

3. Analízis modell kidolgozása

3.1 Objektum katalógus

Bank
Building
Car
City
CityElement
Hideout
Intersection
Police
RoadBlock
Robber

3.1.1 Bank

Innen indul a bankrabló, a megszerzett zsákmánnyal.

3.1.2 Building

A városban található egyéb épületek.

3.1.3 Car

A városban közlekedő autók. Az ő felelőssége nyilvántartani az életét, sebességét, és eldönteni, merre szeretne továbbmenni.

3.1.4 City

Város objektum. Ő tartalmazza a várost alkotó „dolgokat” (járművek, épületek, utak, csomópontok). Ő felelőssége a pálya generálása, és a „léptetés”, amivel a világot értesíti az idő múlásáról.

3.1.5 CityElement

Városban lévő „dolgok” gyűjtőhelye. Ebből fűzünk listát. Azért kell, hogy egy helyen tudjuk tárolni az objektumainkat, és egyszerre tudjuk őket értesíteni az idő múlásáról.

3.1.7 Hideout

Bankrabló rejtékhelye. Ha a bankrabló beér, megnyeri a kört.

3.1.9 Intersection

Kereszteződés. Összekapcsolja az utakat, épületeket. Megmondja az autónak, merre felé lehet belőle menni.

3.1.10 Police

Rendőrk. Ő üldözi a rablót, ha utoléri, elkapja.

3.1.11 RoadBlock

A rács, amiből az út felépül.

3.1.12 Robber

A rabló. Őt lehet irányítani a városban egyedül.

3.2 Osztályok leírása

3.2.1 Bank

A bank épülete, a kiindulópontot jelenti.

- **Felelősség**
- **Ősosztályok**
Building
- **Interfészek**
nincs
- **Attribútumok**
nincs
- **Metódusok**
nincs

3.2.2 Building

Ősosztály az épületek számára.

- **Felelősség**
- **Ősosztályok**
nincs
- **Interfészek**
nincs
- **Attribútumok**
nincs
- **Metódusok**
nincs

3.2.3 Car

A városban haladó autók.

- **Felelősség**
A haladás mértékének meghatározása a sebessége alapján. A haladási irányának eldöntése.
- **Ősosztályok**
Vehicle
- **Interfészek**
nincs
- **Attribútumok**
 - **int speed** : a jármű sebességét jelenti, mennyi útegységet tesz meg időegység alatt
- **Metódusok**
 - **int getSpeed()** : visszaadja a speed attribútum értékét
 - **void setSpeed(s : int)** : beállítja a speed attribútum értékét

3.2.4 City

A várost reprezentáló objektum.

- **Felelősség**
Nyilvántartani a várost felépítő utakat, épületeket és a városban haladó autókat. A forgalmat szabályozni.
- **Ősosztályok**
nincs
- **Interfészek**
nincs
- **Attribútumok**
 - **LinkedList<CityElement> cityElements** : a várost alkotó utak, épületek tárolása
 - **LinkedList<Vehicle> vehicles** : a városban haladó autók tárolása
- **Metódusok**
 - **void Step()** a városban mozgó objektumok mozgatásért felel

3.2.5 CityElement

A várost felépítő objektumok ősosztálya.

- **Felelősség**
Informálja az autókat a közlekedési viszonyokról, tárolja az autók pozícióját. Előzéseknel meghívja az autók megfelelő függvényeit.

- **Ősosztályok**
nincs
- **Interfészek**
nincs
- **Attribútumok**
 - **Vehicle& vehicle** : a rajta lévő autóra mutató referencia
- **Metódusok**
 - **void Step()** : a mozgásért felelős függvény (lámpák változása)

3.2.6 Hideout

A cél pozíció, a rabló menedékhelye.

- **Felelősség**
nincs
- **Ősosztályok**
Building
- **Interfészek**
nincs
- **Attribútumok**
nincs
- **Metódusok**
nincs

3.2.7 Intersection

A kereszteződést reprezentáló osztály.

- **Felelősség**
Megadni a lehetséges kimeneti utak irányát az autónak.
- **Ősosztályok**
RoadBlock
- **Interfészek**
nincs
- **Attribútumok**
 - **CityElement& up** : a felfele irányban található útra mutató referencia
 - **CityElement& down** : a lefele irányban található útra mutató referencia

- **Metódusok**
 - `void get()`

3.2.8 Police

A városban haladó rendőrautók.

- **Felelősség**
Ha rabló próbál mellette elhaladni elkapja.
- **Ősosztályok**
Car
- **Interfészek**
nincs
- **Attribútumok**
nincs
- **Metódusok**
 - `void isPassed(Robber r)` : Ha a mellette elhaladni akaró autó egy rabló akkor elkapja.

3.2.9 RoadBlock

Az útelemeket reprezentáló osztály.

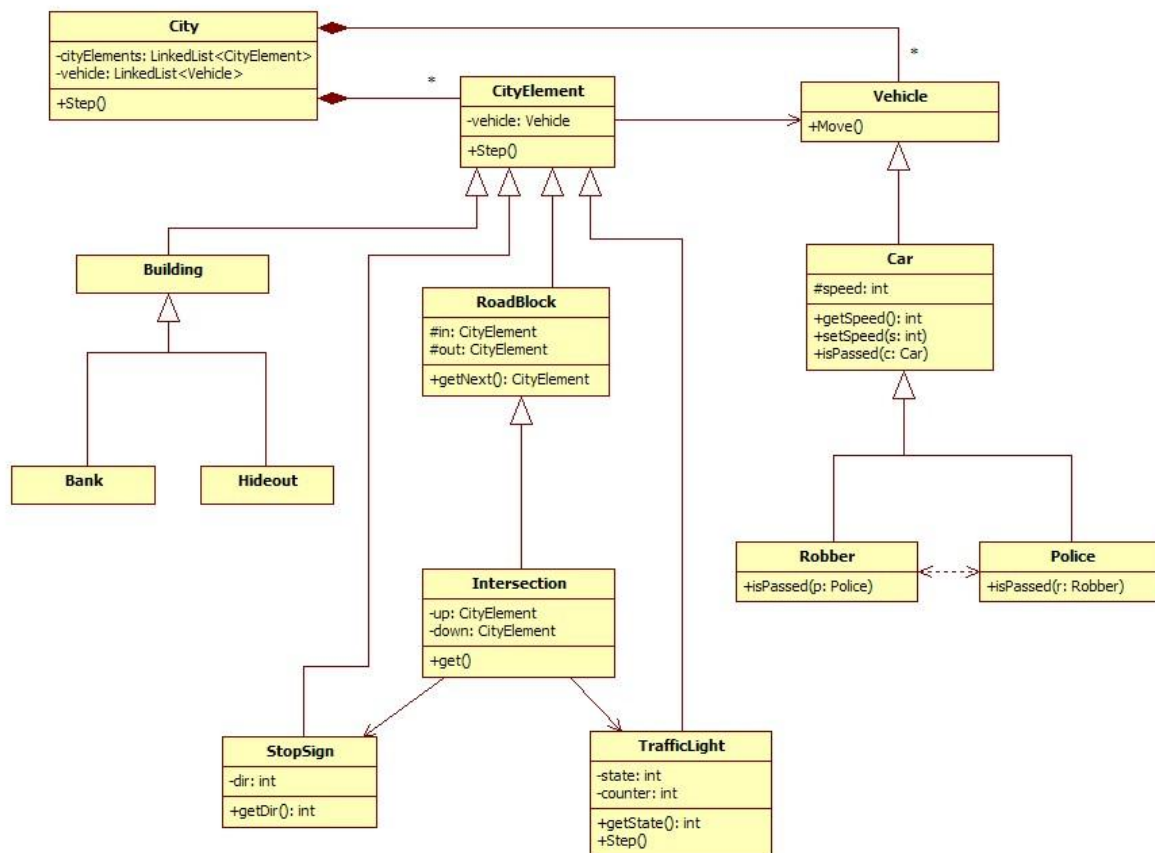
- **Felelősség**
Figyelni a rajta haladó autókat.
- **Ősosztályok**
CityElement
- **Interfészek**
- **Attribútumok**
 - `CityElement& in` : az előző útelemre mutató referencia
 - `CityElement& out` : az előző útelemre mutató referencia
- **Metódusok**
 - `CityElement getNext()` : megadja a következő útelemet

3.2.10 Robber

A rabló osztály a játékos irányítja.

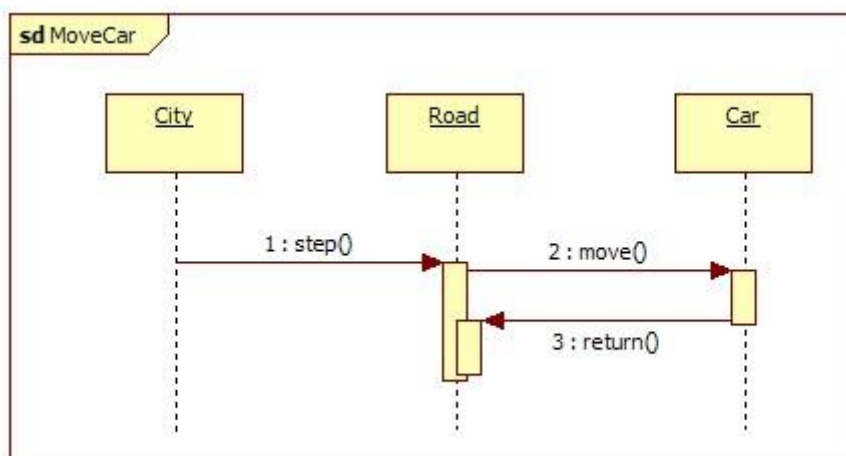
- **Felelősség**
- **Ősosztályok**
Car
- **Interfészek**
nincs
- **Attribútumok**
nincs
- **Metódusok**
 - **isPassed(Police p)** : Ha a mellette elhaladó autó egy rendőr akkor az elkapja

3.3 Statikus struktúra diagramok

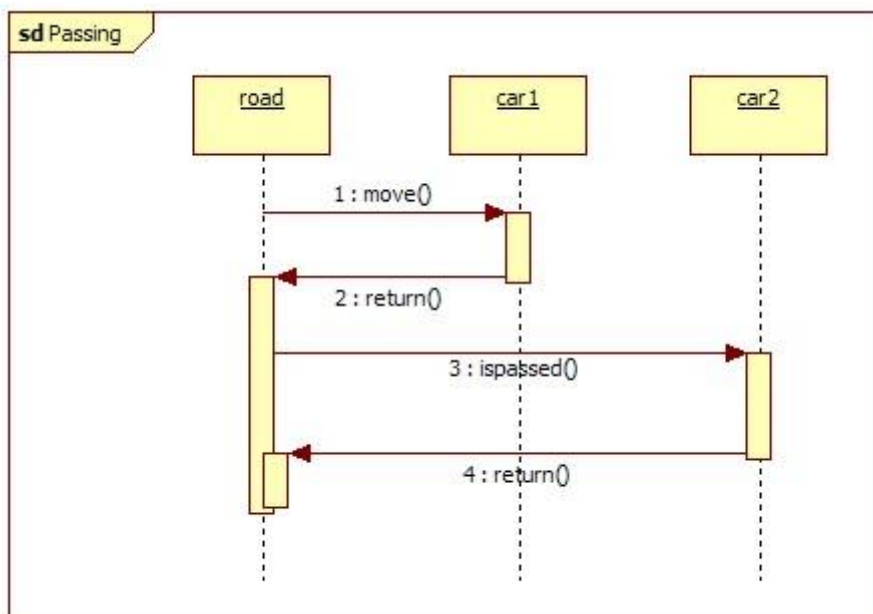


1. ábra: A játék osztálydiagramja

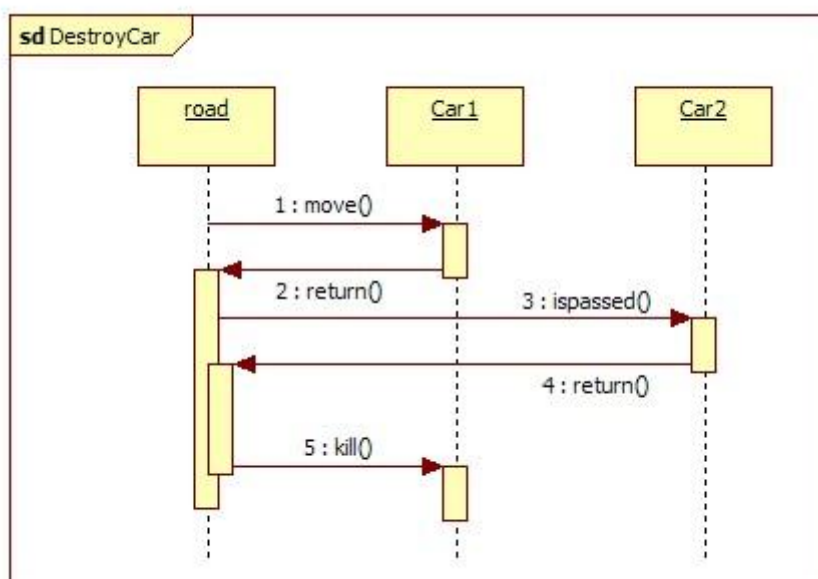
3.4 Szekvencia diagramok



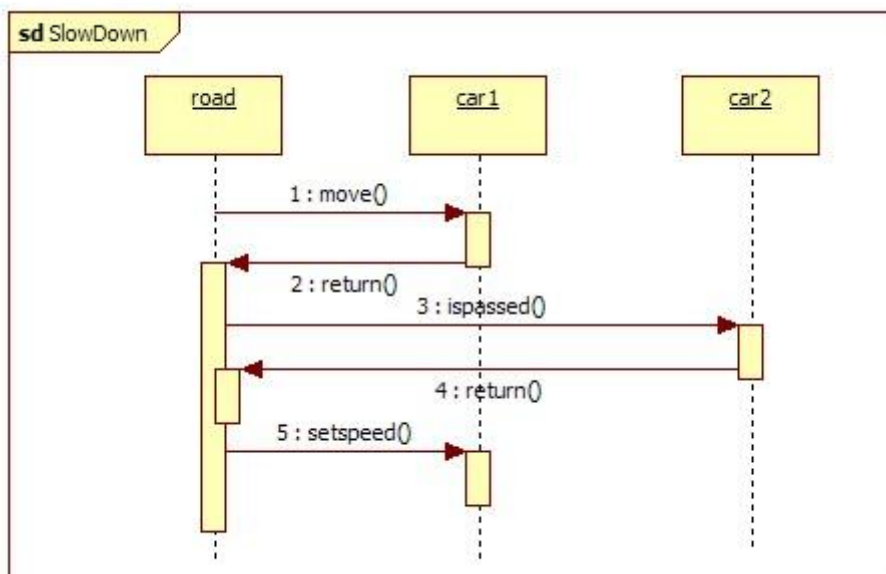
2. ábra: Szekvencia diagram autó mozgására



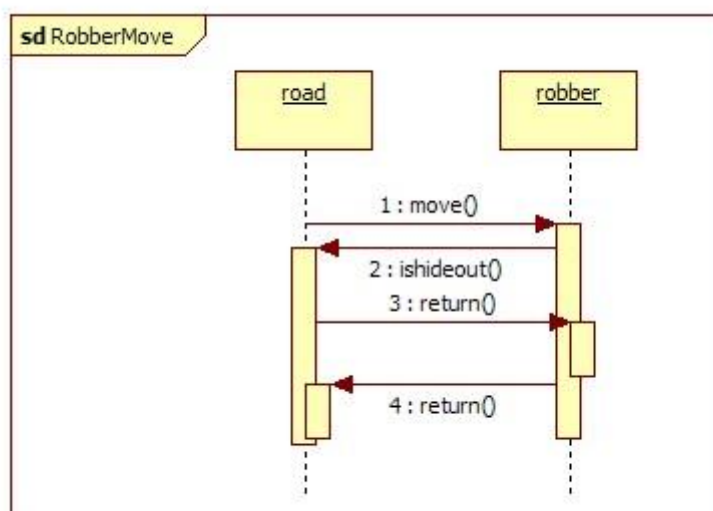
3. ábra: Szekvencia diagram előzésre



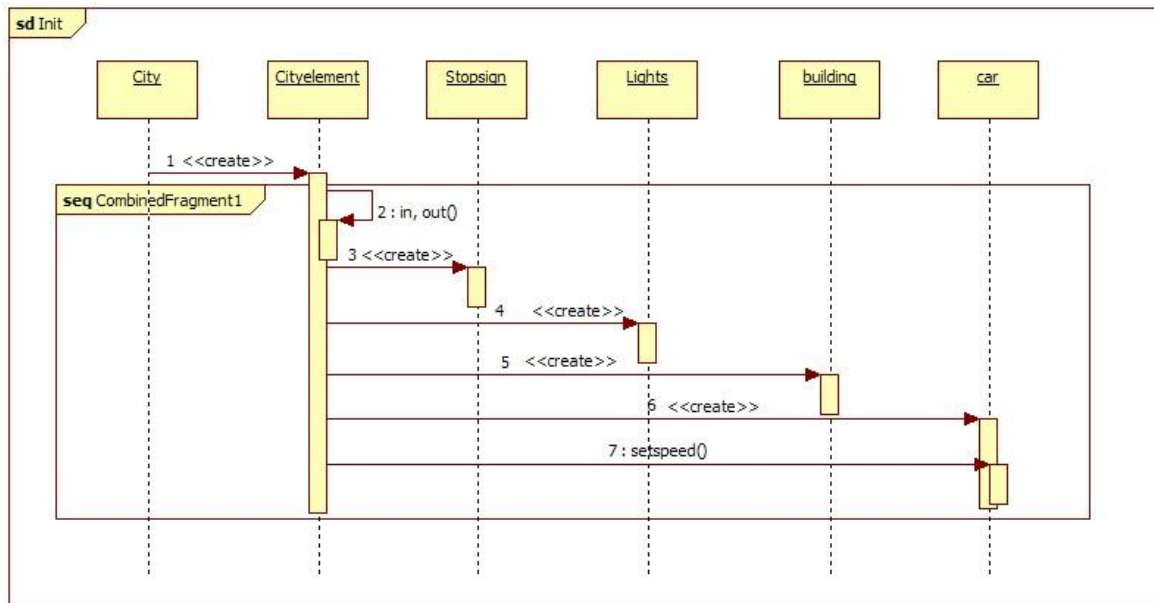
4. ábra: Szekvencia diagram autó megsemmisítésre (pl. egy rendőr elkapja a rablót)



5. ábra: Szekvencia diagram lassításra

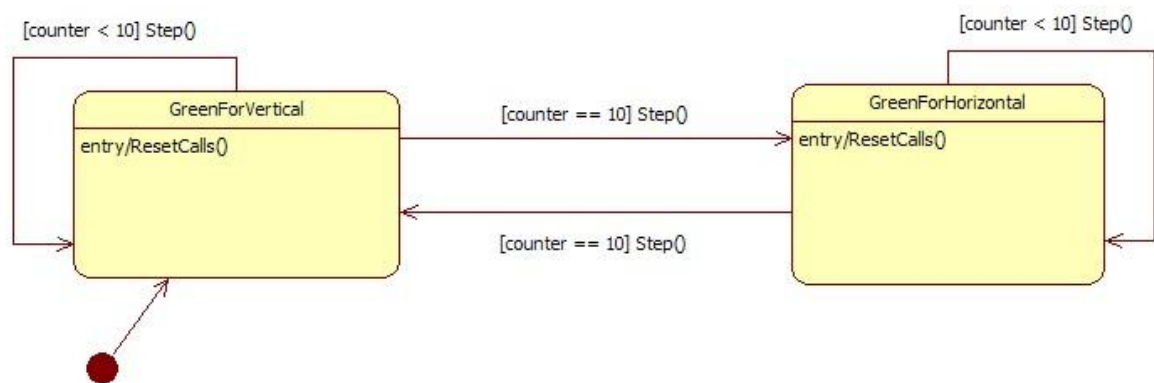


6. ábra: Szekvencia diagram rabló mozgására

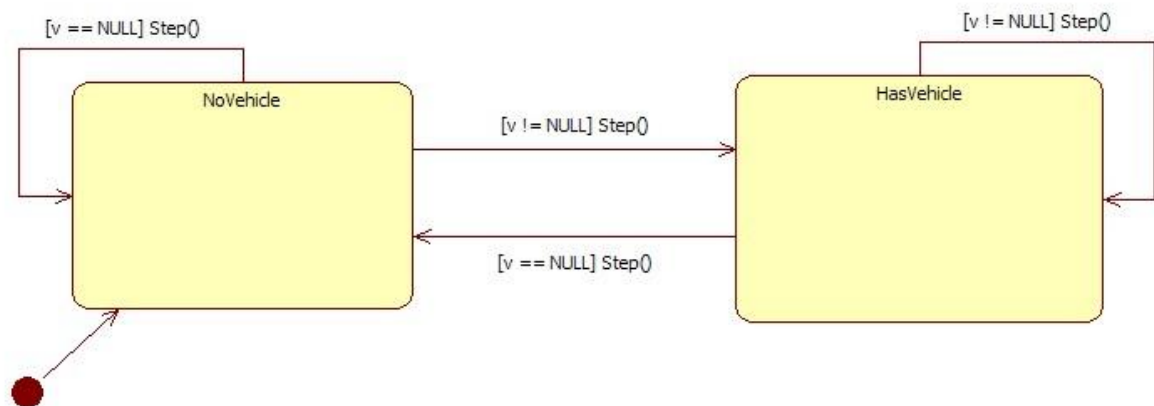


7. ábra: Szekvencia diagram inicializálásra

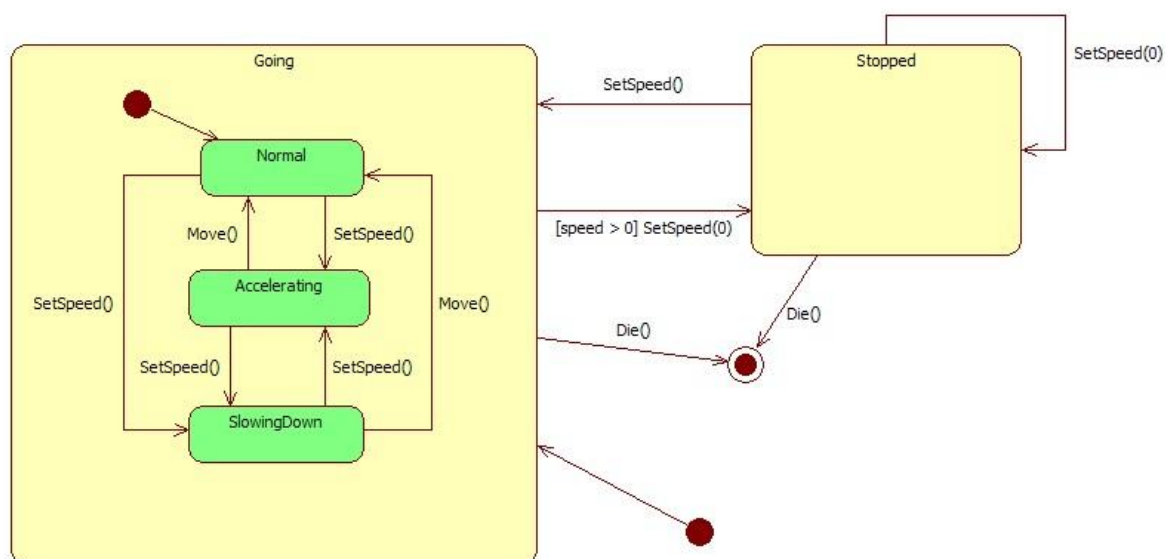
3.5 State-chartok



8. ábra: A jelzőlámpa állapotdiagramja



9. ábra: Az útelem állapotdiagramja



10. ábra: Az autó állapotdiagramja

3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2010.02.22. 21:35	1 óra	Rapp Takács Boros Molnár	Értekezlet. Döntés: Takács elkészíti a Statikus struktúra diagramokat és a State-chartokat, Boros elkészíti az Osztályok leírását, Rapp elkészíti az Objektum katalógust, Molnár elkészíti a Szekvencia diagramokat.
2010.0.24. 16:00	2 óra	Boros	Tevékenység: Egy kezdetleges osztálydiagramot készít.
2010.0.24. 20:15	2 óra	Rapp	Tevékenység: Objektum katalógust készít.
2010.0.24. 20:00	3 óra	Takács	Tevékenység: State chartok, és a végleges osztálydiagram elkészítése.
2010.0.24. 21:00	3 óra	Boros	Tevékenység: Elkészíti az osztályok leírását.
2010.0.24. 21:00	3 óra	Molnár	Tevékenység: Elkészíti a szekvencia diagramokat.
2010.0.25. 10:00	1 óra	Takács	Tevékenység: Formázza, véglegesíti a dokumentumt.