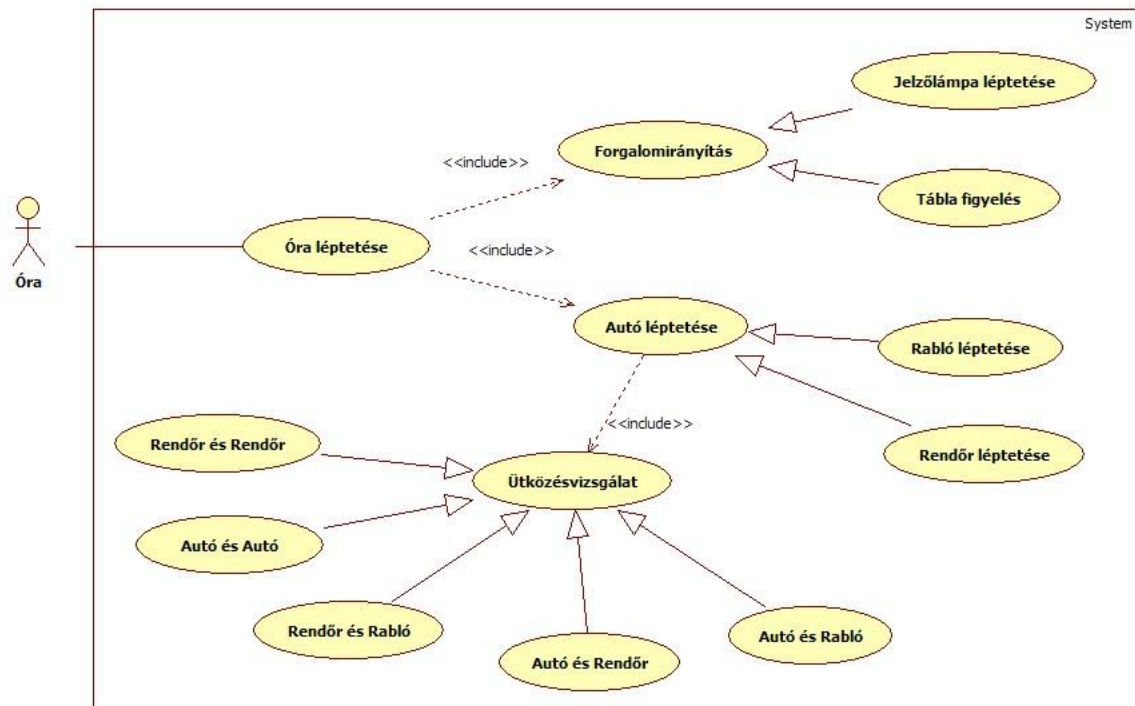


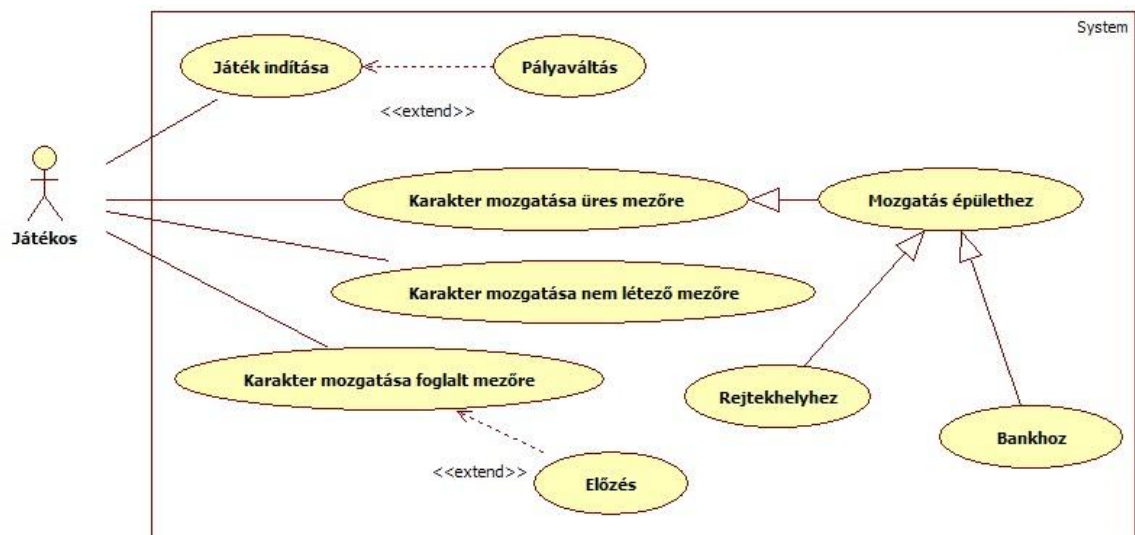
## 5. Szkeleton tervezése

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

#### 5.1.1 Use-case diagram



1. ábra: Az idő múlásával irányított egységek



2. ábra: A játékos által vezérelt egységek

### 5.1.2 Use-case leírások

<b>Use-case neve</b>	Óra léptetése
<b>Rövid leírás</b>	Amit nem a játékos irányít, azt az óra vezérli.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	A játék egy meghatározott idő eltelte után léptet egyet a pálya szereplőin.

<b>Use-case neve</b>	Forgalomirányítás
<b>Rövid leírás</b>	Az útkereszteződésekben forgalomirányítás van.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és az útkereszteződésekben található forgalomirányításnak is jelezzük, hogy új kör van.

<b>Use-case neve</b>	Jelzőlámpa léptetése
<b>Rövid leírás</b>	A jelzőlámpa időnként vált.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és az útkereszteződésekben található jelzőlámpáknak is jelezzük, hogy új kör van. Ezek időnként váltanak.

<b>Use-case neve</b>	Tábla figyelés
<b>Rövid leírás</b>	A tábla jelez, ha be lehet hajtani a kereszteződésbe.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és az útkereszteződésekben található tábláknak is jelezzük, hogy új kör van. A tábla megvizsgálja, hogy szabad-e behajtani a kereszteződésbe.

<b>Use-case neve</b>	Autó léptetése
<b>Rövid leírás</b>	Az utakon autók vannak, ezeket mozgatni kell.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és az autóknak is jelezzük, hogy léphetnek.

<b>Use-case neve</b>	Rabló léptetése
<b>Rövid leírás</b>	A rablóra nem vonatkoznak a szabályok.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és a rablónak is jelezzük, hogy léphetnek. A rablónak nem kell foglalkoznia a szabályokkal.

<b>Use-case neve</b>	Rendőrléptetése
<b>Rövid leírás</b>	A rendőr a rablót keresi, vele ütközni szeretne.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és a rendőröknek is jelezzük ezt. Ha a rendőr előtt rabló halad, akkor ütköznek.

<b>Use-case neve</b>	Ütközésvizsgálat
<b>Rövid leírás</b>	Ha két autó egymás előtt halad, akkor valamit tenni kell.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és ezt tudatjuk az autókkal is. Ők megvizsgálják, hogy ahova lépni szeretnének oda léphetnek-e. Ha nem, akkor különbözőképp reagálnak.

<b>Use-case neve</b>	Rendőr és {Rendőr, Autó, Rabló}
<b>Rövid leírás</b>	Amikor rendőr ütközne valakivel.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és az autók is lépni szeretnének. A rendőr lelassít az előtte haladó rendőr vagy civil sebességére, de a rablóval ütközik.

<b>Use-case neve</b>	Rabló és {Rendőr, Autó}
<b>Rövid leírás</b>	Amikor rabló ütközne valakivel.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és az autók is lépni szeretnének. A rabló megelőzi az előtte haladó járművet.

<b>Use-case neve</b>	Autó és {Autó, Rendőr, Rabló}
<b>Rövid leírás</b>	Amikor két civil ütközne.
<b>Aktorok</b>	Óra
<b>Forgatókönyv</b>	Az óra lép egyet, és az autók is lépni szeretnének. Az autó minden esetben lelassít az előtte haladó sebességére.

<b>Use-case neve</b>	Játék indítása
<b>Rövid leírás</b>	A játékos új játékot kezd.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos futtatja a programot, majd a menüpontok közül kiválasztja a startot.

<b>Use-case neve</b>	Pályaváltás
<b>Rövid leírás</b>	A játékos pályát választ.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos elindítja a játékot, majd pályát választ.

<b>Use-case neve</b>	Karakter mozgatása üres mezőre
<b>Rövid leírás</b>	A rabló szabad útdarabkára lép.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos olyan útdarabkára irányítja a rablót, amin nem áll más autó.

<b>Use-case neve</b>	Mozgatás épülethez
<b>Rövid leírás</b>	A rabló épülethez ér.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos olyan útdarabkára irányítja a rablót, ahol épület is található.

<b>Use-case neve</b>	Rejtekhelyhez
<b>Rövid leírás</b>	A rabló győz.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos a rejtekhelyet tartalmazó útdarabkára irányítja a rablót. A pályának vége, a rabló győzött.

<b>Use-case neve</b>	Bankhoz
<b>Rövid leírás</b>	A rabló kirabolja a bankot.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos a bankot tartalmazó útdarabkára irányítja a játékost. A rabló kirabolja a bankot.

<b>Use-case neve</b>	Karakter mozgatása nem létező mezőre
<b>Rövid leírás</b>	A rabló ki akar menni a városhatárról.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos olyan mezőre akar lépni, amely nem létezik. A rabló nem mehet ki.

<b>Use-case neve</b>	Karakter mozgatása foglalt mezőre
<b>Rövid leírás</b>	A rabló olyan mezőre szeretne lépni, ahol áll egy autó.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos olyan mezőre akar lépni, ahol valamilyen autó áll. Rabló esetén ilyenkor előzés történik.

<b>Use-case neve</b>	Előzés
<b>Rövid leírás</b>	A rabló megelőzi az előtte haladót.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	A játékos olyan mezőre lépne ahol autó áll, ezért megelőzi azt.

## 5.2 Architektúra.

A szkeleton verzió arra hivatott, hogy ellenőrizni lehessen vele a korábban definiált szekvencia diagramok, és use-case modellek működését. Az egyes forgatókönyveket előre elkészített tesztpályákon lehet kipróbálni, melyet a tesztelő felhasználó tud kiválasztani a programból. A program ezen verziója egyszálú, az egyes utasítások, illetve függvényhívások egymás után hívódnak meg.

A teszteléshez létrehoztunk egy osztályt, mely a felhasználóval való kommunikációt biztosítja számunkra. Az egymással analóg eseteket egyazon osztályba soroltuk, a könnyebb áttekinthetőség végett. Az alább található leírásokban olvasható, hogy az egyes pályák mely eseteket mutatják be, illetve mi található rajtuk.

A csapat minden pályán csak a minimálisan szükséges objektumpéldányosításokat végezte el, hiszen most nem a játszhatóságot vagyunk hivatottak bemutatni, hanem az egyes objektumok közötti kommunikációt.

### 1. tesztpálya:

- Bemutatja egy jelzőlámpás, vagy táblás útkereszteződés működését.
- A pályán található egy 4 ágú útkereszteződés, melynek minden pontjához csatlakozik egy útdarabka. A felhasználó döntése szerint helyezkednek el az autók rajtuk, illetve szintén az ő döntése, hogy milyen típusú forgalomirányítás van a kereszteződésben.

### 2. tesztpálya:

- Bemutatja, hogy mi történik, mikor a rabló eléri a rejtekhelyet vagy a bankot.
- A pályán található két útdarabka, ebből az egyikhez csatlakozik a rejtekhely vagy a bank a felhasználó választása szerint.

### 3. tesztpálya:

- Bemutat egy egyszerű léptetést, előzést, lassítást, vagy ütközést a lehelyezett autók függvényében.
- A pályán 3 útdarabka található, ezek egymás után pakolva. Az első két útdarabkán a felhasználó által megadott autók állnak, útdarabkánként maximum 1 darab. A harmadik útdarab üres, nem áll rajta autó.

### 5.3 A szkeleton kezelői felületének terve, dialógusok

A szkeleton, mint program célja, hogy bemutassa a belső működést, illetve a felvázolt modell működőképességét. Ebben a programban minden objektum szerepel, de azoknak jobbra csak az interfésze definiált. Minden metódushíváskor kiíródik a képernyőre az őt hívó objektum típusa, azonosítója, illetve a saját (és osztályának) neve, paraméterlistája. Ezután a metódus meghívja a működéshez szükséges további metódusokat. Minden olyan pálya indítása előtt, ahol többféle szituáció állhat elő, a program bekéri a felhasználótól a szituációt jellemző adatokat a konzolon. Ha az adott pálya lejátszása közben dönteni kell, akkor a program a felhasználótól kéri azt be egy egyszerű, eldöntendő kérdés formájában. A bekérések alatt a lejátszás szünetel, hiszen csak a kérdésre adott válasz alapján folytatódhat a program. Ha a felhasználó válasza értelmezhetetlen vagy szabálytalan (pl. 2 rablót akar betenni egy szituációba), akkor egy hibaüzenet jelenik meg, majd a program megismétli a kérdést.

A szkeletonnak alkalmasnak kell lennie a szekvencia diagramok ellenőrzésére. Ezt biztosítja a karakteres felület áttekinthetősége, illetve egyszerűsége. A különféle formátumok alatt olvashatóak.

A kiírás a következő formátumban történik **metódushívás** esetén:

<b>city.car[0]</b>	<b>: Robber</b>	<b>##CALL##</b>	<b>city.road[0].setCar(city.car[0])</b>
hívó azonosítója	hívó típusa	ez egy hívás	a hívott, és paraméterei

A kiírás a következő formátumban történik **példányosítás** esetén:

<b>city</b>	<b>: City</b>	<b>##CREATE##</b>	<b>city.car[0]</b>	<b>: Robber</b>
hívó azonosítója	hívó típusa	példányosítás	a példány neve	és típusa

A kiírás a következő formátumban történik metódusból **visszatérés** esetén:

<b>city.road[1].getBuilding()</b>	<b>##RETURN##</b>	<b>city.building[1]</b>	<b>: Hideout</b>
metódus azonosítója	visszatérés	visszatérési érték	és típusa

A kiírás a következő formátumban történik **kérdés** feltevése esetén:

<b>##QUESTION##</b>	<b>Do you want to step?</b>	<b>(Y/N)</b>	<b>Y</b>
ez egy kérdés	kérdés szövege	válaszlehetőségek	tesztelő válasza

A kiírás a következő formátumban történik **hiba** közlése esetén:

<b>##ERROR##</b>	<b>You can have only one robber.</b>
ez egy hiba	üzenet

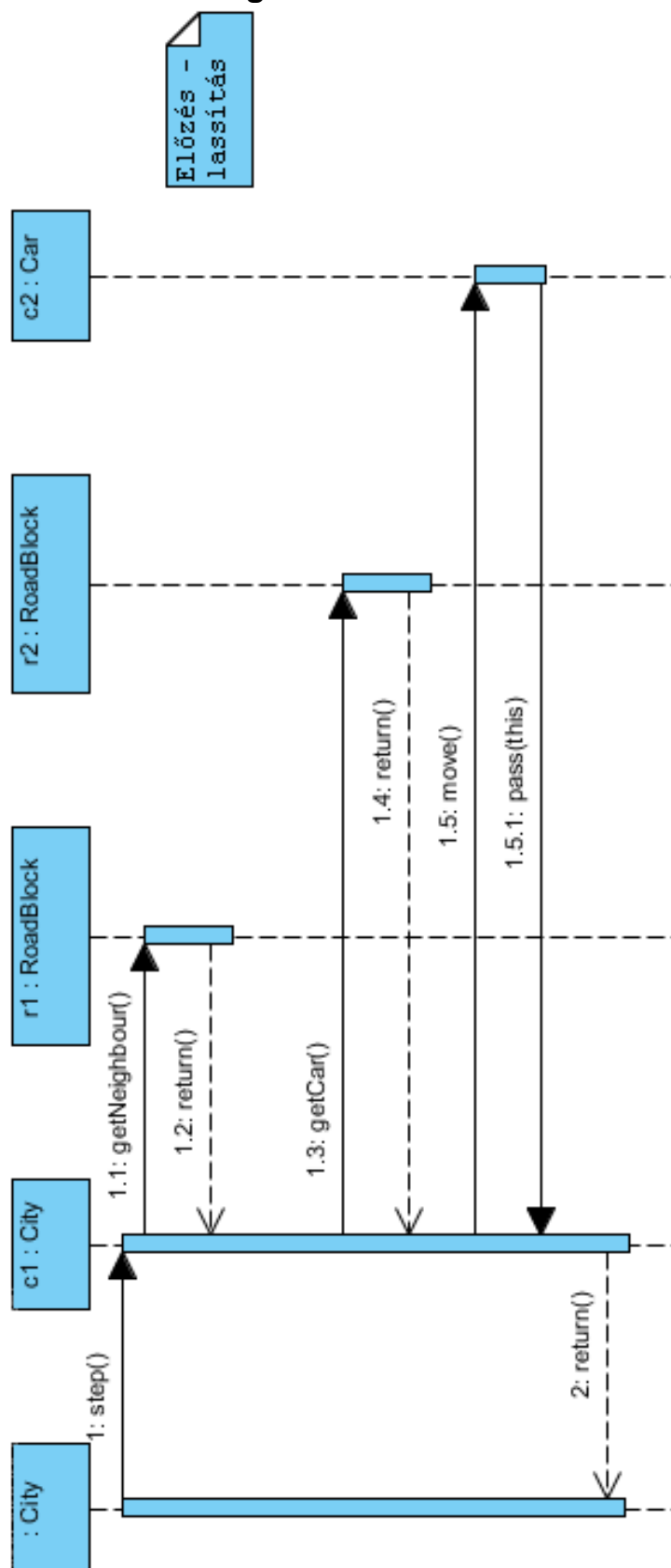
A következő oldalon egy példa log olvasható.

**Példa:**

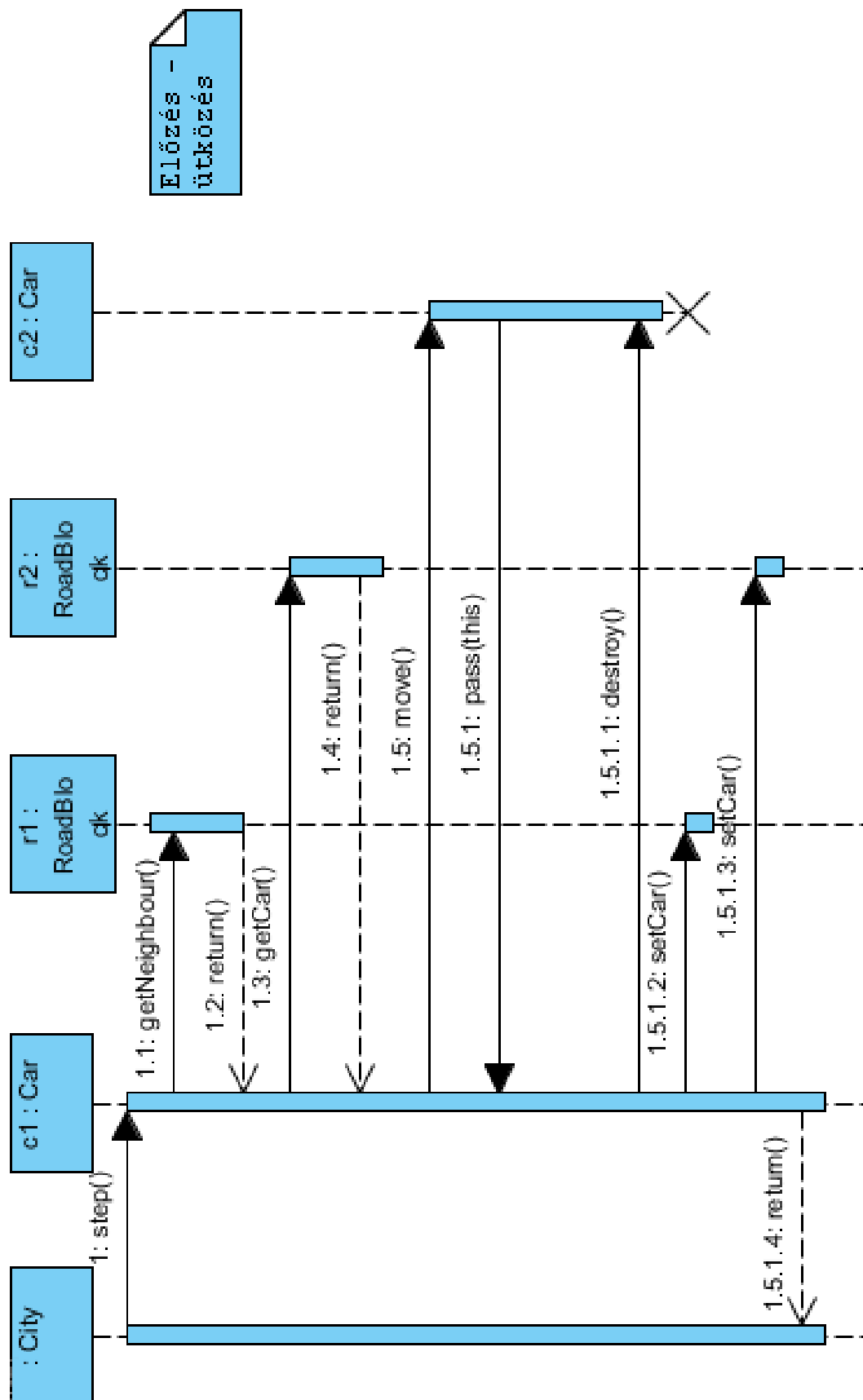
```
##QUESTION## Do you want to step? (Y/N) Y

city: City ##CALL## car[0].step()
city.car[0] : Robber ##CALL## city.road[0].getNeighbour()
city.road[0].getNeighbour() ##RETURN## city.road[1] : RoadBlock
city.car[0] : Robber ##CALL## city.road[1].getCar()
city.road[1].getCar() ##RETURN## NULL
city.car[0] : Robber ##CALL## city.road[0].setCar(NULL)
city.car[0] : Robber ##CALL## city.road[1].setCar(city.car[0])
city.car[0] : Robber ##CALL## city.road[1].getTraffic()
city.road[1].getTraffic() ##RETURN## NULL
city.car[0] : Robber ##CALL## city.road[1].getBuilding()
city.road[1].getBuilding() ##RETURN## city.building[1] : Hideout
```

### 5.4 Szekvencia diagramok a belső működésre







## 5.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2010.03.08. 18:00	0.5 óra	Boros Takács Rapp	Döntés: Takács elkészíti a játékos use-caset, az architektúrát, és a felület tervet. Döntés: Boros elkészíti a belső szekvenciadiagrammokat. Döntés: Rapp elkészíti az óra use-caset.
2010.03.09 20:30	3 óra	Takács	Elkészíti az architektúrát, felület tervet és dialógusokat
2010.03.10. 20:30	1.5 óra	Rapp	Elkészíti a use-case diagramokat
2010.03.10. 22:30	1.5 óra	Takács	Kiegészíti, javítja a use-case diagramokat
2010.03.11. 00:00	1.5 óra	Boros	Elkészíti a szekvencia diagramokat, formázza a dokumentumot