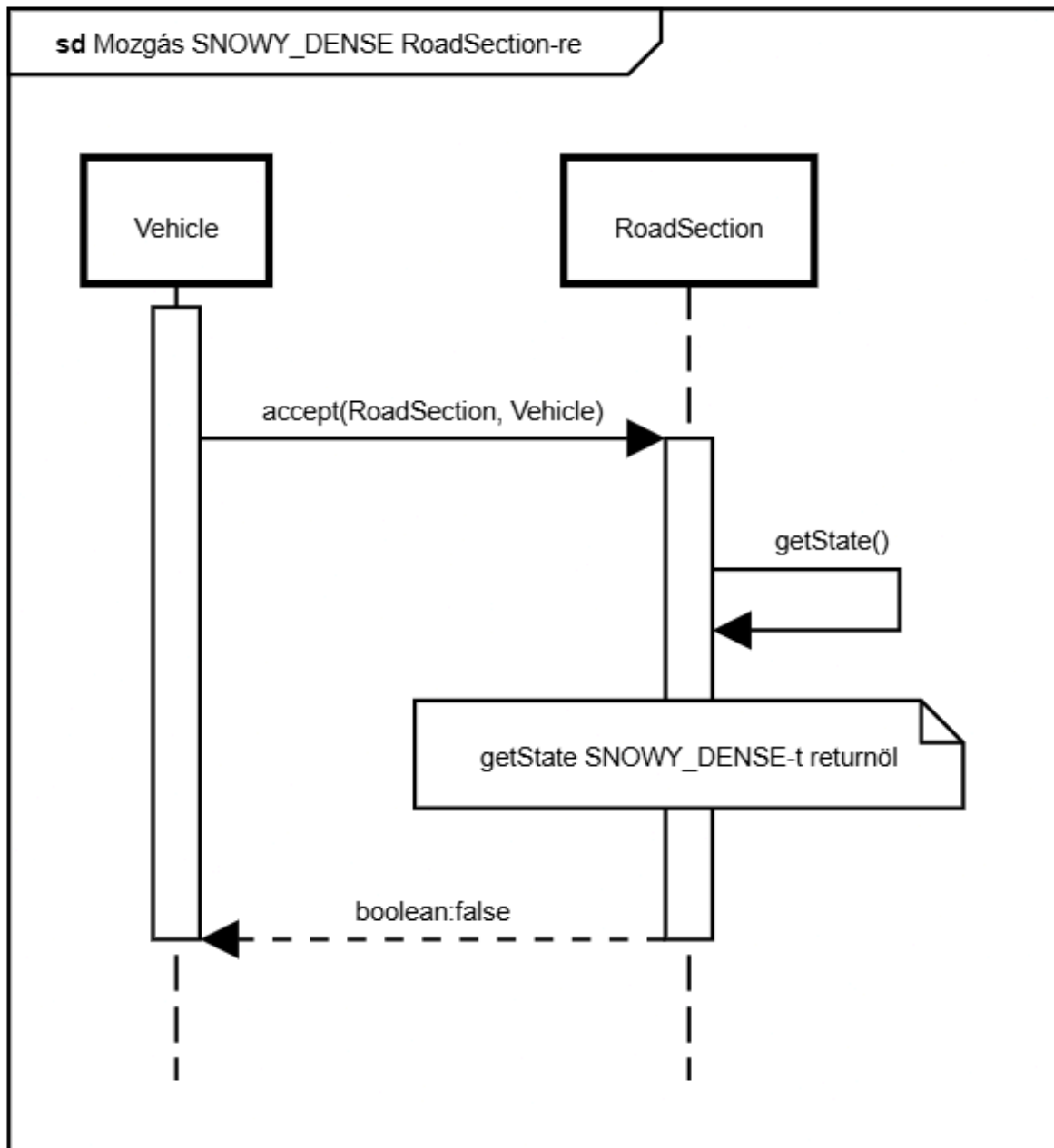


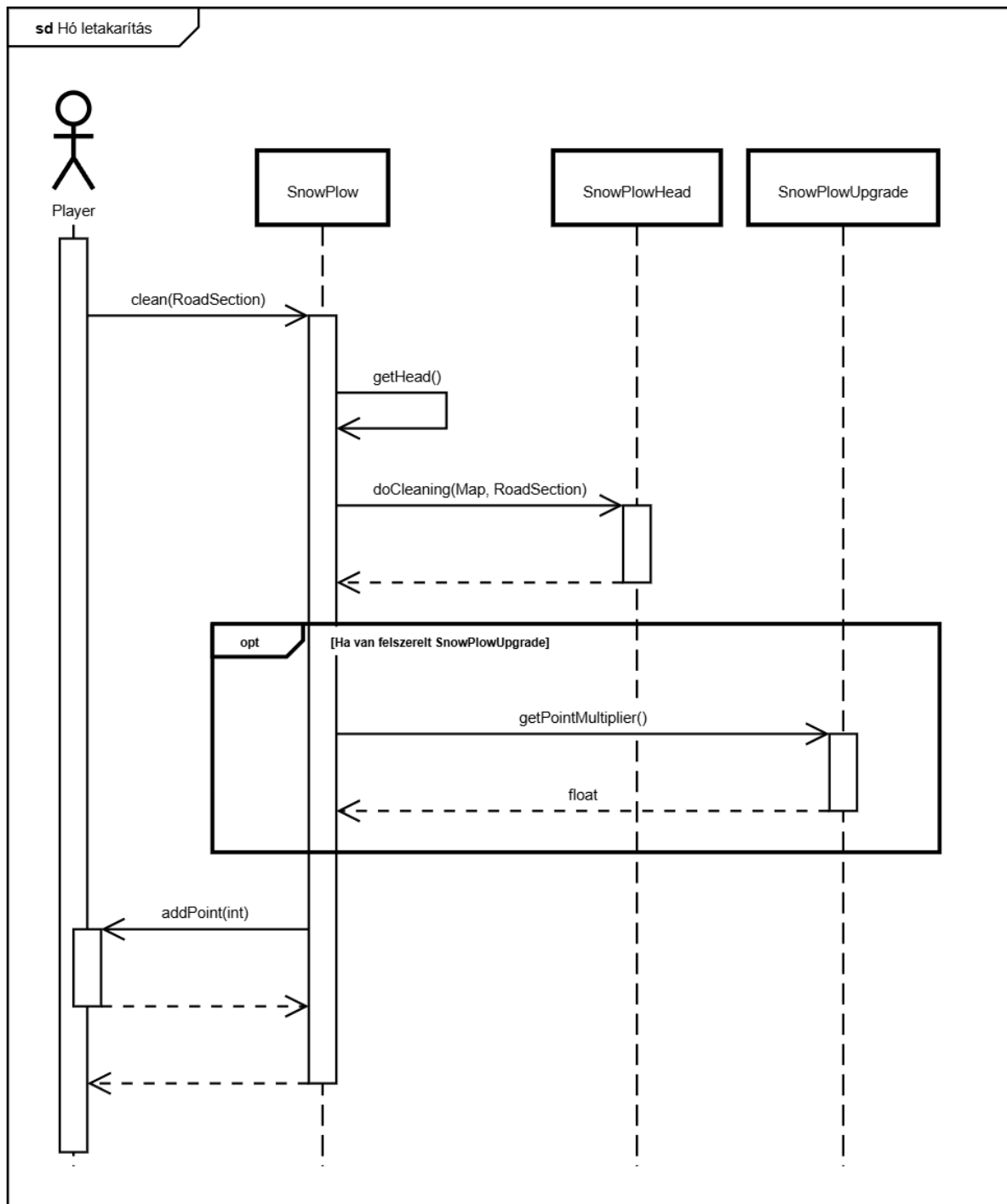


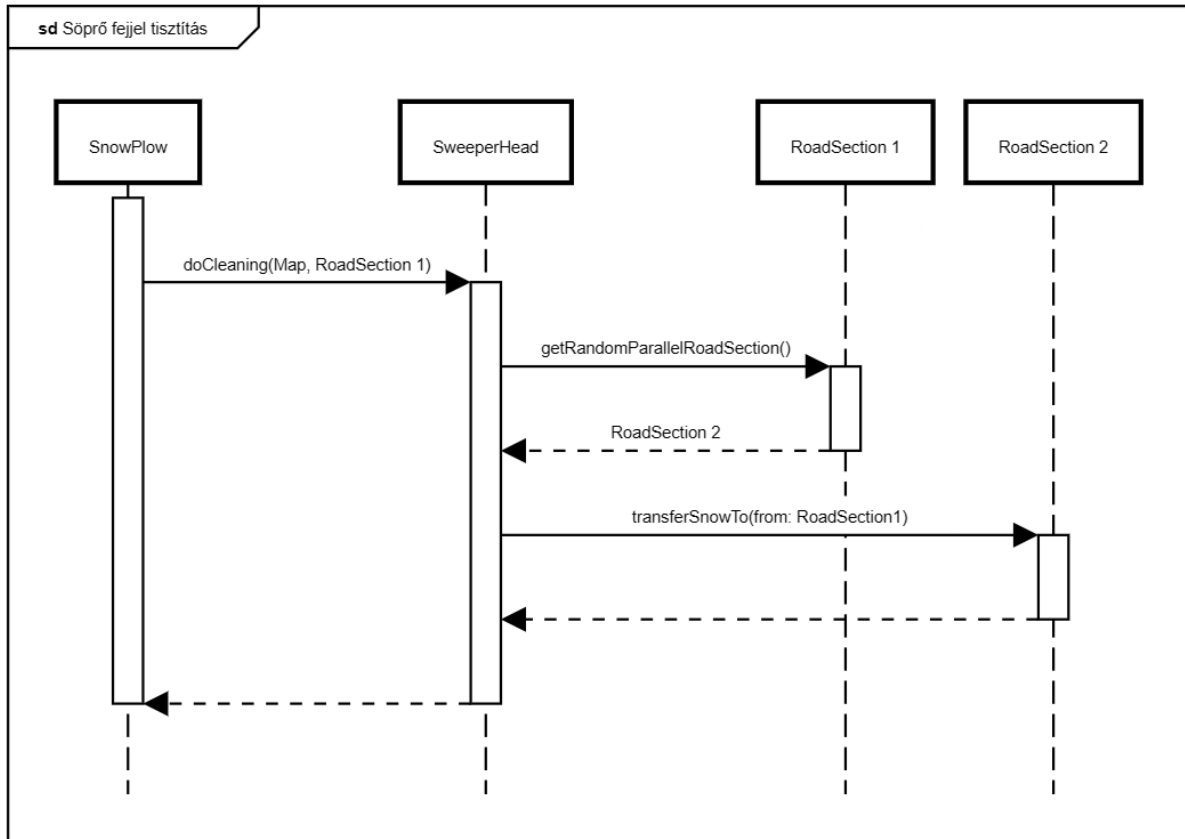


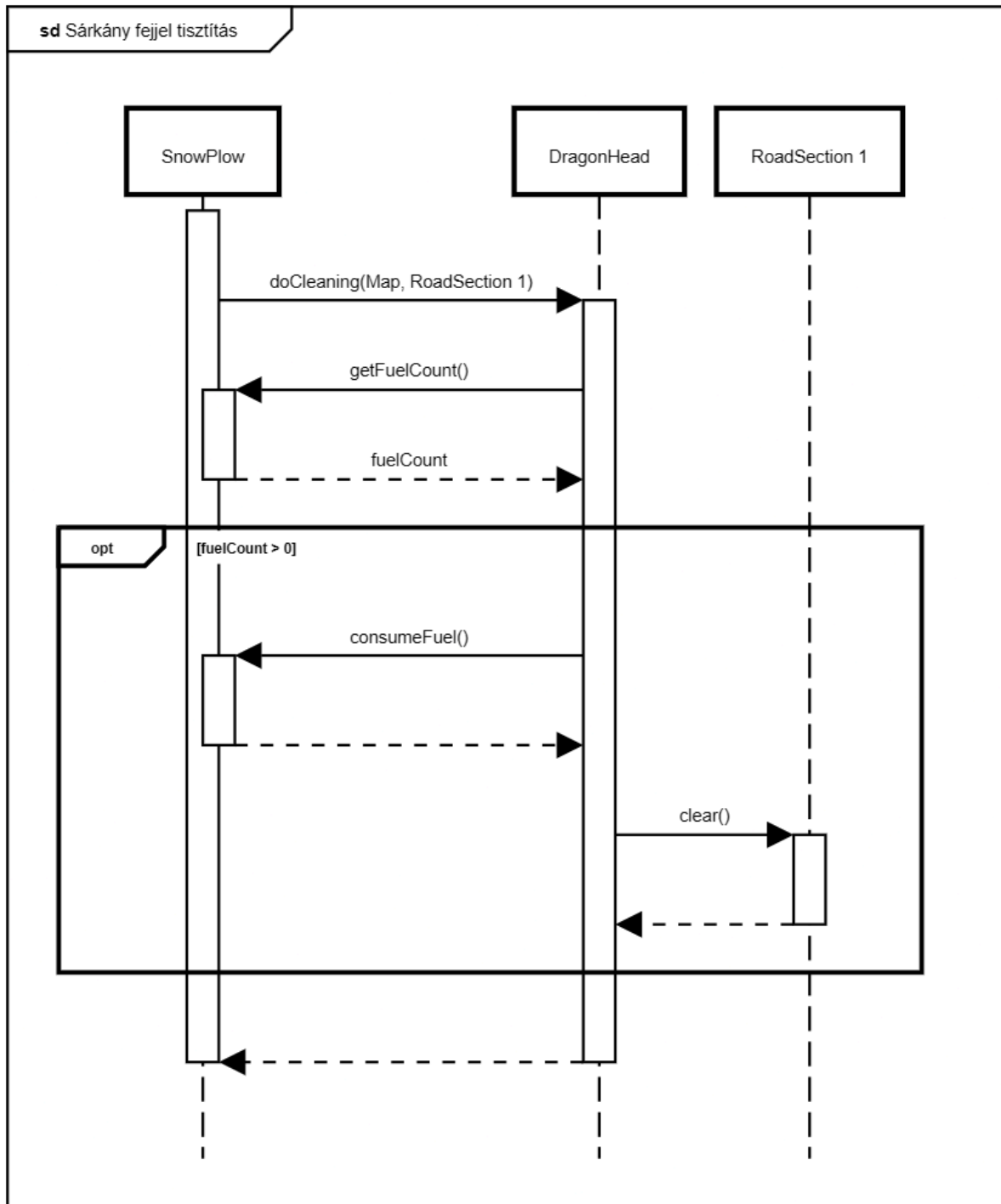


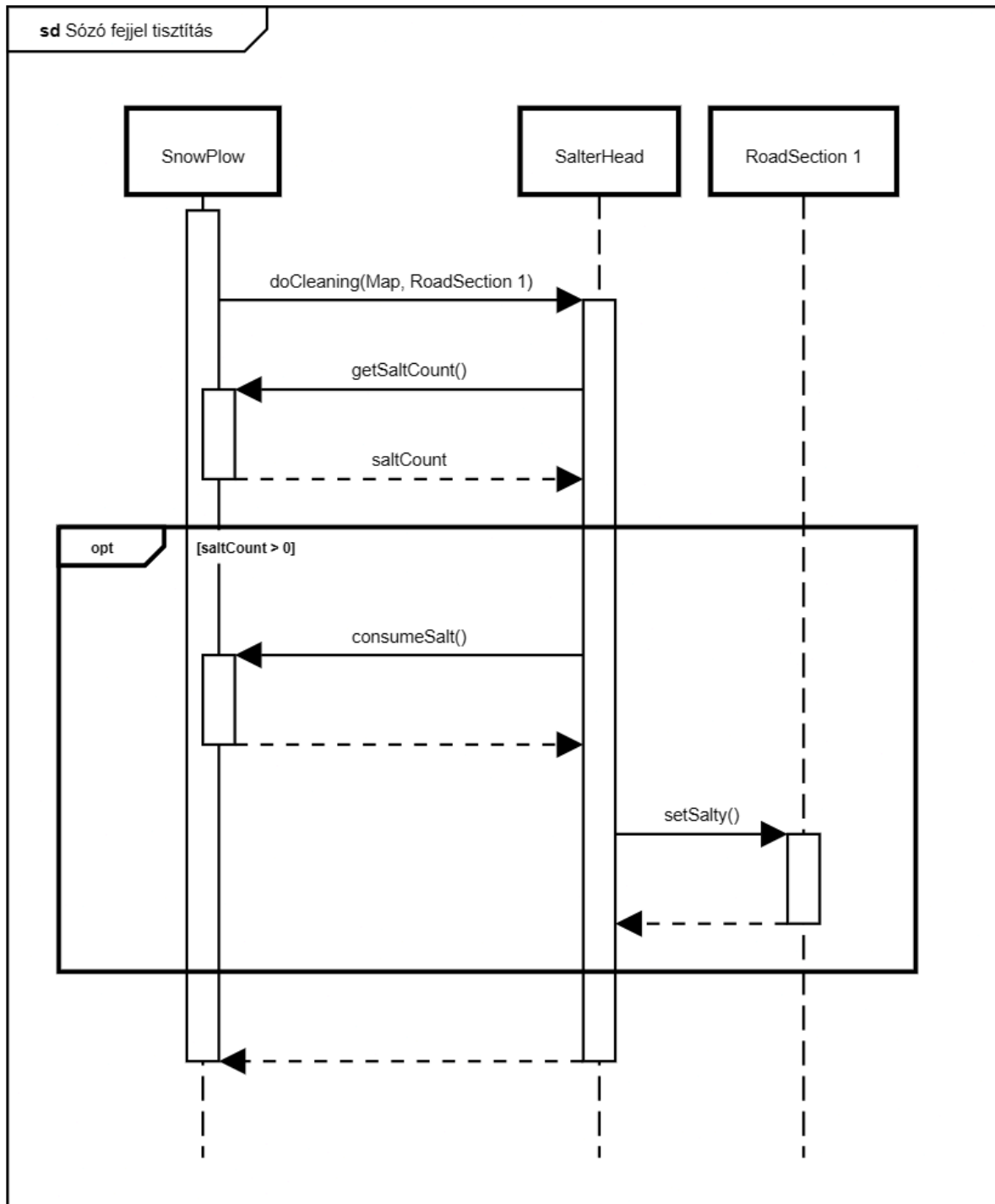


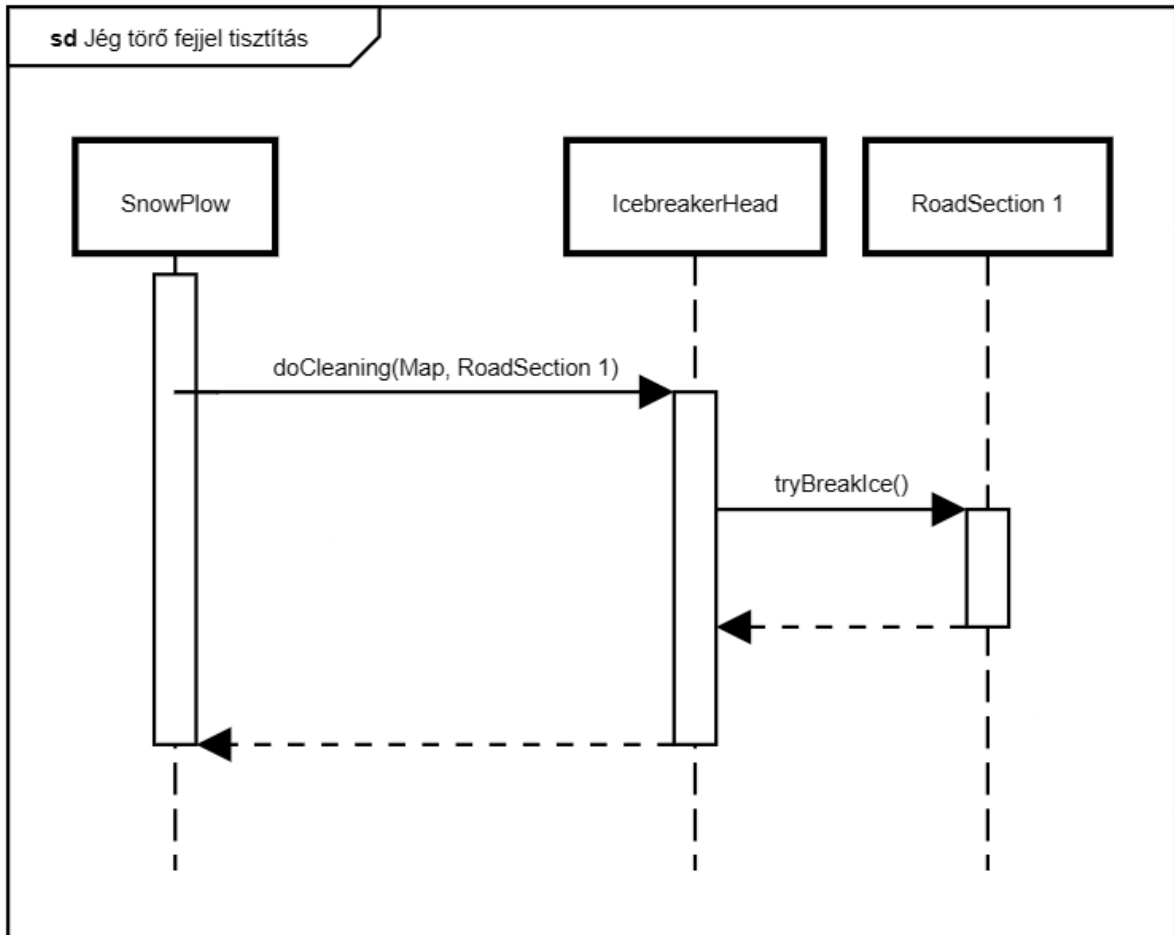


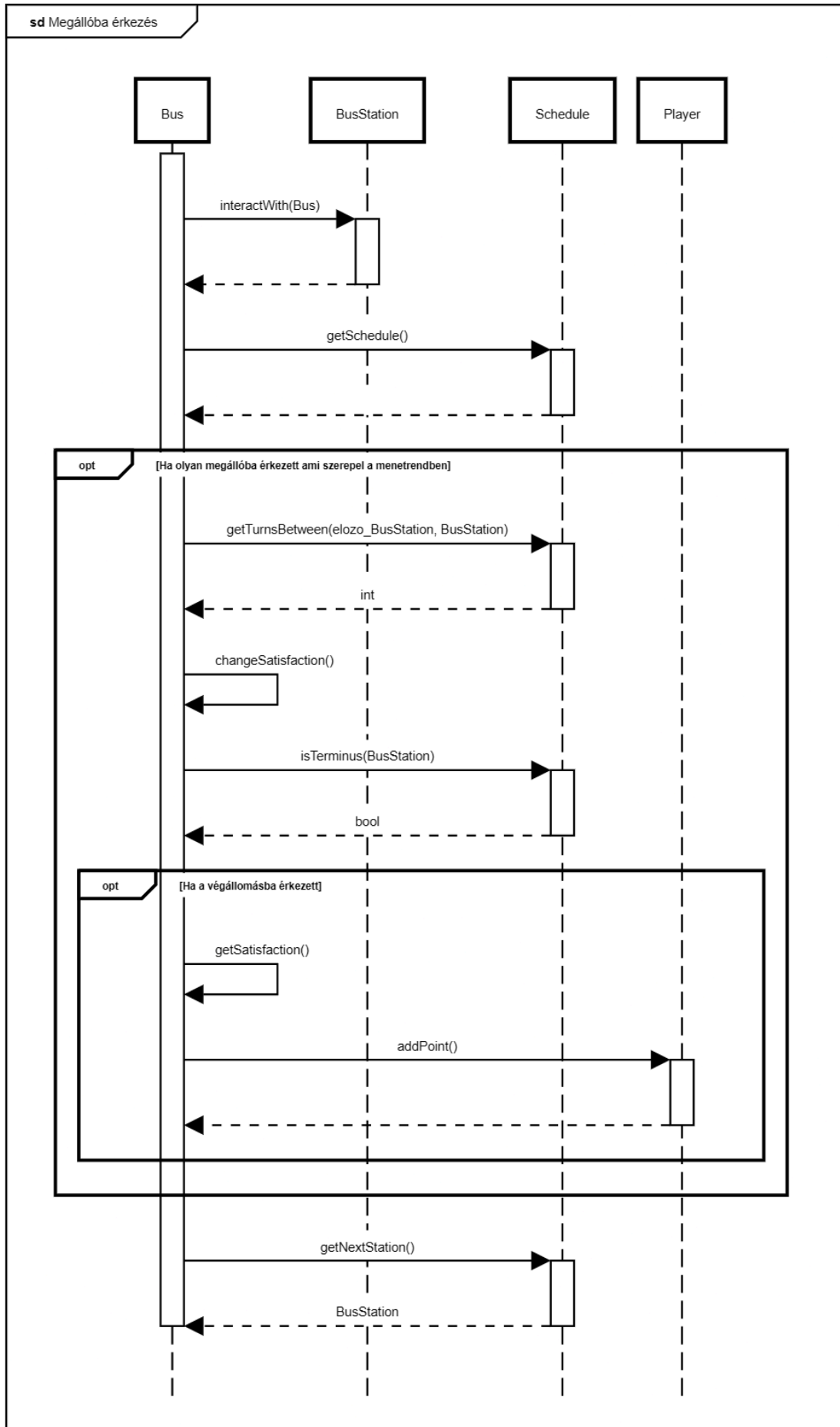


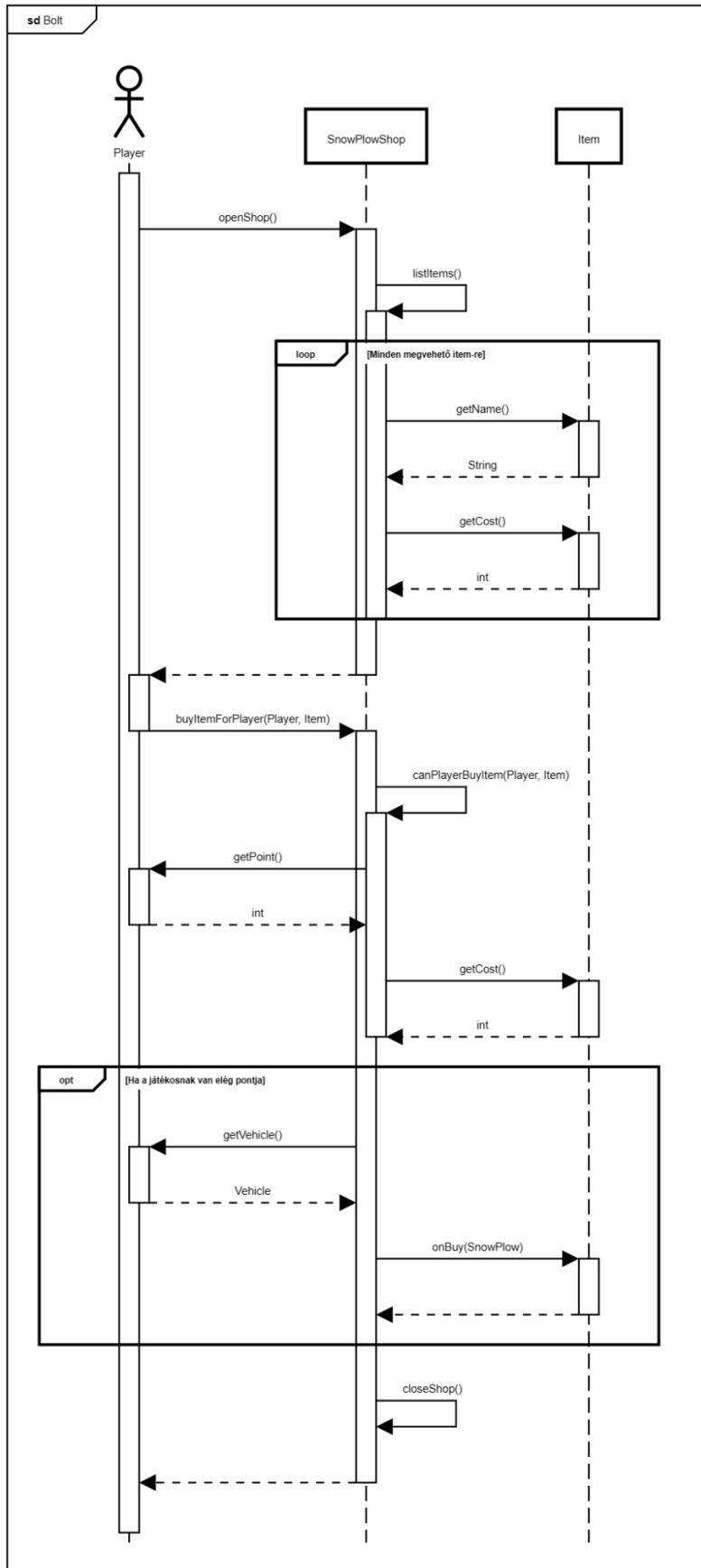


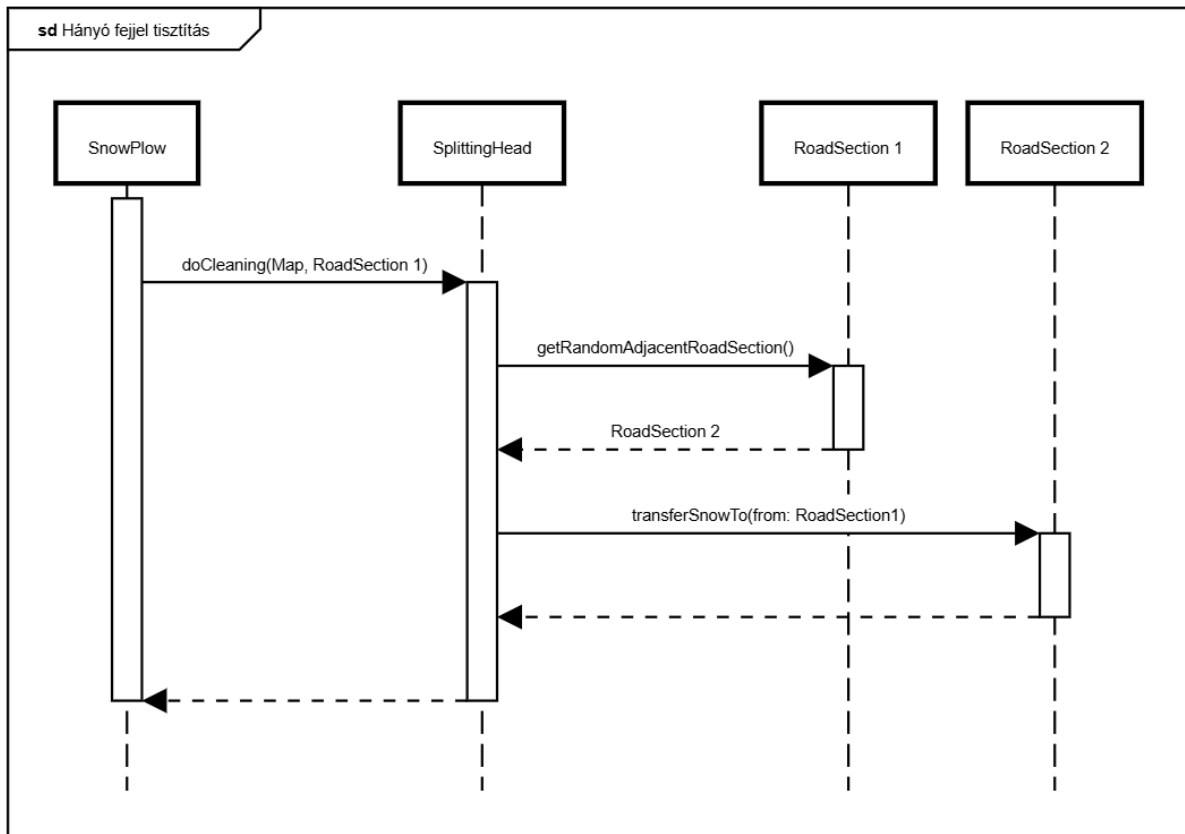




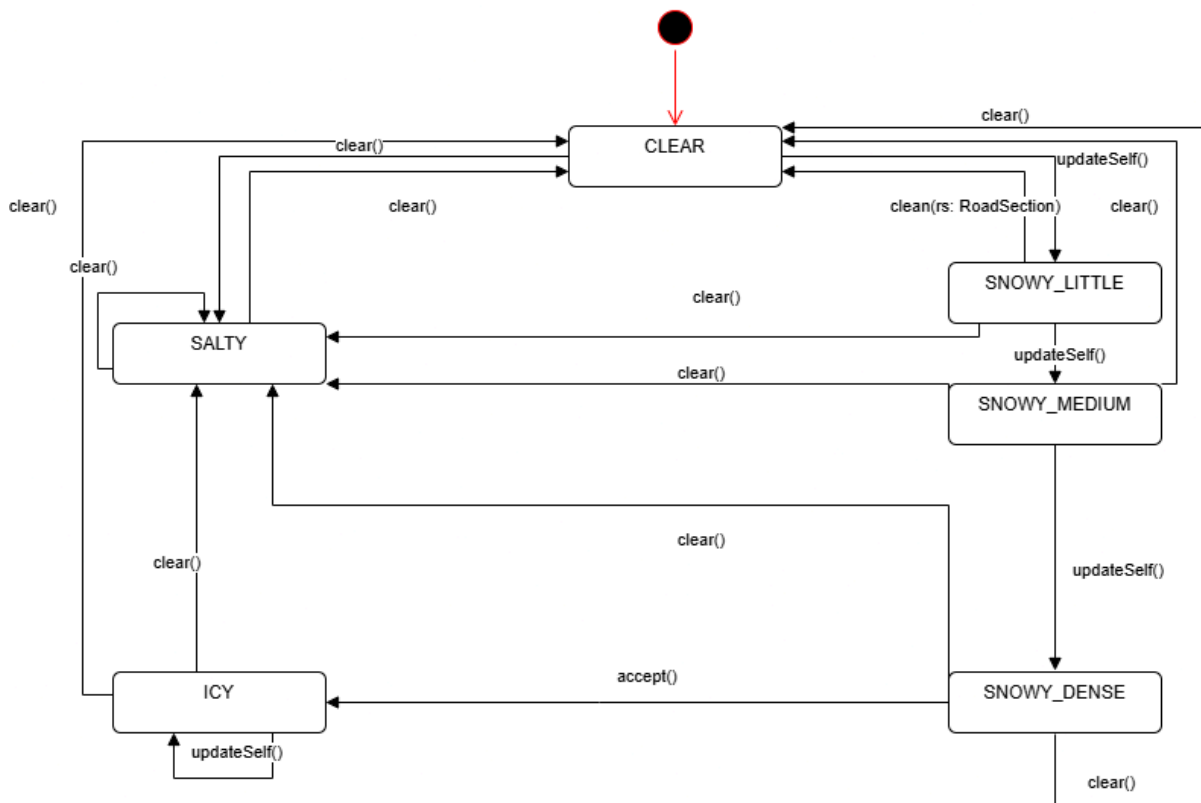








4.5 State-chartok



4.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2026.03.11. 16:30:00	0,75 óra	Funk	Tevékenység: Osztálydiagram átalakítása, objektumkatalógus átalakítása.
2026.03.11. 16:30:00	0,75 óra	Heltai	Tevékenység: Osztálydiagram átalakítása.
2026.03.12. 16:00:00	2,1 óra	Funk	Tevékenység: Szekvenciadiagramok készítése
2026.03.12. 16:00:00	2,1 óra	Csordás	Tevékenység: Szekvenciadiagramok készítése
2026.03.13. 11:00:00	0,9 óra	Heltai	Tevékenység: Járműütközések menetén átdolgozása az osztálydiagrammon; kisebb módosítgatások; pár átírás az objektum katalógusban és a leírásokban.
2026.03.14. 14:00:00	0,5 óra	<i>Mindenki</i>	Értekezlet. Döntés: Review feladatok kiosztása.
2026.03.14. 14:30:00	0,3 óra	Funk	Tevékenység: Szekvenciadiagramok önellenőrzés
2026.03.14. 15:00:00	0,3 óra	Csordás	Tevékenység: Mozgás szekvenciadiagramok javítása.

2026.03.14. 17:00:00	1 óra	Rózsa	Tevékenység: osztály leírások frissítése a javított osztálydiagramm alapján
2026.03.15. 14:00:00	0,3 óra	Heltai	Tevékenység: Szekvenciadiagrammok ellenőrzése; apró hibák orvoslása.
2026.03.15. 14:00:00	0,75 óra	Funk	Tevékenység: Osztályleírás review
2026.03.15. 18:30:00	2,2 óra	Heltai	Tevékenység: Osztályleírás formázásának javítása (helyenként sikertelenül); osztálydiagramrészek külön szedése, beszúrása.
2026.03.15. 21:00:00	0,2 óra	Rózsa	Tevékenység: szekvencia review
2026.03.15. 23:00:00	2 óra	Fekete	Tevékenység: Osztályleírások átírása, multiplicitás kiegészítése diagramban és leírásban, szekvenci diagram kiegészítése

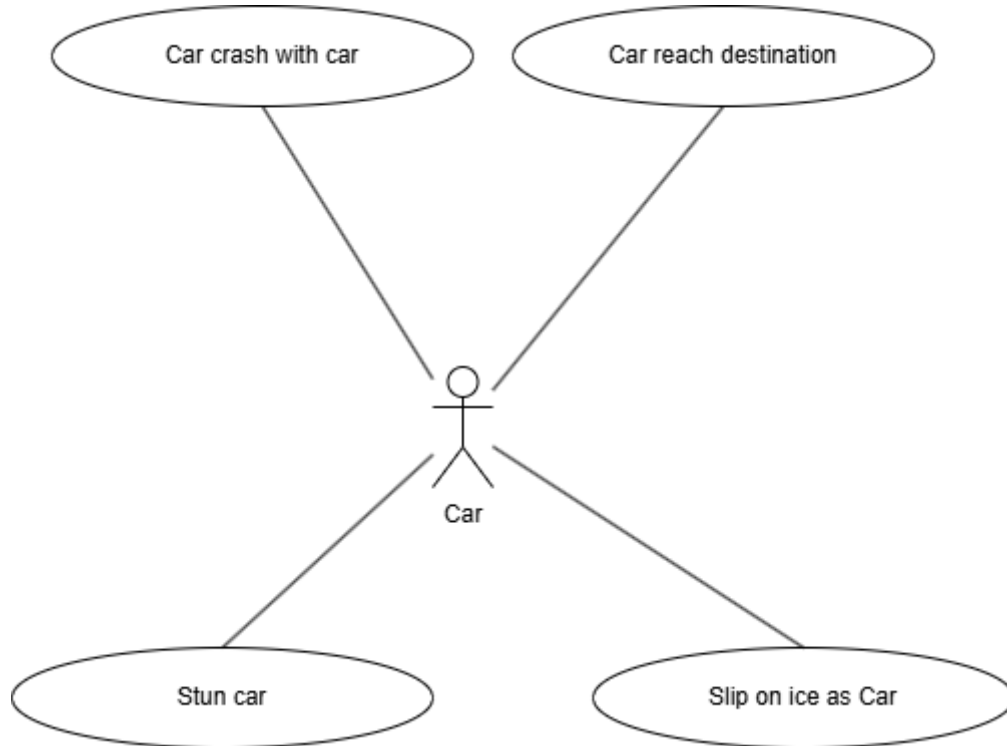
5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ei

5.1.1 Use-case diagram



A car-é:



5.1.2 Use-case leírások

Use-case neve	View shop
Rövid leírás	A hókotrós játékos megnézi a boltot.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a bolt megnyitása során.

Use-case neve	Reach station
Rövid leírás	A buszvezető elér a járatának egy megállójába.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak az állomás elérésének során.

Use-case neve	Move player-vehicle to intersection
Rövid leírás	A játékos az irányított járművét mozgatja egy elágazásra.
Aktorok	Player
Forgatókönyv	1. A Player a járművet a szomszédos elágazásra lépteti.
Forgatókönyv	2. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a mozgás során.

Use-case neve	Move player-vehicle to roadsection
Rövid leírás	A játékos az irányított járművét mozgatja egy új útszakaszra.
Aktorok	Player
Forgatókönyv	1. A Player a járművet a szomszédos útra lépteti.

Forgatókönyv	2. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a mozgás során.
---------------------	---

Use-case neve	Choose player type
Rövid leírás	A játékos kiválasztja hogy busz vagy takarító akar lenni.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a járműválasztás során.
Forgatókönyv	2. A program odaadja a Playernak a választott járművet.

Use-case neve	Clear road section
Rövid leírás	A hókotró takarító letisztít egy adott útszakaszt.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a tisztítás során.

Use-case neve	Change SnowPlowHead
Rövid leírás	A hókotró irányító leváltja a jelenlegi hókotró fejet egy másikra.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a fej váltása során
Forgatókönyv	2. A program kicseréli a jelenlegi fejet a választottra.

Use-case neve	Buy item
Rövid leírás	A hókotró irányító a bolt megnyitása után vásárol is ott valamit.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a vásárlás során
Forgatókönyv	2. A program odaadja a megvásárolt item-et.

Use-case neve	Player crash with car
Rövid leírás	A Player ütközik egy autóval, majd mozgásképtelen lesz egy rövid ideig.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak az ütközés során
Forgatókönyv	2. A program mozgásképtelenné teszi a Playert az adott időre.
Forgatókönyv	3. A kocsit ott marad az úton akadályként egy ideig.

Use-case neve	Crash with bus
Rövid leírás	A játékos ütközik egy másik játékosal, majd mozgásképtelen lesz egy rövid ideig
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak az ütközés során
Forgatókönyv	2. A program mozgásképtelenné teszi mindkét játékost az adott időre.

Use-case neve	Crash with snowplow
Rövid leírás	A hókotró irányító ütközik egy másik hókotró irányítóval, majd mozgásképtelen lesz egy rövid ideig
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak az ütközés során
Forgatókönyv	2. A program mozgásképtelenné teszi mindkét hókotró irányítót az adott időre

Use-case neve	Car crash with car
Rövid leírás	Két autó ütközik az úton így mindkettő akadályként marad a pályán az adott ideig.
Aktorok	Car
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak az ütközés során.
Forgatókönyv	2. A program akadályként ott hagyja a két autót az adott ideig.

Use-case neve	Car reaches destination
Rövid leírás	A kocsí eléri az adott célját.
Aktorok	Car
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak ha eléri a célt.

Use-case neve	Salt road
Rövid leírás	A hókotró irányító lesózza az adott útszakaszt, ha só áll rendelkezésre.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak az út sózása során.
Forgatókönyv	2. A program sós típusúvá teszi az utat.

Use-case neve	Use DragonHead on road
Rövid leírás	A hókotró irányító a sárkányfejet használja az út letisztítására ha van kerozin nála
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak az tisztítása során
Forgatókönyv	2. A program tiszta típusúvá teszi az utat

Use-case neve	Slip on ice
Rövid leírás	A jármű (játékosé) megcsúszik az úton és elveszti az irányítást a kör fölött.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik a csúszás során lefutnak

Use-case neve	Slip on ice as Car
Rövid leírás	A jármű megcsúszik az úton és elveszti az irányítást a kör fölött.
Aktorok	Car
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik a csúszás során lefutnak

Use-case neve	Get stuck in dense snow
Rövid leírás	A Player vagy kocszi elakad a sűrű hóban és nem tudnak tovább haladni
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, az elakadással kapcsolatosak

Use-case neve	Break ice
Rövid leírás	A hókotró irányító feltöri a jeget az úton
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a jég feltörése során
Forgatókönyv	2. A program az utat tiszta állapotúra teszi

Use-case neve	Get stunned
Rövid leírás	A játékos kimarad egy a körből.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a kimaradás során

Use-case neve	Stun car
Rövid leírás	Az autó leragad egy körre, és akadállyá válik.
Aktorok	Car
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a kimaradás során

Use-case neve	Update world
Rövid leírás	A játékos a körét lezárja, és erre a világ reagál.
Aktorok	Player
Forgatókönyv	1. A program kiírja a függvényhívásokat a képernyőre, amik lefutnak a havazás (világ frissítés) során

5.2 A szkeleton kezelői felületének terve, dialógusok

A szkeleton kezelői felülete egy szöveges konzolos felület, amelyen keresztül a felhasználó lefuttathatja az előre definiált teszteseteket.

A felület célja, hogy programunk követhető és tesztelhető legyen és az eddig meghatározott use case-eink futtathatók legyenek.

A program indításakor kilistázza az elérhető, előre definiált forgatókönyveket, amelyek közül választani lehet. Ez a következőképp néz ki:

```
[<sorszám>: <forgatókönyv neve>; <rövid leírás> \n]*
```

Példa: 1. Update world; A játékos a körét lezárja, és erre a világ reagál. *soremelés, majd a többi forgatókönyv következik*

Miután kiválasztottunk egy forgatókönyvet, az alábbi információk jelennek meg, majd lefut a teszt: (Egy vagy több aktor lehet.)

```
<forgatókönyv neve> \n <aktor>[, <aktor>]* \n
```

A forgatókönyvekben *objektumok* vesznek részt, amelyekbe az aktorok is beletartoznak. A *metódusok* az objektumok által meghívott publikus metódusok. Lefutás során az alábbi információk jelennek meg:

```
[\t* <hívó objektum neve>.<meghívott metódus neve> \n]*
```

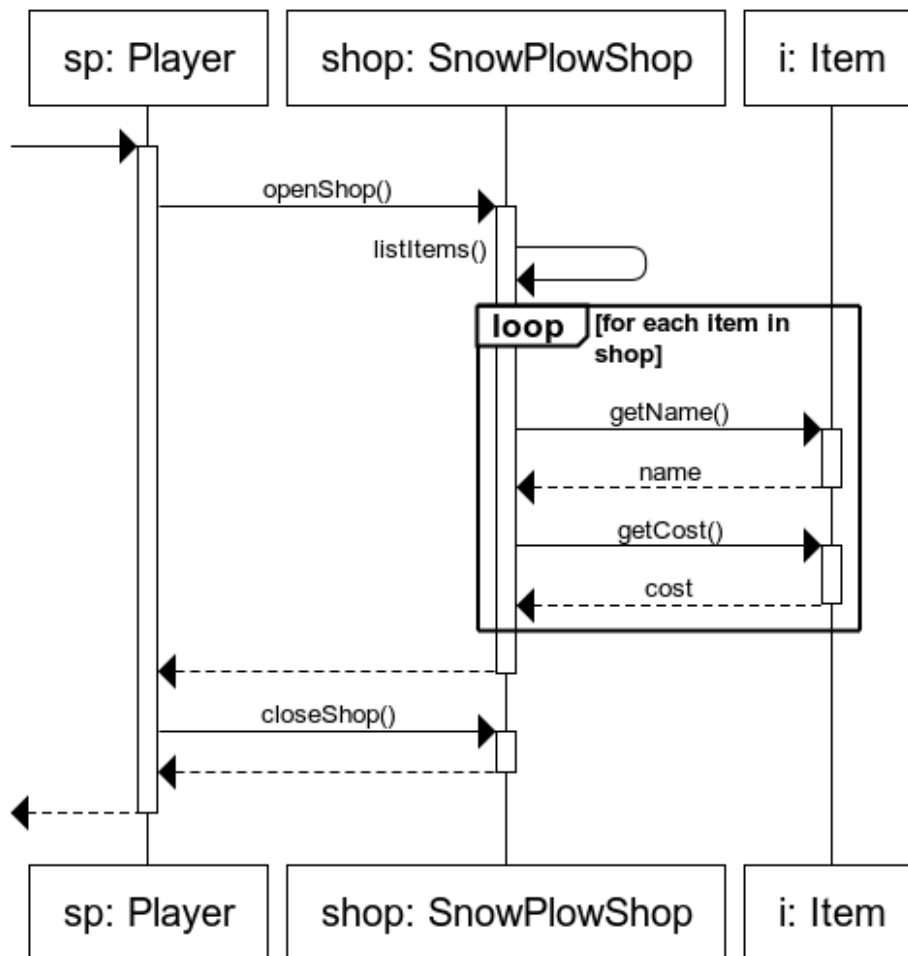
A tabulációk mértéke attól függ, hogy milyen mélyen hívódik meg a metódus.

Ezen kívül, ha felhasználói interakcióra van szükség, akkor az egy kérdésként jelenik meg, az alábbi módon:

```
\t* <kérdés>: felhasználó válasza
```

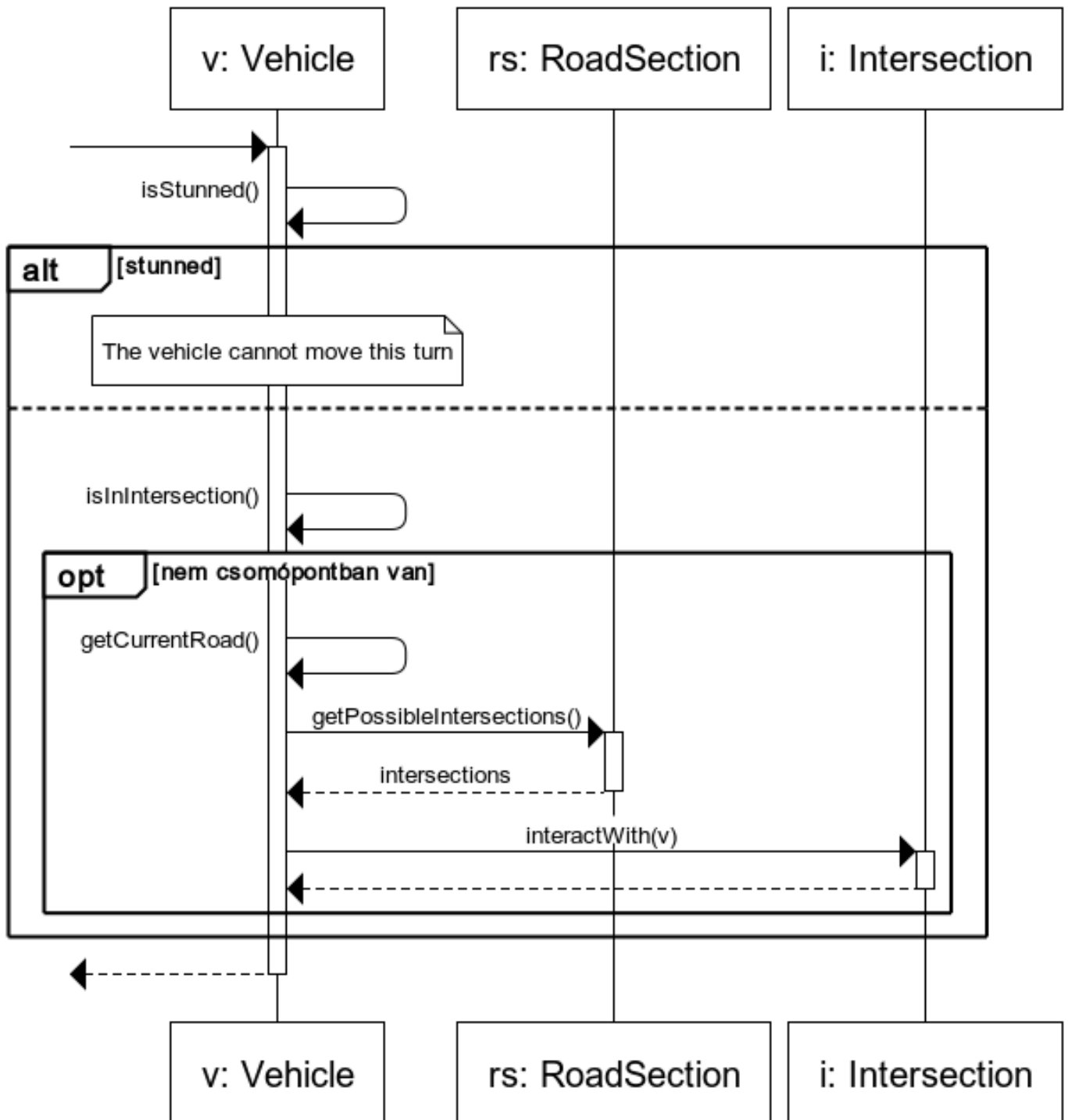
5.3 Szekvencia diagramok a belső működésre

View shop

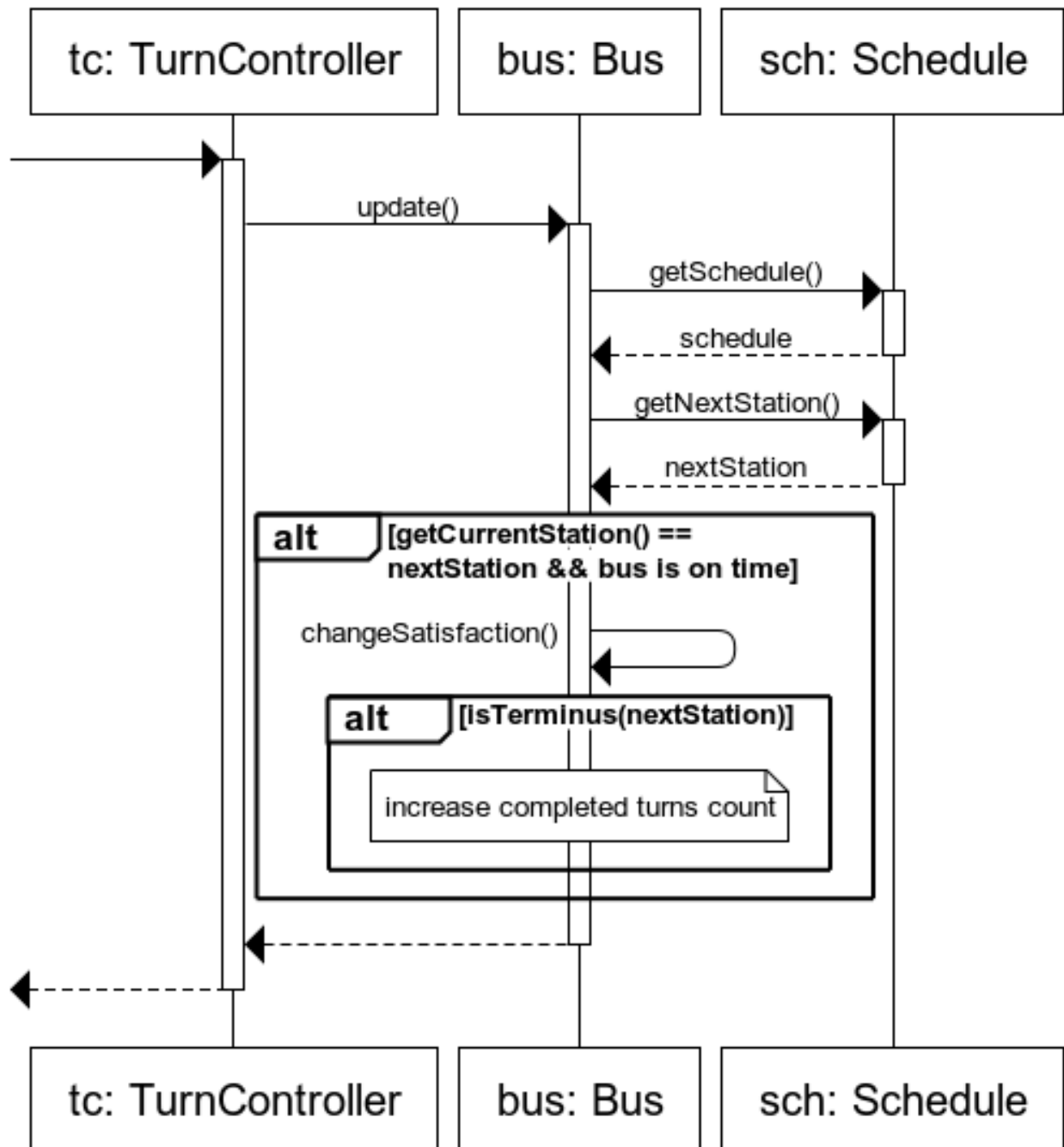


www.websequencediagrams.com

Move player-vehicle to intersection

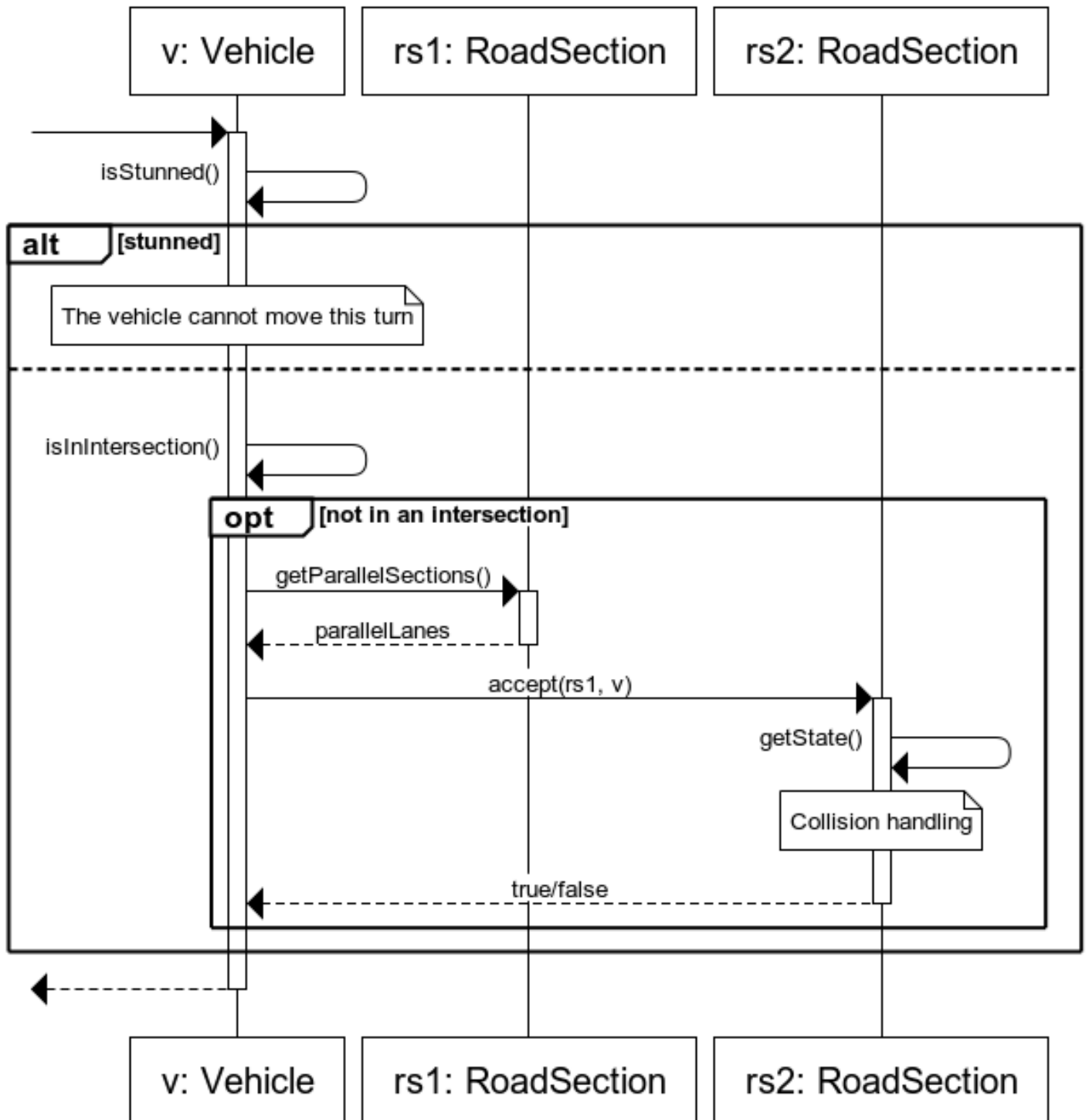


Reach station



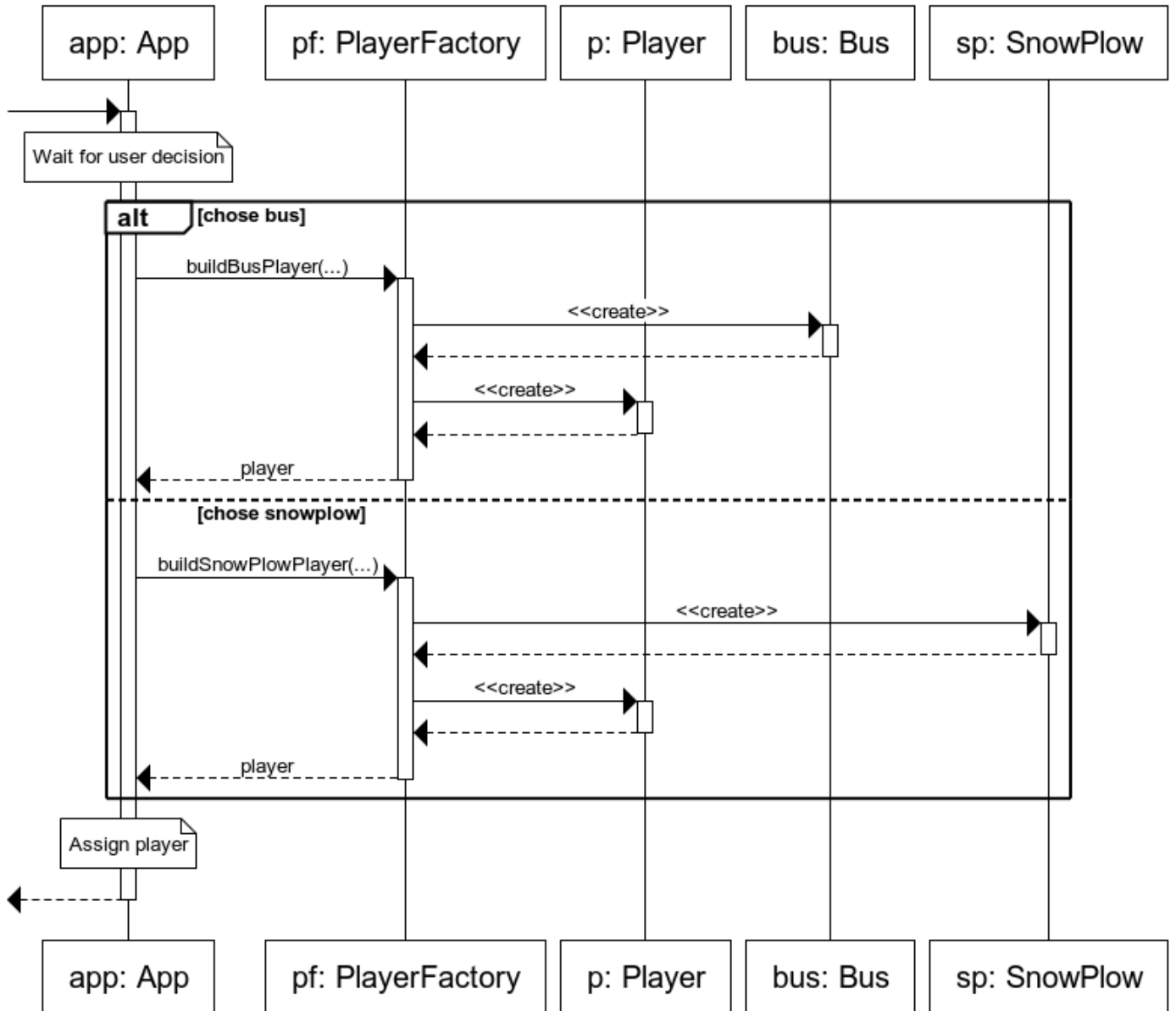
www.websequencediagrams.com

Move player-vehicle to roadsection



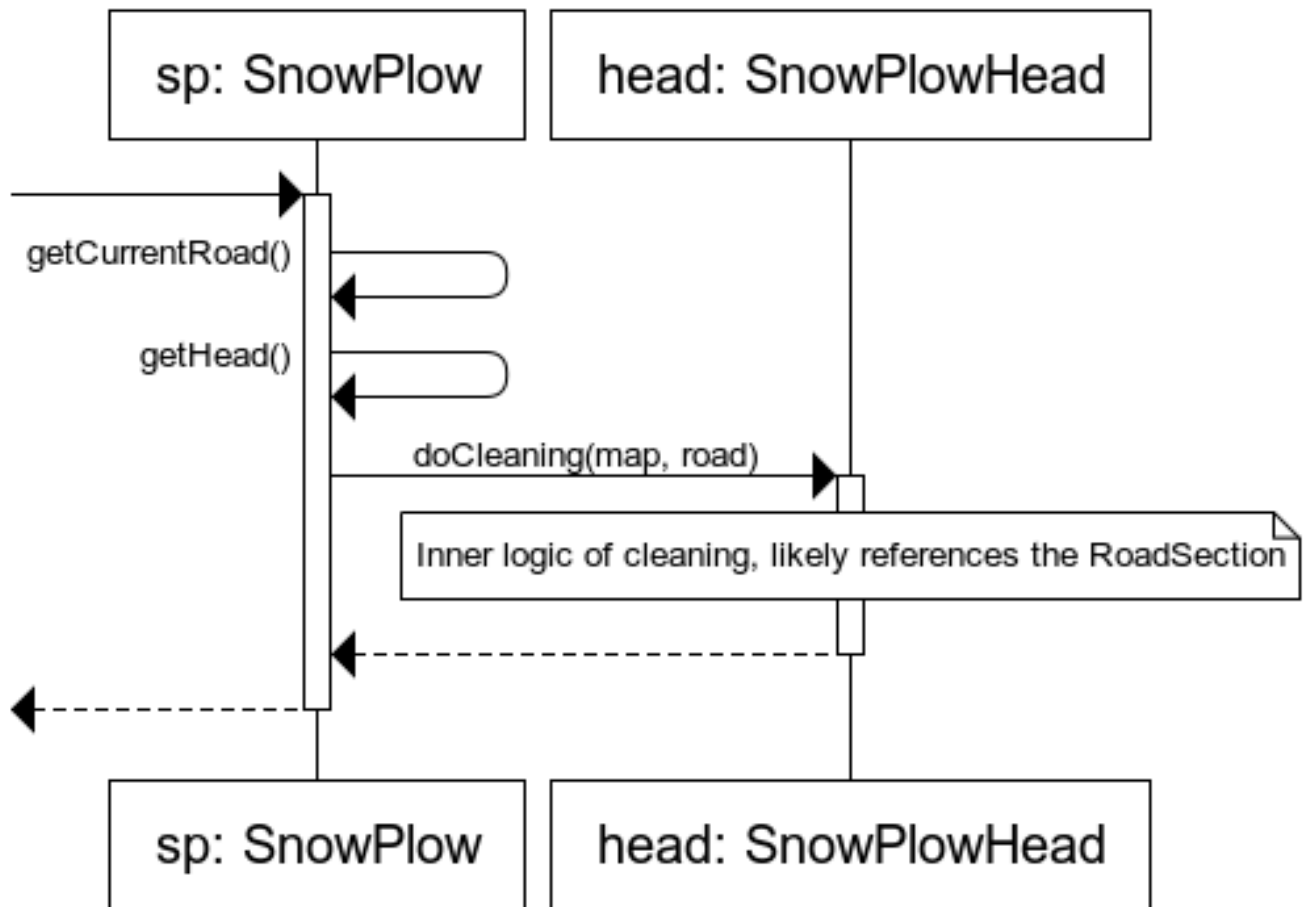
www.websequencediagrams.com

Choose player type



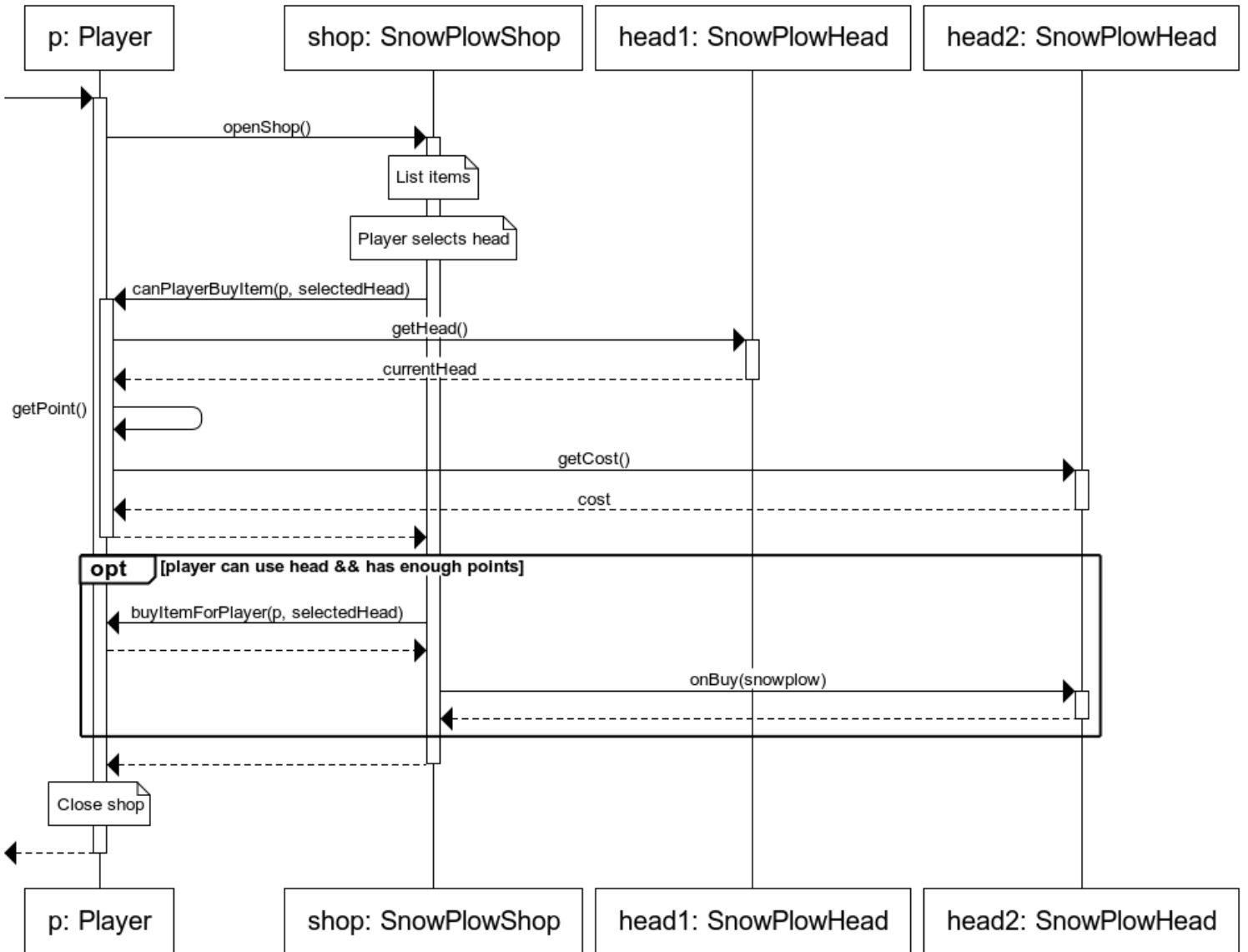
www.websequencediagrams.com

Clear road section



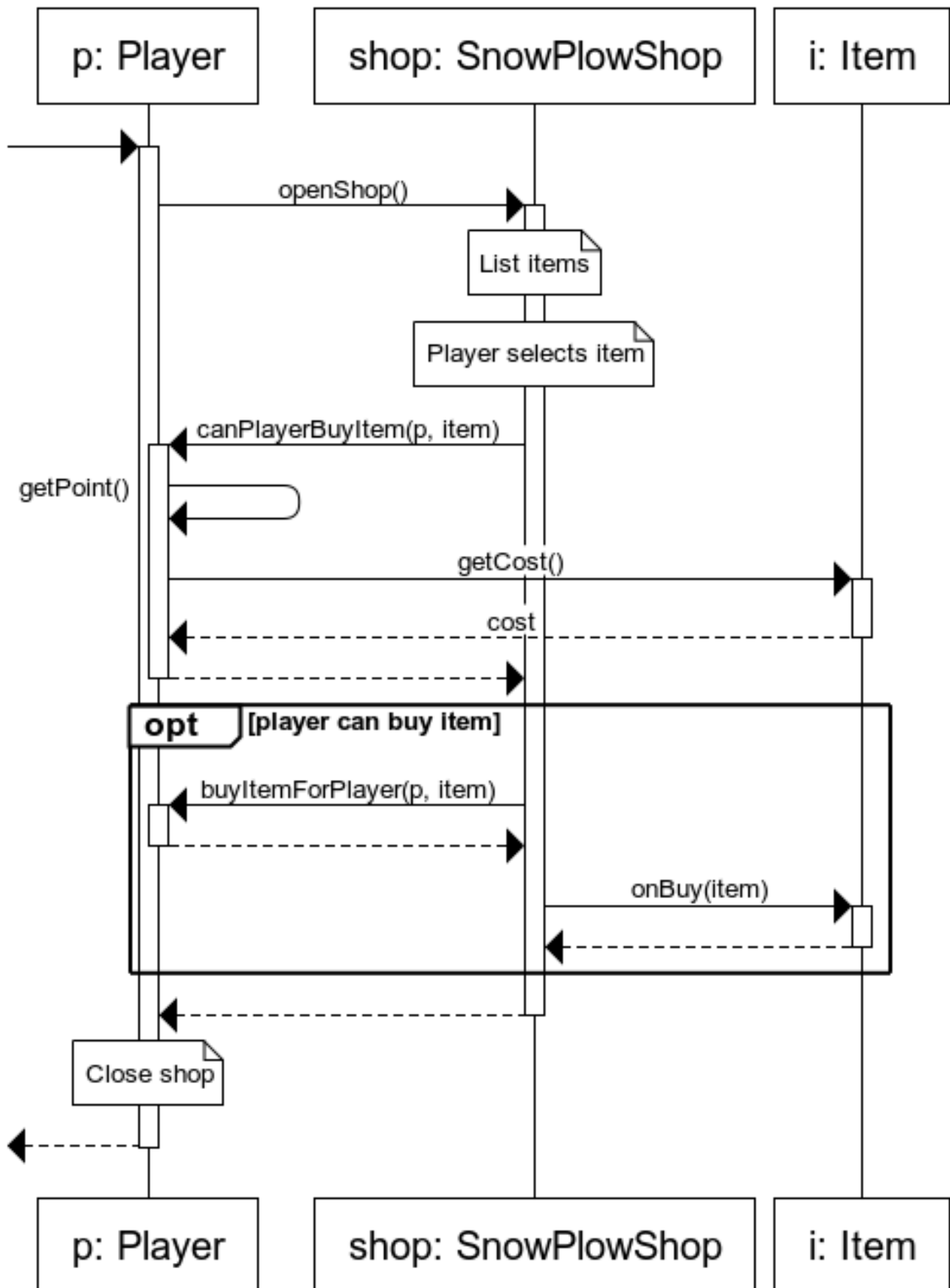
www.websequencediagrams.com

Change SnowPlowHead

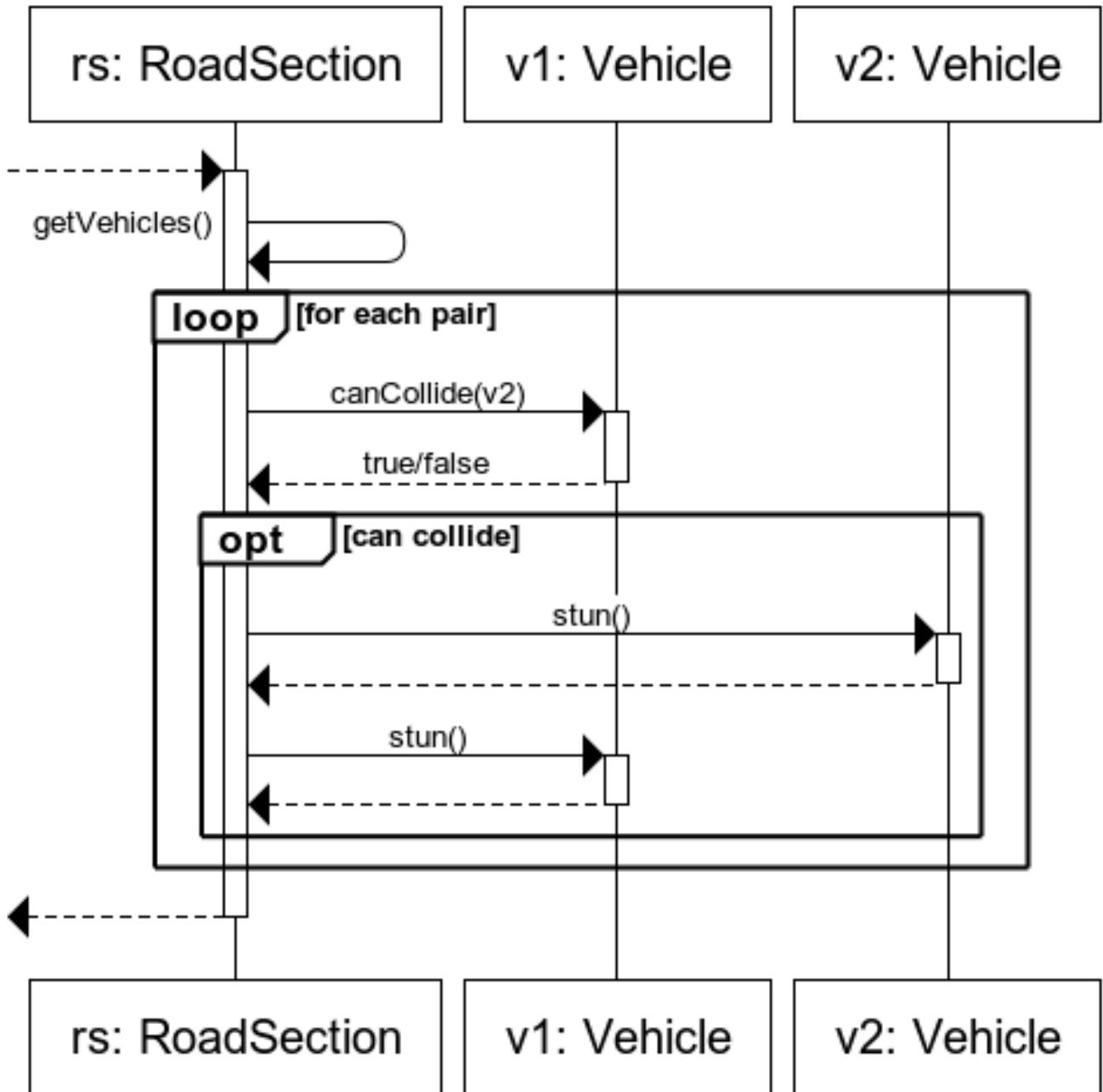


www.websequencediagrams.com

Buy item

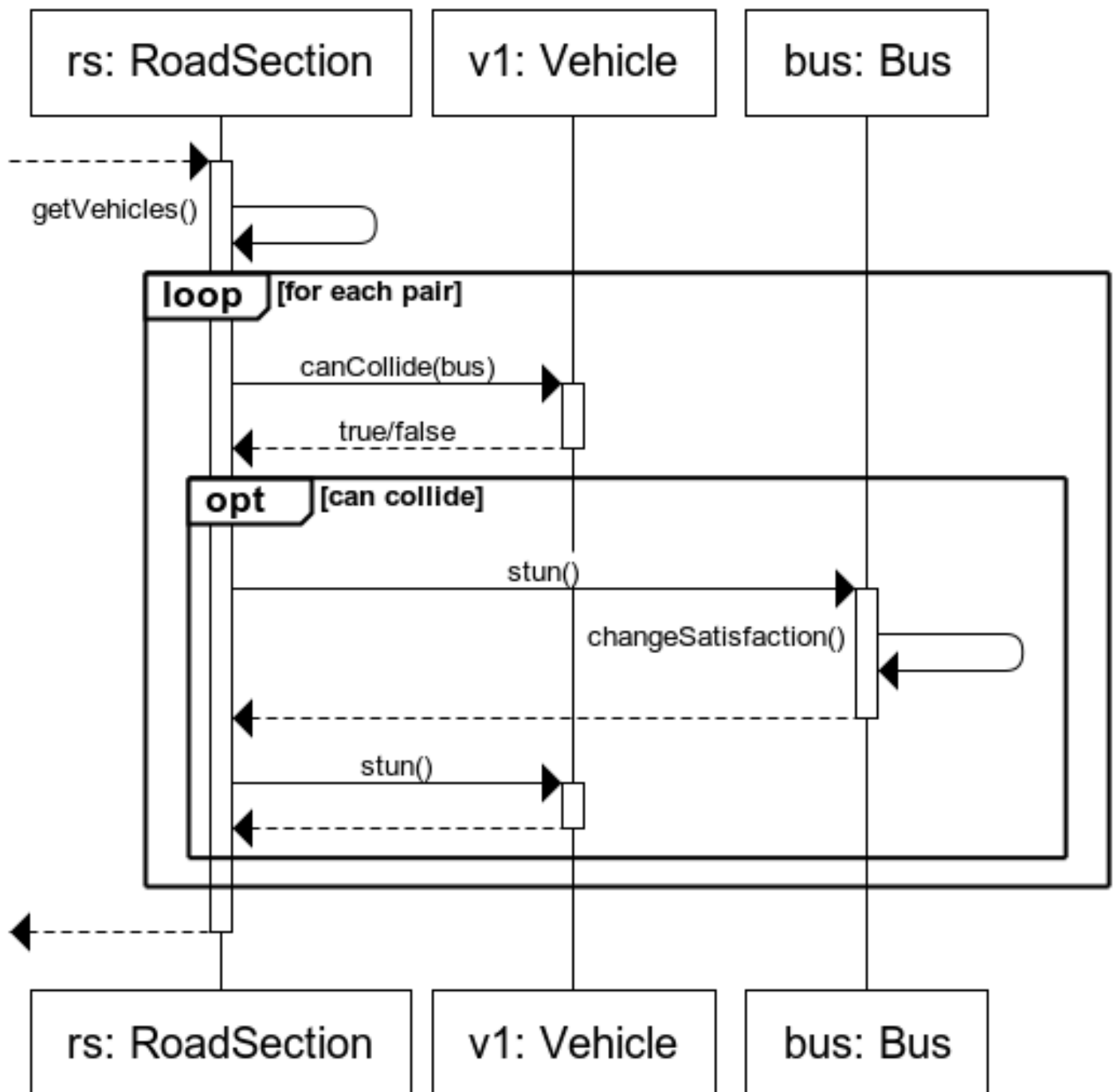


Generic crash



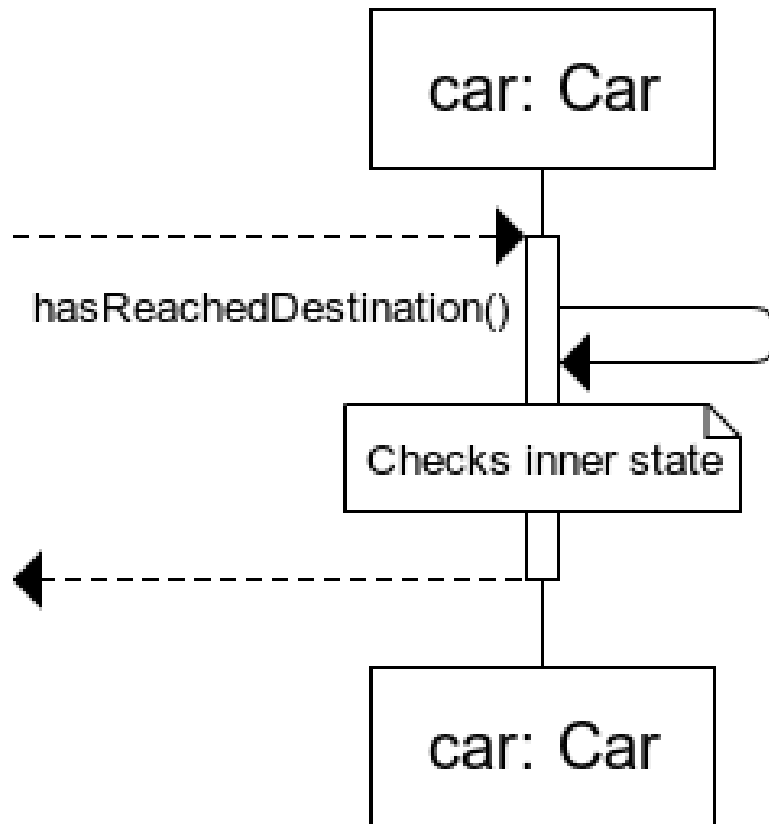
www.websequencediagrams.com

Crash with bus



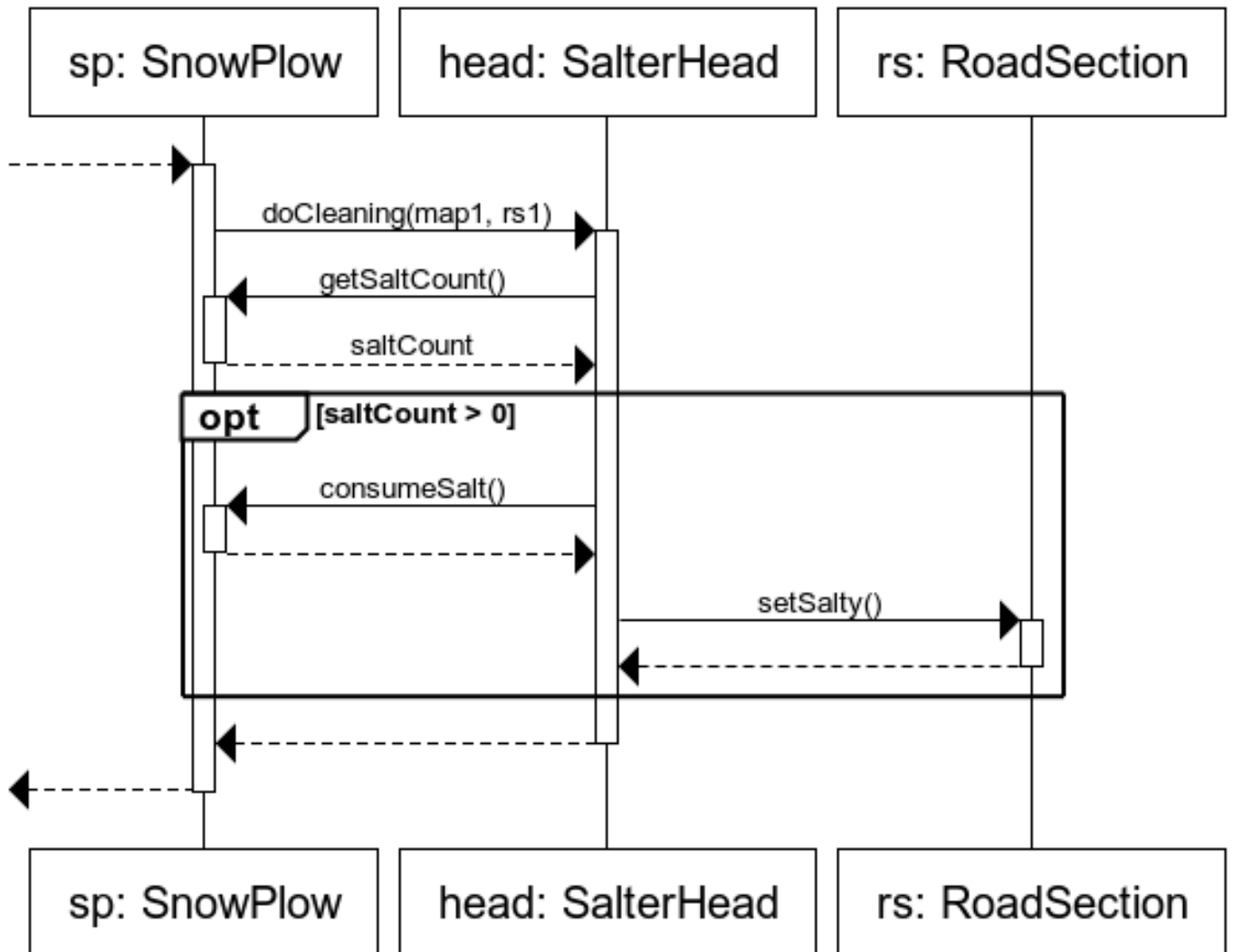
www.websequencediagrams.com

Car reaches destination



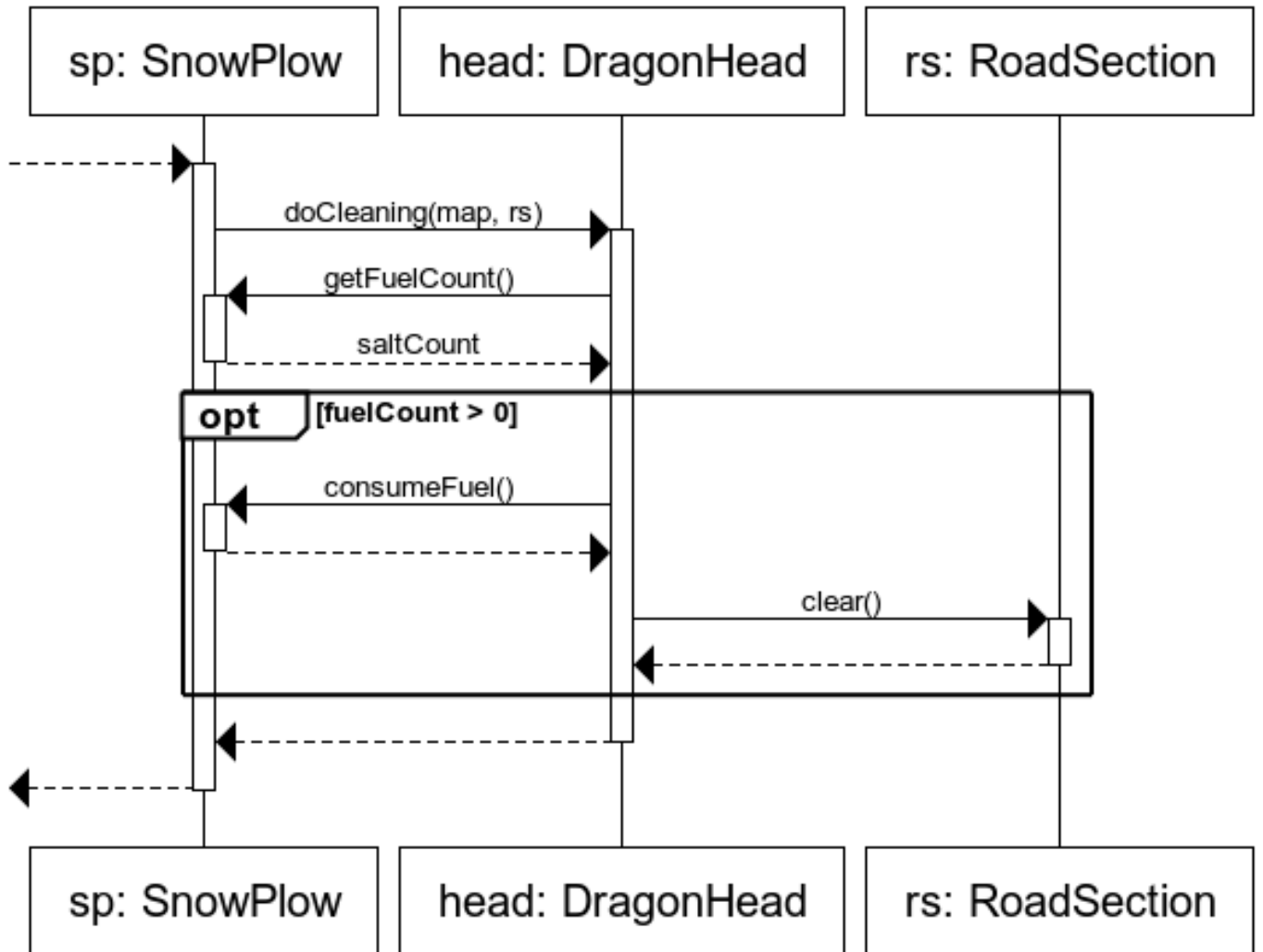
www.websequencediagrams.com

Salt road

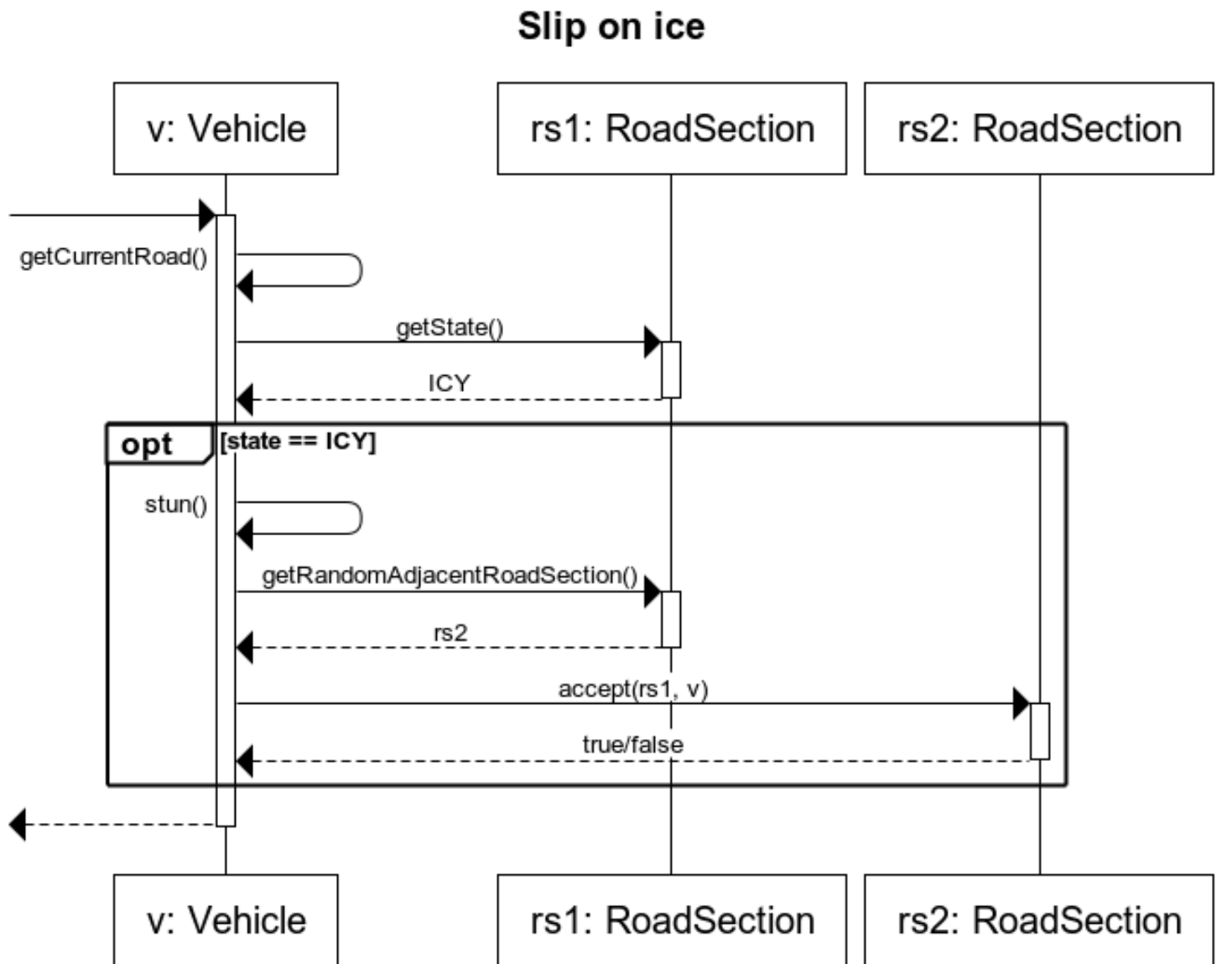


www.websequencediagrams.com

Use DragonHead on road

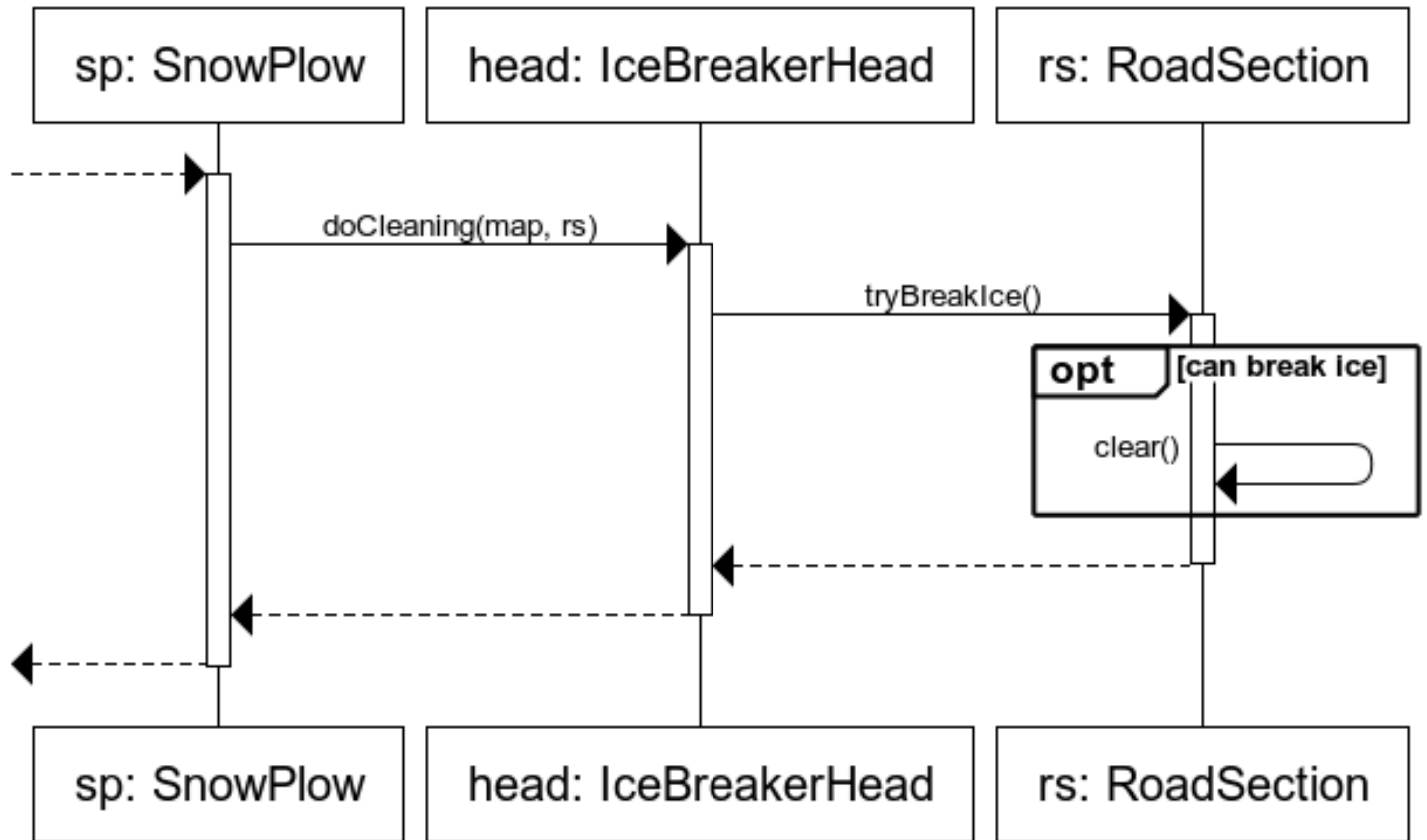


www.websequencediagrams.com



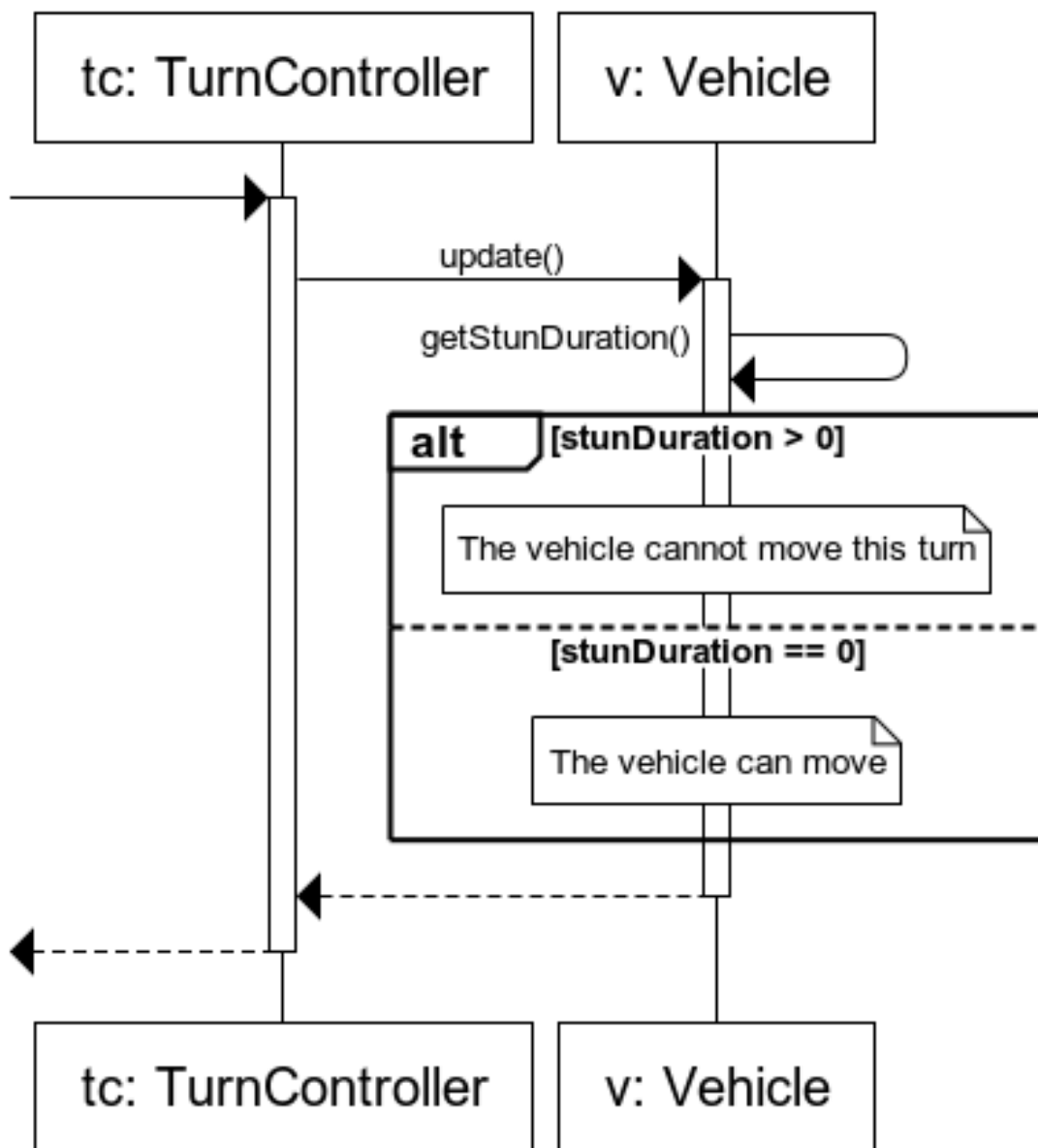
www.websequencediagrams.com

Break ice



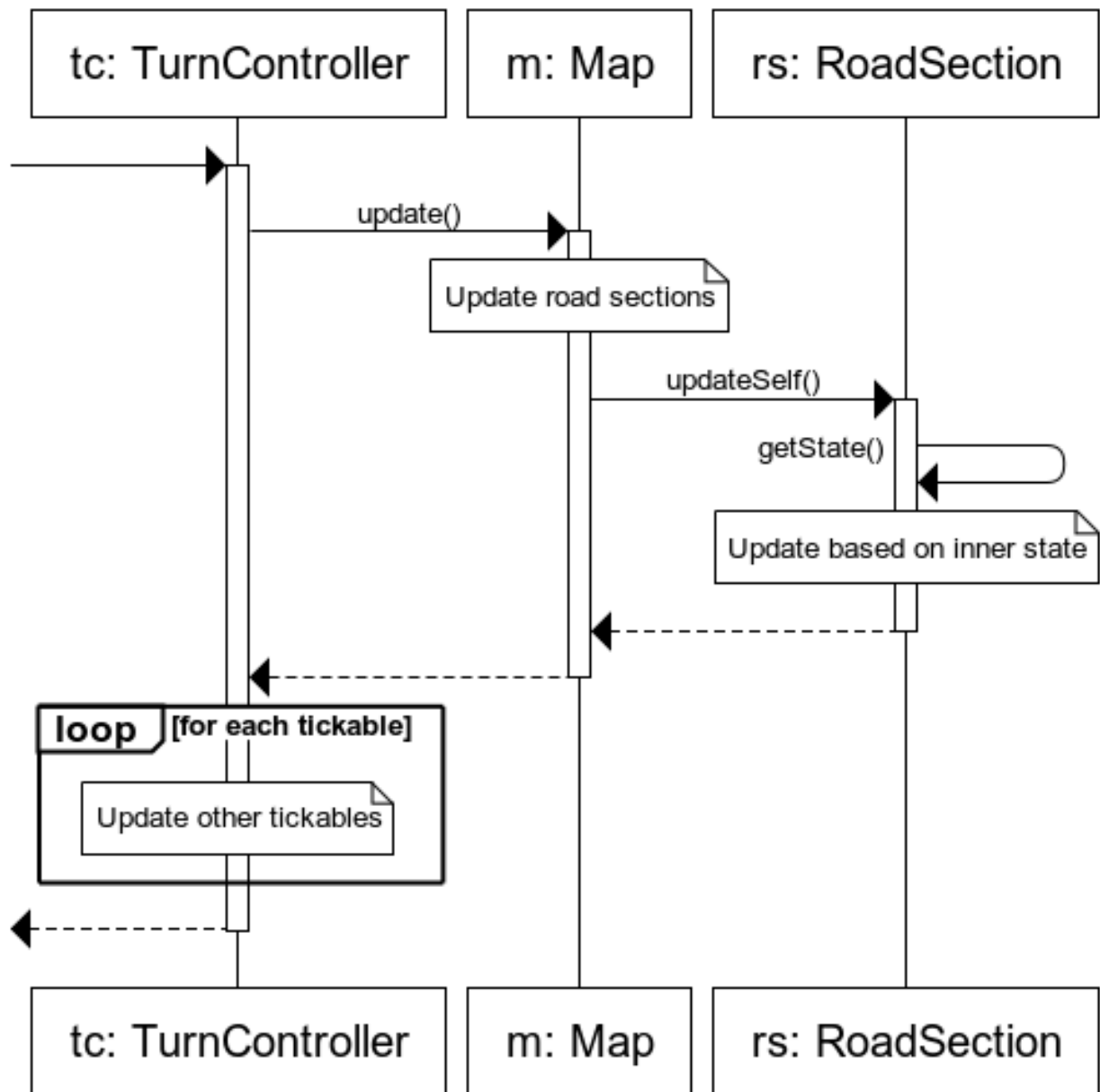
www.websequencediagrams.com

Get stunned



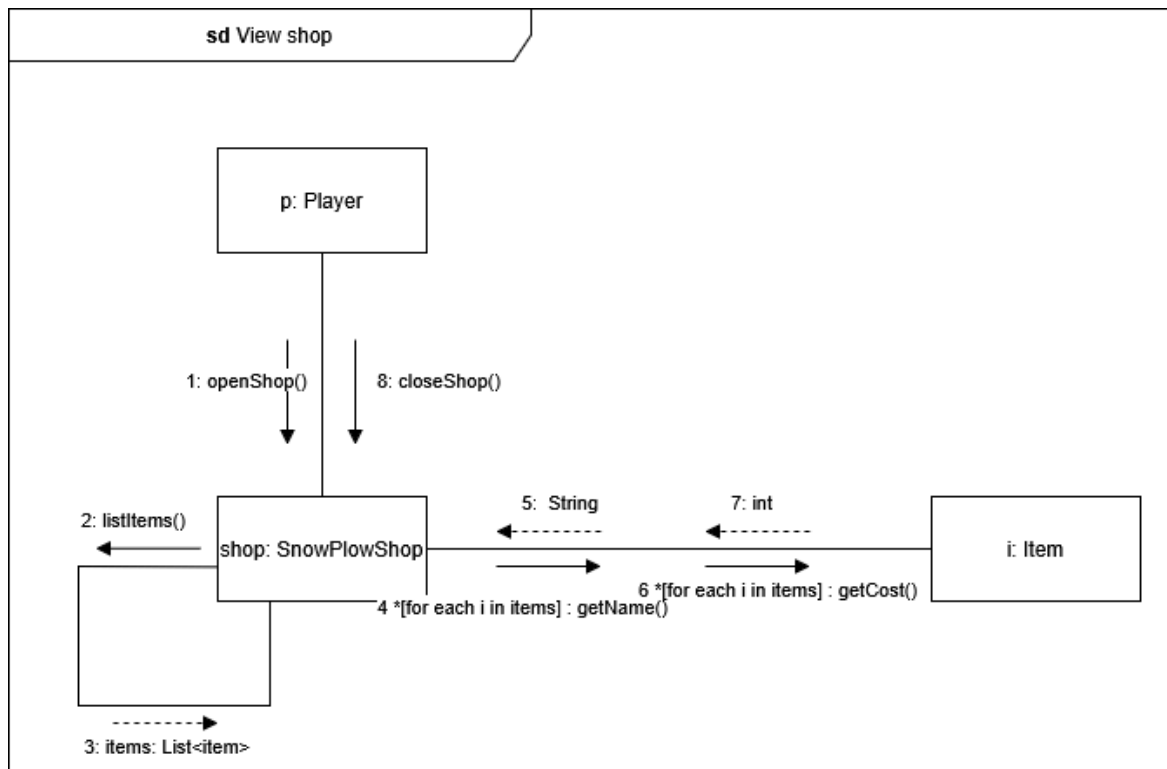
www.websequencediagrams.com

Update world

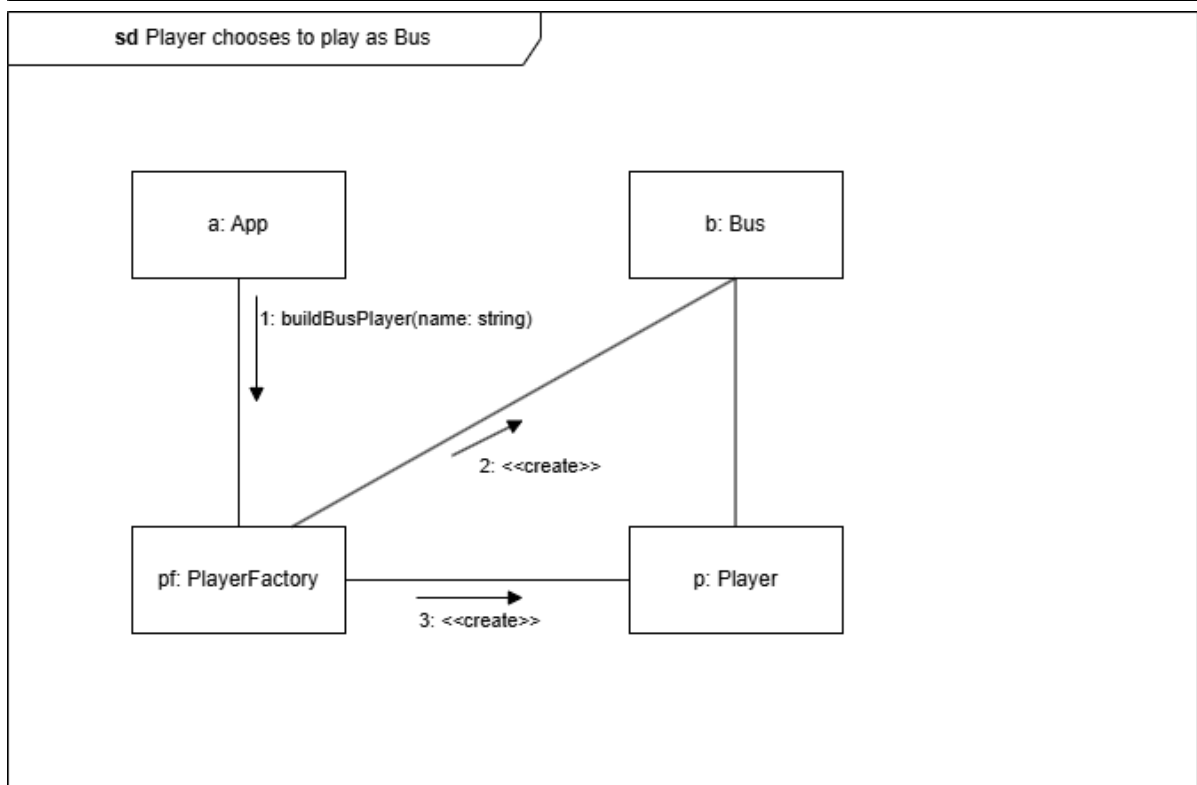
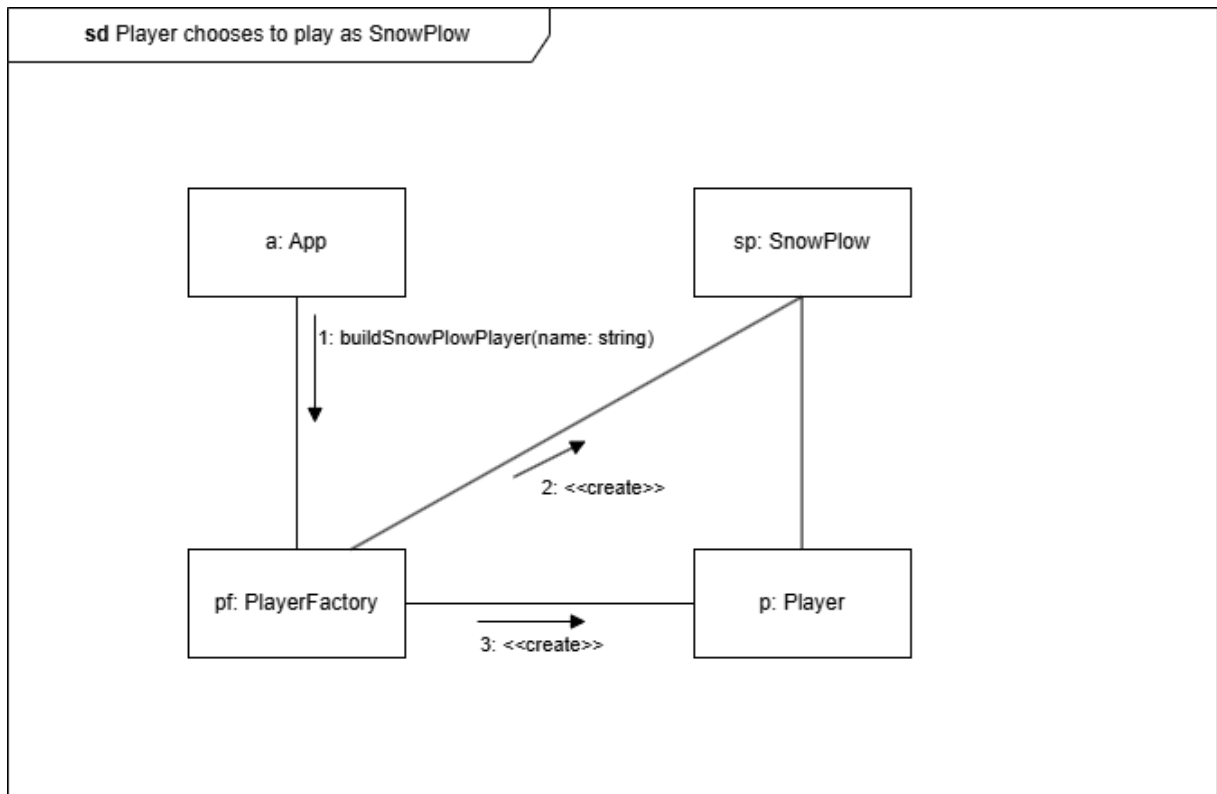


www.websequencediagrams.com

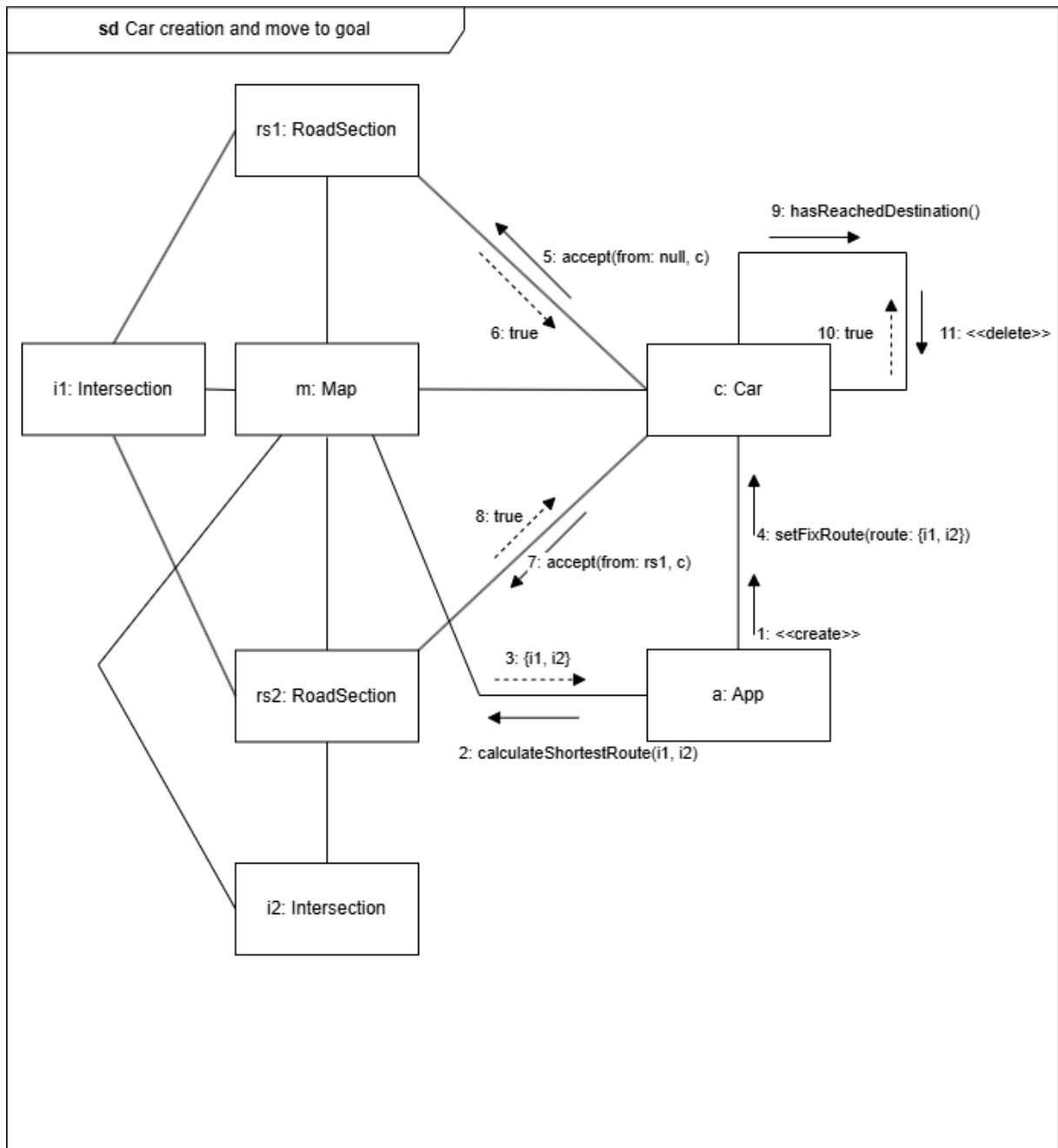
5.4 Kommunikációs diagramok



A View shop szekvenciához tartozik.



Mindkettő a Choose Player Type-nál fontos, illetve mindenhol, ahol a játékosok relevánsak (mozgás, crash stb.).



Az összes Car-ral kapcsolatos szekvenciához tartozik.

Napló

Kezdet	Időtartam	Résztevők	Leírás
2026.03.19. 15:00:00	1,5 óra	Rózsa	Tevékenység: Szekvencia diagrammok készítése
2026.03.20. 9:00:00	0,6 óra	Funk	Tevékenység: Use case-k létrehozása.
2026.03.20. 19:00:00	0,25 óra	Heltai	Tevékenység: Skeleton megbeszélése; szekvenciadiagrammo k elosztása.
2026.03.20. 19:00:00	0,25 óra	Rózsa	Tevékenység: Skeleton megbeszélése; szekvenciadiagrammo k elosztása.
2026.03.21. 8:00:00	1,3 óra	Rózsa	Tevékenység: Szekvencia diagrammok készítése
2026.03.21. 14:00:00	0,6 óra	Mindenki	Értekezlet. Általános kérdéseket beszéltünk meg. Döntés: ki melyik diagrammot csinálja.
2026.03.21. 14:40:00	1 óra	Heltai	Tevékenység: Ismerkedés a websequencediagrams szintaxisával; két szekvenciadiagram megcsinálása.
2026.03.21. 14:40:00	0,3 óra	Funk	Tevékenység: Actor diagramok létrehozása + review.
2026.03.22. 10:00:00	2,5 óra	Heltai	Tevékenység: Szekvencia diagrammok készítése; review; formázás.
2026.03.22. 9:10:00	1,5 óra	Funk	Tevékenység: Kommunikáció diagramok készítése + review

2026.03.22. 11:30:00	0,2 óra	Funk	Tevékenység: Use case-k kiegészítése.
2026.03.22. 18:40:00	0,7 óra	Heltai	Tevékenység: Szekvencia diagrammok
2026.03.22. 19:10:00	1,3 óra	Rózsa	Tevékenység: Szekvencia diagrammok
2026.03.22. 22:00:00	3,5 óra	Fekete	Tevékenység: Use Case-ek hozzáadása/kiegészítése, kommunikációs diagramok hozzáadás
2026.03.22. 22:00:00	3,5 óra	Csordás	Tevékenység: Kommunikáció diagramok készítése/kiegészítése, eltérések javítása, review

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Megjegyzés: a “src” kezdetű elérési utak a ./src/main/java/javallerok utat rövidítik. Illetve a legtöbb keletkezési dátum a Git miatt “helytelen.” Mivel minden pull-t követve megváltoztatta a keletkezések dátumát is.

Fájl neve	Méret (bytes)	Keletkezés ideje	Tartalom
./pom.xml	3988	2026. 02. 09.	Maven config fájlja.
./build.sh	882	2026. 03. 25.	Build script Unix rendszerekhez.
./build.bat	1026	2026. 03. 25.	Build script Windows rendszerekhez.
src/Main.java	283	2026. 03. 29.	A program entry pointja: ez indítja el a forgatókönyv választót.
src/core/App.java	4858	2026. 03. 29.	A játék központi osztálya, ez köti össze a “központi” osztályokat.
src/core/ITickable.java	303	2026. 03. 29.	Az ITickable interfész.
src/core/Player.java	1707	2026. 03. 29.	A játékos osztály.
src/core/PlayerFactory.java	1569	2026. 03. 29.	A PlayerFactory osztály.
src/core/TurnController.java	1658	2026. 03. 29.	A TurnController osztály.
src/gamemap/Bridge.java	1276	2026. 03. 29.	A Bridge osztály.
src/gamemap/BusStation.java	2100	2026. 03. 29.	A BusStation osztály.
src/gamemap/Intersection.java	1603	2026. 03. 29.	Az Intersection osztály.
src/gamemap/Map.java	3432	2026. 03. 29.	A Map osztály.
src/gamemap/NPCHomeStation.java	618	2026. 03. 29.	Az NPCHomeStation osztály.
src/gamemap/RoadSection.java	6356	2026. 03. 29.	A RoadSection osztály.

Fájl neve	Méret (bytes)	Keletkezés ideje	Tartalom
src/gamemap/RoadState.java	736	2026. 03. 29.	A RoadSection osztály.
src/gamemap/Tunnel.java	1047	2026. 03. 29.	A Tunnel osztály.
src/items/Biokerosene.java	933	2026. 03. 29.	A Biokerosene osztály.
src/items/DragonHead.java	1645	2026. 03. 29.	A DragonHead osztály.
src/items/IceBreakerHead.java	1325	2026. 03. 29.	Az IceBreakerHead osztály.
src/items/Item.java	1115	2026. 03. 29.	Az Item osztály.
src/items/Salt.java	818	2026. 03. 29.	A Salt osztály.
src/items/SalterHead.java	1539	2026. 03. 29.	A SalterHead osztály.
src/items/SnowPlowHead.java	1134	2026. 03. 29.	A SnowPlowHead osztály.
src/items/SnowPlowUpgrade.java	1094	2026. 03. 29.	A SnowPlowUpgrade osztály.
src/items/SnowPlowUpgradeNormal.java	848	2026. 03. 29.	A SnowPlowUpgradeNormal osztály.
src/items/SnowPlowUpgradePremium.java	827	2026. 03. 29.	A SnowPlowUpgradePremium osztály.
src/items/SplittingHead.java	1488	2026. 03. 29.	A SplittingHead osztály.
src/items/SweeperHead.java	1485	2026. 03. 29.	A SweeperHead osztály.
src/logging/Loggable.java	392	2026. 03. 28.	A Loggable annotáció.
src/logging/SL.java	6996	2026. 03. 29.	A "Szkeleton Logger" osztály, amely a konzolraírást végzi megfelelő formátumban..
src/schedule/Schedule.java	2429	2026. 03. 29.	A Schedule osztály.

Fájl neve	Méret (bytes)	Keletkezés ideje	Tartalom
src/scripts/Script.java	1557	2026. 03. 29.	A forgatókönyvek alapsosztálya.
src/scripts/ScriptManager.java	21575	2026. 03. 29.	A forgatókönyveket kezelő osztály.
src/shop/SnowPlowShop.java	2977	2026. 03. 29.	A SnowPlowShop osztály.
src/vehicles/Bus.java	3580	2026. 03. 29.	A Bus osztály.
src/vehicles/Car.java	2328	2026. 03. 29.	A Car osztály.
src/vehicles/SnowPlow.java	4027	2026. 03. 29.	A SnowPlow osztály.
src/vehicles/Vehicle.java	5806	2026. 03. 29.	A Vehicle osztály.

6.1.2 Fordítás

Előfeltétel a Java JDK és JRE 11 - vagy modernebb - binary-k megléte és elérhetősége az eszközön. Az alább taglalt műveleteket (parancsokat) a projekt gyökeréből kell futtatni.

A fordításhoz rendelkezésre áll Windows-os futtatáshoz egy `build.bat` fájl, amely generál egy JAR fájlt az `out` mappába `projlabs.jar` néven.

Parancs: `build.bat` (vagy dupla kattintás a fájlra)

Fordító szkript Unix alapú rendszerekhez elérhető `build.sh` néven, outputja megegyzik a Windows-oséval.

Parancs: `./build.sh`

Illetve harmadik lehetőség, a Maven build system használata, amennyiben ez rendelkezésre áll az eszközön. A Maven egy `target` nevű mappába fog generálni egy `projlabs-alpha-skeleton.jar` nevű fájlt.

Parancs: `mvn compile package`

6.1.3 Futtatás

A projekt gyökeréből kell futtatni.

Szkriptek esetén: `java -jar out/projlabs.jar`

Maven esetén: `java -jar target/projlabs-alpha-skeleton.jar`

6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
		20,34
		22,06
		18,41
		19,84
		19,35

Napló

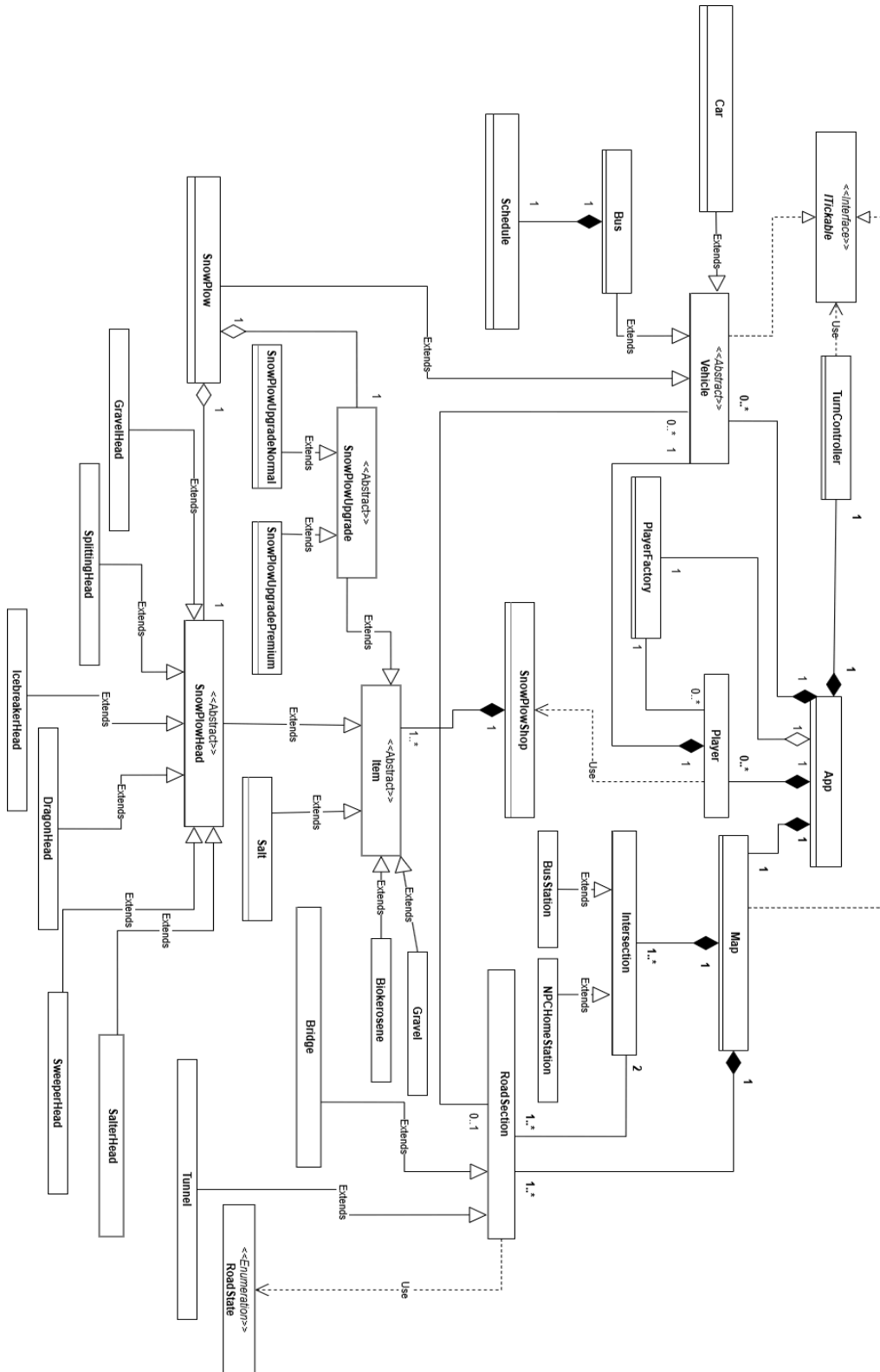
Kezdet	Időtartam	Résztevők	Leírás
2026.03.25. 12:30:00	1,5	Heltai	Tevékenység: Dokumentációk feltöltése a repo-ba; build scriptek írása; szkeleton implementációjának megkezdése.
2026.03.26. 17:50:00	1,5	Heltai	Tevékenység: Szkeleton javítása, dokumentálás.
2026.03.26. 17:50:00	1,2	Funk	Tevékenység: SnowPlow + Itemek szkeleton implementálása
2026.03.27. 14:00:00	1,3	Csordás	Tevékenység: Core szkeleton implementálása
2026.03.27. 15:00:00	1,6	Fekete	Tevékenység: gamemap szkeleton implementálása és car class
2026.03.27. 22:00:00	1,8	Rózsa	Tevékenység: shop package elkezdése és javítgatások
2026.03.28. 8:00:00	1	Rózsa	Tevékenység: refaktor
2026.03.28. 10:00:00	0,5	Funk	Tevékenység: Skeleton review
2026.03.28. 15:00:00	2,8	Funk	Tevékenység: Kommentelés + Playbookok hozzáadása. Review
2026.03.28. 16:00:00	2,4	Heltai	Tevékenység: Helyes(ebb) logika implementációja; formázás.
2026.03.28. 19:00:00	0,95	<i>Mindenki</i>	Értekezlet. Döntés: Csak általánosan a projektről beszéltünk, létező buggokat javítottunk ki.
2026.03.28. 19:00:00	0,2	Heltai	Tevékenység: Fordítási és futtatási útmutatók

			elkészítése.
2026.03.28. 19:00:00	3	Fekete	Tevékenység: gamemap osztályok befejezése, schedule elkészítése, Bus kiegészítése, egyéb javítások
2026.03.28. 19:30:00	2	Csordás	Tevékenység: Core osztályokhoz kommentek, playbookok javítása és hozzáadása
2026.03.29. 0:00:00	2	Rózsa	Tevékenység: forgatókönyvek + kód
2026.03.29. 10:30:00	0,5	Heltai	Tevékenység: Megmaradt forgatókönyv sablonjainak elkészítése; build és fordítási utmutató frissítése.
2026.03.29. 14:00:00	0,7	Rózsa	Tevékenység: forgatókönyvek és kommentek
2026.03.29. 16:00:00	0,55	Funk	Tevékenység: forgatókönyvek hozzáadása
2026.03.29. 16:35:00	0,5	Funk	Tevékenység: forgatókönyvek hozzáadása
2026.03.29. 16:40:00	0,7	Heltai	Tevékenység: Kommentelés.
2026.03.29. 17:30:00	3,5	Csordás	Tevékenység: mozgás logikák, forgatókönyvek
2026.03.29. 18:30:00	2	Fekete	Tevékenység: Kommentek és egyéb változtatások a gamemap-ban

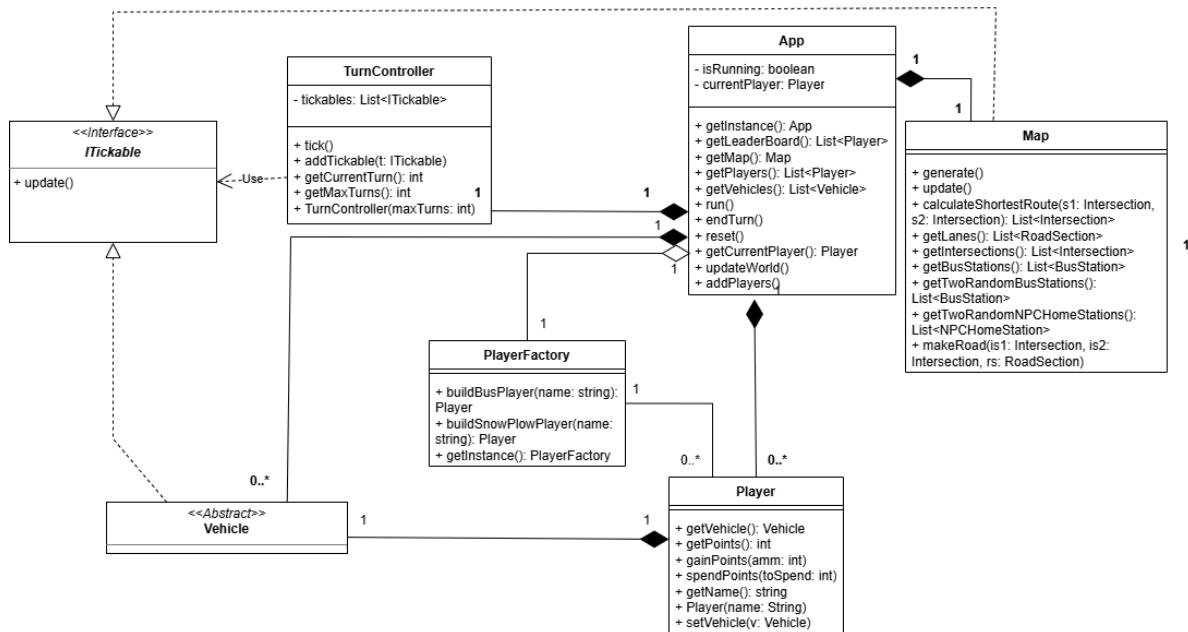
7. Prototípus koncepciója

7.0 Változás hatása a modellre

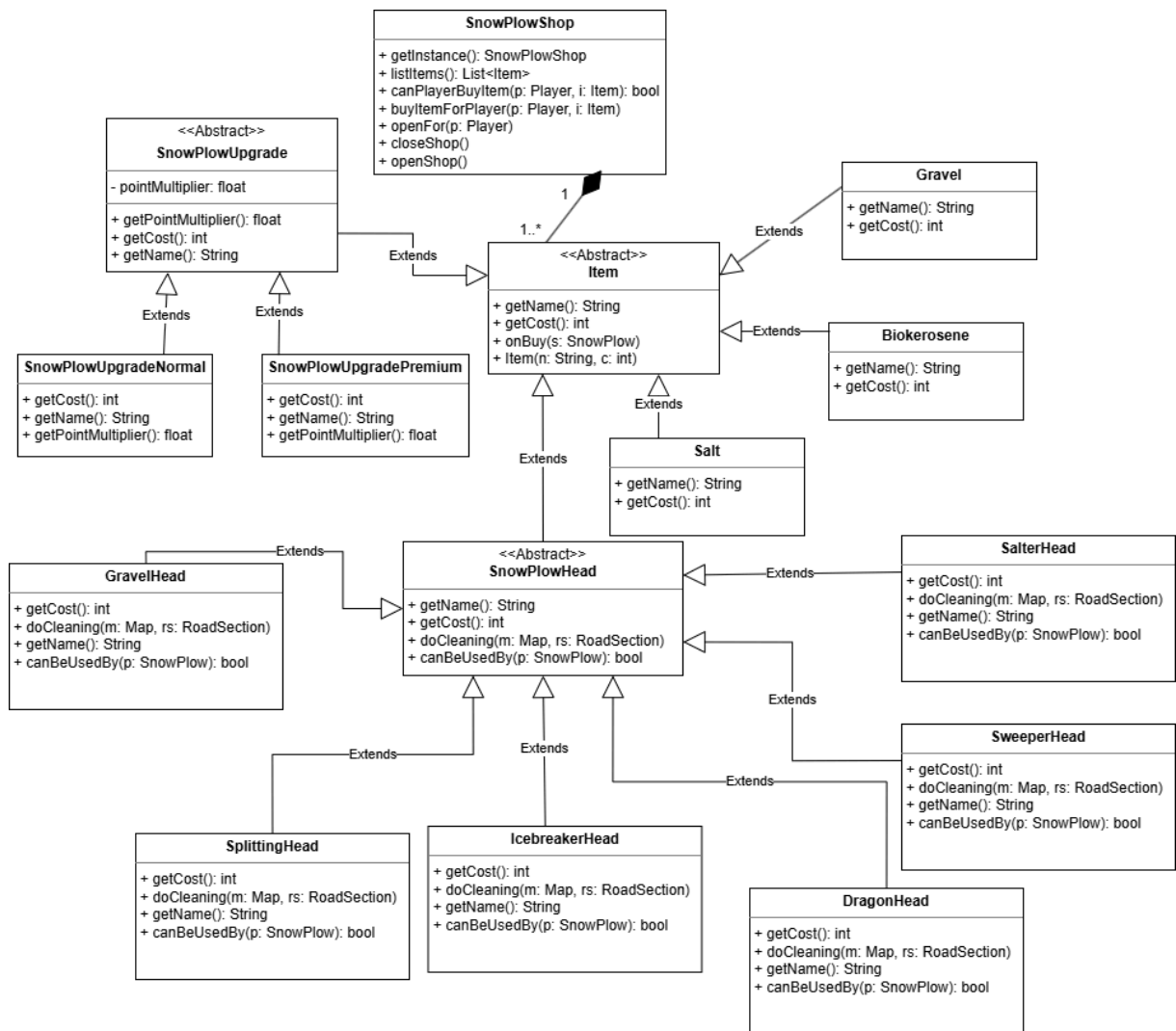
7.0.1 Módosult osztálydiagram



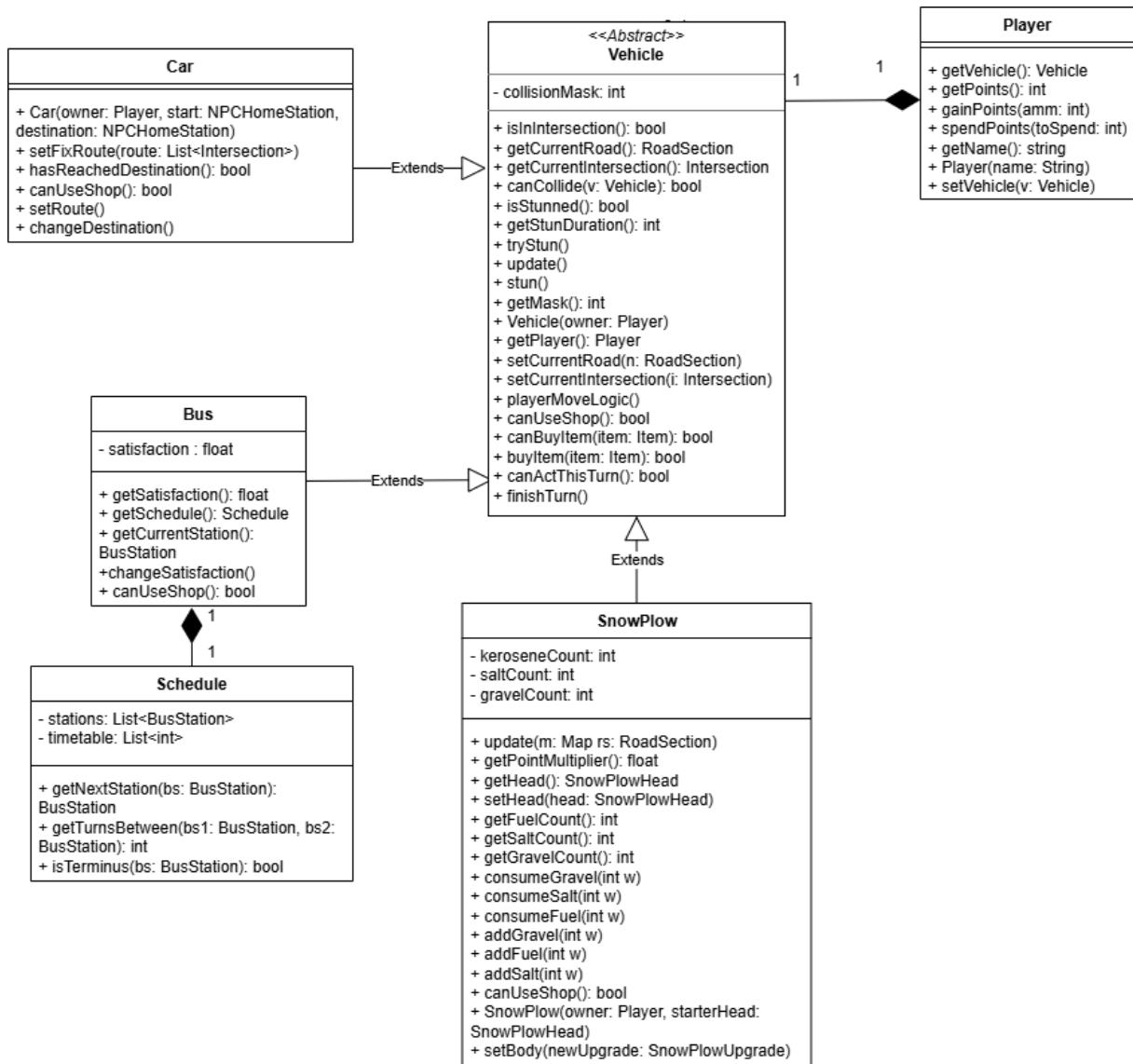
Teljes diagram (csak nevek)



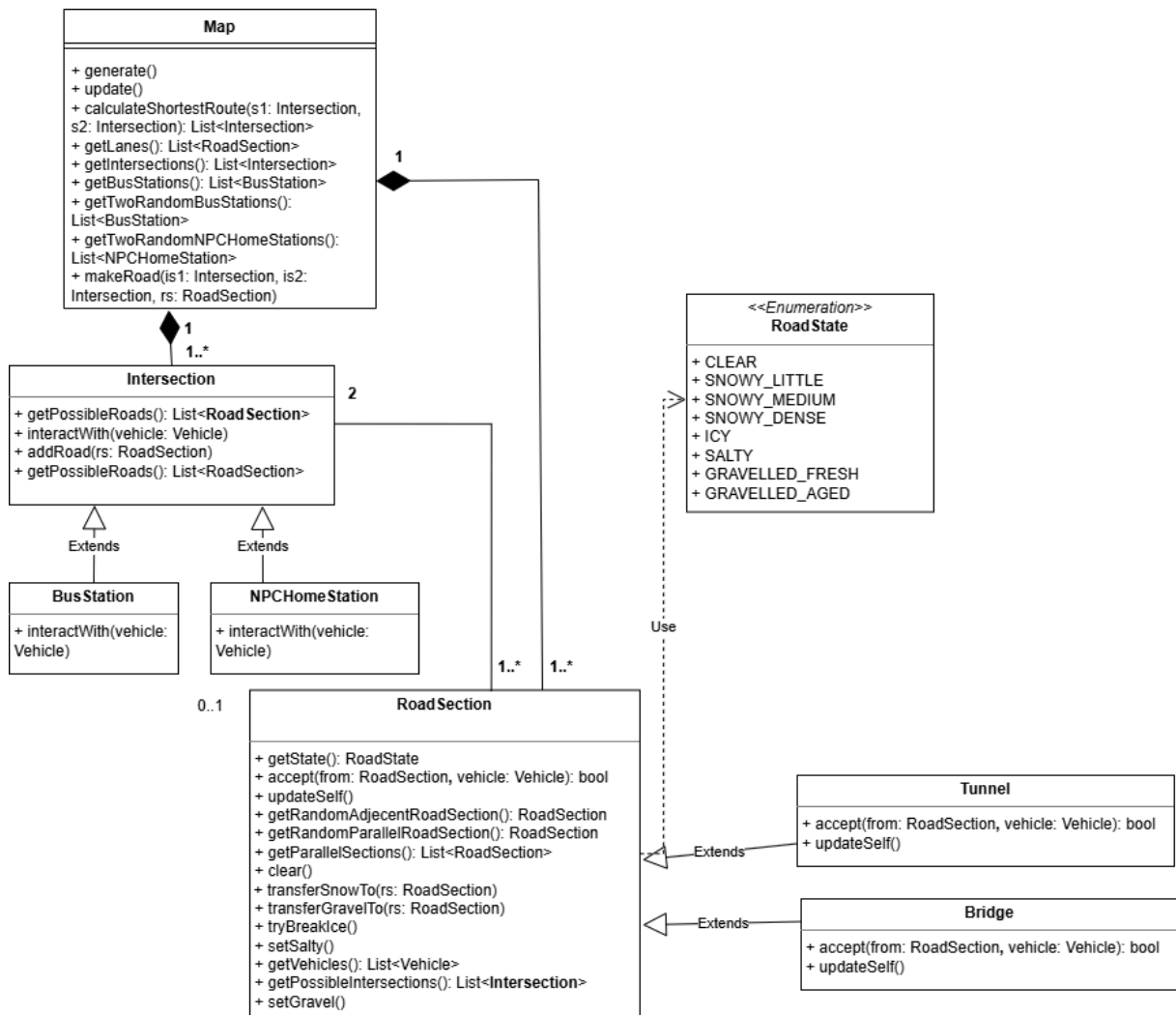
Core



Shop / Tárgyak



Járművek



Pálya

7.0.2 Új vagy megváltozó metódusok

7.0.3 Gravel

- **Felelősség**

A *GravelHead* üzemanyaga, amely nélkül a fej használhatatlan.

- **Asszociációk (Itemen keresztül)**

- **kompozíció-SnowPlowShop:** A *SnowPlowShop*on keresztül lehet vásárolni biokerozint, így a bolt tárol (legalább) egy közös Gravel példányt (1-1..*).

- **Interfészek**

Nincs

- **Ősosztályok**

Item

- **Attribútumok**

Nincs

- **Metódusok**
 - **int getCost():** A tárgy árát adja vissza.
 - **String getName():** A tárgy nevét adja vissza (“Gravel”).

7.0.4 RoadState

- **Felelősség**

Egy enumeráció, amelynek értékei a lehetséges útállapotok.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Értékek**
 - **CLEAR:** Tiszta út.
 - **SNOWY_LITTLE:** Enyhén havas út.
 - **SNOWY_MEDIUM:** Közepesen havas út.
 - **SNOWY_DENSE:** Sűrűn havas út.
 - **SALTY:** Sóval beszórt út.
 - **ICY:** Jeges út.
 - **GRAVELLED_FRESH:** Frissen zúzalékozott út.
 - **GRAVELLED_AGED:** Régen zúzalékozott út.

7.0.5 GravelHead

- **Felelősség**

Egy fajta hókotró fej. Csak akkor használható, ha a játékos rendelkezik *Gravel*-al. Ekkor ennek segítségével az adott sávban képes zúzalékot szórni, ami megszünteti a jeget és meggátolja egy ideig, hogy havas legyen az út, de végül a zúzalékot is befedi a hó.

- **Asszociációk (Itemen és keresztül)**
 - **kompozíció-SnowPlowShop:** A *SnowPlowShop*-ban lehet venni egyet vagy többet belőle (1..*-1).
 - **aggregáció-SnowPlow:** Minden *SnowPlow*-hoz pontosan egy fej tartozhat (1-1).
- **Interfészek**
Nincs
- **Ősosztályok**
SnowPlowHead → Item
- **Attribútumok**
Nincs
- **Metódusok**
 - **int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
 - **string getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
 - **void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow* ezt a fejet használva.

- **bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et. Igaz, ha a *SnowPlow*-nak van elég Gravel-je.

Egyenkénti új/létrejött metódusok:

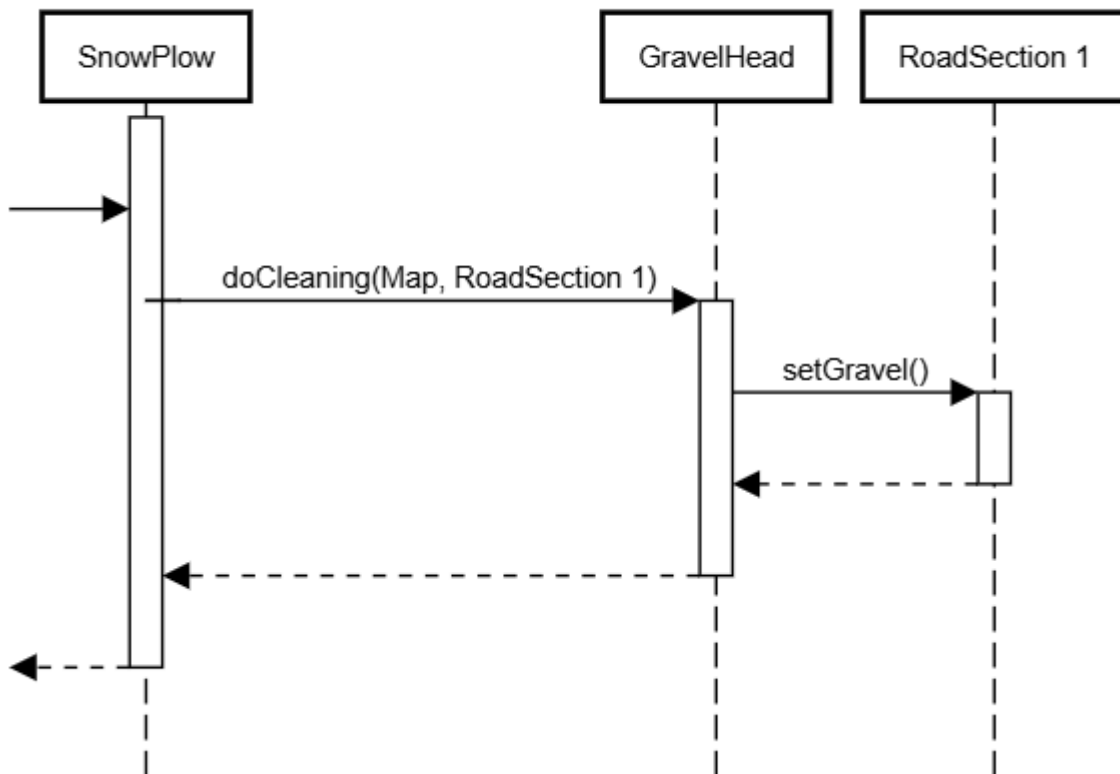
- **void RoadSection.setGravel():** Zúzalékot helyez el a jelenlegi úton.
- **void RoadSection.transferGravelTo(RoadSection from):** Zúzalékot helyez át egy másik útról a jelenlegi útra úgy, hogy az feltorlódhat.
- **int SnowPlow.getGravelCount():** Megkapja, hogy mennyi zúzalékja van a játékosnak. (A *SnowPlow*-nak lett egy per-instance privát adattagja is, a *gravelCount*, amely eltárolja, hogy éppen mennyi zúzalékja van annak a játékosnak)
- **void SnowPlow.addGravel():** Hozzáad egy vásárolt zúzalékot a játékos készletéhez, amennyiben az így nem haladja meg a *MAX_GRAVEL_COUNT* számú zúzalékot.
- **void SnowPlow.consumeGravel():** Elvesz egy vásárolt zúzalékot a játékos készletéből azért, hogy a zúzalékos fej jól működjön.

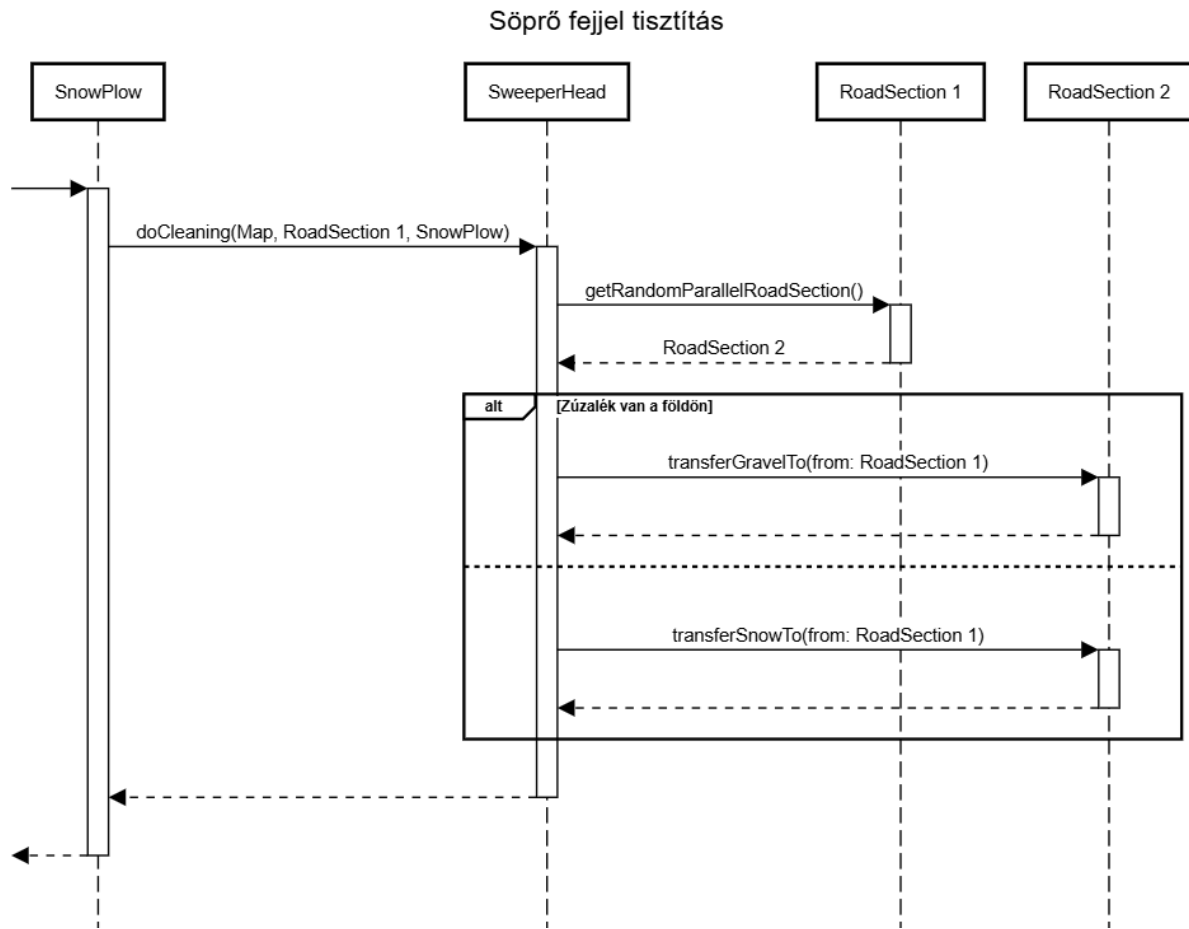
Egyenkénti megváltozott metódusok:

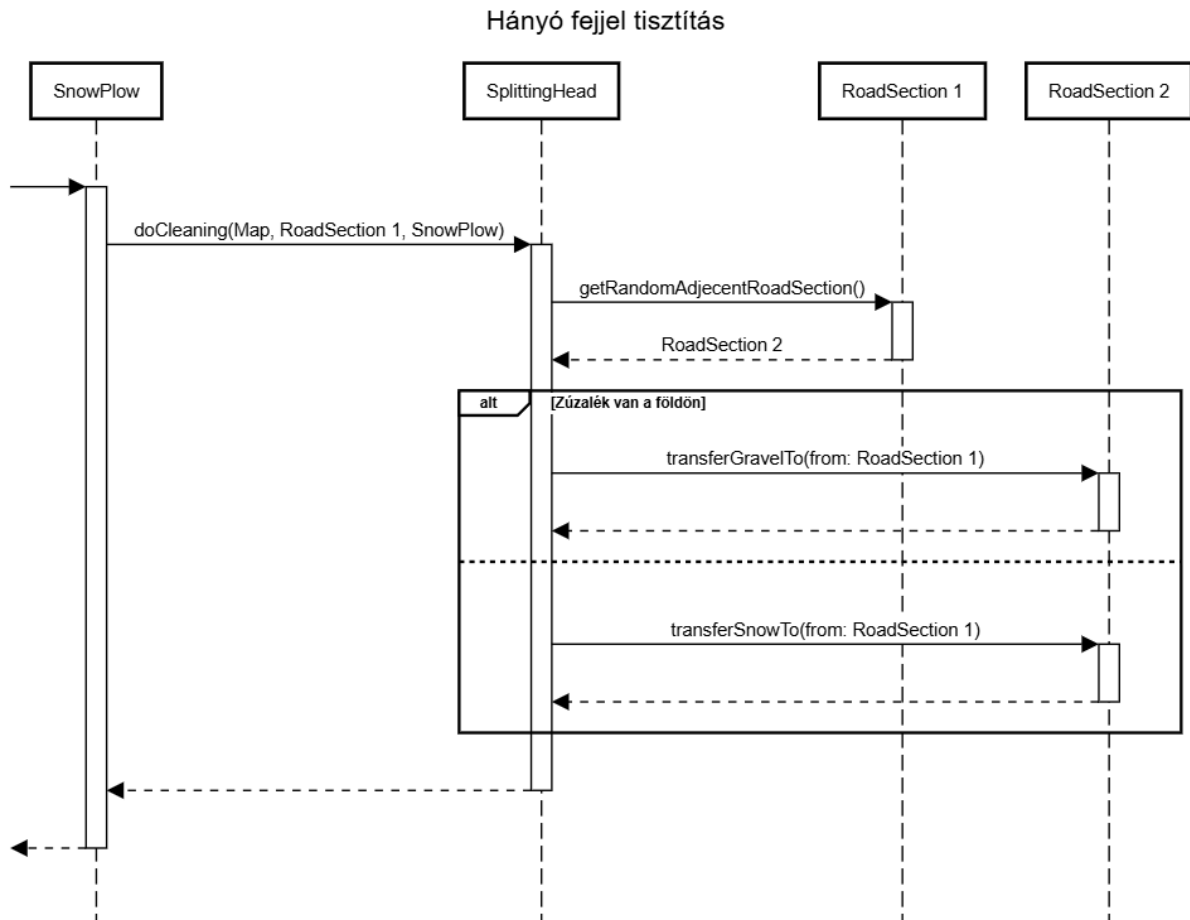
- **void RoadSection.setSalty():** Sót helyez el az útra, amennyiben az út nem zúzalékozott.
- **void SweeperHead.doCleaning(Map map, RoadSection rs, SnowPlow p):** Annyi változott, hogy ha zúzalék van a földön, akkor azt söpri egy párhuzamos útra, másképpen pedig a havat.
- **void SplittingHead.doCleaning(Map map, RoadSection rs, SnowPlow p):** Annyi változott, hogy ha zúzalék van a földön, akkor azt hánnya egy másik random érintkező útra, másképpen pedig a havat.
- **void DragonHead.doCleaning(Map map, RoadSection rs, SnowPlow p):** Annyi változott, hogy a sárkányfej már nem olvasztja el a zúzalékot.

7.0.6 Szekvencia-diagramok

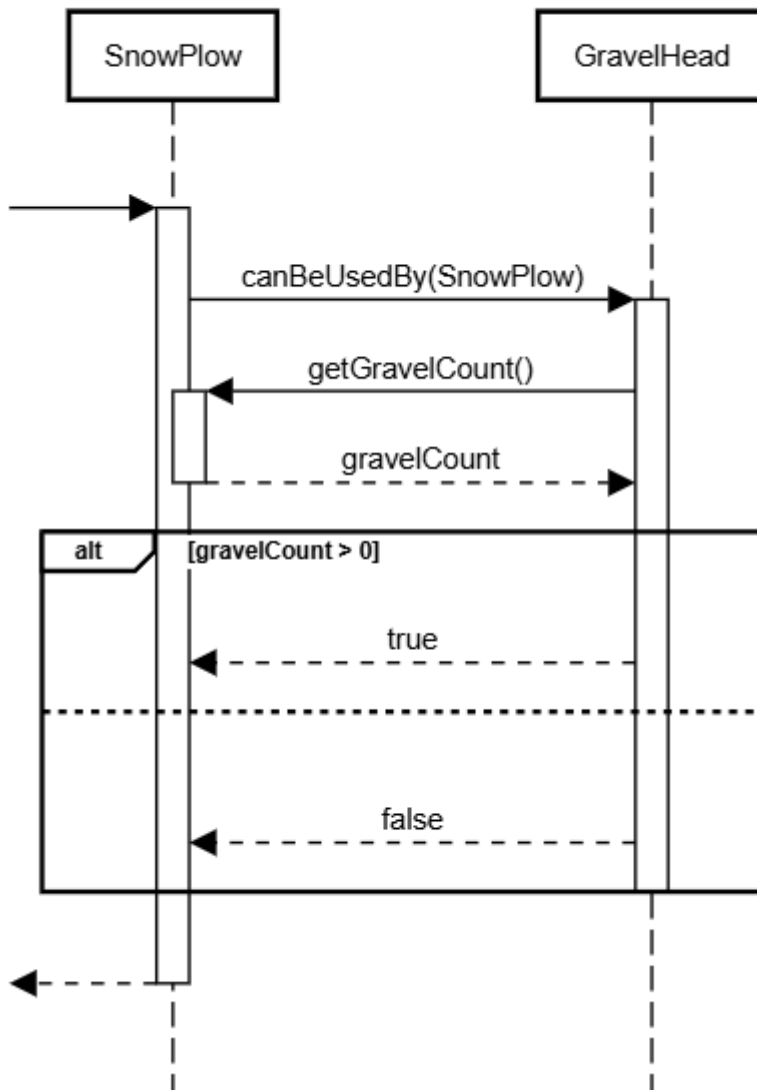
Zúzalék fejjel tisztítás



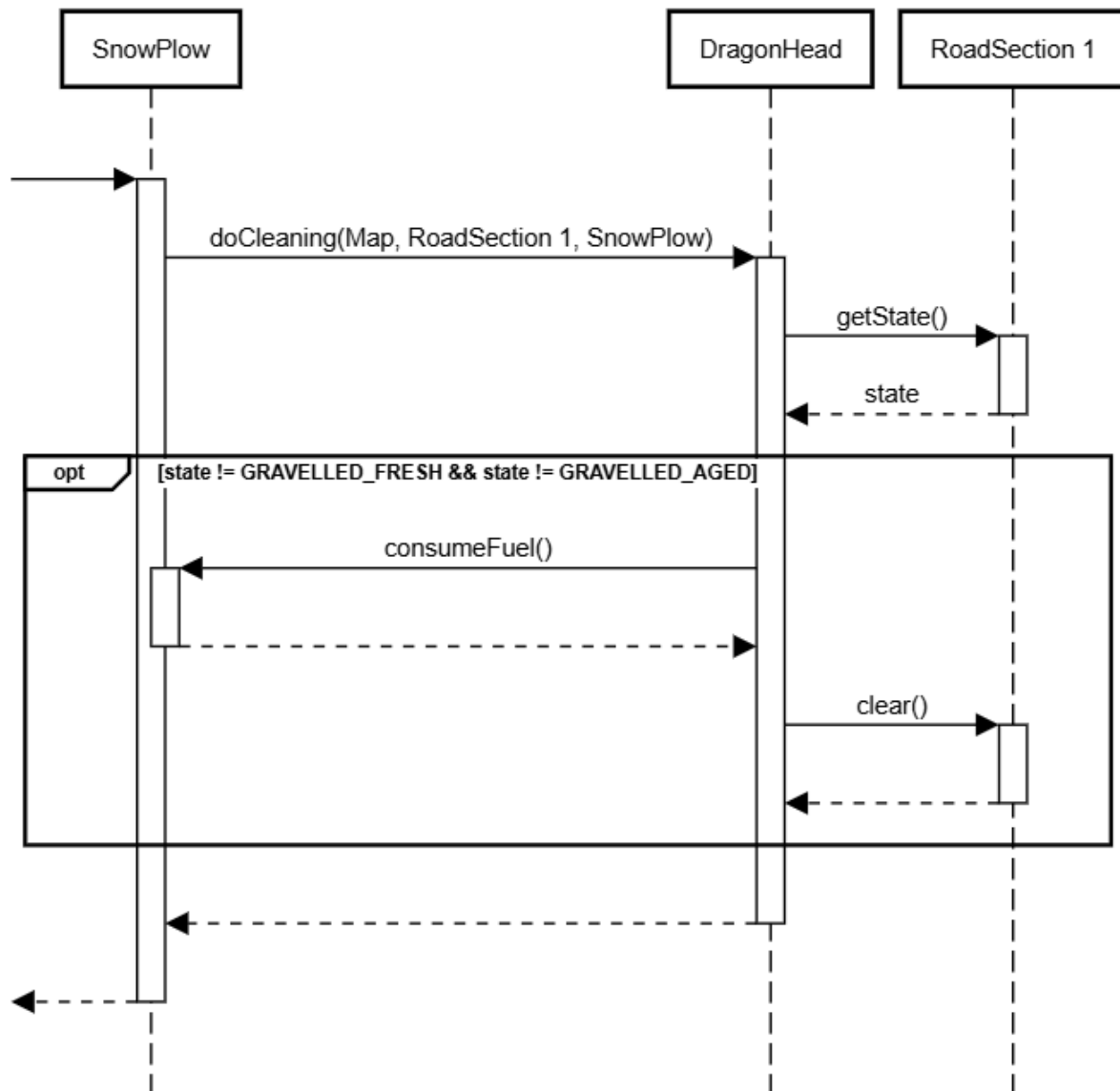




Zúzalékszóró fej használható-e



Sárkány fejjel tisztítás (Frissített)



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A program prototípusát a parancssoron keresztül lehet kezelni. A program parancsokkal irányítható és belső állapotának egy része is módosítható, illetve információk kérhetőek le róla. A parancsok szintaxisa a Bash-re hajaz. Egyenkénti parancskiadás mellett van lehetőség “parancsfájlból” (egyszerű szöveges fájl) egyszerre több parancs beolvasására, és futtatására egy úgynevezett “headless” módban. Egyes parancsok hibaüzenettel is visszatérhetnek, ha futtatásuk során hiba történt.

A parancsok kimenete fájlba menthető, mely után helyessége ellenőrizhető.

7.1.2 Bemeneti nyelv

Az alábbi parancsok leírása során a playername paraméter minden esetben a szóban forgó játékos nevét jelenti.

createplayer

Leírás: Új játékost hoz létre a megadott névvel és járműtípussal. A létrehozott játékos eltárolásra kerül, így későbbi parancsokban név alapján lehet rá hivatkozni. A második paraméter határozza meg a létrejött jármű típusát:

- **bus:** busz játékost hoz létre
- **snowplow:** hókotró játékost hoz létre.

Opciók: `createplayer <playername> <"bus" | "snowplow">`

placevehicle

Leírás:A megadott játékos a járművét egy adott sávra vagy útkereszteződésbe rakhatja. A második paraméter határozza meg, hogy hova történjen az elhelyezés:

- **lane:** a jármű egy adott sávra kerül
- **intersection:** a jármű egy adott kereszteződésbe kerül

A harmadik paraméter az adott sáv vagy kereszteződés indexe.

Opciók: `placevehicle <playername> <"lane" | "intersection"> <index>`

move

Leírás: A megadott játékos járművét egy lépésben mozgatja. A jármű mozgása a típusától és az aktuális pálya állapottól függ. A parancs végrehajtása során további interaktív inputot is kérhet, például:

- párhuzamos út vagy kereszteződés választása
- célút kiválasztása
- egyéb, a jármű mozgásához szükséges döntések

Opciók: `move <playername>`

openshop

Leírás: megnyitja a boltot egy olyan játékos számára, akinek van rá joga. Jelenlegi rendszerben ezzel a joggal a hókotró játékosok rendelkeznek.

Opció: `openshop <playername>`

closeshop

Leírás: bezárja a boltot, ha a játékos még a shop felületén van.

Opció: `closeshop`

tick

Leírás: A világ állapotát lépteti egy körrel, vagyis frissíti a szükséges objektumok állapotát.

Opció: `tick`

listitems

Leírás: kilistázza a shopban elérhető itemeket az indexükkel, nevükkel és árukkal. Csak a boltot megtekintésére feljogosult játékosok tekinthetik meg a

bolt tartalmát, azaz a hókotró vezető játékosok. A parancs nem vár paramétert. Célszerű a buy parancs használata előtt lefuttatni.

Opció: listitems

buy

Leírás: a hókotró játékosok kiválaszthatnak egy itemet a shopból, amit meg szeretnének vásárolni. Sikeres vásárlás esetén a pontok levonásra kerülnek és a tárgy hatása azonnal érvényesül a járművön. A második paraméter annak a játékosnak a neve, aki a vásárlást végzi, az item index paraméter a tárgy sorszám, ami megtekinthető a *listitems* segítségével.

Opció: buy <playername> <item index>

setbusroute

Leírás: a megadott busz játékos útvonalának beállítására szolgáló parancs. A busz két végállomást tárol és egy optimális időt (kört), amin belül érdemes teljesíteni a szakaszt a maximális elégedettségéhez. A megálló1index, megálló2index a két buszmegálló indexe, ami között kell közlekednie. A timetocomplete az az optimális körszám, ami alatt teljesítenie kell a távot.

Opció: setbusroute <playername> <station1index>
<station2index> <timetocomplete>

sethead

Leírás: a megadott hókotró játékos fejének cseréjére szolgáló parancs. A headtype pedig annak a konkrét fejnek a neve amire cserélni szeretnénk.

Opció: sethead <playername> <headtype>

setupgrade

Leírás: a megadott hókotró játékos fejlesztésének “cseréjére” szolgáló parancs. Az upgradetype pedig annak a konkrét fejlesztésnek a neve amit be szeretnénk állítani.

Opció: setupgrade <playername> <upgradetype>

setlanestate

Leírás: az adott útszakasz állapotát állítja be. Az első paraméter annak az útszakasznak az indexe, amelynek az állapotát változtatni szeretnénk. A második paraméter annak az állapotnak a neve, amire állítani szeretnénk.(pl. clear, snowy_little, snowy_medium, gravelled_fresh, stb.)

Opció: setlanestate <laneindex> <state>

spawn car

Leírás: NPC autót hoz létre, ami a két megadott NPC station között közlekedő Car-t helyez a világba. Útvonalat számít magának és a célba érve meg tud fordulni és újraszámolni az útvonalát. Az első paraméter az NPC-t azonosítja és az utolsó két paraméter az NPC végállomásai.

Opció: spawn car <ID vagy név> <NPCstartstation>
<NPCdeststation>

loadmap

Leírás: Pálya paraméterként megadott layout alapján vagy pedig fájlból, amely tartalmazza ezt a layoutot. A fájlnak .maplayout kiterjesztése kell legyen.

Opciók: loadmap <fájlnév | layout-sztring>

A layout szintaxisára példa:

```
is 0 npc;
is 1 npc;
is 2 stop;
is 3;
is 4 stop;
lane 0 0 1;
lane 1 0 1 bridge;
lane 2 1 3 tunnel;
lane 3 3 4;
```

A példa 4 db kereszteződést (*is*), és 4 db sávot (*lane*) definiál. Minden kereszteződés és sáv első paramétere egy azonosító, ami alapján később hivatkozni lehet rá.

A kereszteződések második paramétere opcionális, lehet “npc”: NPCHomeStation, vagy pedig “stop”: BusStation.

A sávok második és harmadik paramétere határozza meg a végpontjaikat, ezek már definiált kereszteződések azonosítói. A negyedik, és egyben utolsó paraméter pedig opcionális, a sáv típusát határozza meg. “bridge”: híd, “tunnel”: alagút, egyébként pedig rendes útsáv.

Minden “sor” pontosvesszővel kell zárni, emiatt egyszerre több “sor” is lehet egy sorban.

Hibás formátum vagy IO hiba esetén megfelelő hibaüzenettel tér vissza.

stat

Leírás: Sáv, játékos vagy NPC állapotának lekérdezése.

Opciók: stat <“lane” | “player” | “npc”> <ID vagy név>

getturn

Leírás: Kiírja a jelenlegi, és utolsó (maximális) kör számát.

Opciók: getturn

leaderboard

Leírás: Kilistázza a játékosok nevét pontjaikkal párba állítva, pontjaik szerint csökkenő sorrendben.

Opciók: leaderboard

clean

Leírás: Megadott hókotrójátékos megpróbálja megtisztítani a sávját a jelenleg aktív fejével. Buszofőr esetén nem lehet használni.

Opciók: clean <játékosnév>

addfuel

Leírás: Valamilyen üzemanyag típusból megadott mennyiségűt ad a megadott hókotrójátékosnak.

Opciók: addfuel <játékosnév> <“salt” | “bio” | “gravel”> <mennyiség>

quit

Leírás: Kilép a proto-programból.

Opciók: `quit`

help

Leírás: Kilistázza az elérhető parancsokat.

Opciók: `help`

7.1.3 Kimeneti nyelv

A legtöbb parancs valamilyen módon, legalább egy státuszüzenet formájában jelzi, hogy sikeres volt-e a lefutás vagy sem. Sikeres lefutás: “Siker”; sikertelen lefutás esetén pedig a hibára vonatkozó üzenettel tér vissza.

Egyedi output formátumok:

help:

Elérhető parancsok:

<parancs>

<parancs>

[...]

getturn:

<jelenlegi kör> / <maximális körszám>

leaderboard:

<játékosnév> <pont>

<játékosnév> <pont>

[...]

stat:

1. Sáv:

<id>: <intersection1>-<intersection2>, <állapot>, <rajta levő járművek vességével elválasztva: játékosnevek és NPC id-k>

2. Játékos:

<név>: <pontszám>, <jármű típusa>, <jelenlegi sávjának id-ja>, <stun időtartam (persze, ha mozoghat, akkor ez 0)>

3. NPC (kocsi):

<id>: <kezdő NPCHomeStation>-<végső NPCHomeStation>, <jelenlegi sávjának id-ja>, <stun időtartam>

move: (futtatás utáni kérdések)

1. Párhuzamos sávra szeretne lépni?

- Ezt választva meghívja a vele párhuzamos sávokra a `stat` parancsot.
- A játékos id alapján tud sávot váltani.

2. Kereszteződésen keresztül szeretne mozogni?

- Ezt választva meghívja a `stat` parancsot saját magán, illetve a közrefogó kereszteződésekben kiinduló többi sávra is.

i. Output formátum:

<stat outputja a jelenlegi sávra> ->

<stat outputja szomszédos (nem párhuzamos) sávra>

<stat outputja szomszédos (nem párhuzamos) sávra>

[...]

- b. A játékos id alapján tud utat (és azon belül sávot) váltani.

listitems:

<index>. <név>: <ár>

<index>. <név>: <ár>

[...]

7.2 Összes részletes use-case

Use-case neve	View shop
Rövid leírás	A hókotró játékos megnyitja a boltot és megtekinti az elérhető tárgyak listáját.
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a az openshop <playername> parancsot 2. A rendszer ellenőrzi, hogy létezik e a megadott nevű játékos 3. A rendszer ellenőrzi a jármű típusát 4. A rendszer ellenőrzi, hogy a játékos jogosult e a vásárlásra 5. Ha jogosult megnyitja a shop-ot a játékos számára 6. A shop kiírja a megvásárolható tárgyakat 7. Ha a játékos nem létezik vagy nem jogosult a shop megnyitására akkor a rendszer hibát dob

Use-case neve	View object statistics
Rövid leírás	A játékos lekéri egy objektum statisztikáját.
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a stat <"lane" "player" "npc"> <ID vagy név> parancsot 2. A rendszer ellenőrzi, hogy létezik e a megadott objektum. 3. A rendszer kiírja az objektum belső statisztikáit.

Use-case neve	View turn statistics
Rövid leírás	A játékos lekéri azt, hogy hanyadik körben van
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a getturn parancsot. 2. A rendszer kiírja azt, hogy hányadik kör van és melyik az utolsó kör.

Use-case neve	View leaderboard
Rövid leírás	A játékos lekéri ponttáblát
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a leaderboard parancsot. 2. A rendszer kiírja a játékosokat rangsor szerint és pontjaikat.

Use-case neve	Clean road
Rövid leírás	A hókotró játékos letisztítja az utat, amin éppen van.
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a clean <játékosnév parancsot> 2. A rendszer megnézi, hogy a játékos létezik-e név szerint. 3. A rendszer letisztítja az utat, amin van a játékos a birtokolt fejjel.

Use-case neve	Add resource
Rövid leírás	A játékos üzemanyagot/fejhez kellő erőforrást kap.
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a addfuel <játékosnév> <"salt" "bio" "gravel"> <mennyiség> parancsot. 2. A rendszer megnézi, hogy a játékos létezik-e név szerint. Itt ellenőrződik az is, hogy a játékosnál elfér-e az a mennyiség, ha van korlát beállítva. 3. A rendszer odaadja a kért erőforrásokat a játékosnak.

Use-case neve	Change head
Rövid leírás	A játékos a hókotrójára új fejet rak úgy, hogy kicseréli az előzőt.
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a setupgrade <playertype> <upgradetype> parancsot. 2. A rendszer megnézi, hogy a játékos létezik-e név szerint és a fej is létezik-e 3. A rendszer kicseréli a kért fejre a játékos hókotrójának a fejét.

Use-case neve	Move player-vehicle to intersection
Rövid leírás	A játékos az általa irányított járművet a választott kereszteződésre irányítja
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a az openshop <playertype> parancsot 2. A rendszer ellenőrzi, hogy létezik-e a megadott nevű játékos 3. A rendszer lekéri a jármű aktuális állapotát és helyzetét 4. Megjelenik a játékos számára a választható mozgási lehetőségek 5. A Player a kereszteződés felé vezető opciót választja 6. Kiíródik az elérhető kereszteződések opciói 7. Kiválaszt egy tetszőlegeset a Player

	8. A rendszer a kiválasztott kereszteződésre mozgatja
	9. Ha a választás érvénytelen vagy nem végrehajtható akkor hibaüzenet íródik ki

Use-case neve	Move player-vehicle to roadsection
Rövid leírás	A játékos az általa irányított járművet a választott kereszteződésre irányítja
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A Player kiadja a az move <playername> parancsot 2. A rendszer ellenőrzi, hogy létezik e a megadott nevű játékos 3. A rendszer lekéri a jármű aktuális állapotát és helyzetét 4. A Player számára megjelennek a választható mozgási opciók 5. A Player kiválaszt egy útszakaszt vagy egy kereszteződésen keresztül juthat el egyre. 6. A rendszer ellenőrzi, hogy a kiválasztott útszakaszra a jármű beléphet e. 7. Ha a mozgás lehetséges a jármű átkerül a kiválasztott útszakaszra 8. Ha az útvonalválasztás érvénytelen vagy nem lehetséges akkor hibaüzenet íródik ki

Use-case neve	Choose player type
Rövid leírás	A játékos kiválasztja, hogy buszt vagy hókotrót szeretne irányítani és a PlayerFactory létrehozza a megfelelő játékost a járművel
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. A player kiadja a createplayer <player name> <bus snowplow> parancsot 2. A rendszer ellenőrzi, hogy a megadott névvel létezik e már játékos 3. Ha a jármű típusa bus, létrehozásra kerül egy busz játékost 4. Ha a jármű típusa snowplwo, létrehozásra kerül egy hókotró játékost 5. A létrehozott játékos eltárolásra kerül 6. Ha a név már foglalt vagy a típus ismeretlen, hibaüzenet íródik ki

Use-case neve	Buy item
Rövid leírás	Egy snowplow játékos megvásárol egy tárgyat a shopból
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. a Player kiadja a buy <playername> <item index> parancsot

	2. A rendszer ellenőrzi a játékos létezését név alapján
	3. A rendszer ellenőrzi, hogy a játékos a járművének típusa alapján igénybe veheti e a shopot
	4. Lekérésre kerül a megvásárolható tárgyak listája
	5. A rendszer ellenőrzi, hogy a megadott item index érvényes e
	6. Kiválasztódik a megvenni kívánt tárgy
	7. A shop ellenőrzi, hogy a hókotró vásárló alkalmas e a tárgy megvásárlására
	8. Ha minden feltétel teljesül, a pontok levonásra kerülnek a játékostól
	9. A rendszer a megvásárolt tárgyat a járműhöz rendeli és aktiválja annak a hatását
	10. Ha a játékos nem létezik, nem fér hozzá a bolthoz, az index hibás, a tárgy nem megvásárolható vagy nincs elég pont, a rendszer hibüzenetet ír ki.

Use-case neve	Update world
Rövid leírás	A játékos lezár egy kört és a világ állapota frissítődik
Aktorok	Player
Forgatókönyv	1. a Player kiadja a step parancsot
	2. megkezdődik a világ egy körrel való frissítése
	3. Frissítődik a pálya állapota
	4. Az automatikusan frissülő objektumok körhöz tartozó műveleteik lefutnak
	5. A rendszer visszajelez a sikeres frissítésről

Use-case neve	Load map
Rövid leírás	A felhasználó betölt egy pályát a prototípusba fájlból vagy közvetlenül megad egy layoutot
Aktorok	Player
Forgatókönyv	1. A Player kiadja a loadmap <fájlnévlayout sztring> parancsot
	2. Ha a paraméter .maplayout kiterjesztésű fájl, a rendszer megpróbálja beolvasni
	3. Ha a paraméter nem fájlnev, a rendszer layout leírásként értelmezi a megadott szöveget
	4. Feldolgozásra kerül a layout
	5. Létrehozódnak a pályához tartozó kereszteződések, megállók, NPC állomások és útszakaszok
	6. Felépül a pálya a beolvasott leírás alapján
	7. A rendszer sikerüzenetet ír ki a pálya felépítéséről

	8. Ha a fájl nem található, a formátum hibás vagy a layout nem dolgozható fel a rendszer hibaüzenetet ír ki
--	---

Use-case neve	Quit
Rövid leírás	A környezet bezárása.
Aktorok	Player
Forgatókönyv	A környezet bezáródik.

Use-case neve	Help
Rövid leírás	A környezet parancsait kilistázza
Aktorok	Player
Forgatókönyv	A környezet kiírja az irányítónak a lehetséges parancsokat.

7.3 Tesztelési terv

Amikor a “Vehicle”/”Jármű”-ről van szó, akkor ott azt értjük, hogy ha az egy SnowPlow, akkor olyan, mint ha nem lenne rajta semmilyen fej.

Teszt-eset neve	1.Vehicle moves to ice.
Rövid leírás	Egy jármű egy jeges útra megy.
Teszt célja	A jármű megcsúszik és irányíthatatlanná válik.

Teszt-eset neve	2.Vehicle moves to clear road section.
Rövid leírás	Egy jármű egy tiszta útra megy.
Teszt célja	A jármű minden akadály nélkül a célhelyre mozog.

Teszt-eset neve	3.Vehicle moves to salted road section.
Rövid leírás	Egy jármű egy sózott útra megy.
Teszt célja	A jármű minden akadály nélkül a célhelyre mozog.

Teszt-eset neve	4.Vehicle moves to gravelled road section.
Rövid leírás	Egy jármű egy zúzalékos útra megy.
Teszt célja	A jármű minden akadály nélkül a célhelyre mozog.

Teszt-eset neve	5.Vehicle moves to dense snowy road section.
Rövid leírás	Egy jármű egy nagyon havas útra megy.
Teszt célja	A jármű megakad és mozgásképtelenné válik.

Teszt-eset neve	6.Vehicle moves to slightly snowy road section.
Rövid leírás	Egy jármű egy enyhén havas útra megy.
Teszt célja	A jármű letapossa a havat és az jéggé válik.

Itt azt feltételezzük, hogy a fejekhez szükséges anyag megvan.

Teszt-eset neve	7.SnowPlow uses DragonHead on snow.
------------------------	-------------------------------------

Rövid leírás	A hókotró a sárkány fejjel havat olvaszt.
Teszt célja	A hó elolvad.

Teszt-eset neve	8.SnowPlow uses DragonHead on gravel
Rövid leírás	A hókotró a sárkány fejjel zúzalékos utat olvaszt.
Teszt célja	A zúzalék megmarad.

Teszt-eset neve	9.SnowPlow uses SalterHead.
Rövid leírás	A hókotró a havas sávra sót rak.
Teszt célja	A hó megszűnik és só marad a helyén.

Teszt-eset neve	10.SnowPlow uses GravelHead on snow.
Rövid leírás	A hókotró a havas sávra zúzalékot rak.
Teszt célja	A hó megszűnik és zúzalék marad a helyén.

Teszt-eset neve	11.SnowPlow uses SweeperHead.
Rövid leírás	A hókotró egy havas/zúzalékos sáv tartalmát át takarítja egy párhuzamos sávra.
Teszt célja	A hó/zúzalék megszűnik és áttolódik a másik sávra.

Teszt-eset neve	12.SnowPlow uses SplitterHead.
Rövid leírás	A hókotró egy havas/zúzalékos sáv tartalmát át takarítja egy véletlenszerű, másik sávra.
Teszt célja	A hó/zúzalék megszűnik és áttolódik a másik sávra.

Ezek a hókotró-specifikus tesztek feltételezik, hogy nincs meg a fejekhez használható alapanyag.

Teszt-eset neve	13.SnowPlow uses Dragonhead with no fuel.
Rövid leírás	A hókotró megpróbálja a fejet használni, de nincs hozzá biokerozin.
Teszt célja	A hókotró nem takarítja le a sávot és rámegy, mint ha egy sima jármű lenne fej interakció nélkül.

Teszt-eset neve	14.SnowPlow uses SalterHead with no fuel.
Rövid leírás	A hókotró megpróbálja a fejet használni, de nincs hozzá só.
Teszt célja	A hókotró nem takarítja le a sávot és rámegy, mint ha egy sima jármű lenne fej interakció nélkül.

Teszt-eset neve	15.SnowPlow uses GravelHead with no fuel.
Rövid leírás	A hókotró megpróbálja a fejet használni, de nincs hozzá zúzalék.
Teszt célja	A hókotró nem takarítja le a sávot és rámegy, mint ha egy sima jármű lenne fej interakció nélkül.

Teszt-eset neve	16.SnowPlow buys SnowPlowUpgradePremium
Rövid leírás	A hókotró játékos egy új hókotrórt vesz, amellyel több pontot tud szerezni.

Teszt célja	A hókotró ára kikerül a játékos pontjából és a fejlesztés applikálódik.
--------------------	---

Teszt-eset neve	17.SnowPlow can't buy SnowPlowUpgradePremium
Rövid leírás	A hókotró játékos egy új hókotrórt vesz, amellyel több pontot tud szerezni, de nincs pénze.
Teszt célja	A hókotró nem tudja megvenni a tárgyat, mert nincs elég pénze.

Teszt-eset neve	18.SnowPlow can't buy SnowPlowHead
Rövid leírás	A hókotró játékos egy új fejet vesz, de nincs pénze.
Teszt célja	A hókotró nem tudja megvenni a tárgyat, mert nincs elég pénze.

Teszt-eset neve	19.SnowPlow buys SnowPlowHead
Rövid leírás	A hókotró játékos egy új fejet vesz, amellyel megváltozik a viselkedése.
Teszt célja	A fej ára kikerül a játékos pontjából és a fejlesztés applikálódik.

Teszt-eset neve	20.Vehicle collision
Rövid leírás	Két jármű ütközik.
Teszt célja	A járművek mozgásképtellené válnak.

Teszt-eset neve	21.Bus reaches station
Rövid leírás	A busz eléri a megállóját, és pontot kap.
Teszt célja	A busz beér a megállóba és teljesítménye alapján pontot kap. Ezen felül a céllálomása megváltozik.

Teszt-eset neve	22.Car reaches destination
Rövid leírás	Az autó eléri a kijelölt célját.
Teszt célja	Az autó eléri a kijelölt célját és az új célállomása az eredeti kiindulópontja lesz.

Teszt-eset neve	23.SnowPlow buys Salt
Rövid leírás	A hókotró játékos sót vesz.
Teszt célja	A só ára kikerül a játékos pontjából és a hókotró sójának száma növekszik.

Teszt-eset neve	24.SnowPlow buys Gravel
Rövid leírás	A hókotró játékos zúzalékot vesz.
Teszt célja	A zúzalék ára kikerül a játékos pontjából és a hókotró zúzalékának száma növekszik.

Teszt-eset neve	25.SnowPlow buys Biokerosene
Rövid leírás	A hókotró játékos biokerozint vesz.
Teszt célja	A biokerozint ára kikerül a játékos pontjából és a hókotró biokerozinjának száma növekszik.

Teszt-eset neve	26.SnowPlow can't buy Gravel
Rövid leírás	A hókotró játékos zúzalékot próbál venni.
Teszt célja	A hókotró nem tud zúzalékot venni, mert elérte a maximumot, amennyit tulajdonolhat.

Teszt-eset neve	27.Snowing
Rövid leírás	Egy eddig tiszta út havas lesz a havazás miatt.
Teszt célja	A vizsgált tiszta útra hó kerül.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A proto-teszteléshez egy Java tesztprogram lesz elérhető, amely összetartozó tesztinput és -output fájlokat ellenőriz. Az input fájlok parancsokat tartalmaznak (akár persze egyszerre többet is), az output fájlok pedig ezen parancsok elvárt kimeneteit.

A tesztprogram indítására egy-egy *bash* és *batch* szkript fog rendelkezésre állni.

A program képes a teszteket egyesével futtatni (a felhasználó ki tudja választani a futtatni kívánt tesztesetet), vagy az összeset egyszerre.

Egy tesztesethez kettő fájl tartozik: `<tesztnév>.testin`, amely az inputparancsokat tartalmazza, és `<tesztnév>.testout`, amely pedig az elvárt végeredményt.

Teszteset futtatását követve a program jelzi annak sikerességét. Sikertelen esetben jelzi, hogy hol nem egyezik meg az elvárt és valós output.

7.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2026.04.03. 12:00:00		0,8 Mindenki	Értekezlet. Döntés: Változtatások + Feladatok kiosztása.
2026.04.03. 18:00:00		1 Funk	Tevékenység: Új elemek kódba implementálása és diagramok kiegészítése.
2026.04.04. 18:00:00		1 Funk	Tevékenység: Mozcás implementálása
2026.04.04. 18:00:00		1 Fekete	Tevékenység: RoadSection frissítése
2026.04.05. 9:00:00		0,9 Funk	Tevékenység: Új szekvenciák létrehozása és új elemek kódba implementálása
2026.04.05. 8:00:00		1,5 Rózsa	Tevékenység: shop logika refaktorálása
2026.04.05. 10:00:00		2 Rózsa	Tevékenység: saját exceptionök felvétele, SL logok kizsedése, belső logikák megvalósítása
2026.04.05. 17:00:00		0,4 Funk	Tevékenység: SL-logok kizsedése
2026.04.06. 19:00:00		1 Heltai	Tevékenység: Protípus általános interface leírás; map config beolvasó.
2026.04.06. 20:30:00		3,5 Csordás	Tevékenység: BFS kialakítása, Car update, általános mozgás kialakítása, BusStation és NPCHomeStation, Bus és Car legenerálása
2026.04.06. 21:00:00		3 Fekete	Tevékenység: Car update, általános mozgás kialakítása, BusStation és

				NPCHomeStation, Bus és Car legenerálásának megalkotása
2026.04.07. 15:45:00	1,5	Heltai		Tevékenység: Proto-parancsfeldolgozó és pár parancs.
2026.04.07. 9:00:00	1	Funk		Tevékenység: A Mi változott szekció kidolgozása.
2026.04.02. 0:00:00	0,42	Rózsa		Tevékenység: refaktor
2026.04.09. 11:00:00	0,3	<i>Mindenki</i>		Értekezlet. Döntés: feladatok kiosztása + review.
2026.04.09. 13:00:00	0,5	Funk		Tevékenység: Kommentek + review
2026.04.09. 13:30:00	0,67	Funk		Tevékenység: Use case-k + tesztek felvétele.
2026.04.09. 18:30:00	0,3	Funk		Tevékenység: Tesztek felsorolásának befejezése.
2026.04.10. 11:00:00	2,8	Rózsa		Tevékenység: Commandok létrehozása + belső működés, újítások tanulmányozása
2026.04.10. 11:00:00	0,5	Heltai		Tevékenység: Tesztelést támogató segéd- és fordítóprogramok specifikálása, proto javítása
2026.04.10. 21:00:00	1,7	Rózsa		Tevékenység: 7.1.2 Bemeneti nyelv és 7.2 részletes use case megírása
2026.04.11. 0:00:00	0,5	Rózsa		Tevékenység: buyItem és BuyCommand
2026.04.11. 9:00:00	1,2	Funk		Tevékenység: UML diagram szinkronizálása.
2026.04.11. 11:20:00	0,7	Funk		Tevékenység: Kommentezések a járműveknél.

2026.04.11. 11:40:00	0,65	Funk	Tevékenység: Review, UML diagramok dokumentációba darabolása.
2026.04.11. 14:30:00	0,6	Heltai	Tevékenység: még több parancs & kimeneti nyelv specifikálása.
2026.04.11. 18:40:00	0,6	Funk	Tevékenység: Use-case-k kiegészítése.
2026.04.12. 21:20:00	0,1	Rózsa	Tevékenység: apróbb javítások

8. Részletes tervek

8.1 Osztályok és metódusok tervei.

8.1.1 App

- **Felelősség**

Az App a játék futásának központi felügyelője. Feladata, hogy a játék indításakor létrehozza és elérhetővé tegye a legfontosabb részeket, mint a pálya, játékosok és járművek és az, hogy a játék futása alatt ezek állapotát egyben tartsa.

Az App tárolja, hogy a játék éppen fut-e, és rajta keresztül lehet lekérni a játék részeit, például a játékosokat és a pályát.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **-bool isRunning:** Fut-e éppen a játék.
- **-Map map:** A pálya.
- **-PlayerFactory playerFactory:** Játékosok létrehozásához használt factory.
- **-List<Player> players:** Játékosok.
- **-TurnController turnController:** Körök kezeléséért felelős kontroller.
- **-List<Vehicle> vehicles:** Játékban példányosított járművek.
- **-Player currentPlayer:** Aktív körrel rendelkező játékos.
- **-static App instance:** Az *App* egy singleton osztály, ez a statikus változója tárolja az instance-t.

- **Metódusok**

- **+static App getInstance():** Az app-instance elérését (és legelső meghíváskor létrehozását) segítő metódus.
- **+static void reset():** Teszteléshez használt metódus; alaphelyzetbe állítja a játékot.
- **+List<Player> getLeaderboard():** Ponttábla lekérdezése.
- **+Map getMap():** Pálya lekérdezése.
- **+List<Vehicle> getVehicles():** A pályán aktív járművek lekérdezése.
- **+Player getCurrentPlayer():** Jelenleg aktív játékos lekérdezése.
- **+void run():** A játék elindításáért és futtatásáért felelős metódus.
- **+void addPlayers():** Játékosok regisztrálása.
- **+void updateWorld():** Könnyebb tesztelhetőség érdekében létezik, kizárólag a `turnController.tick()`-et hívja meg.

8.1.2 Biokerosene

- **Felelősség**

A *DragonHead* üzemanyaga, nélküle a fej használhatatlan.

- **Interfészek**

Nincs

- **Őosztályok**

Item

- **Attribútumok**

Csak örökölt.

- **Metódusok**
 - **+void onBuy(SnowPlow):** Vétel után a vevő hókotró számára üzemanyagot biztosít.

8.1.3 Bridge

- **Felelősség**

A *Bridge* egy olyan sáv (vagy útszakasz), amely a föld fölött haladva elkerüli a többi sávval való találkozást, így nem alakulnak ki keresztezések.

- **Őosztályok**

RoadSection

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+bool accept(RoadSection from, Vehicle vehicle):** Visszaadja logikai értéként hogy az attribútumként kapott *RoadSection* felől az attribútumként kapott *Vehicle* át tud-e menni ezen a *Bridge*-en.
- **+void updateSelf():** Ez kezeli és frissíti a sáv állapotát körről-körre. (Statechart a *RoadState*-nél).

8.1.4 Bus

- **Felelősség**

A *Bus* a buszsofőr játékosok járműve. Minden busznak van egy menetrendje, amit teljesítenie kell, és minden fordulóért pontot kap.

- **Őosztályok**

Vehicle

- **Interfészek**

Nincs

- **Attribútumok**

- **-BusStation destination1:** A forduló egyik végállomása.
- **-BusStation destination2:** A forduló másik végállomása.

- **-int timeToComplete:** A maximális elégedettséghez legfeljebb ennyi kör alatt kell teljesítenie a fordulót.
 - **-float satisfaction:** Utasok elégedettségi mutatója.
 - **-BusStation lastVisited:** Az előző megálló, ahol megállt.
 - **-int turnsCompleted:** Megtett fordulók száma.
 - **-int travelTime:** A előző megálló óta eltelt körök száma.
- **Metódusok**
 - **+float getSatisfaction():** Az elégedettségi szorzó lekérdezése. A pontszámításnál fontos.
 - **+BusStation getLastVisited():** Előző megálló lekérdezése.
 - **+void setLastVisited(BusStation):** Előző megálló beállítása.
 - **+void addTurn():** Egyel növeli a megtett fordulók számát.
 - **+int getTurns():** Megtett fordulók számának lekérdezése.
 - **+void resetTravelTime():** Eltelt körök számának alaphelyzetbe (0) állítása.
 - **+void increaseTravelTime():** Egyeli növeli a travelTime értékét.
 - **+int getTravelTime():** travelTime lekérdezése.
 - **+void changeSatisfaction(float):** Elégedettség módosítása (additív).
 - **+boolean isDestination(BusStation):** Az kapott megálló valamely végállomás-e.
 - **+BusStation getDestination1():** destination1 lekérdezése.
 - **+BusStation getDestination2():** destination1 lekérdezése.
 - **+void setDestinations(BusStation, BusStation):** Végállomások beállítása.
 - **+int getTimeToComplete():** timeToComplete lekérdezése.
 - **+void setTimeToComplete(int):** timeToComplete beállítása.
 - **+boolean isCompletedInTime():** Időben sikerült-e a busznak megtennie a távot.
 - **+void update():** Busz mozgását kezeli.

8.1.5 BusStation

- **Felelősség**

A *BusStation*-ök a buszsofőrök céljai. Ha elérték a kijelöltet, akkor pontot kapnak érte. Valamint *Intersection*-ként is funkcionál.

- **Össztályok**

Intersection

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+void interactWith(Vehicle vehicle):** A megálló ezen keresztül interaktál a járművekkkel, pl. : megáll ott.

8.1.6 Car

- **Felelősség**

A *Car* az NPC autósokat jelöli. Minden autónak van egy fix útvonala, ami mindig a legrövidebb útvonal a lakása és munkahelye között, majd fordítva.

- **Ősosztályok**

Vehicle

- **Interfészek**

ITickable (Vehicle-n keresztül)

- **Attribútumok**

- -NPCHomeStation start: Kezdeti “állomás.”
- -NPCHomeStation destination: “Célállomás.”
- -List<Intersection> route: A meghatározott út, amin keresztül a kocsi eléri a célját.
- -Intersection lastIntersection: Legutóbb meglátogatott kereszteződés.

- **Metódusok**

- +void setRoute(): Meghatározza, és beállítja a legrövidebb útvonalat.
- +bool hasReachedDestination(): Elérte-e az úticélját.
- +List<Intersection> getRoute(): Útvonal lekérdezése.
- +void changeDestination(): Kezdő és végállomás felcserélése.
- +void update(): Mozgás, ütközés stb. kezelése.

8.1.7 CannotBuyItemException

- **Felelősség**

Runtime hiba, amelyet a SnowPlowShop dob, amikor nem hókotró akar vásárolni.

- **Ősosztályok**

RuntimeException

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölt.

- **Metódusok**

Csak örökölt.

8.1.8 ClearState

- **Felelősség**

A tiszta út állapot osztálya.

- **Őosztályok**

RoadState

- **Interfészek**

Nincs

- **Attribútumok**

- **+static INSTANCE:** Állapotváltoztatásnál használt példány.

- **Metódusok**

- **+void transferTo(RoadSection from, RoadSection to):** Hó áthelyezése *from* sávról *to* sávra.

8.1.9 DragonHead

- **Felelősség**

A hókotró ennek segítségével az adott sávban képes megolvasztani a jeget és a havat, további elkostrásra nincs szükség. Csak akkor használható, ha a játékos rendelkezik *Biokerosene*-al.

- **Interfészek**

Nincs

- **Őosztályok**

SnowPlowHead → Item

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+void doCleaning(Map m, RoadSection rs, SnowPlow p):** Ezzel a metódussal tisztítja le az adott sávot a *SnowPlow*.
- **+bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal mondható meg, hogy az adott *SnowPlow* használhatja-e ezt a fejet.

8.1.10 Gravel

- **Felelősség**

GravelHead által használt zúzalék “üzemanyag.”

- **Őosztályok**

Item

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölték.

- **Metódusok**

- **+void onBuy(SnowPlow):** Vásárlás esetén növeli a hókotró üzemanyag mennyiségét. (Figyel a maximális tároló kapacitásra.)

8.1.11 GravelHead

- **Felelősség**

Gravelt használó hókotrófej. Használatával az adott sávban a jég és a hó zúzalékká válik, így nincs szükség további takarításra.

- **Interfészek**

Nincs

- **Őszosztályok**

SnowPlowHead → *Item*

- **Attribútumok**

Nincs

- **Metódusok**

- **+void doCleaning(Map m, RoadSection rs, SnowPlow p):** Ezzel a metódussal tisztítja le az adott sávot a *SnowPlow* zúzalékkal.
- **+bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et: van-e zúzaléka.

8.1.12 GravelledAgedState

- **Felelősség**

A havas-zúzalékos út állapot osztálya.

- **Őszosztályok**

RoadState

- **Interfészek**

Nincs

- **Attribútumok**

- **+static INSTANCE:** Állapotváltoztatásnál használt példány.

- **Metódusok**

- **+void transferTo(RoadSection from, RoadSection to):** Zúzalék áthelyezése *from* sávról *to* sávra.

8.1.13 GravelledFreshState

- **Felelősség**

A friss zúzalékos út állapot osztálya.

- **Őszosztályok**

RoadState

- **Interfészek**

Nincs

- **Attribútumok**

- **+static INSTANCE:** Állapotváltoztatásnál használt példány.

- **Metódusok**

- **+void transferTo(RoadSection from, RoadSection to):** Zúzalék áthelyezése *from* sávról *to* sávra.

8.1.14 IcebreakerHead

- **Felelősség**

Jégzúzó hókotrófej. Feltöri a jeget, de nem biztos hogy el is távolítja azt.

- **Interfészek**

- **Őosztályok**

SnowPlowHead → Item

- **Attribútumok**

Nincs

- **Metódusok**

- **+void doCleaning(Map m, RoadSection rs, SnowPlow p):** Ezzel a metódussal tisztítja le az adott sávot a *SnowPlow*.
- **+bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et: mindig igaz, csak nem biztos hogy tényleg történik valami.

8.1.15 IcyState

- **Felelősség**

A jeges út állapot osztálya.

- **Őosztályok**

RoadState

- **Interfészek**

Nincs

- **Attribútumok**

- **+static INSTANCE:** Állapotváltoztatásnál használt példány.

- **Metódusok**

- **+void transferTo(RoadSection from, RoadSection to):** Hó áthelyezése *from* sávról *to* sávra.

8.1.16 Item

- **Felelősség**

Egy absztrakt osztály, amiből a különböző tárgyak öröklődnek, és valósítják meg.

- **Asszociációk**

Nincs

- **Ősosztályok**

Nincs

- **Attribútumok**

- **#String name:** Tárgy neve.
- **#int cost:**

- **Metódusok**

- **+int getCost():** Ezzel a metódussal lekérdezhető a tárgy ára.
- **+String getName():** Ezzel a metódussal lekérdezhető a tárgy neve.
- **+void onBuy(SnowPlow):** A tárgy megvásárlásakor végrehajtandó művelet.

8.1.17 Intersection

- **Felelősség**

Az *Intersection* egy olyan elem ahol több *RoadSection* találkozik, így interakció végezhető el vele, azaz kiválaszthatjuk melyik úton haladnánk tovább.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **+List<RoadSection> roads:** Ebből a kereszteződésből kifutó útszakaszok (sávok).
- **+int intersectionID:** Egyedi azonosító teszteléshez.

- **Metódusok**

- **+List<RoadSection> getPossibleRoads():** Megadja az *Intersection*-ből elérhető lehetséges *RoadSection*-öket.
- **+void interactWith(Vehicle vehicle):** Ennek a függvénynek segítségével interaktálódik a paraméterben megadott *Vehicle* az adott *Intersection*-el.
- **+void addRoad(RoadSection):** Új sáv hozzárendelése.
- **+String toString():** Tesztelésnél kiíráshoz használt.

8.1.18 ITickable

- **Felelősség**

Ezt az interfészt azok az osztályok implementálják, amely körről-körre valamilyen módon frissítik a belső állapotukat.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

- **Metódusok**

- **+void update(TurnController tc):** Az implementáló osztályok állapotának frissítése.

8.1.19 Map

- **Felelősség**

A *Map* osztály felelős a játék pályájának kezeléséért és tárolásáért. A játék elején le kell generálnia a pályát, a játék során pedig folyamatosan frissítenie kell a változó részeit.

- **Ősosztályok**

Nincs

- **Interfészek**

ITickable

- **Attribútumok**

- **-List<RoadSection> lanes:** Pályán lévő sávok.
- **-List<Intersection> intersections:** Pályán lévő kereszteződések.
- **-List<BusStation> busStations:** Pályán lévő buszmegállók.
- **-List<NPCHomeStation> npcHomeStations:** Pályán lévő autó cél- és indulási pontok.

- **Metódusok**

- **+void update():** Tickenként futtatja a TurnController, frissíti a környezeti hatásokat.
- **+List<RoadSection> getLanes():** Pályán lévő sávok lekérdezése.
- **+List<Intersection> getIntersections():** Pályán lévő kereszteződések lekérdezése.
- **+void makeRoad(Intersection, Intersection, RoadSection>:** Két kereszteződés összekötése megadott sávval.
- **+void buildFromLayout(MapLayout):** Parse-olt .maplayout fájl alapján tölti fel a pályát.
- **+List<Intersection> calculateShortestRoute(Intersection, Intersection):** Legrövidebb út meghatározása két kereszteződés közt BFS-sel.

- **+List<BusStation> getTwoRandomBusStations():** Két véletlenszerű buszmegálló kiválasztása. Kritérium: legalább 6 sávnyi távolságra kell lenniük egymástól.
- **+List<NPCHomeStation> getTwoRandomNPCHomeStations():** Két véletlenszerű NPCHomeStation kiválasztása. Kritérium: legalább 6 sávnyi távolságra kell lenniük egymástól.

8.1.20 MapLayout

- **Felelősség**

MapLayoutReader által használt kimeneti “rekord,” vagyis egy ilyen struktúrába gyűjti ki a .maplayout fájlból olvasott adatokat. Ezt később a Map tudja felhasználni.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **-List<RoadSection> lanes:** Pályán lévő sávok.
- **-List<Intersection> intersections:** Pályán lévő kereszteződések.
- **-List<BusStation> busStations:** Pályán lévő buszmegállók.
- **-List<NPCHomeStation> npcHomeStations:** Pályán lévő autó cél- és indulási pontok.

- **Metódusok**

- **+List<RoadSection> getLanes:** Pályán lévő sávok lekérdezése.
- **+List<Intersection> getIntersections:** Pályán lévő kereszteződések lekérdezése.
- **+List<BusStation> getBusStations:** Pályán lévő buszmegállók lekérdezése.
- **+List<NPCHomeStation> getNpcHomeStations:** Pályán lévő autó cél- és indulási pontok lekérdezése.

8.1.21 MapLayoutParseException

- **Felelősség**

MapLayoutReader által dobott runtime hiba típus. Kiírja, hogy melyik sorban történt parse hiba, és egy addicionális üzenetet.

- **Ősosztályok**

RuntimeException

- **Interfészek**

Nincs

- **Attribútumok**

- **-int lineNumber:** A .maplayout ezen sorában van szintaxishiba.

- **Metódusok**
 - **+int getLineNumber():** Szintaxishiba sorának lekérdezése.

8.1.22 MapLayoutReader

- **Felelősség**
.maplayout fájlok feldolgozásáért felelős osztály.
- **Ősosztályok**
Nincs
- **Interfészek**
Nincs
- **Attribútumok**
 - **-MapLayout mapLayout:** Ezzel a MapLayout-tal fog végül visszatérni valamelyik *read* metódus. Azért adattag, hogy ne kelljen a metódusok közt paraméterként mindig átadni.
- **Metódusok**
 - **-int parseId(String token, int line):** Megpróbál feldolgozni egy ID típusú tokent. Ha nem sikerül, akkor MapLayoutParseException-t dob.
 - **-void parseIntersection(String[] token, int line):** Megpróbál feldolgozni egy “sort”, amely egy kereszteződés leírását tartalmazza. Végül módosítja a mapLayout adattagot. Ha nem sikerül, akkor MapLayoutParseException-t dob.
 - **-void parseLane(String[] token, int line):** Megpróbál feldolgozni egy “sort”, amely egy sáv leírását tartalmazza. Végül módosítja a mapLayout adattagot. Ha nem sikerül, akkor MapLayoutParseException-t dob.
 - **+MapLayout readFromFile(String filePath):** Adott .maplayout fájl feldolgozása.
 - **+MapLayout readFromString(String text):** Adott .maplayout szintaktikájú sztring feldolgozása.

8.1.23 NotEnoughPointsException

- **Felelősség**
Runtime hiba, amelyet a SnowPlowShop dob, ha játékosnak nincs elég pontja a vásárláshoz.
- **Ősosztályok**
RuntimeException
- **Interfészek**
Nincs
- **Attribútumok**
Csak örökölt.
- **Metódusok**
Csak örökölt.

8.1.24 NPCHomeStation

- **Felelősség**

Az *NPCHomeStation*-ök között járnak az autók.

- **Ősosztályok**

Intersection

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+void interactWith(Vehicle vehicle):** Ha ez egy autó végállomása, akkor megfordítja az autó menetirányát.

8.1.25 Player

- **Felelősség**

A *Player* osztály felelős a játékosok járművének tárolásáért, továbbá rajta keresztül kezelhető egy játékos pontja.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **-Vehicle vehicle:** A hozzárendelt jármű.
- **-String name:** A játékos neve.
- **-int points:** Az összegyűjtött pontok száma.

- **Metódusok**

- **+Vehicle getVehicle():** Visszaadja a játékos járművét.
- **+void setVehicle():** Beállítja a játékos járművét.
- **+int getPoints():** Visszaadja a játékos pontszámát.
- **+void spendPoints(int p):** Paraméterben kapott mennyiségű pontot von le a játékostól, de negatívba nem mehet.
- **+void gainPoints(int p):** Paraméterben kapott mennyiségű pontot ad a játékosnak.
- **+String getName():** Visszaadja a játékos nevét.

8.1.26 PlayerFactory

- **Felelősség**

A PlayerFactory a játékosok létrehozásáért felel, és hozzájuk rendeli a kezdő járművet.

- **Ósosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **-PlayerFactory instance:** Singleton példány.

- **Metódusok**

- **+Player buildBusPlayer(String name, BusStation dest1, BusStation dest2, int timeToComplete):** Létrehoz egy buszvezető játékost a megadott névvel, kezdő és végcéllal, illetve, hogy mennyi ideje van a teljesítésre. Visszaadja a létrejött Player példányt.
- **+Player buildSnowPlowPlayer(String name):** Létrehoz egy takarító (hókotró) játékost a megadott névvel és visszaadja a létrejött Player példányt.

8.1.27 Random

- **Felelősség**

Saját Random-wrapper, amely lehetővé teszi a java.util.Random determinisztikus működését (ki- és bekapcsolható a randomizálás).

- **Ósosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **-static java.util.Random javaRandom:** Az internal java.util.Random példány, amelyet a random generáláshoz használ fel.
- **-static boolean isDisabled:** Ki van-e kapcsolva a randomizálás.

- **Metódusok**

- **-static void disable():** Randomizáció kikapcsolása.
- **-static void enable():** Randomizáció bekapcsolása.
- **-static void seed():** Random-seed beállítása.
- **-static float nextFloat():** Visszaad egy véletlen nem-egész számot (0-1 között), ha be van kapcsolva a randomizálás, egyébként pedig mindig 1.0f-et ad vissza.
- **-static boolean nextBoolean():** Visszaad egy véletlen logikai értéket, ha be van kapcsolva a randomizálás, egyébként pedig mindig true-t ad vissza.

- **-static int nextInt(int upperBound):** Visszaad egy véletlen egész számot a megadott határértékig (exklúzívan), ha be van kapcsolva a randomizálás, egyébként pedig mindig 1-et ad vissza.

8.1.28 RoadSection

- **Felelősség**

Egy *RoadSection* két kereszteződés közötti bejárható sáv. Összeköttetést biztosít a kereszteződések között, és tárolja az összeköttetés állapotát.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **#RoadState state:** Az út jelenlegi állapota.
- **-List<Intersection> intersections:** A két végpontja (közbe fogó kereszteződések).
- **-List<Vehicle> vehicles:** A sávon tartózkodó járművek.
- **-List<RoadSection> parallelRoadSections:** A vele párhuzamos sávok.
- **-List<RoadSection> adjacentRoadSections:** A vele szomszédos sávok.
- **-int laneID:** Proto-programban használt ID.

- **Metódusok**

- **+RoadState getState():** Visszaadja az adott útszakasz állapotát.
- **+void addSection(Intersection):** Új végpont (kereszteződés) hozzáadása.
- **-void fillParallelAndAdjacent():** Párhuzamos és szomszédos sávok (újra) meghatározása.
- **+bool accept(RoadSection from, Vehicle vehicle):** Visszaadja logikai értéként hogy az attribútumként kapott *RoadSection* felől az attribútumként kapott *Vehicle* át tud-e menni erre a *RoadSection*-re.
- **+void updateSelf():** Tickenként hívódik meg, frissíti az útszakasz a saját állapotát (pl. havasabb lesz). (Statechart a *RoadState*-nél).
- **+RoadSection getRandomAdjacentRoadSection():** Véletlenszerű szomszédos sáv kiválasztása.
- **+RoadSection getRandomParallelRoadSection():** Véletlenszerű párhuzamos sáv kiválasztása.
- **+List<Vehicle> getVehicles():** A sávon tartózkodó járművek lekérdezése.
- **+List<Intersection> getPossibleIntersections():** Közbe fogó kereszteződések lekérdezése.
- **+String getLaneId():** Proto-programban használt ID lekérdezése.

8.1.29 RoadState

- **Felelősség**

A különböző állapot-osztályok absztrakt osztálya. *(Régen enum volt.)*

- **Őosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

Nincs

- **Metódusok**

- **+RoadState update(Map<RoadState, RoadState> map):** Frissíti az állapotot a megadott leképezés alapján. Ha a leképezés tartalmazza ezt az állapotot, a hozzárendelt értékkel tér vissza, különben önmagát (a jelenlegi állapotot) adja vissza.
- **+void transferTo(RoadSection from, RoadSection to):** Átviszi az állapot tartalmát az egyik útszakaszról a másikra.

8.1.30 Salt

- **Felelősség**

A *SalterHead* üzemanyaga. Nélküle a fej használhatatlan.

- **Interfészek**

Nincs

- **Őosztályok**

Item

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+void onBuy(SnowPlow):** Vásárláskor hozzáad az adott SnowPlow üzemanyagkészletéhez.

8.1.31 SalterHead

- **Felelősség**

Ha van *Salt* az adott játékosnál és ezt a fejet használja, akkor besózhatja az adott *RoadSection*-t.

- **Interfészek**

Nincs

- **Őosztályok**

SnowPlowHead → Item

- **Attribútumok**

Csak örökölt.

- **Metódusok**
 - **+void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow*.
 - **+bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

8.1.32 SaltyState

- **Felelősség**

A sós út állapot osztálya.
- **Ősosztályok**

RoadState
- **Interfészek**

Nincs
- **Attribútumok**
 - **+static INSTANCE:** Állapotváltoztatásnál használt példány.
- **Metódusok**
 - **+void transferTo(RoadSection from, RoadSection to):** Hó áthelyezése *from* sávról *to* sávra.

8.1.33 ShopAccessDeniedException

- **Felelősség**

Runtime hiba, amelyet a *SnowPlowShop* dob, ha olyan játékos akarja a boltot megnyitni, akinek nincsen ehhez engedélye.
- **Ősosztályok**

RuntimeException
- **Interfészek**

Nincs
- **Attribútumok**

Csak örökölt.
- **Metódusok**

Csak örökölt.

8.1.34 SnowPlow

- **Felelősség**

A *SnowPlow* a hókotró játékosok járműve. A hókotrókat fejleszteni lehet, és új fejeket lehet venni hozzájuk.

- **Ósosztályok**

Vehicle

- **Interfészek**

ITickable (*Vehicle*-n keresztül)

- **Attribútumok**

- **-SnowPlowUpgrade body:** A hókotró jelenlegi fejlesztése.
- **-SnowPlowHead head:** A hókotró jelenleg felszerelt feje.
- **-int keroseneCount:** A hókotró által tárolt biokerozin-mennyiség.
- **-int saltCount:** A hókotró által tárolt sómennyiség.
- **-int gravelCount:** A hókotró által tárolt zúzalékmennyiség.
- **+static int MAX_GRAVEL_COUNT:** A maximálisan tárolható zúzalékmennyiség (100).

- **Metódusok**

- **+void update():** Mozgást kezelő metódus.
- **+float getPointMultiplier():** Fejlesztés pontszorzójának lekérdezése.
- **+void setHead(SnowPlowHead newHead):** Fej beállítása.
- **+SnowPlowHead getHead():** Jelenleg használt fej lekérdezése.
- **+void setBody(SnowPlowUpgrade newUpgrade):** Fejlesztés beállítása.
- **+void consumeSalt():** Só felhasználása.
- **+void consumeFuel():** Biokerozin felhasználása.
- **+void consumeGravel():** Zúzalék felhasználása.
- **+void addSalt():** Só hozzáadása.
- **+void addFuel():** Biokerozin hozzáadása.
- **+void addGravel():** Zúzalék hozzáadása.
- **+int getFuelCount():** Biokerozin-mennyiség lekérdezése.
- **+int getSaltCount():** Só-mennyiség lekérdezése.
- **+int getGravelCount():** Zúzalék-mennyiség lekérdezése.
- **+bool canUseShop():** Használhatja-e a tulajdonos játékos a boltot (igen).
- **+bool canBuyItem(Item item):** Meg tudja-e venni a tulajdonos játékos az adott tárgyat.
- **+void buyItem(Item item):** Meghívja az adott tárgy onBuy metódusát *this* paraméterrel.

8.1.35 SnowPlowHead

- **Felelősség**

A *SnowPlow*-oknak ennek fajtáit lehet feltenni az eleire, majd ezek külön-külön hatást fejtenek ki az útra a helyzettől függően

- **Interfészek**

Nincs

- **Ősosztályok**

Item

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow*.
- **+bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.
- **+bool onBuy(SnowPlow s):** Vétel után felszereli magát az adott hókotróra.

8.1.36 SnowPlowShop

- **Felelősség**

Hókotrók játékosok által használható bolt. Ezen keresztül tudnak fejlesztéseket, fejeket, és üzemanyagot venni.

- **Interfészek**

Nincs

- **Ősosztályok**

Nincs

- **Attribútumok**

- **-static instance:** A singleton példány.
- **-List<Item> items:** Vásárolható tárgyak listája.

- **Metódusok**

- **+void openFor(Player p):** Megnyitja a boltot a megadott játékosnak (amennyiben engedélyezett).
- **+static SnowPlowShop getInstance():** Singleton példány lekérdezése.
- **-void openShop():** A konkrét “menünyitó” logika.
- **+void closeShop():** Bolt bezárása.
- **+List<Item> listItems():** Vásárolható tárgyak listájának lekérdezése.
- **+bool canPlayerBuyItem(Player p, Item i):** Megmondja hogy az adott *p Player* meg tudja-e venni a jelenlegi pontjaiból az adott *i Item*-et.
- **+void buyItemForPlayer(Player p, Item i) :** A *p Player* számára megveszi az adott *i Item*-et.
- **+void buyItemForPlayer(Player p, int idx) :** A *p Player* számára megveszi az adott indexszel rendelkező tárgyat.
- **+Item getItemByIndex(int idx):** Tárgy elérése index alapján.

8.1.37 SnowPlowUpgrade

- **Felelősség**

Ősosztály, amiből származnak a *SnowPlow*-hoz tartozó fejlesztések.

- **Interfészek**

Nincs

- **Ősosztályok**

Item

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+float getPointMultiplier():** Fejlesztés által nyújtott pontszorzó lekérdezése.
- **+void onBuy(SnowPlow s):** Vétel hatására beállítja az adott hókotrónak ezt a fejlesztést.

8.1.38 SnowPlowUpgradeNormal

- **Felelősség**

A normál verziója a *SnowPlow* fejlesztéseinek.

- **Interfészek**

Nincs

- **Ősosztályok**

SnowPlowUpgrade → Item

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+float getPointMultiplier():** Visszatér a normál pontszorzóval: 1 (a sima fejlesztés nem befolyásolja a pontgyűjtést).

8.1.39 SnowPlowUpgradePremium

- **Felelősség**

A prémium verziója a *SnowPlow* fejlesztéseinek.

- **Interfészek**

Nincs

- **Ősosztályok**

SnowPlowUpgrade → Item

- **Attribútumok**

Csak örökölt.

- **Metódusok**
 - **+float getPointMultiplier():** Visszatér a prémium pontszorzóval: 2 (duplázza a szerzett pontokat).

8.1.40 SnowyDenseState

- **Felelősség**

A nagyon havas út állapot osztálya.
- **Ősosztályok**

RoadState
- **Interfészek**

Nincs
- **Attribútumok**
 - **+static INSTANCE:** Állapotváltoztatásnál használt példány.
- **Metódusok**
 - **+void transferTo(RoadSection from, RoadSection to):** Hó áthelyezése *from* sávról *to* sávra.

8.1.41 SnowyLittleState

- **Felelősség**

A kicsit havas út állapot osztálya.
- **Ősosztályok**

RoadState
- **Interfészek**

Nincs
- **Attribútumok**
 - **+static INSTANCE:** Állapotváltoztatásnál használt példány.
- **Metódusok**
 - **+void transferTo(RoadSection from, RoadSection to):** Hó áthelyezése *from* sávról *to* sávra.

8.1.42 SnowyMediumState

- **Felelősség**

A közepesen havas út állapot osztálya.

- **Össztályok**

RoadState

- **Interfészek**

Nincs

- **Attribútumok**

- **+static INSTANCE:** Állapotváltoztatásnál használt példány.

- **Metódusok**

- **+void transferTo(RoadSection from, RoadSection to):** Hó áthelyezése *from* sávról *to* sávra.

8.1.43 SplittingHead

- **Felelősség**

Lényegében úgy funkcionál mint egy *SweeperHead* azonban ez képes átlökni a havat és a jeget egy párhuzamos sávba, vagy szomszédos útszakaszra.

- **Interfészek**

Nincs

- **Össztályok**

SnowPlowHead → Item

- **Attribútumok**

Nincs

- **Metódusok**

- **+void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow*.
- **+bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

8.1.44 SweeperHead

- **Felelősség**

Letakarítja a havat és a feltört jeget az útról.

- **Interfészek**

Nincs

- **Össztályok**

SnowPlowHead → Item

- **Attribútumok**

Nincs

- **Metódusok**
- **+void doCleaning(Map m, RoadSection rs):** Ezzel a metódussal tisztítja le az adott útszakaszt a *SnowPlow*.
- **+bool canBeUsedBy(SnowPlow sp):** Ezzel a metódussal adható meg hogy az adott *SnowPlow* használhatja-e ezt a *SnowPlowHead*-et.

8.1.45 Tunnel

- **Felelősség**

A *Tunnel* egy olyan sáv, amely a föld alatt haladva elkerüli a többi sávval való találkozást. Nem tud havassá válni.

- **Ősosztályok**

RoadSection

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölt.

- **Metódusok**

- **+bool accept(RoadSection from, Vehicle vehicle):** Visszaadja logikai értéként hogy a kapott *RoadSection* felől a kapott *Vehicle* át tud-e menni ezen a *Tunnel*-ön.
- **+void updateSelf():** Tickenként hívódik meg, frissíti az alagút saját állapotát. A hó nincs rá hatással. (Statechart a *RoadState*-nél).

8.1.46 TurnController

- **Felelősség**

Ez az osztály felel mindazon osztályok belső állapotának frissítéséért, amelyek implementálják az *ITickable* interfészt.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **-List<ITickable> tickables:** A frissítendő osztály példányok nyilvántartása.
- **-int currentTurn:** Jelenlegi kör száma.
- **-int maxTurn:** Ennyi körből áll a játék.

- **Metódusok**

- **+void tick():** Nyilvántartásba vett példányok frissítése.
- **+void addTickable():** Új példány nyilvántartásba vétele.

- **+int getCurrentTurn():** Visszaadja a jelenlegi kör (turn) számát.
- **+int getMaxTurn():** Visszaadja, hogy hány körből áll a játék.

8.1.47 Vehicle

- **Felelősség**

A *Vehicle* a járművek közös absztrakt őse. Ez az osztály felel a járművek ütközésének kezeléséért (legalábbis megadja a lehetőséget), illetve megteremti a járművek interfészének alapját.

- **Ősosztályok**

Nincs

- **Interfészek**

ITickable

- **Attribútumok**

- **#RoadSection currentRoadSection:** Jelenlegi sáv, amin a jármű tartózkodik.
- **#Intersection currentIntersection:** Jelenlegi kereszteződés, amin a jármű tartózkodik.
- **#int stunTimer:** Mozgásképtelen idő számlálója.
- **#int collisionMask:** Ütközéshez használt maszk.
- **#Player owner:** A játékos, akihez tartozik.
- **-bool isInIntersection():** Jelenleg kereszteződésben van-e.

- **Metódusok**

- **+bool isInIntersection():** Éppen egy kereszteződésben van-e.
- **+RoadSection getCurrentRoad():** Jelenleg melyik útszakaszon (azon belül sávon) tartózkodik - ez egy lépés (turn) közben releváns. Ha éppen nem lépés közben van, akkor null.
- **+void setCurrentRoad(RoadSection):** Beállítja, hogy éppen melyik sávon tartózkodjon a jármű.
- **+void setCurrentIntersection(Intersection):** Beállítja, hogy éppen melyik kereszteződésben tartózkodjon a jármű.
- **+Intersection getCurrentIntersection():** Jelenleg melyik kereszteződésben tartózkodik.
- **+bool canCollide(Vehicle v):** A *v* jármű tud-e ezzel a járművel összecsúszni.
- **+int getMask():** Ütközéshez használt maszk lekérdezése.
- **+Player getPlayer():** Tulajdonos játékos lekérdezése.
- **+bool isStunned():** Egy összecsúszás következtében tud-e mozogni, vagy sem.
- **+int getStunDuration():** Még hány körig nem mozoghat a legutolsó összecsúszás következtében.
- **+bool canUseShop():** Ezzel a járművel rendelkező játékos használhatja-e a boltot.
- **+bool canBuyItem(Item):** A tulajdonos játékos meg tudja-e venni az adott tárgyat.
- **+void buyItem(Item):** Adott tárgy megvétele a játékosnak.
- **+void stun():** Mozgásképtelenné teszi a járművet.
- **+void update():** Jármű mozgás-logikáját kezeli.

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

A tesztesetekenél használt fájlok:

```
t1.maplayout:
is 0 stop;
is 1;
is 2 stop;
lane 0 0 1;
lane 1 1 2;
```

8.2.1 Teszteset1 - Vehicle moves to ice.

- **Leírás**

Egy jármű egy jeges útra megy.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Jármű mozgásképtelenség logika

- **Bemenet**

```
loadmap t1.maplayout
setlanestate 1 icy
createplayer P1 bus
placevehicle P1 lane 0
move P1 intersection 1 1
stat player P1
```

- **Elvárt kimenet**

```
Vehicle{owner=P1, stunned=true, stunTimer=10, points=0, inIntersection=false,
currentRoadSection=lane 0, currentIntersection=null}
```

8.2.2 Teszteset2 - Vehicle moves to clear road section.

- **Leírás**

Egy jármű egy tiszta útra megy.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Jármű sikeresen át tud mozogni egy tiszta útra.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 bus
placevehicle P1 lane 0
move P1 intersection 1 1
stat player P1
```

- **Elvárt kimenet**

```
Vehicle{owner=P1, stunned=false, stunTimer=0, points=0, inIntersection=false,
currentRoadSection=lane 1, currentIntersection=null}
```

8.2.3 Teszteset3 - Vehicle moves to salted road section.

- **Leírás**

Egy jármű egy sózott útra megy.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Jármű sikeresen át tud mozogni egy szózott útra.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 bus
placevehicle P1 lane 0
setlanestate 1 salty
move P1 intersection 1 1
stat player P1
```

- **Elvárt kimenet**

```
Vehicle{owner=P1, stunned=false, stunTimer=0, points=0, inIntersection=false,
currentRoadSection=lane 1, currentIntersection=null}
```

8.2.4 Teszteset4 - Vehicle moves to gravelled road section.

- **Leírás**

Egy jármű egy zúzalékos útra megy.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Jármű sikeresen át tud mozogni egy zúzalékos útra.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 bus
placevehicle P1 lane 0
setlanestate 1 gravelled_fresh
move P1 intersection 1 1
stat player P1
```

- **Elvárt kimenet**

```
Vehicle{owner=P1, stunned=false, stunTimer=0, points=0, inIntersection=false,
currentRoadSection=lane 1, currentIntersection=null}
```

8.2.5 Teszteset5 - Vehicles moves to dense snowy road section.

- **Leírás**

Egy jármű egy nagyon havas útra próbál menni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A jármű megakad és nem mozog át.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 bus
placevehicle P1 lane 0
setlanestate 1 snowy_dense
move P1 intersection 1 1
stat player P1
```

- **Elvárt kimenet**

```
Vehicle{owner=P1, stunned=false, stunTimer=0, points=0, inIntersection=false,
currentRoadSection=lane 0, currentIntersection=null}
```

8.2.6 Teszteset6 - Vehicle moves to slightly snowy road section

- **Leírás**

Egy jármű egy enyhén havas útra megy.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A jármű letapossa a havat és az jéggé válik.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 bus
placevehicle P1 lane 0
setlanestate 1 snowy_medium
move P1 intersection 1 1
tick
stat player P1
stat lane 1
```

- **Elvárt kimenet**

```
Vehicle{owner=P1, stunned=false, stunTimer=0, points=0, inIntersection=false,
currentRoadSection=lane 1, currentIntersection=null}
Roadsection{state=ICY}
```

8.2.7 Teszteset7 - SnowPlow uses DragonHead on snow.

- **Leírás**

A hókotró a sárkány fejjel havat olvaszt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hó elolvad.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 dragonhead
addfuel P1 bio 10
placevehicle P1 lane 0
setlanestate 1 snowy_little
move P1 intersection 1 1
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection{state=CLEAR}
```

8.2.8 Teszteset8 - SnowPlow uses DragonHead on gravel.

- **Leírás**

A hókotró a sárkány fejjel zúzalékos utat olvaszt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A zúzalék megmarad.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 dragonhead
addfuel P1 bio 10
placevehicle P1 lane 0
setlanestate 1 gravelled_fresh
move P1 intersection 1 1
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection{state=GRAVELLED_FRESH}
```

8.2.9 Teszteset9 - SnowPlow uses SalterHead.

- **Leírás**

A hókotró a havas sávra sót rak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hó megszűnik és só marad a helyén.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 salterhead
addfuel P1 salt 10
placevehicle P1 lane 0
setlanestate 1 snowy_medium
move P1 intersection 1 1
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection{state=SALTY}
```

8.2.10 Teszteset10 - SnowPlow uses GravelHead on snow.

- **Leírás**

A hókotró a havas sávra zúzalékot rak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hó megszűnik és zúzalék marad a helyén.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 gravelhead
addfuel P1 gravel 10
placevehicle P1 lane 0
setlanestate 1 snowy_medium
move P1 intersection 1 1
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection{state=GRAVELLED_FRESH}
```

8.2.11 Teszteset11 - SnowPlow uses SweeperHead.

- **Leírás**

A hókotró egy havas/zúzalékos sáv tartalmát letakarítja.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hó/zúzalék megszűnik.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 sweeperhead
placevehicle P1 lane 0
setlanestate 1 snowy_medium
```

```
move P1 intersection 1 1
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection{state=CLEAR}
```

8.2.12 Teszteset12 - SnowPlow uses SplitterHead

- **Leírás**

A hókotró egy havas/zúzalékos sáv tartalmát át takarítja egy véletlenszerű, másik sávra.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hó/zúzalék megszűnik és áttolódik a másik sávra.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 splittinghead
placevehicle P1 lane 0
setlanestate 1 snowy_medium
move P1 intersection 1 1
stat lane 1
stat lane 0
```

- **Elvárt kimenet**

```
Roadsection{state=CLEAR}
Roadsection{state=SNOWY_MEDIUM}
```

8.2.13 Teszteset13 - SnowPlow uses Dragonhead with no fuel.

- **Leírás**

A hókotró megpróbálja a fejet használni, de nincs hozzá biokerozin.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hókotró nem takarítja le a sávot és rámegy, mint ha egy sima jármű lenne fej interakció nélkül.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 dragonhead
placevehicle P1 lane 0
setlanestate 1 snowy_little
move P1 intersection 1 1
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection{state=SNOWY_LITTLE}
```

8.2.14 Teszteset14 - SnowPlow uses SalterHead with no fuel.

- **Leírás**

A hókotró megpróbálja a fejet használni, de nincs hozzá só.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hókotró nem takarítja le a sávot és rámegy, mint ha egy sima jármű lenne fej interakció nélkül.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 salterhead
placevehicle P1 lane 0
setlanestate 1 snowy_little
move P1 intersection 1 1
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection{state=SNOWY_LITTLE}
```

8.2.15 Teszteset15 - SnowPlow uses GravelHead with no fuel.

- **Leírás**

A hókotró megpróbálja a fejet használni, de nincs hozzá zúzalék.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hókotró nem takarítja le a sávot és rámegy, mint ha egy sima jármű lenne fej interakció nélkül.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
sethead P1 gravelhead
placevehicle P1 lane 0
setlanestate 1 snowy_little
move P1 intersection 1 1
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection{state=SNOWY_LITTLE}
```

8.2.16 Teszteset16 - SnowPlow buys SnowPlowUpgradePremium.

- **Leírás**

A játékos egy prémium hókotrórt vásárol.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékosnak most már prémium a hókotrója.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
addmoney P1 4000
openshop P1
buy P1 9
stat item P1
```

- **Elvárt kimenet**

```
Items{hull=premium, head=splittinghead, salt=0, biokerosene=0, gravel=0}
```

8.2.17 Teszteset17 - SnowPlow can't buy SnowPlowUpgradePremium.

- **Leírás**

A játékos prémium hókotrórt szeretne vásárolni, de nem tud

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos hiába próbálkozott megvenni a prémium hókotrót, nincs elég pénze így marad a normálnál

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
addmoney P1 200
openshop P1
buy P1 9
stat item P1
```

- **Elvárt kimenet**

```
Items {hull=normal, head=splittinghead, salt=0, biokerosene=0, gravel=0}
```

8.2.18 Teszteset18 - SnowPlow can't buy SnowPlowHead.

- **Leírás**

A játékos hókotró fejet vásárolna a boltban, de nem tud.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos hiába próbál hókotró fejet vásárolni, nincs elég pénze, így nem tud.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
addmoney P1 200
openshop P1
buy P1 4
stat item P1
```

- **Elvárt kimenet**

```
Items {hull=normal, head=splittinghead, salt=0, biokerosene=0, gravel=0}
```

8.2.19 Teszteset19 - SnowPlow buys SnowPlowHead.

- **Leírás**

A játékos hókotró fejet vásárol a boltban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos megvásárolja a hókotró fejet, és meg is jelenik nála, hogy az új fej a jelenlegi amit használ

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
addmoney P1 1000
openshop P1
buy P1 4
stat item P1
```

- **Elvárt kimenet**

```
Items {hull=normal, head=sweeperhead, salt=0, biokerosene=0, gravel=0}
```

8.2.20 Teszteset20 - Vehicle collision.

- **Leírás**

Két jármű ütközik az úton.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ha a két jármű ütközhetett az úton, akkor mindkettő stun-olódik egy időre, azaz mozgásképtelenek lesznek.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
createpalyer P2 snowplow
move P1 intersection 1 1
move P2 intersection 1 1
stat player P1
stat player P2
```

- **Elvárt kimenet**

```
Vehicle{owner=P1, stunned=true, stunTimer=10, points=0, inIntersection=false,
currentRoadSection=lane 0, currentIntersection=null}
```

```
Vehicle{owner=P2, stunned=true, stunTimer=10, points=0, inIntersection=false,
currentRoadSection=lane 0, currentIntersection=null}
```

8.2.21 Teszteset21 - Bus reaches station.

- **Leírás**

A játékos a busszal beér a végállomásba, így pontot szerez és a kezdő állomása lesz most a végállomása.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos az elégedettségi szintjének megfelelő mennyiségű pontot kap, valamint az elégedettsége és változik a beérés függvényében, majd a végcél megváltozik a jelenlegi kezdőpontra.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 bus
stat bus P1
move P1 intersection 1 1
move P1 intersection 2 1
stat bus P1
```

- **Elvárt kimenet**

```
BusData{start=is 0, destination=is 3, satisfaction=0}
```

```
BusData{start=is 3, destination=is 0, satisfaction=10}
```

8.2.22 Teszteset22 - Car reaches destination.

- **Leírás**

A kocszi eléri a végcélját, majd megfordul és a kezdőpontja lesz az új végcél, és az új kezdőpont pedig a régi végcél.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A kocszi, kicserélte a két NPCHomeStation adattagját, így a végcélből lett a kezdőpont és fordítva, ezek után folytatja az útját a kiszámított legrövidebb úton.

- **Bemenet**

```
loadmap t1.maplayout
```

```

createcar C1
stat car C1
tick
tick
tick
stat car C1
  • Elvárt kimenet
CarData{start=is 0, destination=is 4}
CarData{start=is 4, destination=is 0}

```

8.2.23 Teszteset23 - SnowPlow buys Salt.

- **Leírás**

A játékos sót vesz a boltban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos megvásárolja a boltban a sót, van elég pénze rá, így be is kerül hozzá ez.

- **Bemenet**

```

loadmap t1.maplayout
createplayer P1 snowplow
addmoney P1 10
openshop P1
buy P1 1
stat item P1

```

- **Elvárt kimenet**

```
Items{hull=normal, head=splittinghead, salt=1, biokerosene=0, gravel=0}
```

8.2.24 Teszteset24 - SnowPlow buys Gravel.

- **Leírás**

A játékos sódert vesz a boltban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos megvásárolja a boltban a sódert, van elég pénze rá, így be is kerül hozzá ez.

- **Bemenet**

```

loadmap t1.maplayout
createplayer P1 snowplow
addmoney P1 10
openshop P1
buy P1 10
stat item P1

```

- **Elvárt kimenet**

```
Items{hull=normal, head=splittinghead, salt=0, biokerosene=0, gravel=1}
```

8.2.25 Teszteset25 - SnowPlow buys Biokerosene.

- **Leírás**

A játékos biokerozint vesz a boltban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos megvásárolja a boltban a biokerozint, van elég pénze rá, így be is kerül hozzá ez.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
addmoney P1 10
openshop P1
buy P1 2
stat item P1
```

- **Elvárt kimenet**

```
Items {hull=normal, head=splittinghead, salt=0, biokerosene=1, gravel=0}
```

8.2.26 Teszteset26 - SnowPlow can't buy Gravel.

- **Leírás**

A játékos nem tud sódert vásárolni a boltba.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos nem tudja megvenni a sódert, mivel nincs elég pénze, így nem kerül be hozzá az.

- **Bemenet**

```
loadmap t1.maplayout
createplayer P1 snowplow
addmoney P1 5
openshop P1
buy P1 10
stat item P1
```

- **Elvárt kimenet**

```
Items {hull=normal, head=splittinghead, salt=0, biokerosene=0, gravel=0}
```

8.2.27 Teszteset27 - Snowing.

- **Leírás**

Egy eddig tiszta út havas lesz a havazás miatt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A vizsgált tiszta útra hó kerül.

- **Bemenet**

```
loadmap t1.maplayout
tick
stat lane 1
```

- **Elvárt kimenet**

```
Roadsection {state=SNOWY_LITTLE}
```

8.3 A tesztelést támogató programok tervei

A tesztelő program alapvetően egy Java program, amely felhasználja a proto-programot.

A tesztelő program egy batch (Windows) és egy bash (Unix) szkript segítségével is fordítható, amelyek egy JAR fájlba csomagolják azt. Ezután pedig a java commandline eszközzel indítható. A program paraméterezése: <tesztek tartalmazó mappa> [opcionális teszteset].

Egy tesztesethez kettő fájl tartozik: <tesztnév>.testin, amely az inputparancsokat tartalmazza, és <tesztnév>.testout, amely pedig az elvárt végeredményt. Végeredmény alatt a parancsok által standard outra írt üzenetek összességét értjük (a tesztelő program képes átirányítani ezt a streamet, hogy aztán értelmezze).

Ha a második paraméter nincs megadva, akkor az összes tesztet lefuttatja a mappán belül, amelyhez tartozik egy megfelelő `.testin` és `.testout` pár. A tesztelő program elsősorban a `.testin` kiterjesztésű fájlokat keresi a mappában.

Egy `.testin` fájl tartalmára példa: (soronként egy parancs)

```
loadmap example.maplayout
createplayer P1 snowplow
stat all
```

A végén nem szükséges a `quit` parancs kiadása,.

Teszteset futtatását követve a program jelzi ezek sikerességét. Sikertelen esetben jelzi, hogy mi volt az elvárt kimenet. Szintaktika:

```
"[" <"HIBA" | "SIKER"> "]" <teszteset neve (.testin nélkül)>
```

Ha sikertelen, akkor az előző sort követik:

```
"Elvárt:" <.testout tartalma>
"Tényleges": <output, amelyet a .testin adott>
```

Ha minden teszt futtatása lett kérve, akkor plusz statisztikát is ír:

```
"---" <sikeres tesztek száma> "sikeres", <sikertelenek száma>
"sikertelen"
```

8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2026.04.16. 15:40:00	1	Heltai	Tevékenység: tesztelő felület lekódolása & tesztelést támogató program részletes terveinek megírása
2026.04.16. 18:00:00	0,5	Funk	Tevékenység: Tesztek leírásának elkezdése és leírónyelv implementálása.
2026.04.17. 10:30:00	2,5	Csordás	Tevékenység: Tesztesetek felének kidolgozása, kód javítása.
2026.04.17. 20:00:00	0,35	Heltai	Tevékenység: Kommentelés & Random-wrapper; osztályleírások elkezdése.
2026.04.18. 17:20:00	0,15	<i>Mindenki</i>	Értekezlet. Tárgy: mi van kész?
2026.04.18. 17:30:00	1	Heltai	Tevékenység: osztályleírások.
2026.04.18. 8:20:00	1,2	Funk	Tevékenység: State enum-ok refactor.
2026.04.19. 10:00:00	0,1	Rózsa	Tevékenység: osztályleírásokhoz sablon létrehozása
2026.04.19. 11:30:00	2	Heltai	Tevékenység: osztályleírások, véleményezés.
2026.04.19. 17:30:00	0,1	Funk	Tevékenység: Review
2026.04.19. 17:30:00	2	Fekete	Tevékenység: Tesztek másik felének megírása, state diagram átdolgozása
2026.04.19. 20:00:00	0,2	Heltai	Tevékenység: megjegyzések feldolgozása, utolsó simítások.

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

Megjegyzés: a “src” kezdetű elérési utak a ./src/main/java/javallerok utat rövidítik. Illetve a legtöbb keletkezési dátum a Git miatt “helytelen,” mivel minden pull-t követve megváltoztatta a keletkezések dátumát is.

10.1.1 Fájllista

Fájl neve	Méret (byte)	Keletkezés ideje	Tartalom
build.bat	1198	2026. 04. 06.	Build script Windows rendszerekhez a proto-program fordításához.
build.sh	882	2026. 04. 01.	Build script Unix rendszerekhez a proto-program fordításához.
build_tests.bat	1763	2026. 05. 01.	Windows batch szkript a tesztprogram fordításához.
build_tests.sh	1345	2026. 05. 01.	Unix shell szkript a tesztprogram fordításához.
tlayouts/t1.maplayout	51	2026. 04. 30.	Teszteknél használt egyik map layout.
tlayouts/t2.maplayout	49	2026. 05. 01.	Teszteknél használt egyik map layout.
tlayouts/test1.maplayout	102	2026. 04. 17.	Teszteknél használt egyik map layout.
tests/busReachesStation.testin	124	2026. 05. 01.	Teszteset input fájlja.
tests/busReachesStation.testout	137	2026. 05. 01.	Teszteset output fájlja.
tests/carReachesDestination.testin	79	2026. 05. 01.	Teszteset input fájlja.
tests/carReachesDestination.testout	372	2026. 05. 03.	Teszteset output fájlja.
tests/snowPlowBuysBiokerosene.testin	92	2026. 05. 01.	Teszteset input fájlja.
tests/snowPlowBuysBiokerosene.testout	101	2026. 05. 01.	Teszteset output fájlja.

tests/snowPlowBuys Gravel.testin	93	2026. 05. 01.	Teszteset input fájlja.
tests/snowPlowBuys Gravel.testout	101	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowBuys Salt.testin	92	2026. 05. 01.	Teszteset input fájlja.
tests/snowPlowBuys Salt.testout	101	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowCant BuyGravel.testin	92	2026. 05. 01.	Teszteset input fájlja.
tests/snowPlowCant BuyGravel.testout	95	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowUses DragonHeadOnGrav el.testin	185	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUses DragonHeadOnGrav el.testout	76	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowUses DragonHeadOnSnow .testin	182	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUses DragonHeadOnSnow .testout	66	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowUses DragonHeadWithNo Fuel.testin	164	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUses DragonHeadWithNo Fuel.testout	67	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowUses GravelHeadOnSnow. testin	185	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUses GravelHeadOnSnow. testout	76	2026. 05. 01.	Teszteset output fájlja.

tests/snowPlowUsesGravelHeadWithNoFuel.testin	164	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUsesGravelHeadWithNoFuel.testout	67	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowUsesSalterHead.testin	183	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUsesSalterHead.testout	66	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowUsesSalterHeadWithNoFuel.testin	164	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUsesSalterHeadWithNoFuel.testout	67	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowUsesSsplitterHead.testin	179	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUsesSsplitterHead.testout	85	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowUsesSweeperHead.testin	165	2026. 04. 30.	Teszteset input fájlja.
tests/snowPlowUsesSweeperHead.testout	60	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowbuysSnowPlowHead.testin	94	2026. 05. 01.	Teszteset input fájlja.
tests/snowPlowbuysSnowPlowHead.testout	99	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowbuysSnowPlowUpgradePremium.testin	93	2026. 05. 01.	Teszteset input fájlja.
tests/snowPlowbuysSnowPlowUpgradePremium.testout	103	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowcantbuySnowPlowHead.testin	92	2026. 05. 01.	Teszteset input fájlja.

tests/snowPlowcantb uySnowPlowHead.te stout	95	2026. 05. 01.	Teszteset output fájlja.
tests/snowPlowcantb uySnowPlowUpgrad ePremium.testin	93	2026. 05. 01.	Teszteset input fájlja.
tests/snowPlowcantb uySnowPlowUpgrad ePremium.testout	95	2026. 05. 01.	Teszteset output fájlja.
tests/snowing.testin	46	2026. 05. 01.	Teszteset input fájlja.
tests/snowing.testout	43	2026. 05. 01.	Teszteset output fájlja.
tests/vehicleCollision .testin	121	2026. 05. 03.	Teszteset input fájlja.
tests/vehicleCollision .testout	332	2026. 05. 03.	Teszteset output fájlja.
tests/vehicleMovesTo ClearRoadsection.test in	112	2026. 04. 30.	Teszteset input fájlja.
tests/vehicleMovesTo ClearRoadsection.test out	154	2026. 04. 30.	Teszteset output fájlja.
tests/vehicleMovesTo DenseSnowyRoadsec tion.testin	139	2026. 04. 30.	Teszteset input fájlja.
tests/vehicleMovesTo DenseSnowyRoadsec tion.testout	160	2026. 04. 30.	Teszteset output fájlja.
tests/vehicleMovesTo GravelledRoadsectio n.testin	143	2026. 04. 30.	Teszteset input fájlja.
tests/vehicleMovesTo GravelledRoadsectio n.testout	160	2026. 04. 30.	Teszteset output fájlja.
tests/vehicleMovesTo Ice.testin	131	2026. 04. 30.	Teszteset input fájlja.
tests/vehicleMovesTo Ice.testout	160	2026. 04. 30.	Teszteset output fájlja.

tests/vehicleMovesToSaltedRoadsection.testin	133	2026. 04. 30.	Teszteset input fájlja.
tests/vehicleMovesToSaltedRoadsection.testout	160	2026. 04. 30.	Teszteset output fájlja.
tests/vehicleMovesToSlightlySnowyRoadsection.testin	157	2026. 05. 01.	Teszteset input fájlja.
tests/vehicleMovesToSlightlySnowyRoadsection.testout	189	2026. 05. 01.	Teszteset output fájlja.
src/Main.java	425	2026. 04. 25.	A (rég) program entry pointja.
src/Utils/Random.java	1389	2026. 05. 01.	A Random osztály.
src/items/Biokerosene.java	845	2026. 04. 09.	A Biokerosene osztály.
src/items/DragonHead.java	2178	2026. 04. 19.	A DragonHead osztály.
src/items/Gravel.java	836	2026. 04. 09.	A Gravel osztály.
src/items/GravelHead.java	2147	2026. 04. 19.	A GravelHead osztály.
src/items/IceBreakerHead.java	1572	2026. 04. 19.	A IceBreakerHead osztály.
src/items/Item.java	1205	2026. 04. 09.	A Item osztály.
src/items/Salt.java	740	2026. 04. 30.	A Salt osztály.
src/items/SalterHead.java	2075	2026. 04. 19.	A SalterHead osztály.
src/items/SnowPlowHead.java	1379	2026. 04. 19.	A SnowPlowHead osztály.
src/items/SnowPlowUpgrade.java	1128	2026. 04. 09.	A SnowPlowUpgrade osztály.
src/items/SnowPlowUpgradeNormal.java	830	2026. 04. 09.	A SnowPlowUpgradeNormal osztály.

src/items/SnowPlowUpgradePremium.java	827	2026. 04. 01.	A SnowPlowUpgradePremium osztály.
src/items/SplittingHead.java	1660	2026. 04. 19.	A SplittingHead osztály.
src/items/SweeperHead.java	1939	2026. 04. 19.	A SweeperHead osztály.
src/shop/CannotBuyItemException.java	96	2026. 04. 07.	A CannotBuyItemException osztály.
src/shop/NotEnoughPointsException.java	98	2026. 04. 07.	A NotEnoughPointsException osztály.
src/shop/ShopAccessDeniedException.java	99	2026. 04. 07.	A ShopAccessDeniedException osztály.
src/shop/SnowPlowShop.java	3980	2026. 05. 01.	A SnowPlowShop osztály.
src/vehicles/Bus.java	4934	2026. 05. 01.	A Bus osztály.
src/vehicles/Car.java	5654	2026. 05. 03.	A Car osztály.
src/vehicles/SnowPlow.java	4229	2026. 05. 03.	A SnowPlow osztály.
src/vehicles/Vehicle.java	7986	2026. 05. 03.	A Vehicle osztály.
src/szkeleton/scripts/Script.java	1567	2026. 04. 07.	A Script osztály.
src/szkeleton/scripts/ScriptManager.java	21925	2026. 04. 07.	A ScriptManager osztály.
src/proto/Proto.java	5869	2026. 05. 01.	A proto-program entry pointja.
src/proto/commands/AddFuelCommand.java	1649	2026. 04. 30.	A AddFuelCommand osztály.
src/proto/commands/AddMoneyCommand.java	1237	2026. 05. 01.	A AddMoneyCommand osztály.
src/proto/commands/BuyCommand.java	1628	2026. 04. 29.	A BuyCommand osztály.

src/proto/commands/ CleanCommand.java	1512	2026. 04. 30.	A CleanCommand osztály.
src/proto/commands/ CloseShopCommand .java	502	2026. 04. 30.	A CloseShopCommand osztály.
src/proto/commands/ Command.java	847	2026. 04. 10.	A Command osztály.
src/proto/commands/ CommandManager.ja va	2506	2026. 04. 29.	A CommandManager osztály.
src/proto/commands/ CreatePlayerComma nd.java	3042	2026. 04. 30.	A CreatePlayerCommand osztály.
src/proto/commands/ GetTurnCommand.ja va	542	2026. 04. 29.	A GetTurnCommand osztály.
src/proto/commands/ LeaderboardComman d.java	528	2026. 04. 29.	A LeaderboardCommand osztály.
src/proto/commands/ ListItemsCommand.j ava	702	2026. 04. 29.	A ListItemsCommand osztály.
src/proto/commands/ LoadMapCommand.j ava	973	2026. 05. 01.	A LoadMapCommand osztály.
src/proto/commands/ MoveCommand.java	3394	2026. 05. 01.	A MoveCommand osztály.
src/proto/commands/ OpenShopCommand. java	993	2026. 04. 29.	A OpenShopCommand osztály.
src/proto/commands/ PlaceVehicleComma nd.java	3004	2026. 04. 29.	A PlaceVehicleCommand osztály.
src/proto/commands/ SetBusRouteComma nd.java	2284	2026. 04. 30.	A SetBusRouteCommand osztály.
src/proto/commands/ SetHeadCommand.ja va	2329	2026. 04. 30.	A SetHeadCommand osztály.

src/proto/commands/SetLaneStateCommand.java	2633	2026. 04. 30.	A SetLaneStateCommand osztály.
src/proto/commands/SetUpgradeCommand.java	1717	2026. 04. 30.	A SetUpgradeCommand osztály.
src/proto/commands/SpawnCarCommand.java	2121	2026. 05. 03.	A SpawnCarCommand osztály.
src/proto/commands/StatCommand.java	4066	2026. 05. 03.	A StatCommand osztály.
src/proto/commands/TickCommand.java	441	2026. 04. 29.	A TickCommand osztály.
src/config/MapLayout.java	1169	2026. 04. 07.	A MapLayout osztály.
src/config/MapLayoutParseException.java	365	2026. 04. 19.	A MapLayoutParseException osztály.
src/config/MapLayoutReader.java	7032	2026. 05. 01.	A MapLayoutReader osztály.
src/logging/Loggable.java	392	2026. 04. 01.	A Loggable interfész.
src/logging/SL.java	7006	2026. 04. 07.	A SL osztály.
src/gamemap/Bridge.java	815	2026. 04. 30.	A Bridge osztály.
src/gamemap/BusStation.java	1413	2026. 05. 01.	A BusStation osztály.
src/gamemap/Intersection.java	1826	2026. 05. 01.	A Intersection osztály.
src/gamemap/Map.java	6719	2026. 05. 01.	A Map osztály.
src/gamemap/NPCHomeStation.java	881	2026. 05. 01.	A NPCHomeStation osztály.
src/gamemap/RoadSection.java	11311	2026. 05. 03.	A RoadSection osztály.
src/gamemap/Tunnel.java	1025	2026. 04. 17.	A Tunnel osztály.

src/gamemap/roadstates/ClearState.java	476	2026. 04. 19.	A ClearState osztály.
src/gamemap/roadstates/GravelledAgedState.java	530	2026. 04. 19.	A GravelledAgedState osztály.
src/gamemap/roadstates/GravelledFreshState.java	535	2026. 04. 19.	A GravelledFreshState osztály.
src/gamemap/roadstates/IcyState.java	465	2026. 04. 19.	A IcyState osztály.
src/gamemap/roadstates/RoadState.java	1675	2026. 04. 28.	A RoadState osztály.
src/gamemap/roadstates/SaltyState.java	474	2026. 04. 19.	A SaltyState osztály.
src/gamemap/roadstates/SnowyDenseState.java	506	2026. 04. 19.	A SnowyDenseState osztály.
src/gamemap/roadstates/SnowyLittleState.java	520	2026. 04. 19.	A SnowyLittleState osztály.
src/gamemap/roadstates/SnowyMediumState.java	511	2026. 04. 19.	A SnowyMediumState osztály.
src/core/App.java	5863	2026. 05. 01.	A App osztály.
src/core/ITickable.java	303	2026. 04. 01.	A ITickable interfész.
src/core/Player.java	1697	2026. 04. 09.	A Player osztály.
src/core/PlayerFactory.java	2017	2026. 04. 17.	A PlayerFactory osztály.
src/core/TurnController.java	1585	2026. 05. 01.	A TurnController osztály.
src/test/java/Tester.java	4903	2026. 05. 03.	A tesztek futtatásáért felelős program entry point-ja.
src/test/java/Tests.java	948	2026. 04. 17.	A tesztek futtatásáért felelős osztály.

10.1.2 Fordítás

A tesztelő program alapvetően egy Java program, amely felhasználja a proto-programot. A tesztelő program egy batch (Windows) és egy bash (Unix) szkript segítségével is fordítható, amelyek egy JAR fájlba csomagolják azt.

A fordításhoz szükséges, hogy elérhető legyen egy Java 11+ JDK az eszközön, illetve hozzáférés a parancssorhoz. A fordító/csomagoló szkriptek a projektgyökérben találhatóak, és onnan kell őket futtatni.

Windows: `build_tests.bat`

Linux: `./build_tests.sh`

Az output JAR fájl a `<projektgyökér>/out-test` mappában lesz elérhető.

10.1.3 Futtatás

A tesztelő program futtatásához szükséges, hogy elérhető legyen egy Java 11+ JRE az eszközön. Mindkét rendszeren a projektgyökérből futtatva:

```
java -jar out-test/projlab-tests.jar <tesztek tartalmazó mapp> [opcionális teszteset]
```

Példa egy darab teszteset futtatására a projekt gyökér könyvtárból:

```
java -jar ./out-test/projlab-tests.jar tests snowPlowcantbuySnowPlowUpgradePremium
```

Egy, a tesztek tartalmazó mappa a projektgyökérben elérhető `tests/` néven. Opcionálisan megadható egy konkrét teszteset neve is. Ha az utóbbi paraméter nincs megadva, akkor a mappában található összes tesztesetet futtatja.

10.2 Tesztek jegyzőkönyvei

10.2.1 Teszteset1 - vehicleMovesTolce

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 03. 16:20

10.2.2 Teszteset2 - vehicleMovesToClearRoadsection

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 03. 16:21

10.2.3 Teszteset3 - vehicleMovesToSaltedRoadsection

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 03. 16:22

10.2.4 Teszteset4 - vehicleMovesToGravelledRoadsection

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 03. 16:23

10.2.5 Teszteset5 - vehicleMovesToDenseSnowyRoadsection

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 03. 16:24

10.2.6 Teszteset6 - vehicleMovesToSlightlySnowyRoadsection

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 03. 16:25

10.2.7 Teszteset7 - snowPlowUsesDragonHeadOnSnow

Tesztelő neve	[REDACTED]
Teszt időpontja	2026.05.03.18:23

10.2.8 Teszteset8 - snowPlowUsesDragonHeadOnGravel

Tesztelő neve	[REDACTED]
Teszt időpontja	2026.05.03. 18:24

10.2.9 Teszteset9 - snowPlowUsesSalterHead

Tesztelő neve	[REDACTED]
Teszt időpontja	2026.05.03. 18:25

10.2.10 Teszteset10 - snowPlowUsesGravelHeadOnSnow

Tesztelő neve	[REDACTED]
Teszt időpontja	2026.05.03 18:27

10.2.11 Teszteset11 - snowPlowUsesSweeperHead

Tesztelő neve	[REDACTED]
Teszt időpontja	2026.05.03. 18:28

10.2.12 Teszteset12 - snowPlowUsesSplitterHead

Tesztelő neve	[REDACTED]
Teszt időpontja	2026.05.03. 18:31

10.2.13 Teszteset13 - nowPlowUsesDragonHeadWithNoFuel

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 02. 11:20

10.2.14 Teszteset14 - snowPlowUsesSalterHeadWithNoFuel

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 02. 11:22

10.2.15 Teszteset15 - snowPlowUsesGravelHeadWithNoFuel

Tesztelő neve	
Teszt időpontja	2026. 05. 02. 11:23

10.2.16 Teszteset16 - snowPlowbuysSnowPlowUpgradePremium

Tesztelő neve	
Teszt időpontja	2026. 05. 02. 11:24

10.2.17 Teszteset17 - snowPlowcantbuySnowPlowUpgradePremium

Tesztelő neve	
Teszt időpontja	2026. 05. 02. 11:26
Teszt eredménye	[HIBA] snowPlowcantbuySnowPlowUpgradePremium A teszt az alábbi hibákat generálta: Nincs elég pont a vásárláshoz.
Lehetséges hibaok	A teszt hibát vár, viszont a tesztkörnyezet akármilyen exception-t hibás tesztnek minősít.
Változtatások	Nem veszi többet figyelembe ezeket a hibaüzeneteket.

Tesztelő neve	
Teszt időpontja	2026. 05. 03. 16:45

10.2.18 Teszteset18 - snowPlowcantbuySnowPlowHead

Tesztelő neve	
Teszt időpontja	2026. 05. 02. 12:33
Teszt eredménye	[HIBA] snowPlowcantbuySnowPlowHead A teszt az alábbi hibákat generálta: Nincs elég pont a vásárláshoz.
Lehetséges hibaok	A teszt hibát vár, viszont a tesztkörnyezet akármilyen exception-t hibás tesztnek minősít.
Változtatások	Nem veszi többet figyelembe ezeket a hibaüzeneteket.

Tesztelő neve	
Teszt időpontja	2026. 05. 03. 16:52

10.2.19 Teszteset19 - snowPlowbuysSnowPlowHead

Tesztelő neve	
Teszt időpontja	2026. 05. 02. 12:34

10.2.20 Teszteset20 - vehicleCollision

Tesztelő neve	
Teszt időpontja	2026. 05. 02. 12:36

Teszt eredménye	[HIBA] vehicleCollision A teszt az alábbi hibákat generálta: Hibás parancs hívás. No such player: P2 No such Player: P2 Hibás parancs hívás.
Lehetséges hibaok	A "createplayer" parancs el van írva: "createpalyer".
Változtatások	Mostmár egy autó és egy játékos által irányított jármű ütközik a leírásnak megfelelően

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 03. 19:50

10.2.21 Teszteset21 - busReachesStation

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 02. 12:38

10.2.22 Teszteset22 - carReachesDestination

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 02. 12:40

10.2.23 Teszteset23 - snowPlowBuysSalt

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 02. 23:00

10.2.24 Teszteset24 - snowPlowBuysGravel

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 02. 23:02

10.2.25 Teszteset25 - snowPlowBuysBiokerosene

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 02. 23:03

10.2.26 Teszteset26 - snowPlowCantBuyGravel

Tesztelő neve	[REDACTED]
Teszt időpontja	2026. 05. 02. 23:04
Teszt eredménye	[HIBA] snowPlowCantBuyGravel

	A teszt az alábbi hibákat generálta: Nincs elég pont a vásárláshoz.
Lehetséges hibaok	A teszt hibát vár, viszont a tesztkörnyezet akármilyen exception-t hibás tesztnek minősít.
Változtatások	Nem veszi többet figyelembe ezeket a hibaüzeneteket.

Tesztelő neve	██████████
Teszt időpontja	2026. 05. 03. 16:53

10.2.27 Teszteset27 - snowing

Tesztelő neve	██████████
Teszt időpontja	2026. 05. 02. 23:06

10.3Értékelés

Tag neve	Tag neptun	Munka százalékban
Rózsa	██████████	19
Csordás	██████████	20
Fekete	██████████	19
Heltai	██████████	21
Funk	██████████	21

10.4 Napló

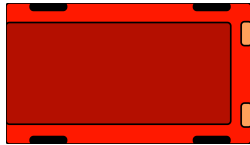
Kezdet	Időtartam	Résztevők	Leírás
2026.04.21. 11:00:00	0,35	Heltai	Tevékenység: fordítási és futtatási útmutató + tesztelő környezet egy hibájának javítása + parancssablonok
2026.04.28. 15:30:00	0,7	Funk	Tevékenység: takarítás state pattern-é alakítása. Architektúra átgondolások ahhoz kapcsolódóan.
2026.04.29. 9:00:00	1,2	Rózsa	Tevékenység: parancsok megírása + ütközéskezelés folytatása + eddigi kódváltoztatások értelmezése
2026.04.29. 15:20:00	0,5	Heltai	Tevékenység: parancsok implementálása
2026.04.29. 16:20:00	0,3	Funk	Tevékenység: code review, viselkedés kiegészítése, ahol hiányosságok voltak.
2026.07.17. 19:00:00	1,1	Rózsa	Tevékenység: parancsok
2026.04.30. 11:00:00	2	Csordás	Tevékenység: tesztesetek implementálása
2026.04.29. 15:30:00	1,25	Heltai	Tevékenység: parancsok implementálása + kisebb fixek
2026.04.29. 19:00:00	3	Csordás	Tevékenység: parancsok, tesztek, viselkedés javítása
2026.05.02. 10:00:00	0,4	<i>Mindenki</i>	Értekezlet. Döntés: ki melyik jegyzőkönyvet csinálja.
2026.05.02. 11:00:00	0,65	Funk	Tevékenység: Tesztjegyzőkönyv

			elkezdés, dokumentáció kiegészítése.
2026.05.02. 12:30:00	0,35	Heltai	Tevékenység: tesztelés + egyéb fixek.
2026.05.03. 18:32:00	0,1	Rózsa	Tevékenység: tesztelés.
2026.05.03. 18:40:00	3	Fekete	Tevékenység: Tesztek implementálása, és egyéb változtatások hogy minden működjön

11. Grafikus felület specifikációja

11.1 A grafikus interfész

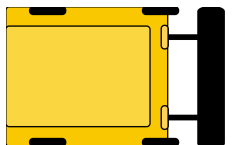
A pálya ezeket az alakzatokat/képeket fogja tartalmazni:



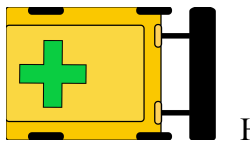
Autó sprite



Busz sprite



Hókotró nem fejlesztett sprite



Hókotró fejlesztett sprite



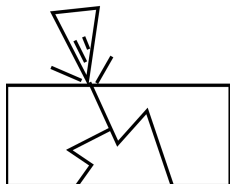
Játék borító ikon



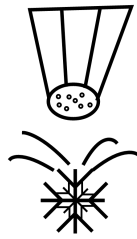
Sárkány fej ikon



Zúzálékos fej ikon



Jégtörő fej ikon



Sózó fej ikon



Hányó fej ikon



Söprő fej ikon

A következő elemek procedurálisan lesznek létrehozva. Az utak egyenesek lesznek, melyeknek a belső színe fogja megmondani, hogy milyen típusú az út. A border színük jelöli a havas vagy más állapotokat. Példa egy havas útra:



Havas sima út



Havas alagút (Az alagút nem lehet havas, de a havas határ jobban illusztrálja az alagút színét ebben a példában)



Havas híd



NPC állomás/kezdőpont

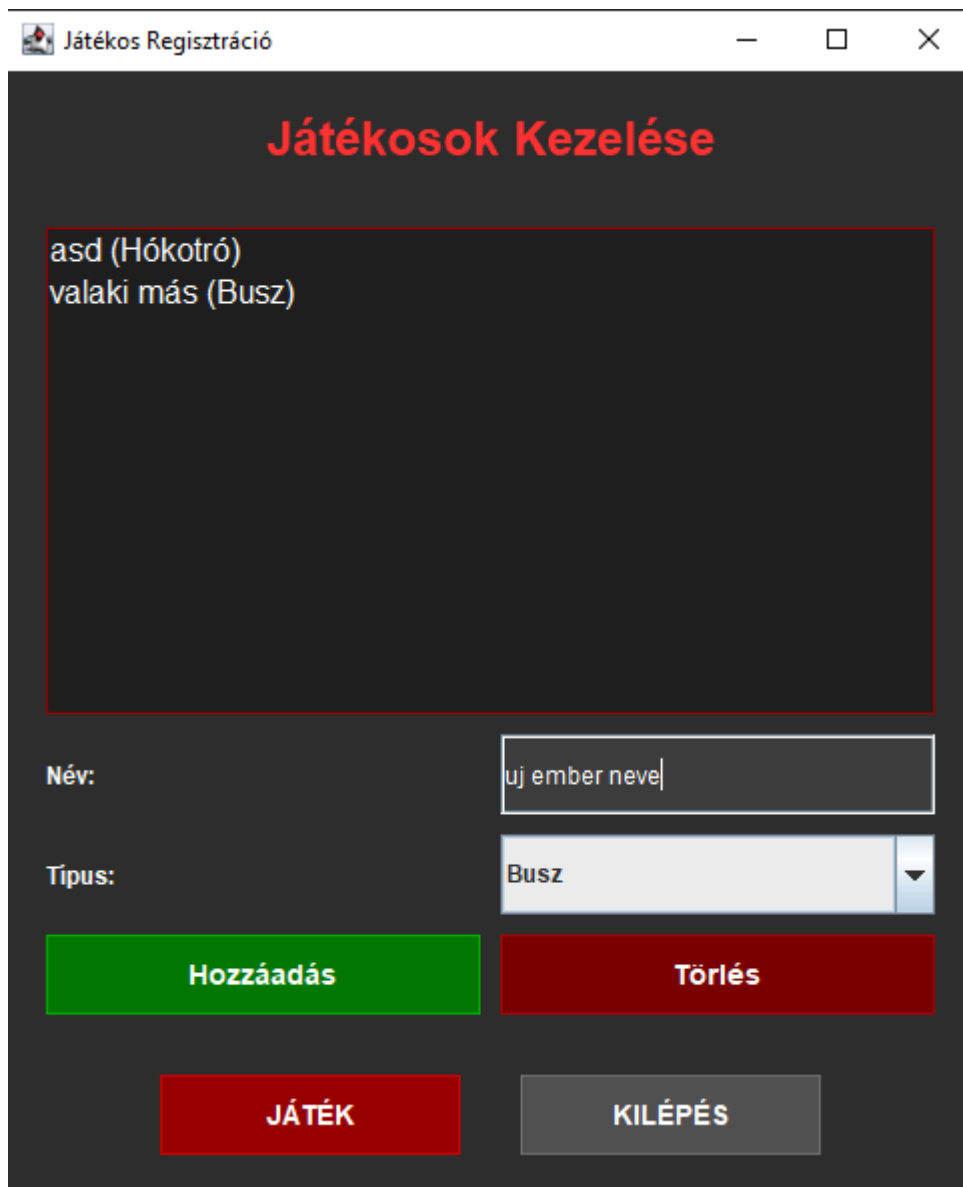


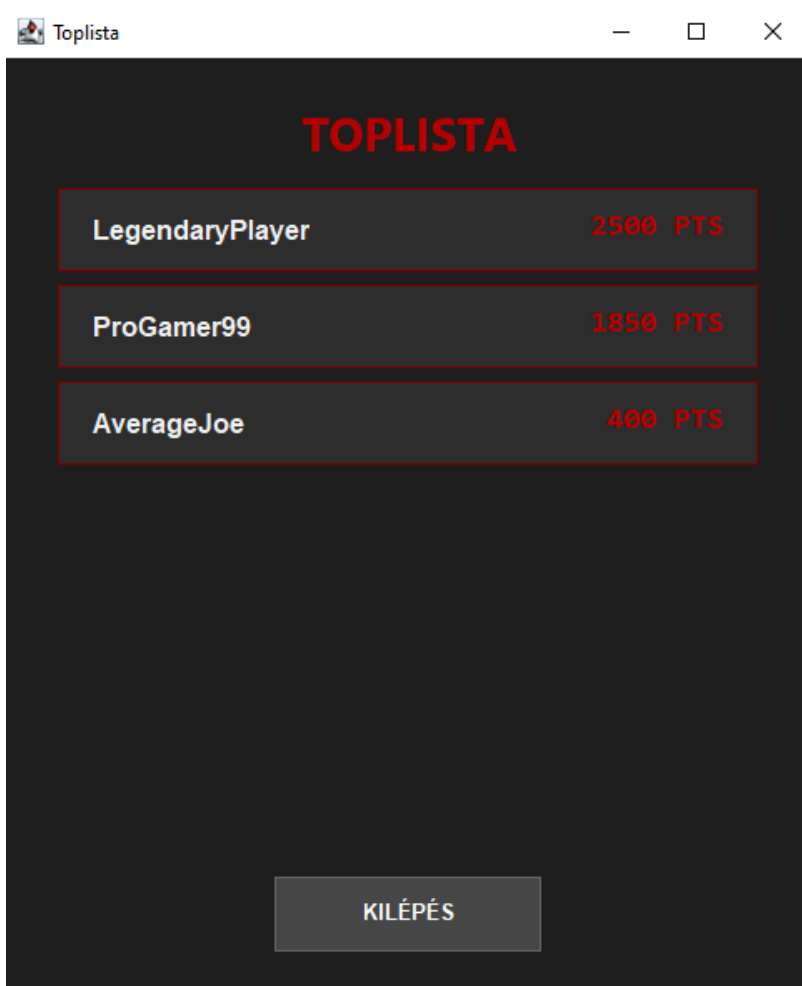
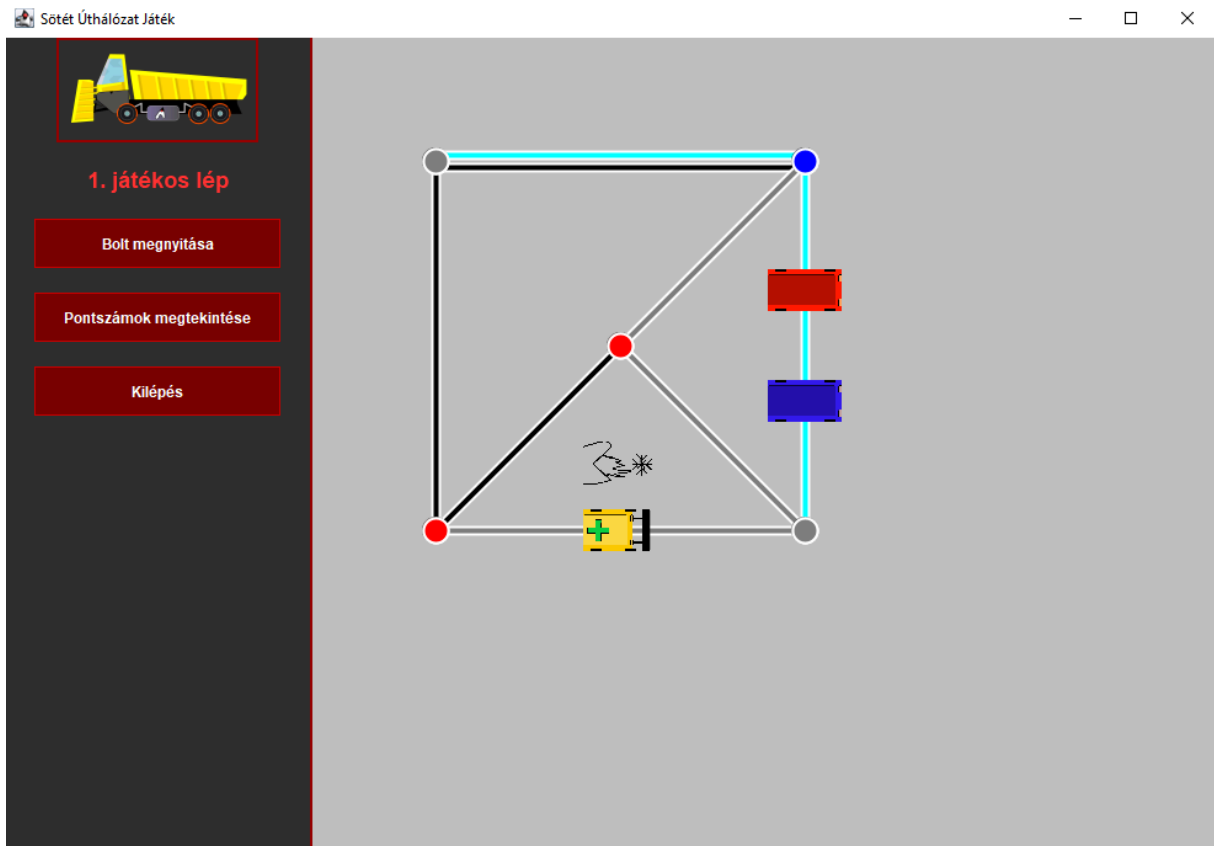
Sima kereszteződés



Buszmegálló

Példa illusztráció a menü részekre:





11.2 A grafikus rendszer architektúrája

A felület tervezése során az MVC tervezési mintát vettük alapul.

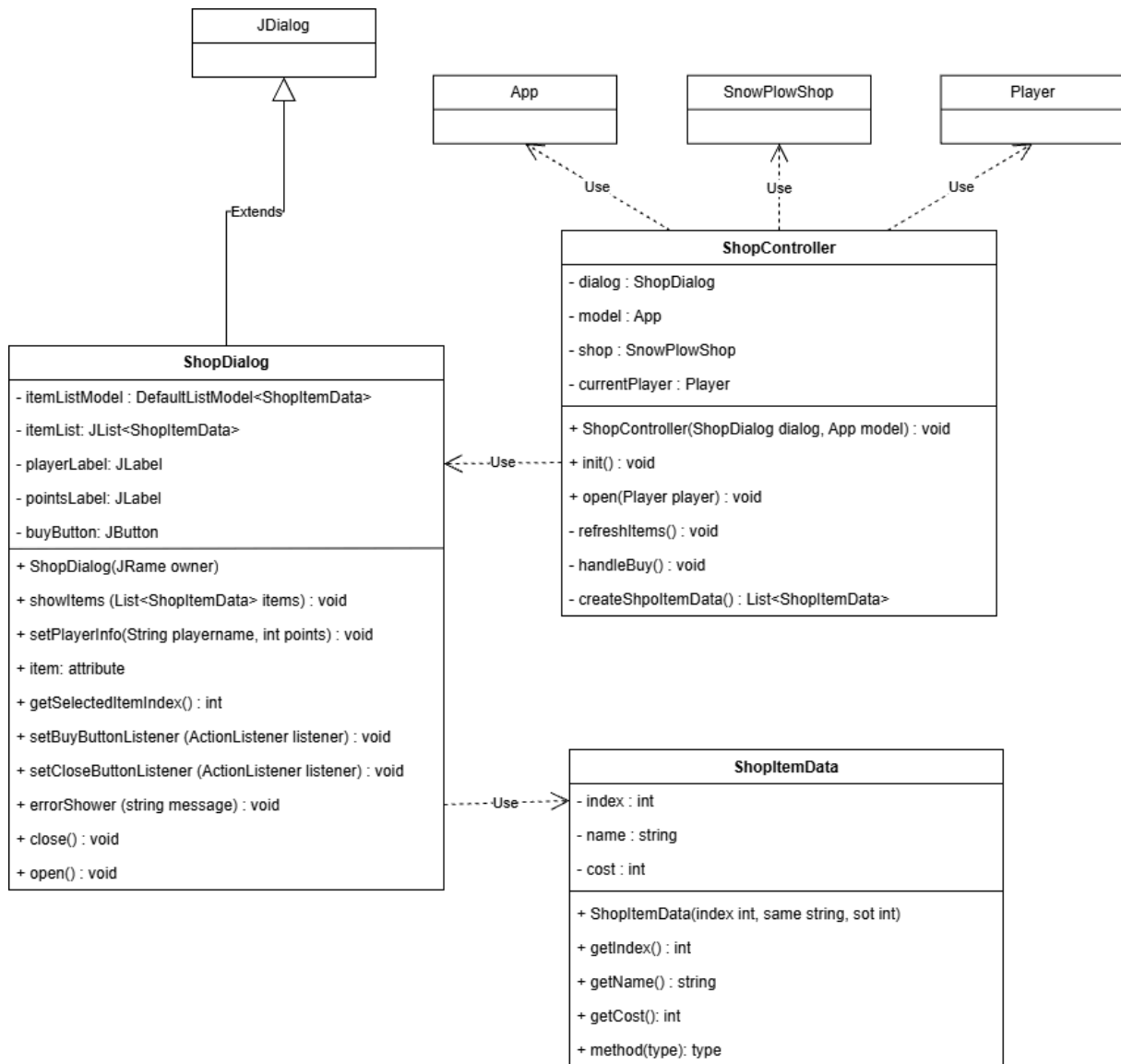
Model: Az eddig elkészített proto-program és hozzátartozó biznissz logika.

View: Több *View* osztály testesíti meg. Minden ilyen osztály egy adott nézethez tartozik: játékos felvétel, a tényleges játék, leaderboard stb. A nézetosztályok úgynevezett *Visual* osztályokat használnak fel a megjelenítendő grafika (képek) megszerzéséhez.

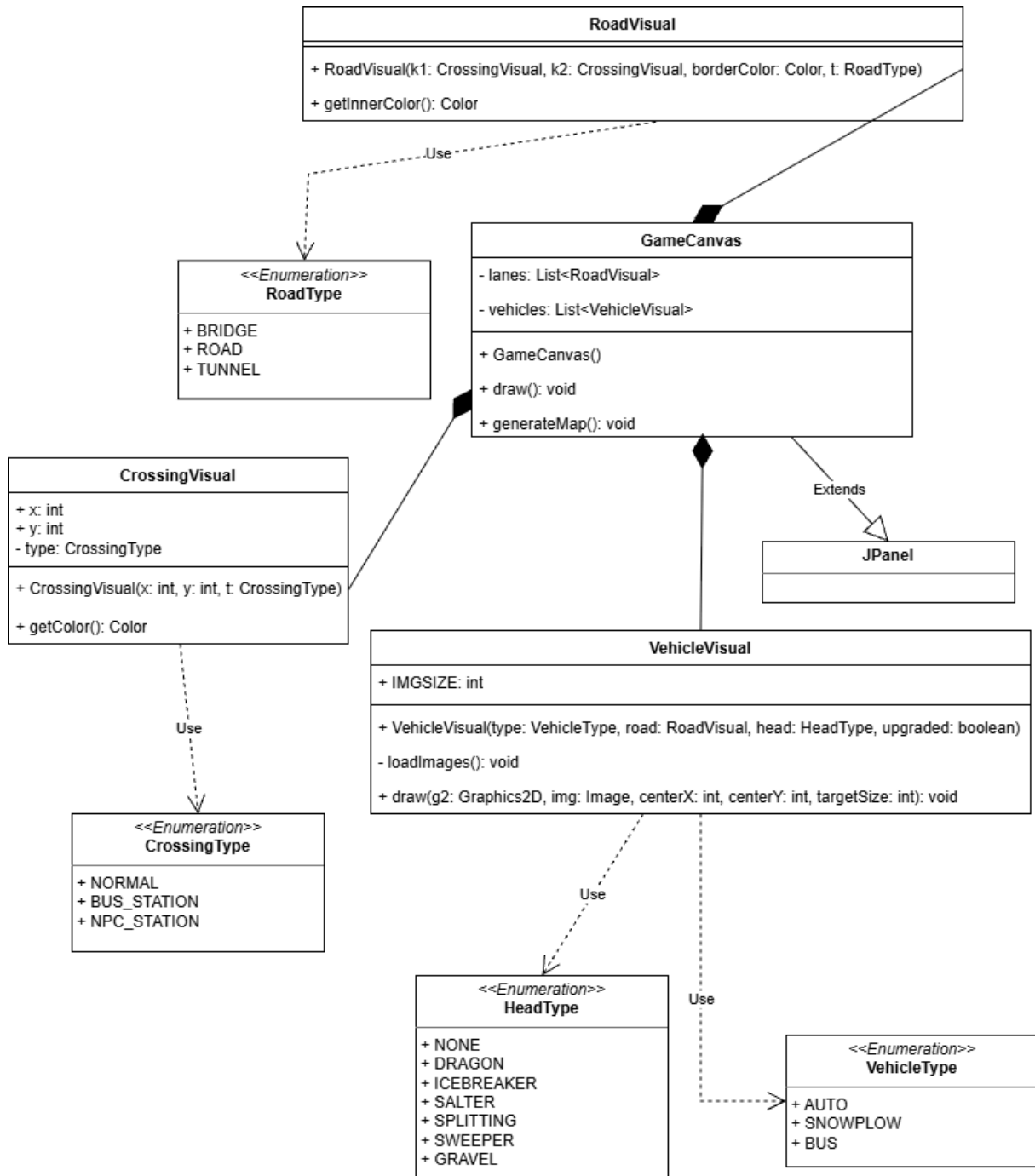
Controller: Minden *View*-hoz egy-egy ilyen tartozik. Kezeli a felhasználók által generált eseményeket, amelyek alapján frissíti a modell állapotát és kéri fel a jelenleg aktív nézetet, hogy frissítse a megjelenést.

11.2.1A felület működési elve

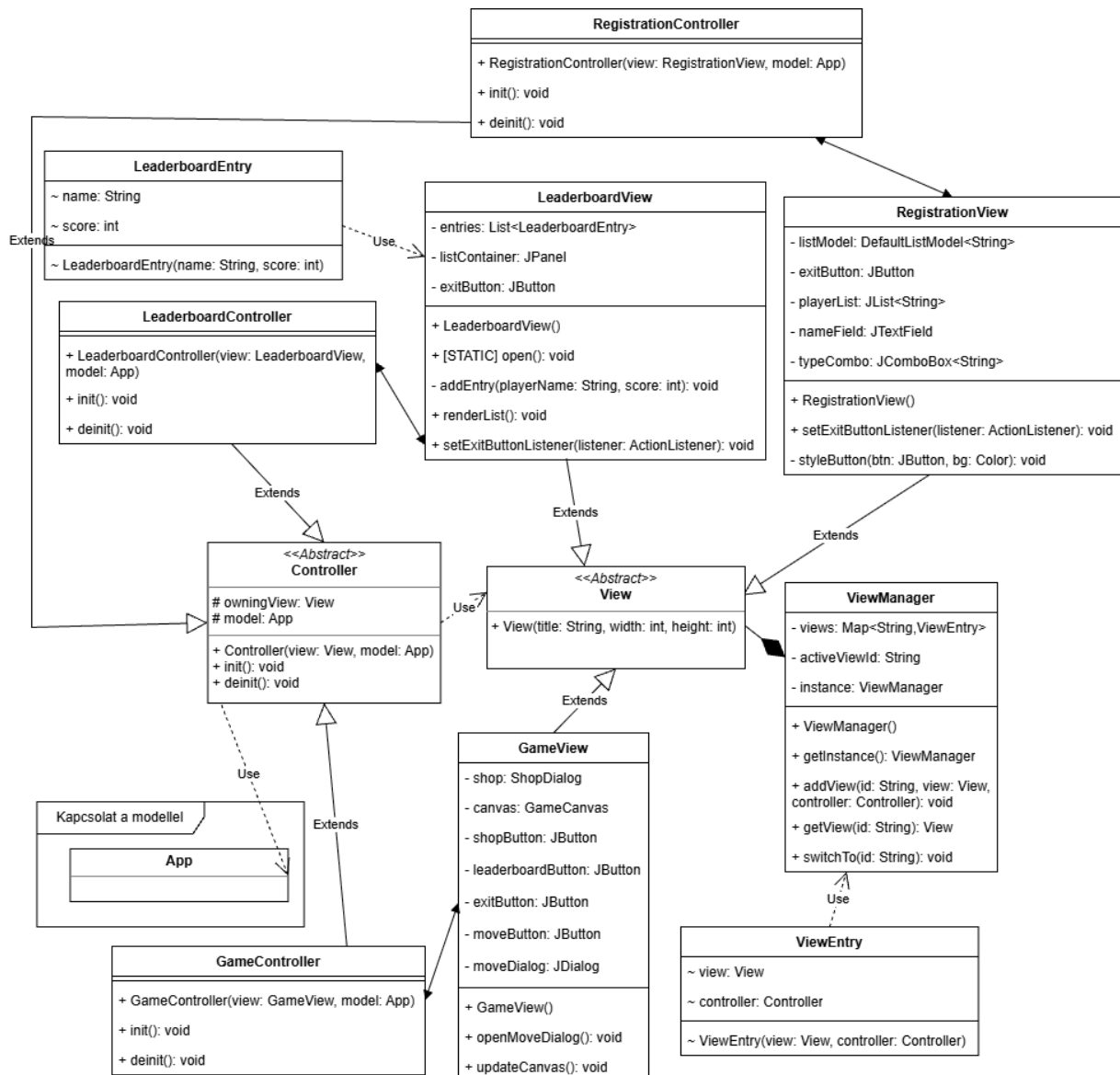
Alapvetően push alapú. A Controller kezeli a felhasználói interakciókat, frissíti a modellt, majd a szükséges adatokat setter metódusokon keresztül a *View*-ba tolja. A tényleges játék során a különböző járműmozgató események, és a "Kör vége" gomb eseménykezelői a Controlleren keresztül frissítik a *View* megjelenését.



Shop UI osztályok



GameCanvas és UI elemei



Viewok és controlok kapcsolata.

11.3 A grafikus objektumok felsorolása

11.3.1 Controller

- **Felelősség**

Ezt az abstract osztályt fogják megvalósítani a különböző kontrollerek, feladatuk összeköttetés létrehozása a modell (App) és a nézetek (GameView, stb.) között.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **#View owningView:** Tárolja a Controllerhez tartozó nézetet.
- **#App model:** Tárolja a Controllerhez tartozó modellt.

- **Metódusok**

- **+void init():** Controller inicializálása, akkor hívódik meg amikor az owningView az aktív nézetté válik.
- **+void deinit():** Controller deinitializálása, akkor hívódik meg amikor az owningView elveszti aktív státuszát.

11.3.2 CrossingType

- **Felelősség**

Enumeráció, amelynek értékei a lehetséges kereszteződés típusok. A CrossingVisual használja a kereszteződés típusának tárolására.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Értékek**

- **NORMAL:** Sima kereszteződés.
- **BUS_STATION:** Buszmegálló.
- **NPC_STATION:** Autó útvonal kezdő/végpont.

11.3.3 CrossingVisual

- **Felelősség**

Egy kereszteződés vizuális reprezentációjára való osztály. A GameCanvas-on jelennek meg.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **+x: int:** A kereszteződés x koordinátája a nézeten belül.
- **+y: int:** A kereszteződés y koordinátája a nézeten belül.
- **-type: CrossingType:** A kereszteződés típusa.

- **Metódusok**

- **+Color getColor():** Visszaadja a kereszteződés színét, a kereszteződés típusa szerint.

11.3.4 GameCanvas

- **Felelősség**

A játék teljes pályájának vizuális reprezentációja, a GameView nézet jeleníti meg. Tárolja az utak és járművek vizualizációjait is, illetve ő felel a pálya vizuális megjelenítéséért.

- **Ősosztályok**

JPanel

- **Interfészek**

Nincs

- **Attribútumok**

- **-List<RoadVisual> lanes:** Tárolja a pályán megjelenítendő sávokat.
- **-List<VehicleVisual> vehicles:** Tárolja a pályán megjelenítendő járműveket.

- **Metódusok**

- **+void draw():** Pálya rajzolása.
- **+void generateMap():** Pálya vizuális legenerálása. (Kiszámítja a kereszteződések optimális pozícióját.)

11.3.5 GameController

- **Felelősség**

A GameView-hoz tartozó Controller.

- **Ősosztályok**

Controller

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölt.

- **Metódusok**

Csak örökölt.

11.3.6 GameView

- **Felelősség**

Ez az osztály a játék fő ablakához tartozó nézet, a konstruktorában beállítódik a játék ikonja, a nézet bal oldalán létrejön egy vezérlőpanel, ahol látszik kinek van éppen a köre, valamint itt találhatóak meg a gombok amikkel meg lehet nyitni a pontlista és a bolt ablakokat, és egy kilépés gomb is található itt. A nézet jobb oldalán pedig megjelenik a GameCanvas.

- **Ősosztályok**

View

- **Interfészek**

Nincs

- **Attribútumok**

- **-ShopDialog shop:** A játék közben megnyitható bolt ablaka.
- **-GameCanvas canvas:** Az utakat, kereszteződéseket és járműveket kirajzoló komponens.
- **-JButton shopButton:** A bolt megnyitására való gomb.
- **-JButton leaderboardButton:** A ranglista megnyitására szolgáló gomb.
- **-JButton exitButton:** A játék nézetből való kilépésre szolgáló gomb.
- **-JButton moveButton:** A mozgás indítására való gomb.
- **-JDialog moveDialog:** A mozgáshoz szükséges dialógusablak.

- **Metódusok**

- **+GameView():** Konstruktor, .

- **+void openMoveDialog():** Megnyitja a mozgáshoz tartozó ablakot.
- **+void updateCanvas():** Frissíti vagy újrarajzolja a játékot megjelenítő GameCanvas komponenst.

11.3.7 HeadType

- **Felelősség**

Enumeráció, értékei a lehetséges hókotró fej típusok. A VehicleVisual használja a fej típusának tárolására.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Értékek**

- **DRAGON:** Sárkány fej.
- **GRAVEL:** Zúzottkő szóró fej.
- **ICEBREAKER:** Jégtörő fej.
- **SALTER:** Sósóró fej.
- **SPLITTING:** Hányó fej.
- **SWEEPER:** Söprő fej.
- **NONE:** Semmilyen fej. (pl. autó vagy busznál)

11.3.8 LeaderboardController

- **Felelősség**

A LeaderboardView-hoz tartozó Controller.

- **Ősosztályok**

Controller

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölt.

- **Metódusok**

Csak örökölt.

11.3.9 LeaderboardEntry

- **Felelősség**

Játékosnév-pont rekordokat tárol. A LeaderboardView és LeaderboardController által használt osztály.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **+String name:** A játékos neve.
- **+int score:** A játékos pontszáma.

- **Metódusok**

Nincs

11.3.10 LeaderboardView

- **Felelősség**

Felelős a ranglista grafikus megjelenítéséért. Megjeleníti a játékosok nevét és pontszámát egy listában vagy panelen és biztosít egy kilépés gombot. A megjelenítendő adatot nem tárolja, megkapja a LeaderBoardControllertől

- **Ősosztályok**

View

- **Interfészek**

Nincs

- **Attribútumok**

- **-List<LeaderboardEntry> entries:** Bejegyzések listája, amiket a ranglistában megjelenít.
- **-JPanel listContainer:** A ranglista elemeit tartalmazó panel.
- **-JButton exitButton:** A ranglistás nézetből való kilépésre vagy visszalépésre szolgáló gomb.

- **Metódusok**

- **+LeaderboardView():** Konstruktor
- **+void open(): void:** Megnyitja a ranglista nézetet.
- **-void addEntry(playerName: String, score: int):** Hozzáad egy játékost a ranglista megjelenített elemeihez(név+pontszám).
- **+void renderList():** Újrarajzolja a ranglista elemeit az entries lista alapján.
- **+void setExitButtonListener(listener: ActionListener):** Beállítja az exitButton eseménykezelőjét.

11.3.11 ShopDialog

- **Felelősség**

Felelős az osztály grafikus megjelenítésért. Egy felugró ablakban megjelennek a megvásárolható elemek, az aktuális játékos neve és pontszáma, valamint a vásárlás és bezárás gombjai.

- **Ősosztályok**

JDialog

- **Interfészek**

Nincs

- **Attribútumok**

- **-DefaultListModel<ShopItemData> itemListModel:** A boltban megjelenő elemek listája
- **-JList<ShopItemData> itemList:** A megvásárolható elemeket megjelenítő lista.
- **-JLabel playerLabel:** Az aktuális játékos nevének megjelenítésére van.
- **-JLabel pointsLabel:** Az aktuális játékos pontszámának megjelenítésére van.
- **-JButton buyButton:** A vásárlást indító gomb.
- **-JButton closeButton:** A bolt bezárására szolgáló gomb.

- **Metódusok**

- **+ShopDialog(owner: JFrame):** Létrehozza a bolt párbeszédablakát a megadott főablakhoz kötve
- **+void showItems(items: List<ShopItemData>):** Frissíti a boltban megjelenített elemek listáját.

- **+void setPlayerInfo(playerName: String, points: int):** Beállítja az aktuális játékos nevét és pontszámát.
- **+int getSelectedItemIndex():** Visszaadja a listában kiválasztott elem indexét.
- **+void setBuyButtonListener(listener: ActionListener):** Beállítja a vásárlás gomb eseménykezelőjét.
- **+void setCloseButtonListener(listener: ActionListener):** Beállítja a bezárás gomb eseménykezelőjét.
- **+void errorShower(message: String):** Hibaüzenetet jelenít meg a felhasználónak.
- **+void open():** Megnyitja a bolt ablakot.
- **+void close():** Bezárja a bolt ablakot.

11.3.12 ShopController

- **Felelősség**

Felelős a bolt működésének vezérléséért. Összeköti a modellt a grafikus felülettel.

- **Össztályok**

nincs

- **Interfészek**

nincs

- **Attribútumok**

- **-ShopDialog dialog:** A controllerhez tartozó dialógusablak.
- **-App model:** A játék fő modellje, amelyből elérhetők a játékosok és a játék állapota.
- **-SnowPlowShop shop:** A bolt modellje, amely tartalmazza a megvásárolható elemeket és a vásárlási logikát.
- **-Player currentPlayer:** Az éppen vásárló játékos.

- **Metódusok**

- **+ShopController(dialog: ShopDialog, model: App):** Beállítja a vezérelt dialógust.
- **+void init():** Inicializálja a controller eseménykezelőit.
- **+void open(player: Player):** Megnyitja a boltot a megadott játékos számára, beállítja az aktuális játékos, frissíti az adatokat és megjeleníti a dialógust.
- **-void refreshItems():** Frissíti a boltban megjelenített elemek listáját.
- **-void refreshPlayerInfo():** Frissíti a játékos nevét és pontszámát(dialógusban).
- **-void handleBuy():** Kezeli a vásárlás gombra kattintást. Lekéri a kiválasztott elemet, ellenőrzi, hogy vásárolható-e, majd meghívja a modell vásárlási műveletét.
- **-List<ShopItemData> createShopItemData():** A modellben található Item objektumból ShopItemData listát készít.

11.3.13 ShopItemdata

- **Felelősség**

ShopDialog számára tartalmazza a megjelenítéshez szükséges adatokat.

- **Össztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **-int index:** Az item boltban használt indexe
- **-String name:** Az item neve
- **-int cost:** Az item ára

- **Metódusok**

- **+ShopItemData(int index, string name):** konstuktor
- **+int getIndex():** Visszaadja az item indexét
- **+int getCost():** Visszaadja az item árát
- **+string toString():** Visszaadja az item megjelenítendő szövegét

11.3.14 RegistrationController

- **Felelőség**

A RegistrationView-hoz tartozó Controller.

- **Össztályok**

Controller

- **Interfészek**

Nincs

- **Attribútumok**

Csak örökölt.

- **Metódusok**

Csak örökölt.

11.3.15 RegistrationView

- **Felelőség**

Ez az osztály felelős a regisztrációs nézet megjelenítéséért. Tartalmazza a játékosnév megadásához szükséges mezőt, a jármű típus kiválasztásához tartozó legördülő listát, a már hozzáadott játékosok listáját stb.

- **Össztályok**

View

- **Interfészek**

Nincs

- **Attribútumok**
 - **-DefaultListModel<String> listModel:** A regisztrált játékosok neveit tároló lista
 - **-JButton exitButton:** A regisztrációs nézetből való kilépésre vagy továbblépésre való gomb.
 - **-JList<String> playerList:** A már regisztrált játékosok listáját megjelenítő komponens.
 - **-JTextField nameField:** A játékosok nevének megadására szolgáló mező.
 - **-JComboBox<String> typeCombo:** A kiválasztandó járműtipusokat tartalmazó legördülő lista
- **Metódusok**
 - **+RegistrationView():** konstruktor
 - **+void setExitButtonListener(ActionListener listener):** Beállítja az exitButton eseménykezelőjét
 - **+void styleButton(Button btn, Color bg):** Beállítja a gombra alkalmazott stílust

11.3.16 RoadType

- **Felelőség**

Enumeráció, értékei a lehetséges út típusok. A RoadVisual használja az út típusának tárolására.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Értékek**

- **BRIDGE:** Híd.
- **ROAD:** Sima út.
- **TUNNEL:** Alagút.

11.3.17 RoadVisual

- **Felelőség**

Felelős az útszakasz grafikus megjelenítéséért

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**
 - **-CrossingVisual k1:** Az útszakasz egyik végpontjához tartozó kereszteződés vizuális reprezentációja.
 - **-CrossingVisual k2:** Az útszakasz másik végpontjához tartozó kereszteződés vizuális reprezentációja.
 - **-borderColor: Color:** Az út szegélyének színe.
 - **-RoadType t:** Az útszakasz típusa.
- **Metódusok**
 - **+RoadVisual(CrossingVisua k1, CrossingVisual k2, Color borderColor, RoadType t):** Konstruktor
 - **+Color getInnerColor():** Visszaadja az útszakasz színét. Ez az érték az úttípus alapján változhat.

11.3.18 VehicleType

- **Felelősség**

Enumeráció, értékei a lehetséges jármű típusok. A VehicleVisual használja a jármű típusának tárolására.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Értékek**
 - **AUTO:** Autó.
 - **SNOWPLOW:** Hókotró.
 - **BUS:** Busz.

11.3.19 VehicleVisual

- **Felelősség**

Egy jármű grafikus megjelenítéséért felelős. Tárolja a jármű pozícióját, méretét, típusát, forgatását.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **+static int IMG_SIZE:** Jármű kirajzolásához használt lépméret
- **-Image image:** A járműhöz tartozó aktuálisan használt kép
- **-VehicleType type:** A megjelenített jármű típusa
- **-HeadType head:** A hókotróhoz tartozó felszerelt fej típusa
- **-boolean upgraded::** Jelzi, hogy a jármű fejlesztett állapotban van e

- **Metódusok**

- **+VehicleVisual(VehicleType, RoadVisual road, HeadType head, boolean upgraded):** konstruktor, beállítja a jármű vizuális adatait
- **-void loadImage():** Betölti a jármű típusához, fejéhez, fejlesztettségéhez tartozó képeket
- **+void draw(Graphics2D g2, int index, int totalOnRoad, int roadOffsetIndex, int totalRoadsBetweenPoints):** Kirajzolja a járművet a megfelelő útszakaszra. Az index és a totalOnRoad értékek alapján több jármű esetén eltolva helyezi el őket ugyanazon az úton.
- **-void drawScaledImage(Graphics2D g2, Image img, int centerX, int centerY, int targetSize):** A megadott képet középre igazítva és méretezve rajzolja ki

11.3.20 View

- **Felelősség**

Abstract osztály, amelyet a nézetek valósítanak meg. Minden nézet egy külön ablak, a view konstruktorában beállítódik az ablak címe, mérete és egyéb potenciális alap tulajdonságai.

- **Őszosztályok**

JFrame

- **Interfészek**

Nincs

11.3.21 ViewEntry

- **Felelősség**

View-Controller rekordokat tárol. A ViewManager egy belső osztálya.

- **Őszosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **+View view:** A tárolt nézet.
- **+Controller controller:** A tárolt nézethez tartozó Controller.

- **Metódusok**

Nincs

11.3.22 ViewManager

- **Felelősség**

A különböző nézetek tárolásáért, és köztük való váltásért felelős singleton osztály.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

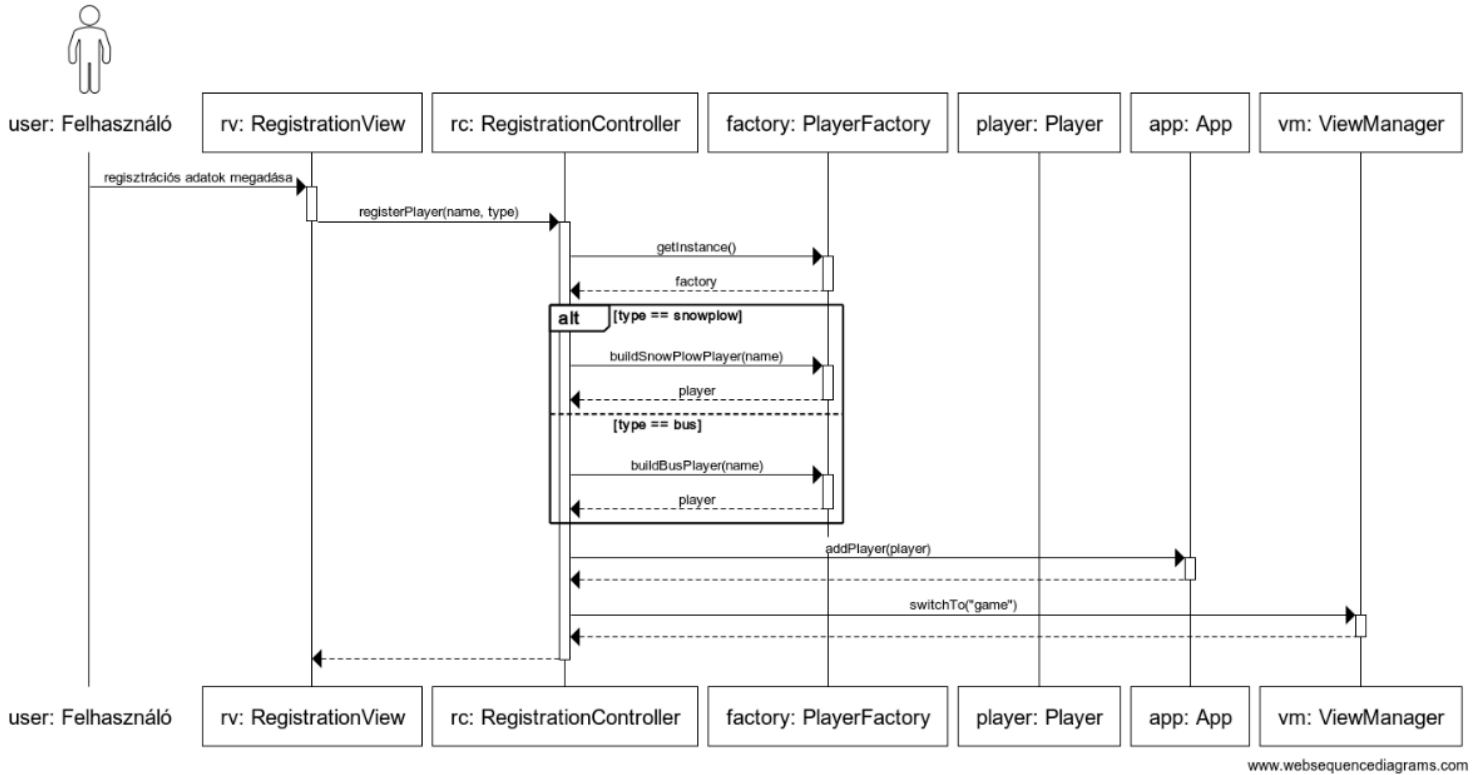
- **-Map<String, ViewEntry> views:** Szöveges azonosítóhoz rendeli az egyes nézeteket, így könnyen elérhetőek.
- **-String activeViewId:** Az aktív nézethez tartozó azonosító.
- **-static ViewManager instance:** A singleton példány.

- **Metódusok**

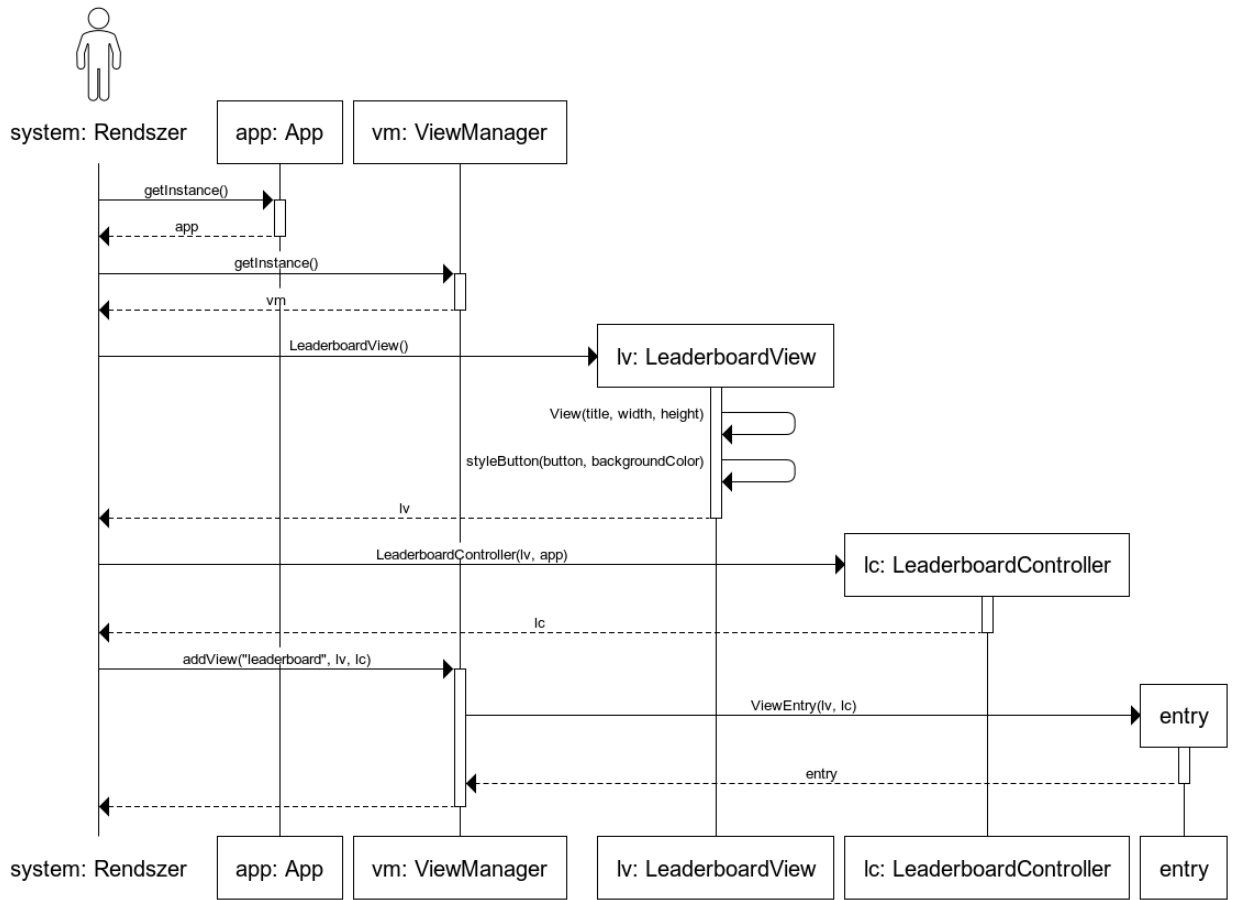
- **+static ViewManager getInstance():** Singleton instance elérése, vagy pedig létrehozása ha még nem létezik.
- **+View getView(String id):** Paraméterben megadott szöveges azonosítóhoz megfelelő nézetet adja vissza.
- **+void switchTo(String id):** Paraméterben megadott szöveges azonosítóhoz megfelelő nézetet teszi az aktív nézetté, közben meghívva a megfelelő életciklus kezelő init és deinit metódusokat.

11.4 Kapcsolat az alkalmazói rendszerrel

Játékos regisztrálása

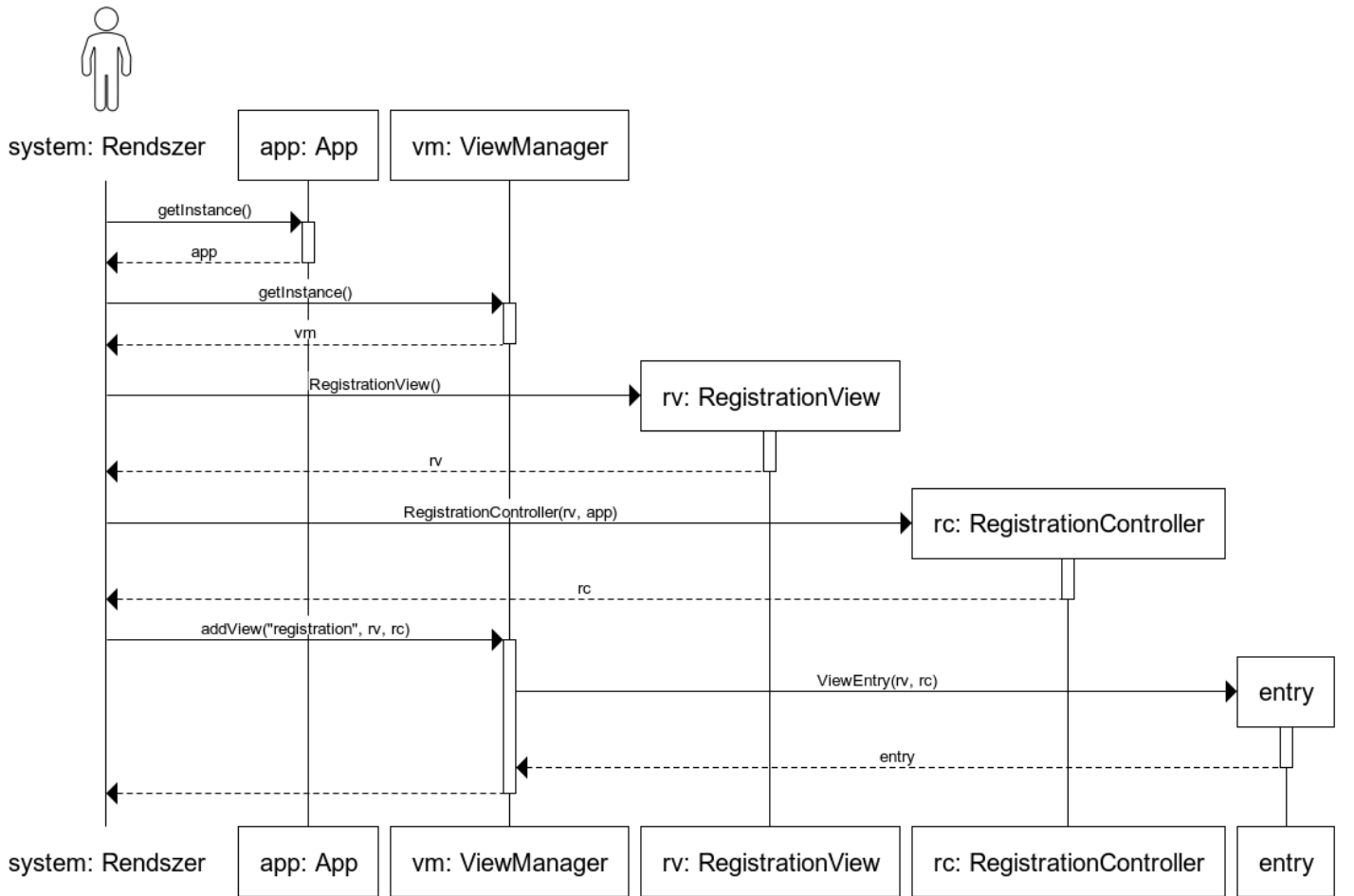


LeaderboardView regisztrálása



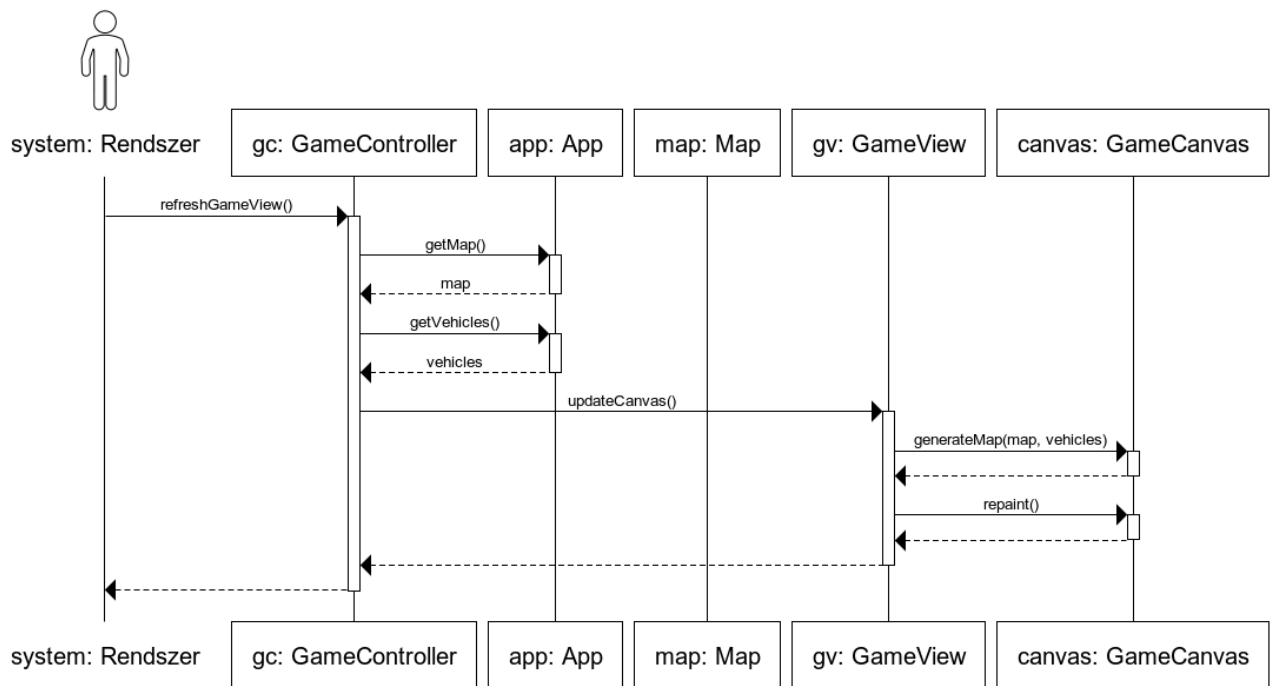
www.websequencediagrams.com

RegistrationView regisztrálása



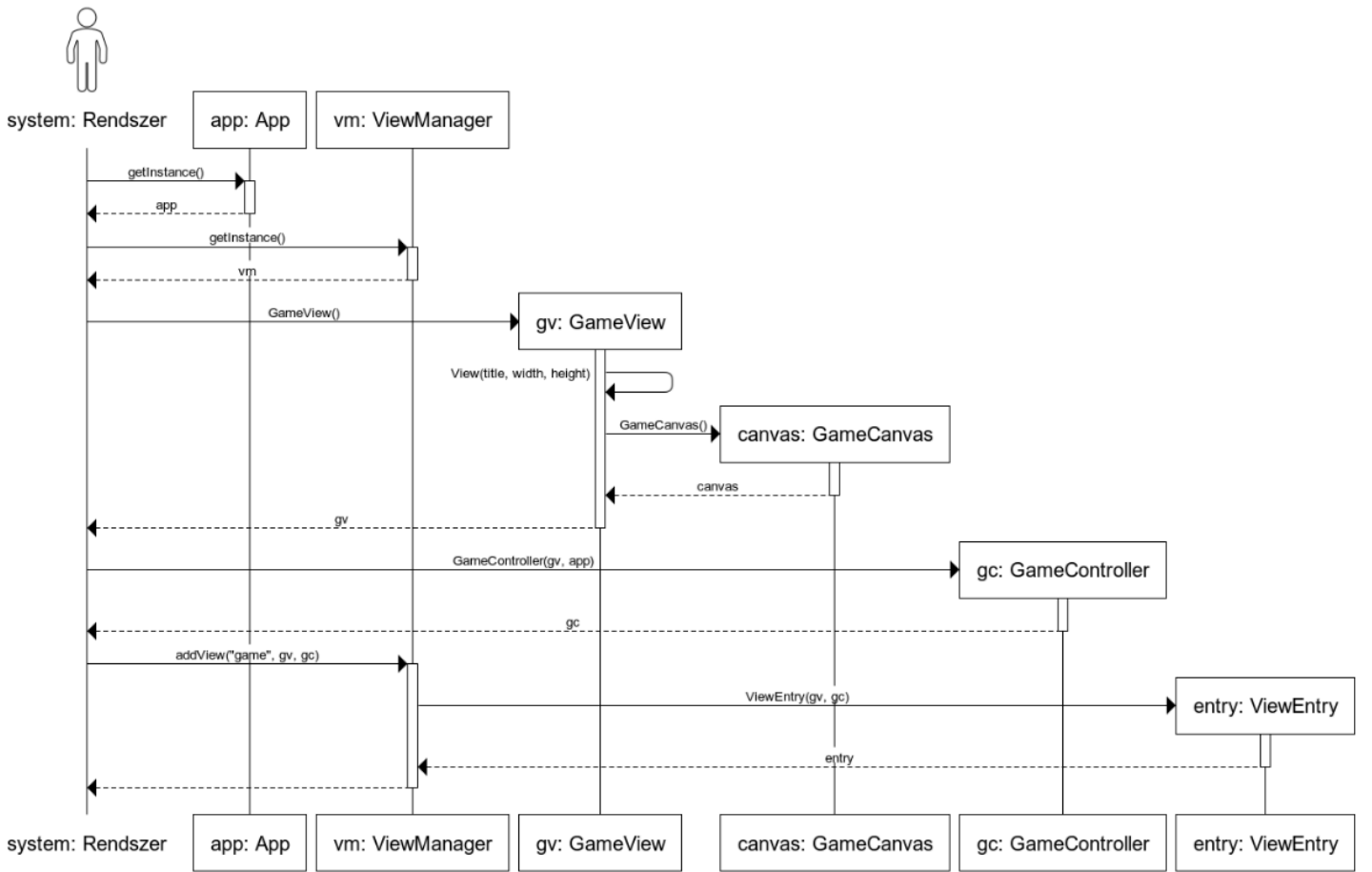
www.websequencediagrams.com

Játéktér frissítése



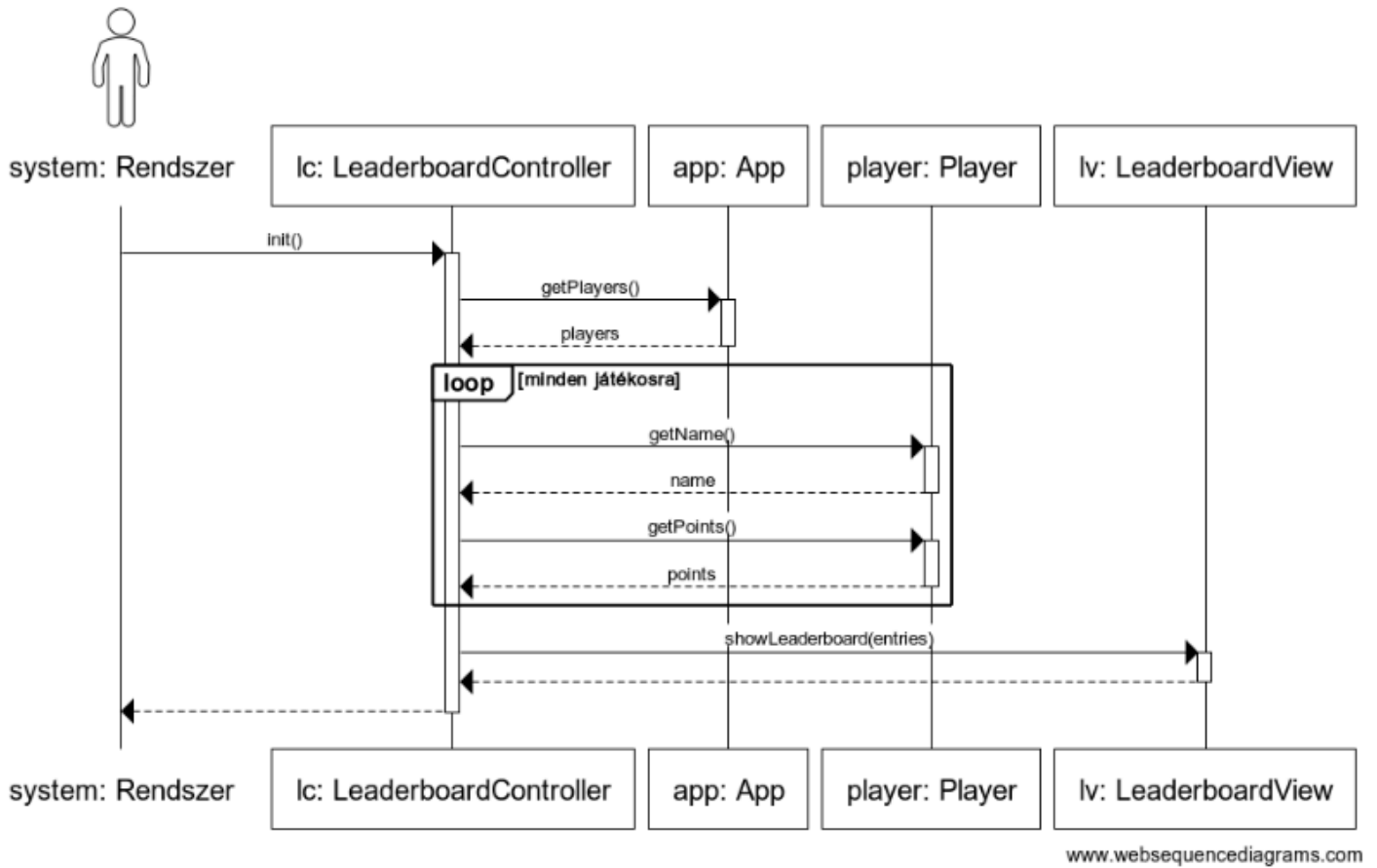
www.websequencediagrams.com

GameView regisztrálása

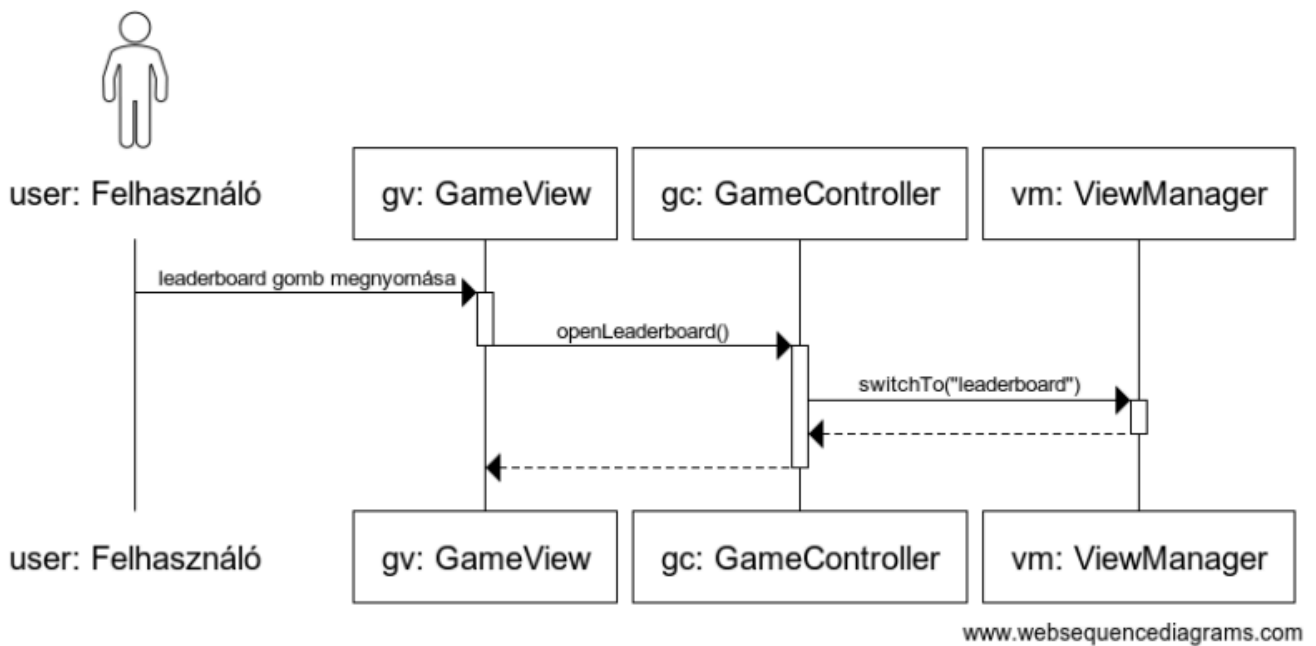


www.websequencediagrams.com

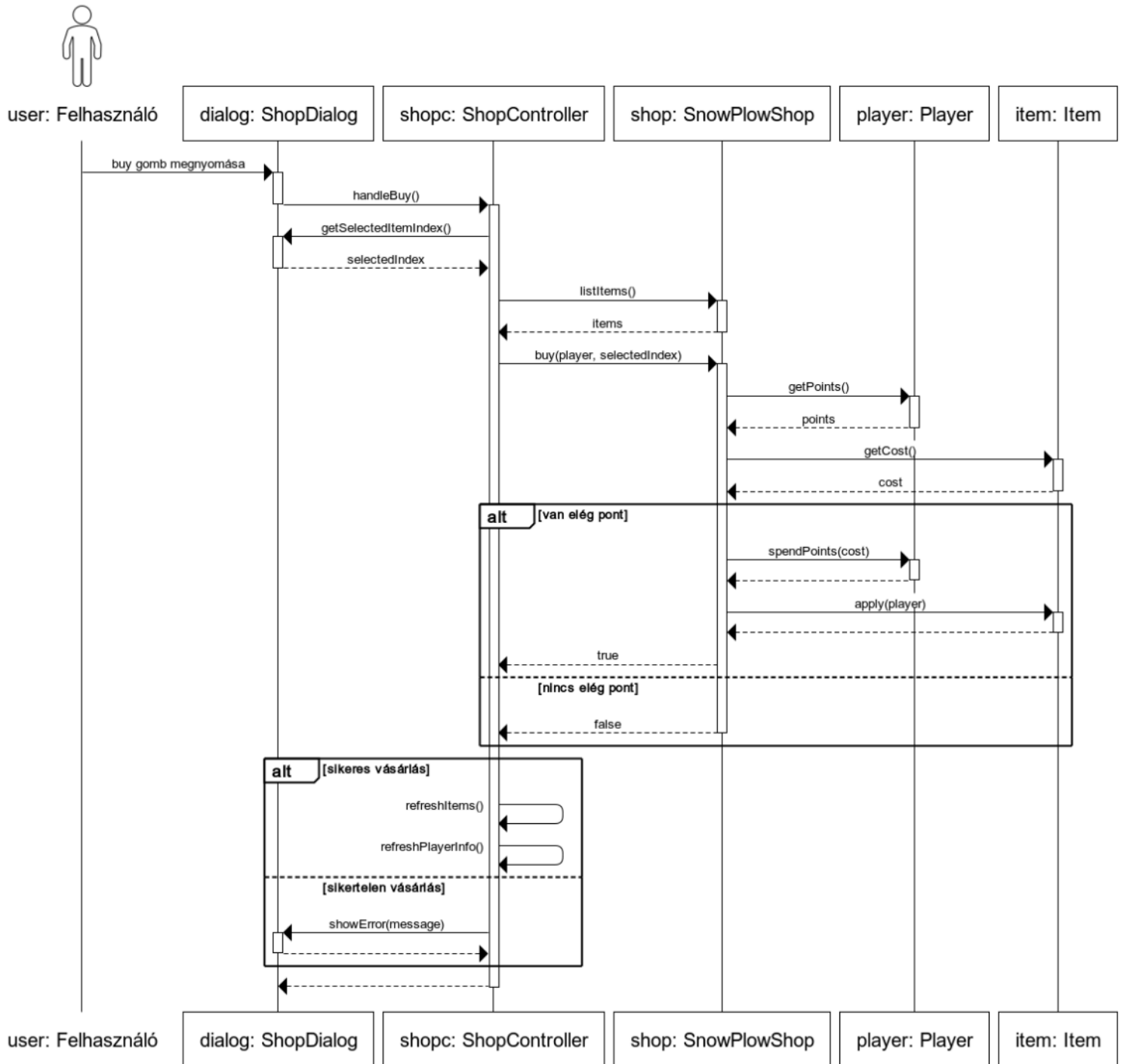
Leaderboard lista frissítése



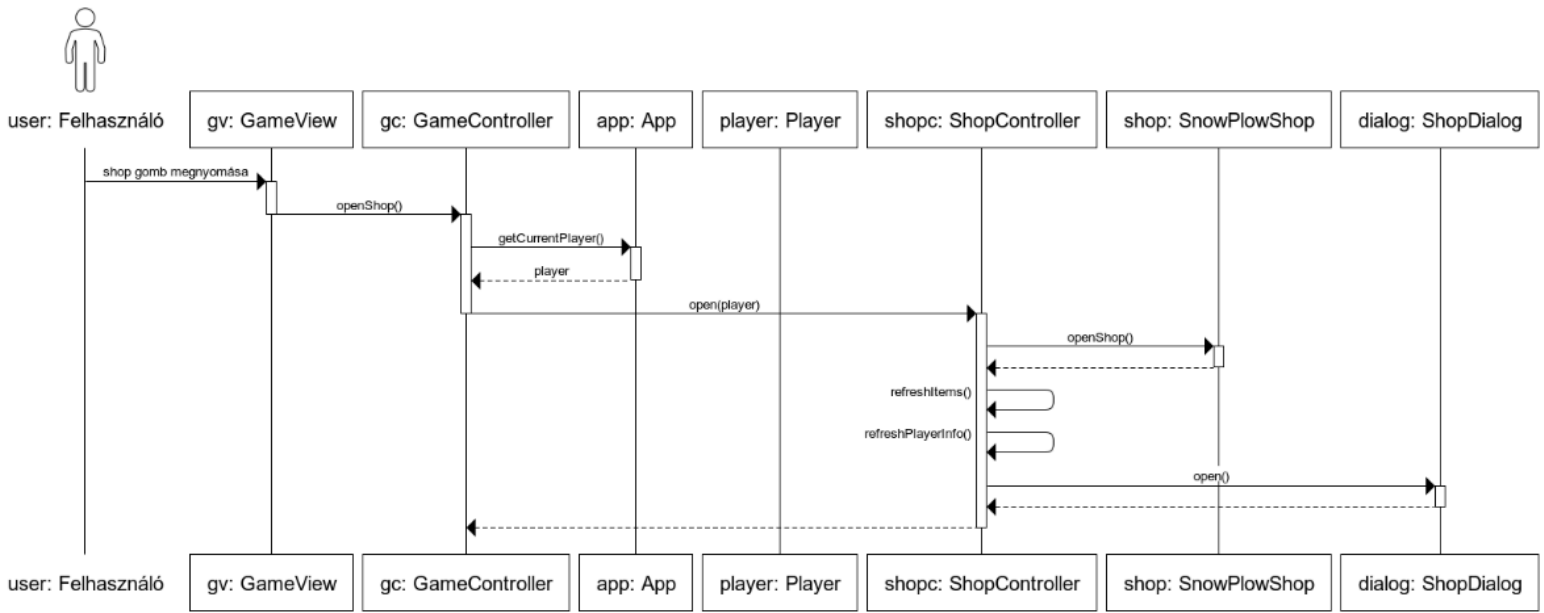
Leaderboard megnyitása



Vásárlás a shopban

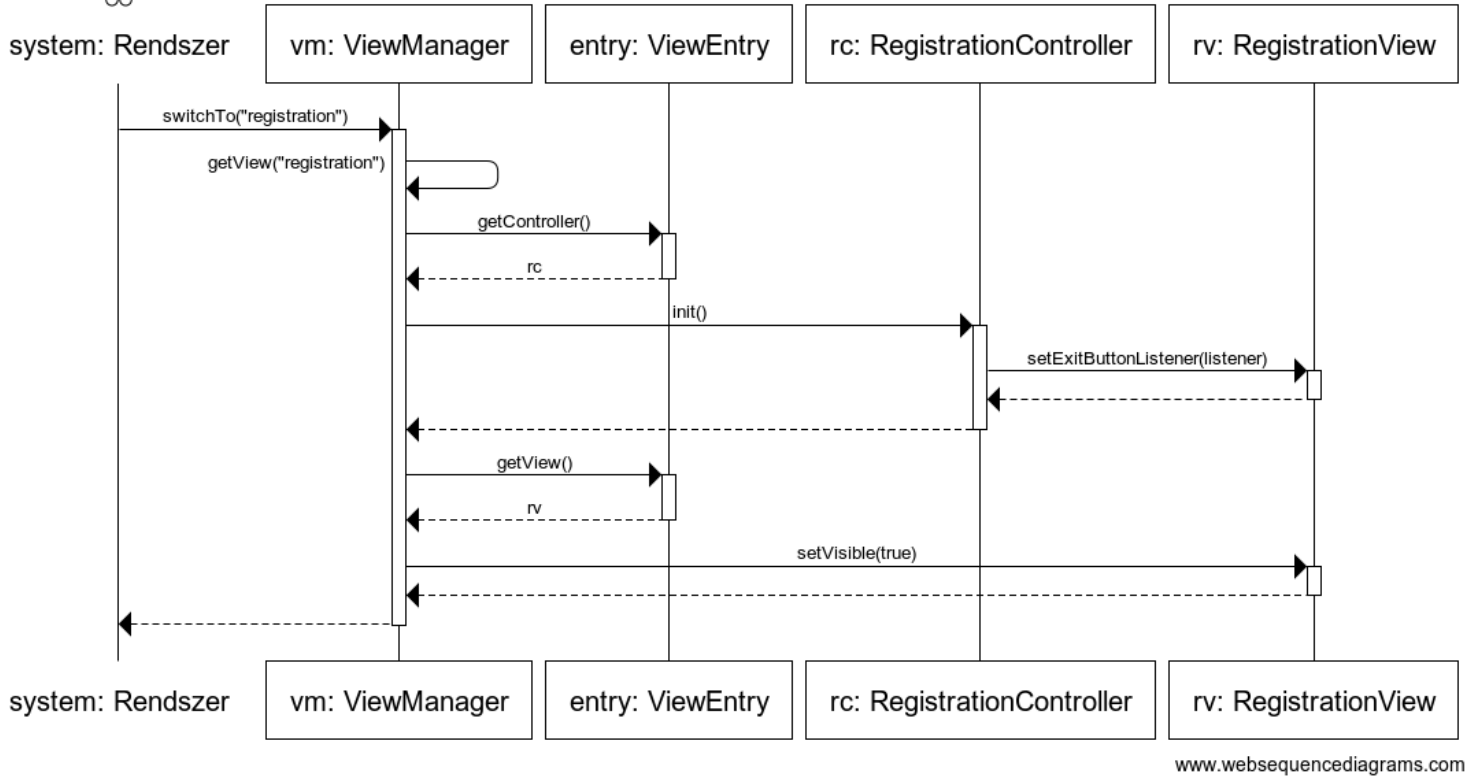


Shop megnyitása



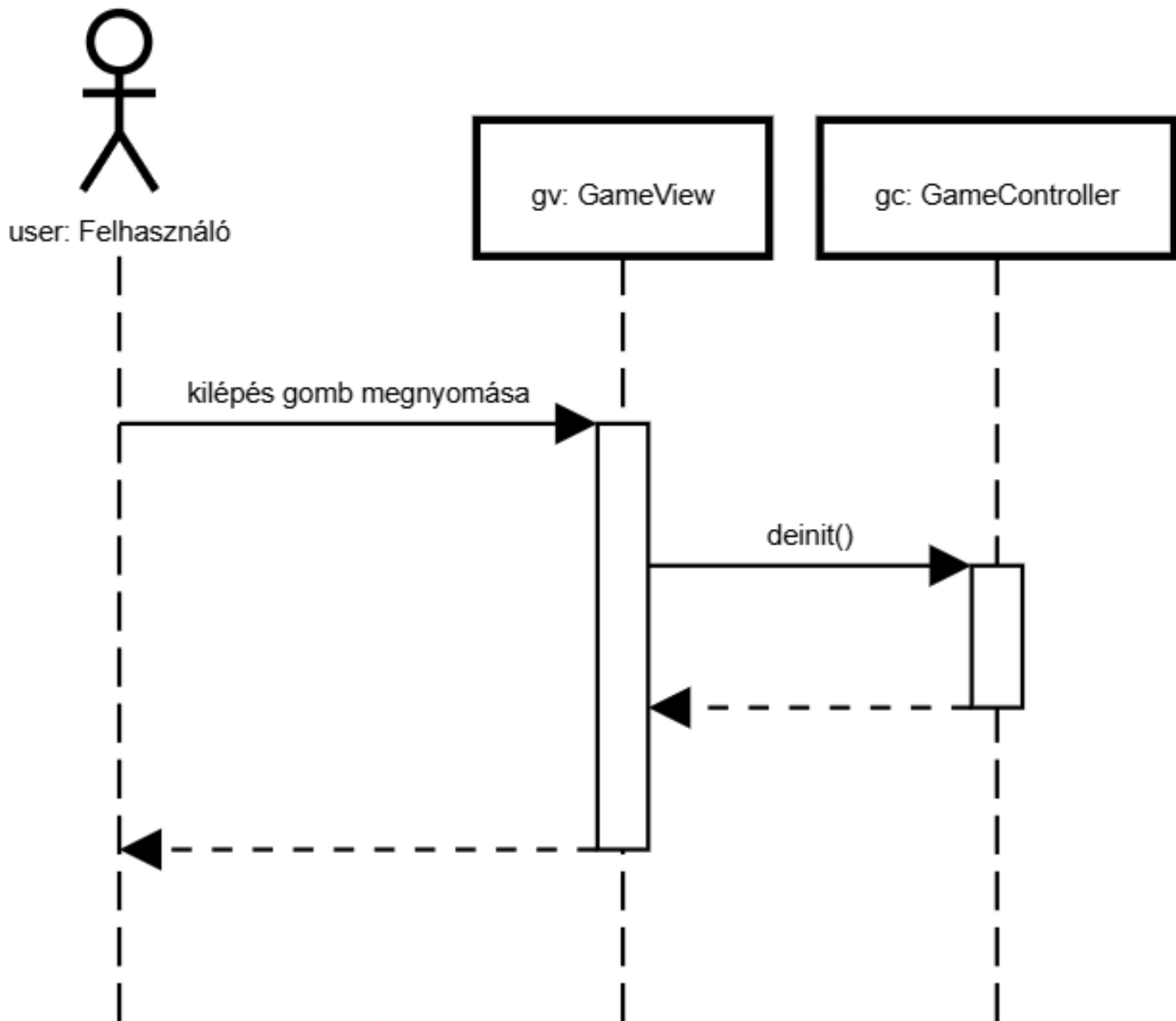
www.websequencediagrams.com

Kezdő nézet aktiválása

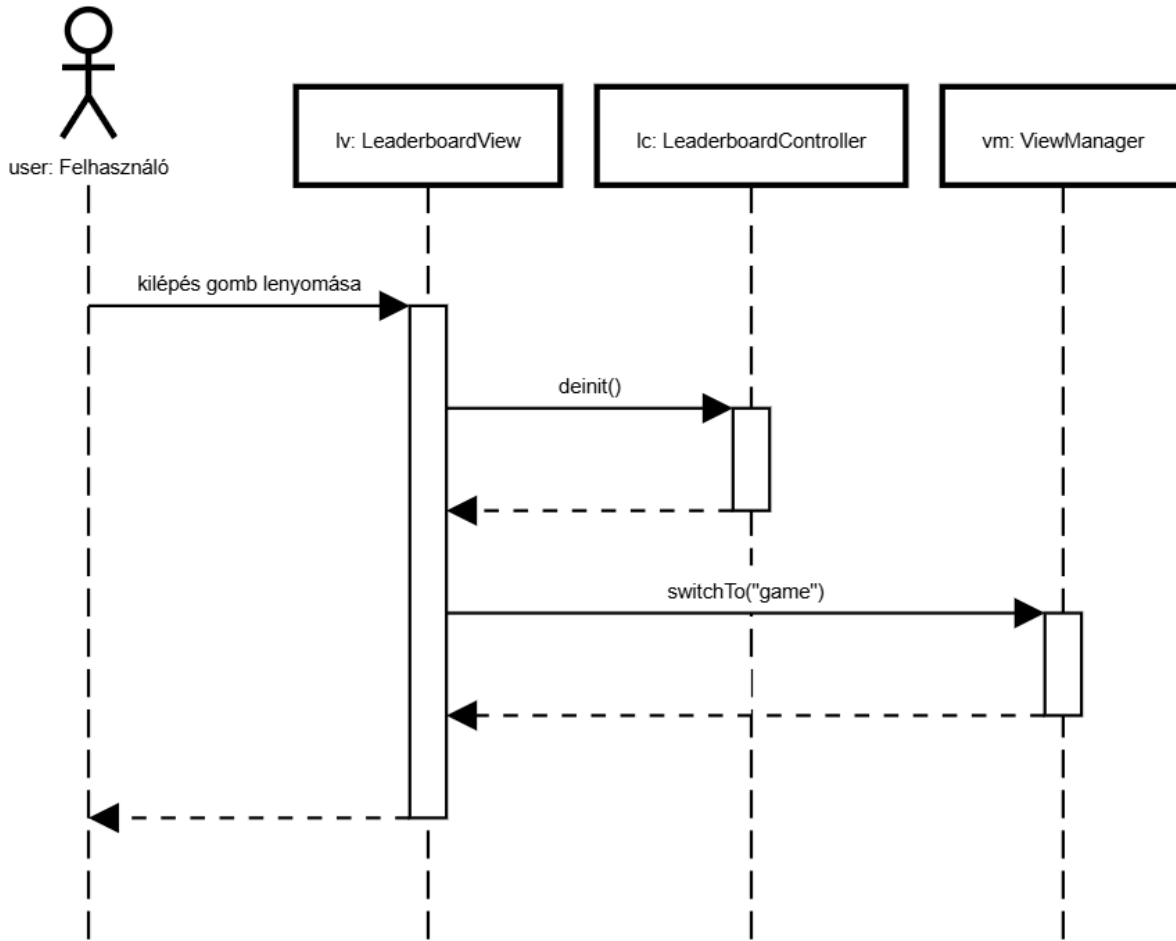


www.websequencediagrams.com

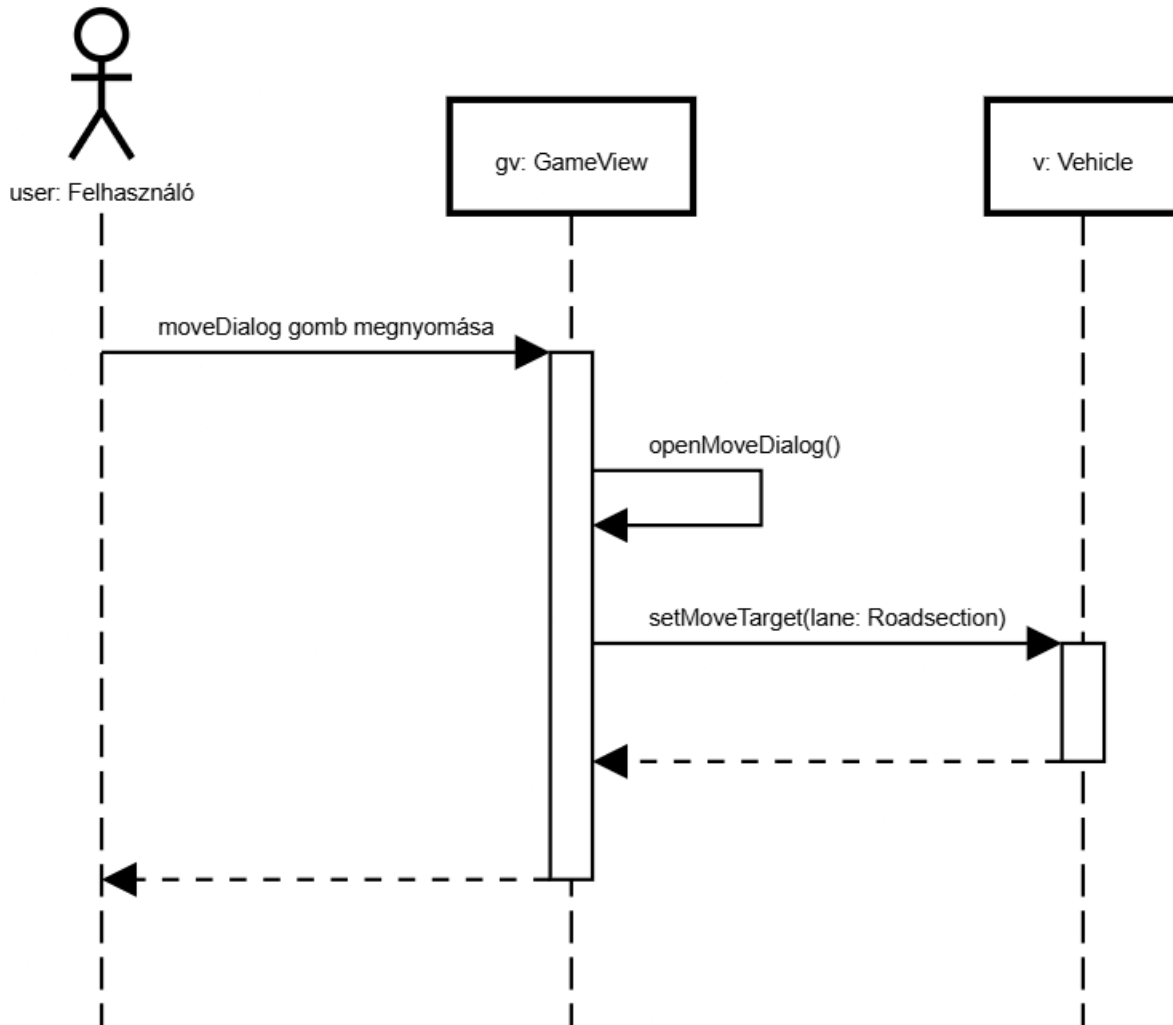
Játék bezárása



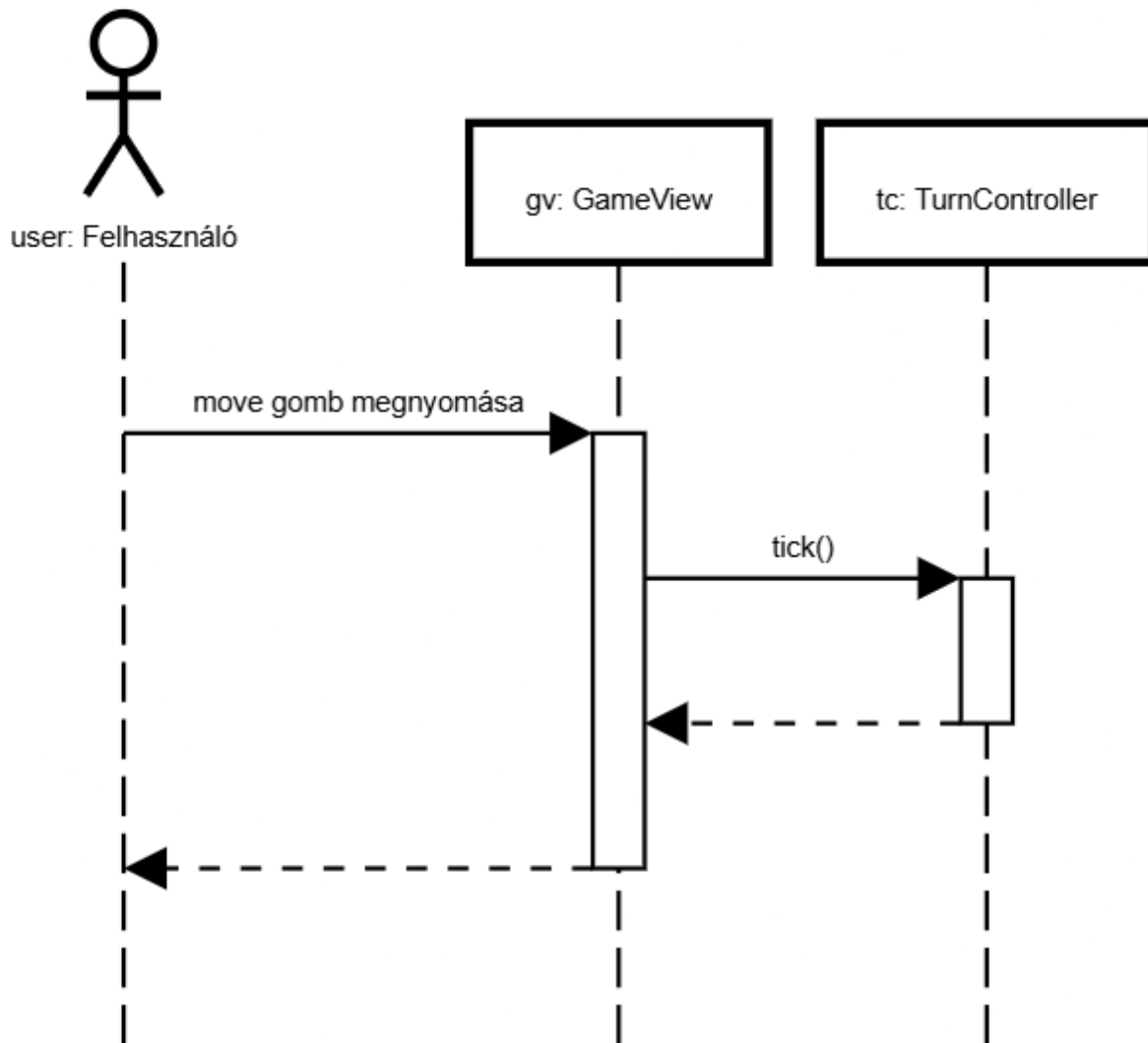
Leaderboard bezárása



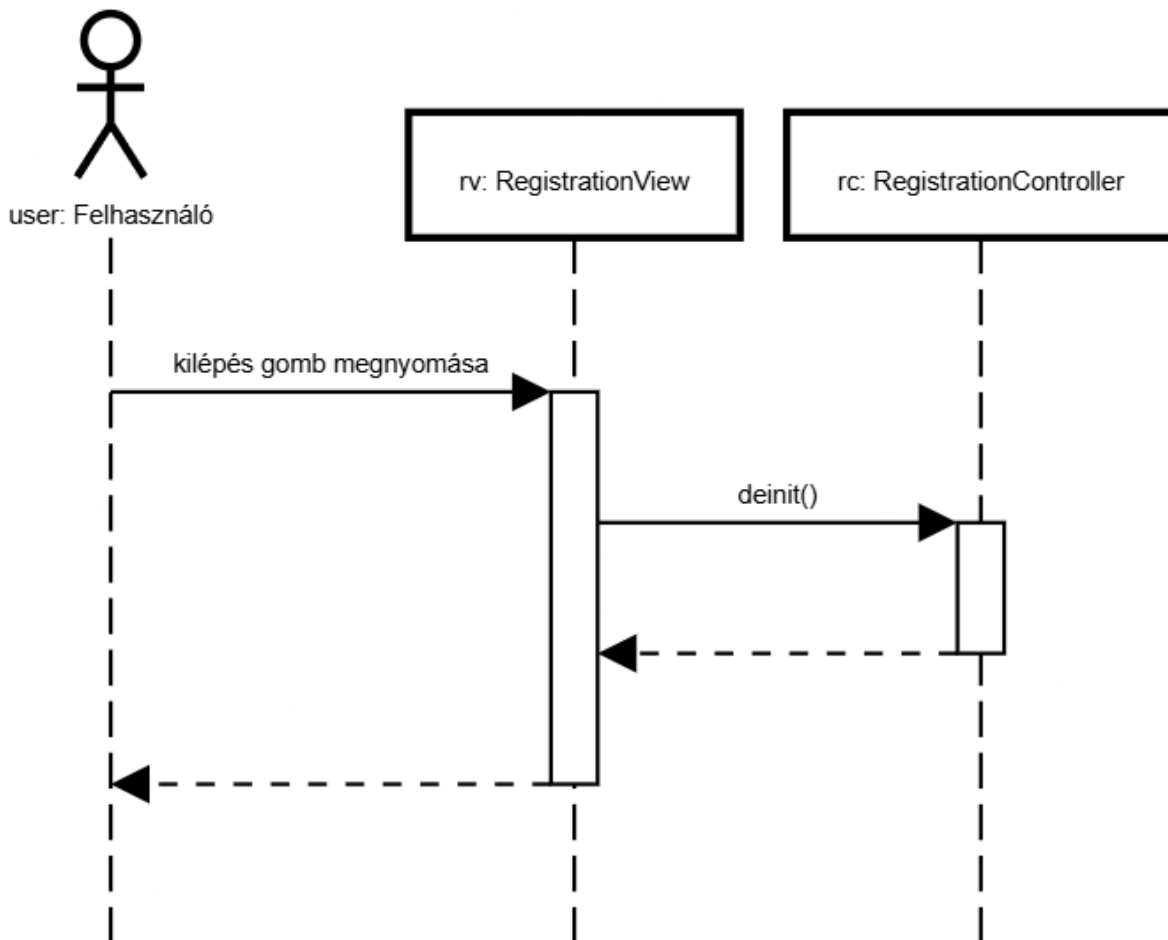
Következő helyre mozgás beállítása



játékos mozgatása



Regisztrálás bezárása



Napló

Kezdet	Időtartam	Résztevők	Leírás
2026.05.06. 12:50:00	1,5	Funk	Tevékenység: UI struktúra kialakítása, grafikai elemek készítése.
2026.05.06. 16:00:00	1,5	Funk	Tevékenység: Képek, UI folytatása.
2026.05.06. 20:00:00	0,5	<i>Mindenki</i>	Értekezlet. Téma: mindenki nagyon ügyes.
2026.05.07. 20:00:00	0,55	Heltai	Tevékenység: modell leírása, struktúra kialakítása.
2026.05.09. 9:00:00	1,1	Funk	Tevékenység: új UI

			elemek, UI UML osztálydiagram, új visualok
2026.05.09. 10:06:00	0,1	Funk	Tevékenység: Új UI felvétele a dokumentációba.
2026.05.09. 14:30:00	1	Csordás	Tevékenység: Osztályleírások kidolgozása.
2026.05.10. 11:00:00	0,5	Rózsa	Tevékenység: Shop UI UML
2026.05.10. 14:00:00	0,5	Heltai	Tevékenység: osztálydiagram és leírások módosítása.
2026.05.10. 13:00:00	2,05	Rózsa	Tevékenység: UI UML, dokumentum formázása, osztályleírások megírása
2026.05.10. 15:00:00	2	Rózsa	Tevékenység: szekvenciák készítése, beszúrása
2026.05.10. 13:00:00	1	Funk	Tevékenység: Osztálydiagramok befejezése
2026.05.10. 15:50:00	0,65	Funk	Tevékenység: Dokumentációba diagramok/képek illesztése
2026.05.10. 0:00:00	1,5	Fekete	Tevékenység: Szekvenciák készítése

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

A legtöbb keletkezési dátum a Git miatt "helytelen," mivel minden pull-t követve megváltoztatta a keletkezések dátumát is.

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
./build.bat	1330	2026. 05. 25.	Build script Windows rendszerekhez.
./build.sh	882	2026. 05. 25.	Build script Unix rendszerekhez.
tlayouts/example.maplayout	9	2026. 05. 25.	Példa pálya.
tlayouts/large.maplayout	819	2026. 05. 25.	Az ajánlott pálya.
tlayouts/t1.maplayout	51	2026. 05. 25.	Példa pálya.
tlayouts/t2.maplayout	49	2026. 05. 25.	Példa pálya.
tlayouts/test1.maplayout	102	2026. 05. 25.	Példa pálya.
assets/img/auto.png	28672	2026. 05. 25.	Asset.
assets/img/auto.svg	3915	2026. 05. 25.	Asset.
assets/img/busz.png	30397	2026. 05. 25.	Asset.
assets/img/busz.svg	3916	2026. 05. 25.	Asset.
assets/img/hokotro.png	111230	2026. 05. 25.	Asset.
assets/img/hokotro.svg	48666	2026. 05. 25.	Asset.
assets/img/snowplow.png	48730	2026. 05. 25.	Asset.
assets/img/snowplow.svg	4896	2026. 05. 25.	Asset.
assets/img/snowplow2.png	61314	2026. 05. 25.	Asset.

assets/img/heads/dragon_head.png	23765	2026. 05. 25.	Asset.
assets/img/heads/dragon_head.svg	9012	2026. 05. 25.	Asset.
assets/img/heads/gravel_head.png	37057	2026. 05. 25.	Asset.
assets/img/heads/gravel_head.svg	11131	2026. 05. 25.	Asset.
assets/img/heads/icebreaker_head.png	19684	2026. 05. 25.	Asset.
assets/img/heads/icebreaker_head.svg	2947	2026. 05. 25.	Asset.
assets/img/heads/salter_head.png	27938	2026. 05. 25.	Asset.
assets/img/heads/salter_head.svg	11292	2026. 05. 25.	Asset.
assets/img/heads/splitting_head.png	23382	2026. 05. 25.	Asset.
assets/img/heads/splitting_head.svg	13907	2026. 05. 25.	Asset.
assets/img/heads/sweeper_head.png	37201	2026. 05. 25.	Asset.
assets/img/heads/sweeper_head.svg	15017	2026. 05. 25.	Asset.
src/main/java/javallerok/Main.java	1359	2026. 05. 25.	A program entry pointja: ez indítja el a játékot.
src/main/java/javallerok/Utils/Random.java	1389	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/Biokerosene.java	845	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/DragonHead.java	2178	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/Gravel.java	836	2026. 05. 25.	Java forrásfájl.

src/main/java/javallerok/items/GravelHead.java	2147	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/IceBreakerHead.java	1572	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/Item.java	1205	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/Salt.java	740	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/SalterHead.java	2075	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/SnowPlowHead.java	1379	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/SnowPlowUpgrade.java	1128	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/SnowPlowUpgradeNormal.java	830	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/SnowPlowUpgradePremium.java	827	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/SplittingHead.java	1660	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/items/SweeperHead.java	1898	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/CrossingVisual.java	1094	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/GameCanvas.java	16909	2026. 05. 25.	Java forrásfájl.

src/main/java/javallerok/ui/RoadVisual.java	2070	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/ShopItemData.java	639	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/VehicleVisual.java	7830	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/views/GameView.java	16546	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/views/LeaderboardView.java	5646	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/views/RegistrationView.java	9827	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/views/ShopDialog.java	5442	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/views/View.java	432	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/views/ViewManager.java	1627	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/controllers/Controller.java	884	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/controllers/GameController.java	13630	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/controllers/LeaderboardController.java	1387	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/controllers/Reg	4371	2026. 05. 25.	Java forrásfájl.

istrationController.java			
src/main/java/javallerok/ui/controllers/ShopController.java	3454	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/types/CrossingType.java	252	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/types/HeadType.java	251	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/types/RoadType.java	211	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/types/Vec.java	2387	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/ui/types/VehicleType.java	212	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/shop/CannotBuyItemException.java	96	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/shop/NotEnoughPointsException.java	98	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/shop/ShopAccessDeniedException.java	99	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/shop/SnowPlowShop.java	3982	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/vehicles/Bus.java	4934	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/vehicles/Car.java	5434	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/vehicles/SnowPlow.java	4341	2026. 05. 25.	Java forrásfájl.

src/main/java/javallerok/vehicles/Vehicle.java	8236	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/szkeleton/scripts/Script.java	1567	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/szkeleton/scripts/ScriptManager.java	21925	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/Proto.java	7219	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/AddFuelCommand.java	1649	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/AddMoneyCommand.java	1237	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/BuyCommand.java	1628	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/CleanCommand.java	1512	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/CloseShopCommand.java	502	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/Command.java	847	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/CommandManager.java	2506	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/CreatePlayerCommand.java	3042	2026. 05. 25.	Java forrásfájl.

src/main/java/javallerok/proto/commands/GetTurnCommand.java	542	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/LeaderboardCommand.java	528	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/ListItemsCommand.java	702	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/LoadMapCommand.java	973	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/MoveCommand.java	4022	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/OpenShopCommand.java	993	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/PlaceVehicleCommand.java	3004	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/SetBusRouteCommand.java	2284	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/SetHeadCommand.java	2329	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/SetLaneStateCommand.java	2633	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/	1717	2026. 05. 25.	Java forrásfájl.

SetUpgradeCommand.java			
src/main/java/javallerok/proto/commands/SpawnCarCommand.java	2121	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/StatCommand.java	4066	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/proto/commands/TickCommand.java	441	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/config/MapLayout.java	1169	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/config/MapLayoutParseException.java	365	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/config/MapLayoutReader.java	7032	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/logging/Loggable.java	392	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/logging/SL.java	7006	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/Bridge.java	815	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/BusStation.java	1413	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/Intersection.java	1826	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/Map.java	6924	2026. 05. 25.	Java forrásfájl.

src/main/java/javallerok/gamemap/NPCHomeStation.java	881	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/RoadSection.java	12628	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/Tunnel.java	1025	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstates/ClearState.java	476	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstates/GravelledAgedState.java	530	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstates/GravelledFreshState.java	535	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstates/IcyState.java	465	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstates/RoadState.java	1675	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstates/SaltyState.java	474	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstates/SnowyDenseState.java	506	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstates/SnowyLittleState.java	520	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/gamemap/roadstat	511	2026. 05. 25.	Java forrásfájl.

es/SnowyMediumState.java			
src/main/java/javallerok/core/App.java	8016	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/core/ITickable.java	303	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/core/Player.java	1697	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/core/PlayerFactory.java	2017	2026. 05. 25.	Java forrásfájl.
src/main/java/javallerok/core/TurnController.java	1585	2026. 05. 25.	Java forrásfájl.
src/test/java/Tester.java	4903	2026. 05. 25.	Java forrásfájl.
src/test/java/Tests.java	948	2026. 05. 25.	Java forrásfájl.

13.1.2 Fordítás és telepítés

A program egy batch (Windows) és egy bash (Unix) szkript segítségével is fordítható, amelyek egy JAR fájlba csomagolják azt.

A fordításhoz szükséges, hogy elérhető legyen egy Java 11+ JDK az eszközön, illetve hozzáférés a parancssorhoz. A fordító/csomagoló szkriptek a projektgyökérben található, és onnan kell őket futtatni.

Windows: build.bat

Linux: ./build.sh

Az output JAR fájl a <projektgyökér>/out mappában lesz elérhető.

13.1.3 Futtatás

A program futtatásához szükséges, hogy elérhető legyen egy Java 11+ JRE az eszközön. Mindkét rendszeren a projektgyökérből futtatva:

```
java -jar out/projlab.jar
```

Ezt követően megjelenik egy ablak, ahol lehetőség van a pálya kiválasztására, majd a játékosok felvételére. Az optimális játékelmény érdekében a large nevű pályát ajánljuk.

13.2Értékelés

Tag neve	Tag neptun	Munka százalékban
Rózsa	██████	20
Csordás	██████	20
Fekete	██████	20
Heltai	██████	20
Funk	██████	20

13.3Napló

Kezdet	Időtartam	Résztevők	Leírás
2026.05.13. 14:00:00	0,1	Heltai	Tevékenység: fordítási & futtatási útmutató
2026.05.21. 19:00:00	4	Heltai	Tevékenység: map generálás
2026.05.22. 8:30:00	0,75	Funk	Tevékenység: Toplista mentés, betöltés, osztályok kommentelése.
2026.05.22. 14:00:00	0,5	<i>Mindenki</i>	Értekezlet. Téma: hiányosságok átbeszélése.
2026.05.22. 20:00:00	1	Funk	Tevékenység: Player model bekötése a regisztrálás és topliistába
2026.05.23. 17:00:00	5	Fekete	Tevékenység: Mozgás implementálása, egyéb bugok fixálása
2026.05.24. 18:00:00	4	Csordás	Tevékenység: Út állapot vizualizálás, stun megjelenítés, map betöltő menü
2026.05.23. 17:00:00	3,5	Rózsa	Tevékenység: shop ui
2026.05.25. 10:27:00	0,75	Funk	Tevékenység: kommentelés, összesítés

2026.05.25. 10:56:00	3,5	Rózsa	Tevékenység: kommentelés és játszadozás
2026.05.25. 11:00:00	5	Fekete	Tevékenység: Mozgás javítása, egyéb bugok fixálása
2026.05.25. 11:00:00	4	Csordás	Tevékenység: Mozgás javítása, egyéb bugok fixálása
2026.05.25. 18:00:00	3	Heltai	Tevékenység: dokumentumok készítése & tesztelés

14. Összefoglalás

14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
[REDACTED]	48,02
[REDACTED]	46,80
[REDACTED]	46,70
[REDACTED]	48,25
[REDACTED]	48,42
Összesen	238,29

- **A feltöltött programok forrásainak száma**

Fázis	Kódsorok száma
Szkeleton	3249
Prototípus	6097
Grafikus változat	9026
Összesen	18372

14.2 • Projekt összegzés

14.2.1 Mit tanultak a projektből konkrétan és általában?

UML diagrammok készítése mostmár mindenkinek megmarad egy életre.

14.2.2 Mi volt a legnehezebb és a legkönnyebb?

A csapatmunka nálunk elég könnyen ment (mivel ismertük egymást és ez segített a kollaborációban). A verziókezelést is értelmesen tudtuk használni, mert bizonyos szabályokat felállítottunk és azt többnyire megtartottuk. A legnehezebb mindig az UML diagramok szinkronizálása a kódhoz volt.

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

Igen.

14.2.4 Ha nem, akkor hol okozott ez nehézséget?

14.2.5 Milyen változtatási javaslatuk van?

Érdemes lenne kevesebb időt fordítani a kezdetleges dokumentációra és inkább 2 analízismodell helyett lehetne 2 skeleton vagy 2 proto, mert addigra már valamennyire megváltozik akaratlanul is az analízismodell.

14.2.6 Milyen feladatot ajánlanának a projektre?

- Cicanevelde (régib mobilos játék)
- Valamilyen Tower Defense játék
- tOOOrpedo (torpedó játék)

14.2.7 Egyéb kritika és javaslat

Nagyon sokat kellett nyomtatni. Elég lehetne csak elektronikusan leadni a feladatokat.