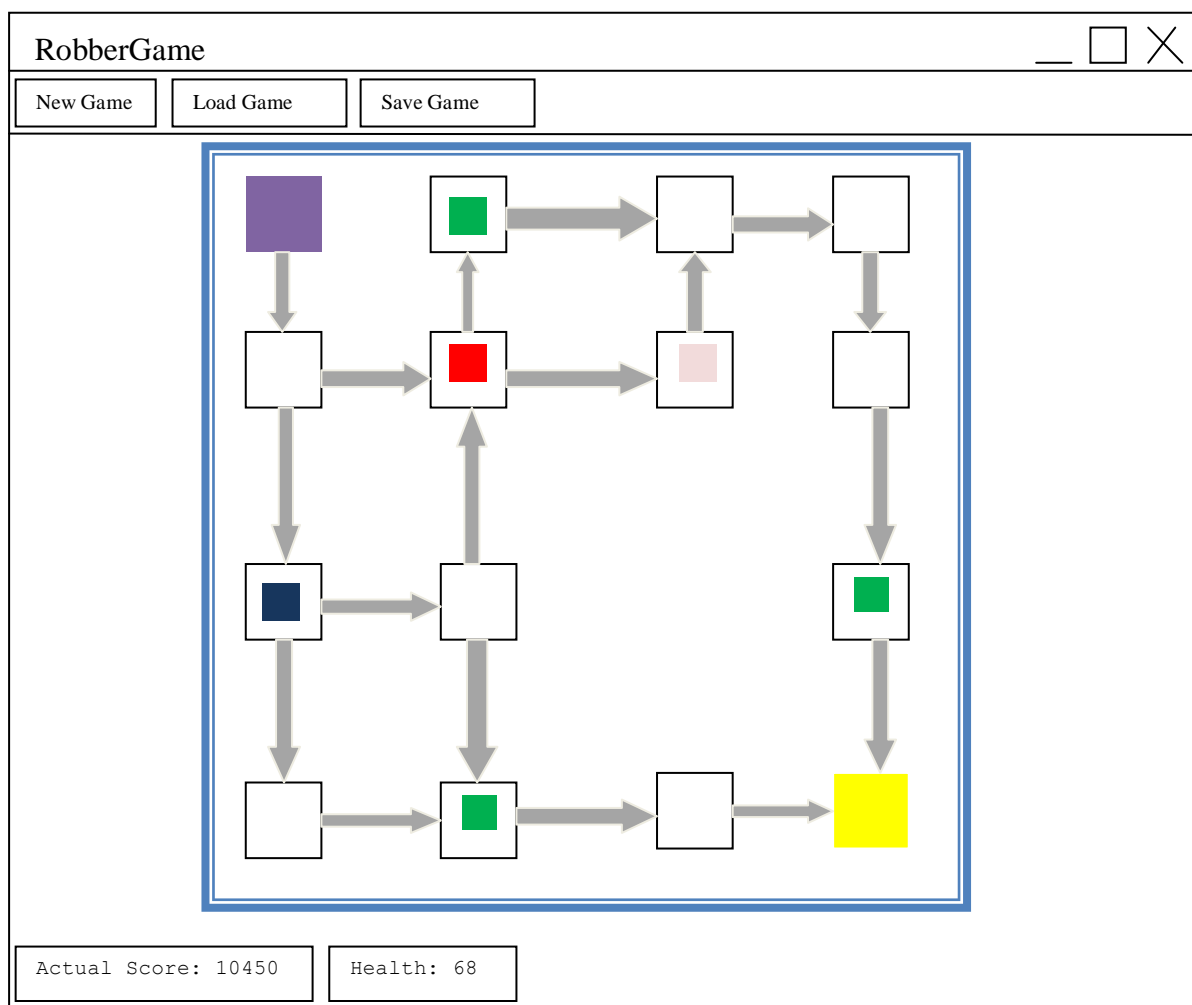


## 11. Grafikus felület specifikációja

### 11.1 A grafikus interfész



#### Jelmagyarázat a képernyőképhez:

A különböző objektumok (épületek, autók, stb..) különböző színnel jelennek meg a játékban:

- rendőr: kék
- rabló: piros
- út: szürke
- nyuszi: rózsaszín
- bank: lila
- rejtékhely: sárga
- civil autók: zöld

## **11.2 A grafikus rendszer architektúrája**

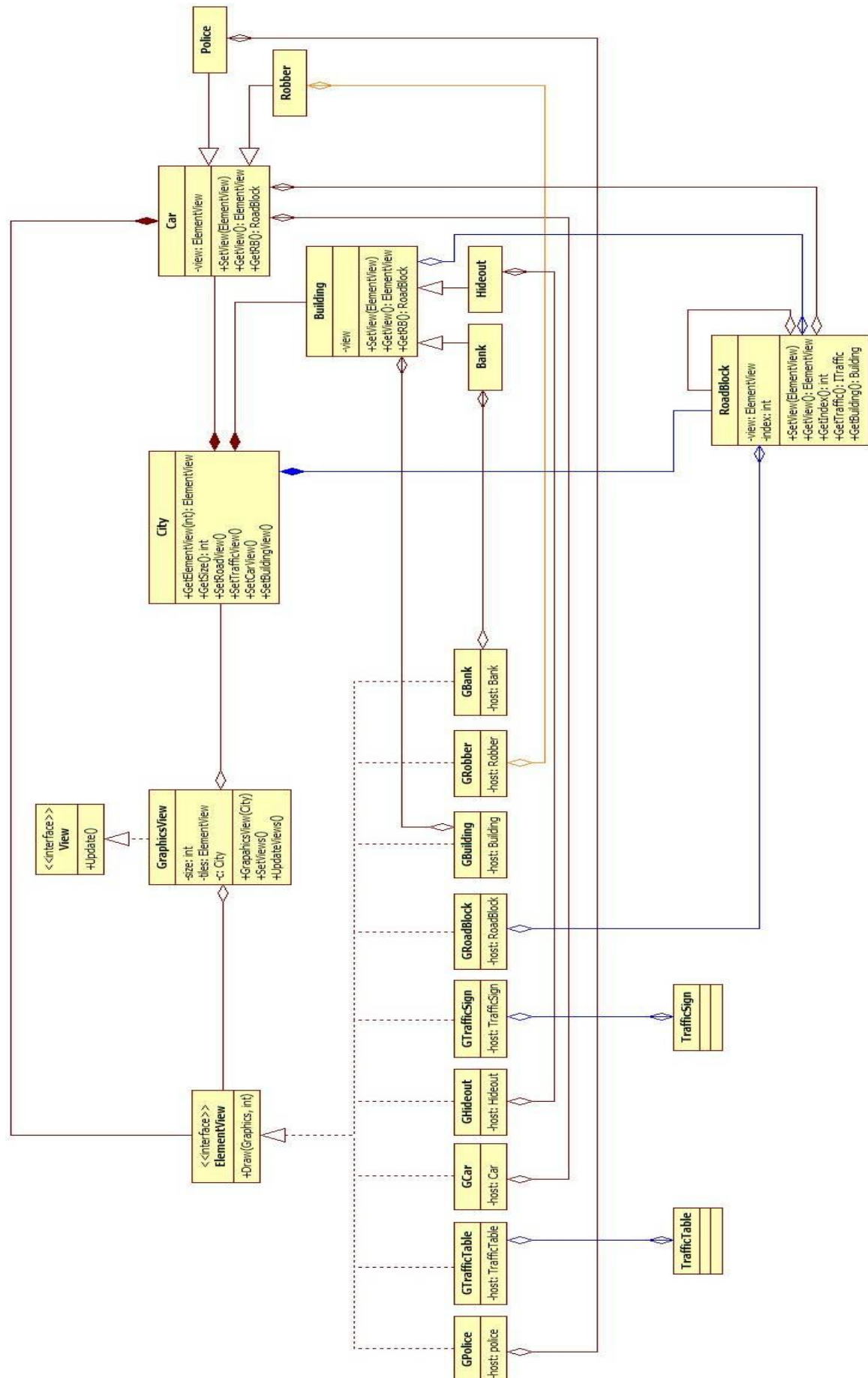
### **11.2.1 A felület működési elve**

A program grafikus változatának elkészítésekor a Model-View-Controller tervezési mintát használjuk. Ezzel el tudjuk választani a program játék logikáját a grafikus megjelenítéstől, ezzel eleget téve a követelményeknek, valamint a program egyes részeinek karbantartása is könnyebbé válik. A grafikus felület mindig frissíteni fogja magát az aktuális játékállapothoz, viszont az egyes elemek saját magukat fogják kirajzolni, így a kevert alapelveket részesítettük előnyben.

### **11.2.2 A felület osztály-struktúrája**

Sajnos utólag kiderült, hogy a prototípus osztályait a megfelelő grafikus felület implementálásához módosítani kell. A módosítás új tagváltozók és tagfüggvények hozzáadásában merül ki, a meglévőket nem módosítottuk. Az alábbi osztálydiagramon a grafikus felülettel kiegészült programváz látható, a régi osztályok esetében csak a módosításokat tüntettük fel.

Ahol a vonalak keresztezik egymást, ott a félreértéseket elkerülvén különböző színekkel jeöltük az érintett vonalakat. Fekete-Fehér nyomtatásban sajnos ez nem látszik.



## 11.3 A grafikus objektumok felsorolása

### 11.3.1 View

- **Felelősség**

Interfészt nyújt a megjelenítő felületek számára.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**

- **+ void Update():** A felület frissítését végzi el.

### 11.3.2 GraphicsView

- **Felelősség**

A játéklogika grafikus felületre való leképezésért felelős.

- **Ősosztályok**

Nincs.

- **Interfészek**

View

- **Attribútumok**

- **- size: int:** A pálya méretét tárolja, a grafikus ablak méretét határozzuk meg belőle.
- **-c: City:** Referenciát tárol a megjelenített városra.
- **-tiles: ElementView:** Ide kérjük le az aktuálisan kirajzolandó objektum grafikus változatát.

- **Metódusok**

- **+GraphicsView(city: City):** Az osztály konstruktora, mely inicializálja a grafikus interfészt is egyben.
- **+void SetViews():** Beállítja az egyes objektumok megjelenését. Megadja, hogy ki hogyan rajzolja ki magát.
- **+void UpdateViews():** Frissíti az egyes objektumok (esetünkben csak a Car típusúak, mert csak ezek változnak) megjelenését. Pl. ha jött egy új autó akkor ad neki megjelenést is.

### 11.3.3 ElementView

- **Felelősség**

Interfészt nyújt a megjeleníthető objektumok kirajzolásához.

- **Ősosztályok**

Nincs.

- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**

- **+ void Draw(Graphics, int):** Kirajzolja az objektum magát az első paraméterben kapott felületre, mely a második paraméterben megadott méretű (utóbbiból számolja ki, hogy pontosan milyen koordinátákba rajzoljon).

### 11.3.4 GPolice, GTrafficTable, GCar, GRobber, GRoadBlock, GHideout, GBank, GBuilding, GTrafficSign

- **Megjegyzés:** A fentieket összevontuk, mert a leírásuk egyezik. Mindössze a Draw() metódus törzse tér el annyiban, hogy más típusú objektumokat hív. A mellékelt szekvencia diagramon ez látható is.

- **Felelősség**

Az egyes objektumok grafikus változatának kirajzolása a képernyőre.

- **Ősosztályok**

Nincs.

- **Interfészek**

ElementView

- **Attribútumok**

- **- host:** Referenciát tárol arra az objektum példányra, akit kirajzol. Típusát szándékosan nem jelöltük az összevonás miatt, hiszen logikusan Car, Police, Building, Robber, Bank, RoadBlock, stb... referenciát tárol attól függően, hogy éppen melyik változatot nézzük (GCar, GPolice, GBuilding, stb...).

- **Metódusok**

- **+void Draw(Graphics, int):** Lásd: ElementView. A rajzolás után előfordulhat, hogy megvizsgáljuk van-e valamilyen tartalmazott objektum (pl. RoadBlock esetében egy az adott útdarabon álló autó, vagy jelzőlámpa, épület, stb...) és azokat is kirajzoljuk a megfelelő Draw() metódusok meghívásával.

### 11.3.5 Módosult: City

- **Új Metódusok**
  - **+ElementView GetElementView(int):** Visszatér az adott sorszámú RoadBlock ElementView típusú tagváltozójával.
  - **+int GetSize():** Visszaadja a pálya méretét.
  - **+void SetRoadView():** A road tömb elemein végiglépkedve beállítja az összes RoadBlock megjelenését a SetView() metódus meghívásával.
  - **+void SetTrafficView():** A traffic tömb elemein végiglépkedve beállítja az összes ITraffic interfészt implementáló objektum megjelenését a SetView() metódus meghívásával.
  - **+void SetCarView:** A car tömb elemein végiglépkedve beállítja az összes Car megjelenését a SetView() metódus meghívásával.
  - **+void SetBuildingView:** A building tömb elemein végiglépkedve beállítja az összes Building megjelenését a SetView() metódus meghívásával.

### 11.3.6 Módosult: Car

- **Új Tagváltozók**
  - **-ElementView view:** Az aktuális objektumra vonatkozó kirajzolásért felelős objektumot tárolja.
- **Új Metódusok**
  - **+ElementView GetView():** A view tagváltozót adja vissza.
  - **+void SetView(ElementView):** Beállítja a view tagváltozót.
  - **+RoadBlock GetRB():** Visszaadja, hogy az útdarabot, amelyen az autó épp áll.

### 11.3.7 Módosult: Building

- **Új Tagváltozók**
  - **-ElementView view:** Az aktuális objektumra vonatkozó kirajzolásért felelős objektumot tárolja.
- **Új Metódusok**
  - **+ElementView GetView():** A view tagváltozót adja vissza.
  - **+void SetView(ElementView):** Beállítja a view tagváltozót.
  - **+RoadBlock GetRB():** Visszaadja, hogy az útdarabot, amelyen az épület áll.

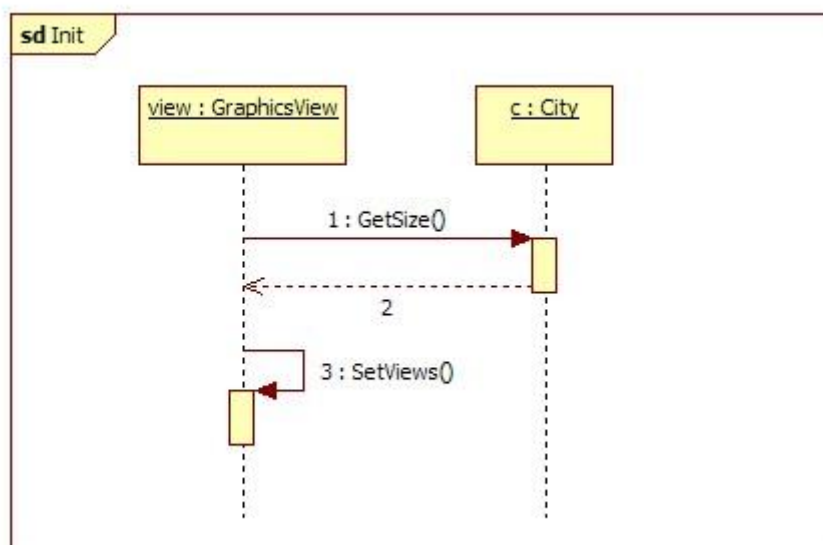
### 11.3.8 Módosult: RoadBlock

- **Új Tagváltozók**
  - **-ElementView view:** Az aktuális objektumra vonatkozó kirajzolásért felelős objektumot tárolja.
  - **-int index:** Az útdarab sorszámát tárolja.

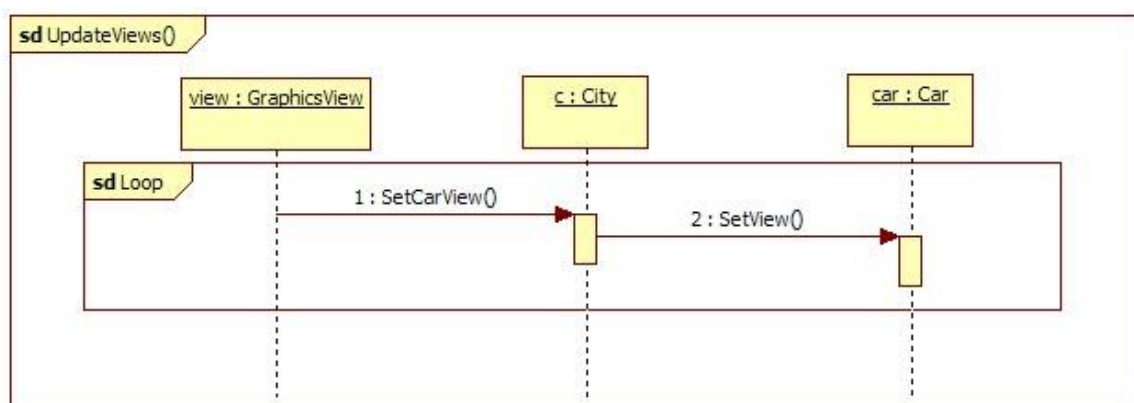
- **Új Metódusok**

- **+ElementView GetView():** A view tagváltozót adja vissza.
- **+void SetView(ElementView):** Beállítja a view tagváltozót.
- **+int GetIndex():** Visszaadja az index tagváltozó értékét.
- **+ITraffic GetTraffic():** Visszaadja a forgalomirányítót, ha van.
- **+Building GetBuilding():** Visszaadja az épületet, ha van.

### 11.4 Kapcsolat az alkalmazói rendszerrel



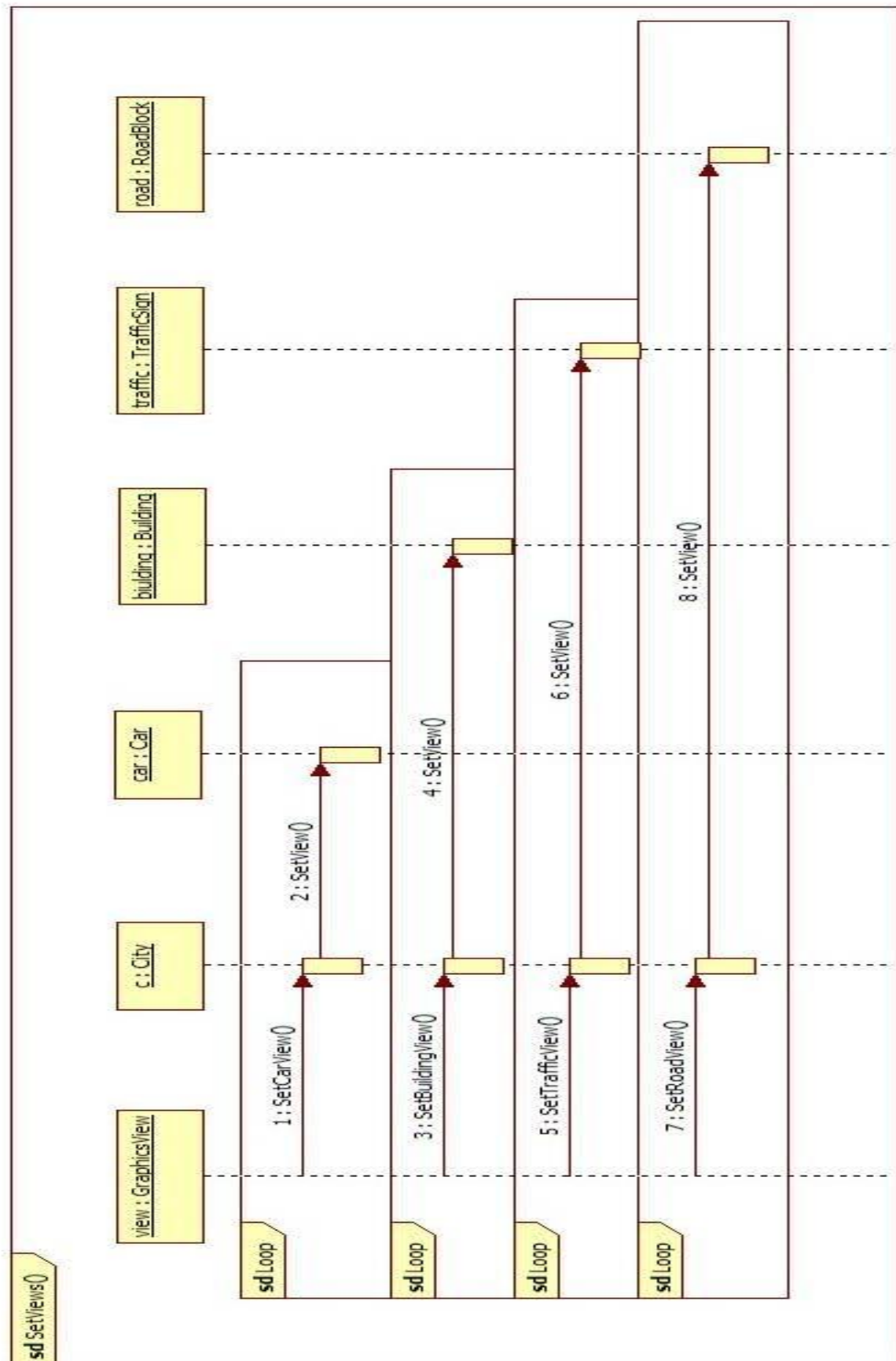
1. ábra: Grafikus felület inicializálása



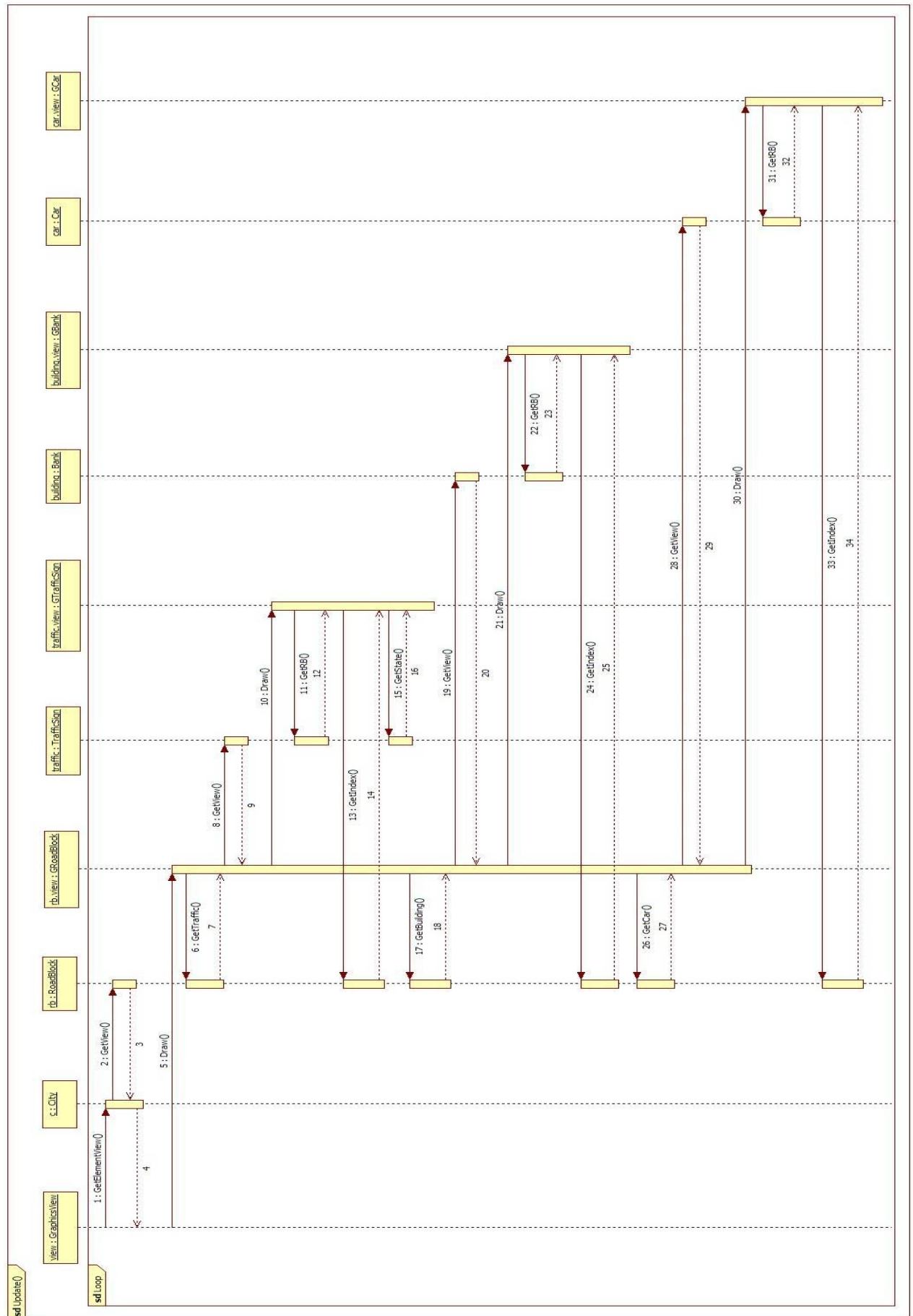
2. ábra: Grafikus felület nézeteinek frissítése

A következő két ábra sorrendben:

- Nézetek beállítása.
- Grafikus felület frissítése, újrarajzolása.







**11.5 Napló**

<b>Kezdet</b>	<b>Időtartam</b>	<b>Résztevők</b>	<b>Leírás</b>
2010.04.19. 20:00	1,5 óra	Rapp Takács	Értekezlet. A grafikus felület elvi felépítésének megbeszélése.
2010.04.21. 20:00	6 óra	Rapp	Tevékenység: 11.1 és 11.2 elkészítése, alapvető célok, elvárások, elvek rögzítése.
2010.04.21. 20:00	4 óra	Takács	Tevékenység: Az osztálydiagram megtervezése, szükséges módosítási pontok felmérése.
2010.04.22. 8:00	4 óra	Takács	Tevékenység: A végleges osztálydiagram elkészítése, objektumok leírása, módosítási pontok dokumentálása, szekvencia diagramok készítése.
2010.04.22 12:00	1 óra	Takács	Tevékenység: A dokumentum formázása, helyesírási hibák javítása, ellenőrzés.