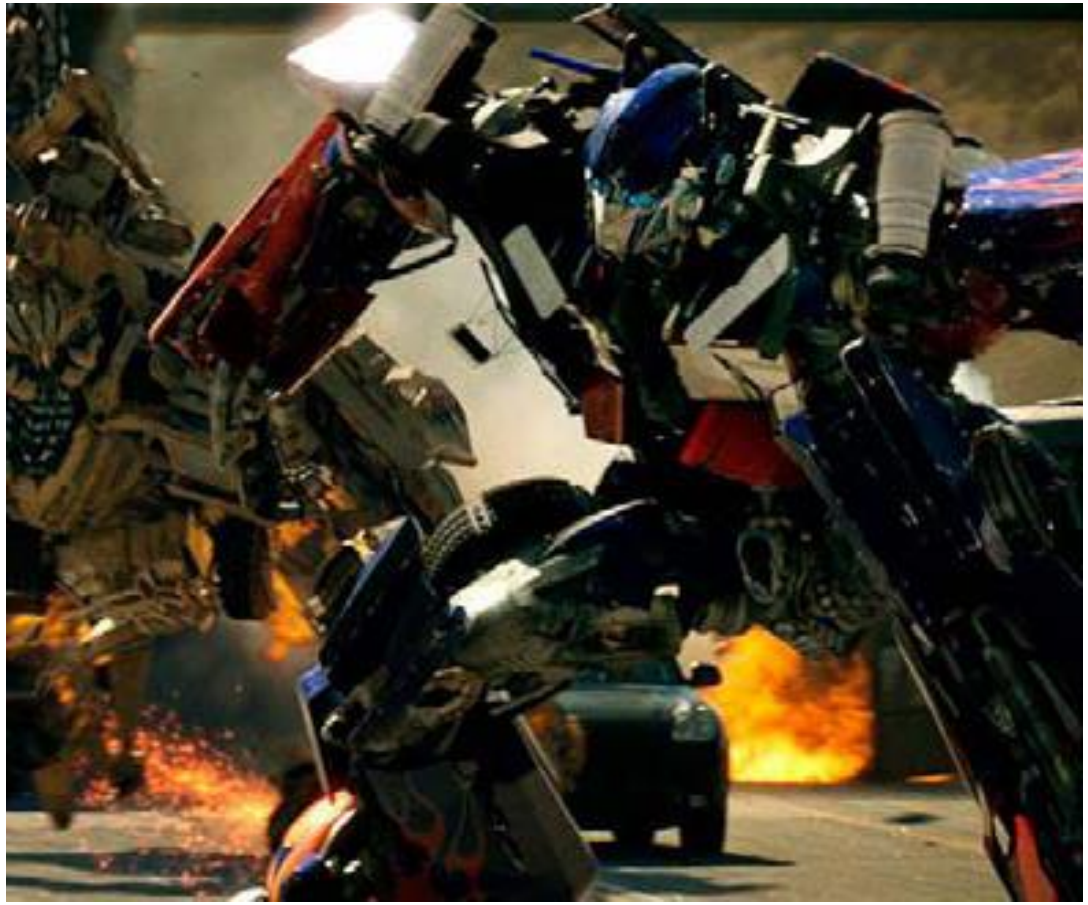
GRAF SCENE

Graf scene

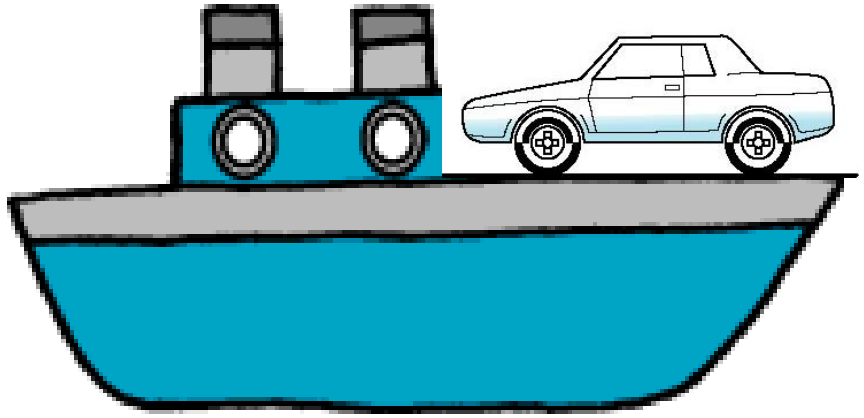
- Predmeti/scene v RG so tipično sestavljeni iz več delov
- Lahko bi jih predstavili neodvisno s seznamom delov
  - vsak je poligonski model
  - vsak ima svoje lastnosti in transformacije, ki jih pri izrisu upoštevamo
- **Vendar**: deli v sceni med seboj velikokrat **niso neodvisni** – npr. premikajo se skupaj oz. relativno en na drugega
  - npr. telo in roke/noge/glava
  - pri neodvisni predstavitvi je to problem, saj je potrebno pri spremembah najti in spremeniti vse odvisne dele

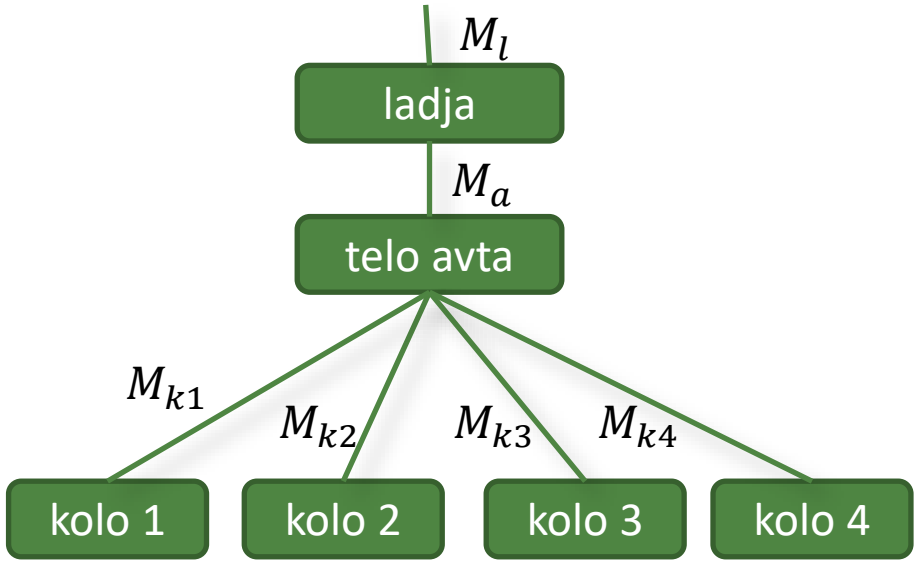


Dreamworks/Paramount—*Transformers*

Graf scene

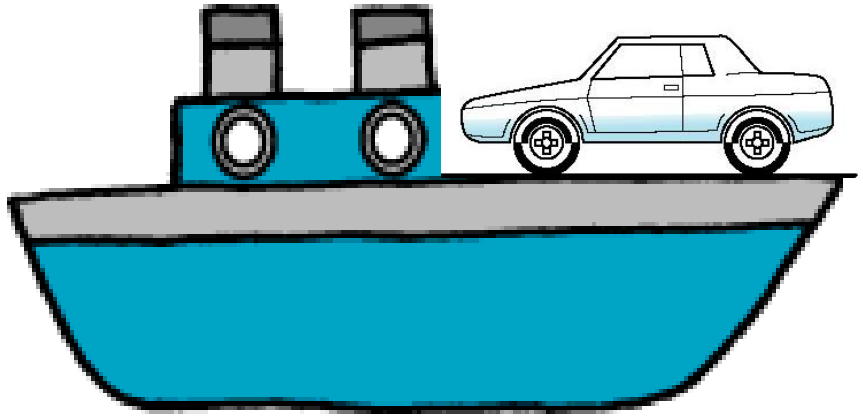
- Rešitev je v združevanju predmetov v **skupine**
- Vsaka skupina ima seznam predmetov (poligonskih modelov) in si deli lastnosti
- Hierarhija -> dobimo drevo
  - določimo “glavne predmete” in “odvisne predmete”
- Povezave nosijo podatke o **relativnem položaju** glede na starša
  - **transformacijske matrike**
- Pri **izrisu** predmeta je transformacijska matrika produkt matrik na poti od korena do predmeta
  - Npr. za kolo 1: $M_l M_a M_{k1}$

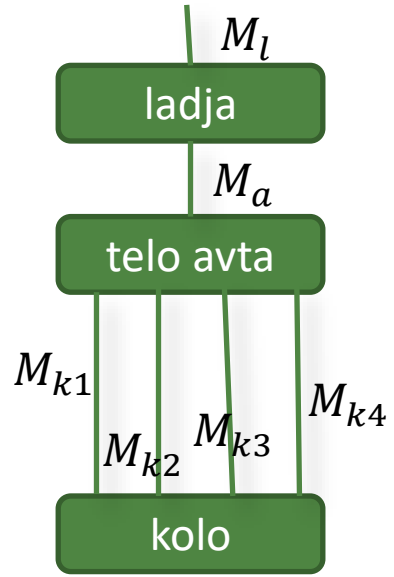




Graf scene - instance

- En predmet je lahko v sceni večkrat izrisan – ima več **instanc**
  - vsako z drugačno transformacijsko matriko
- Iz drevesa dobimo **graf**
  - usmerjen, acikličen
- Transformacije se ravno tako akumulirajo na poti od korena do predmeta

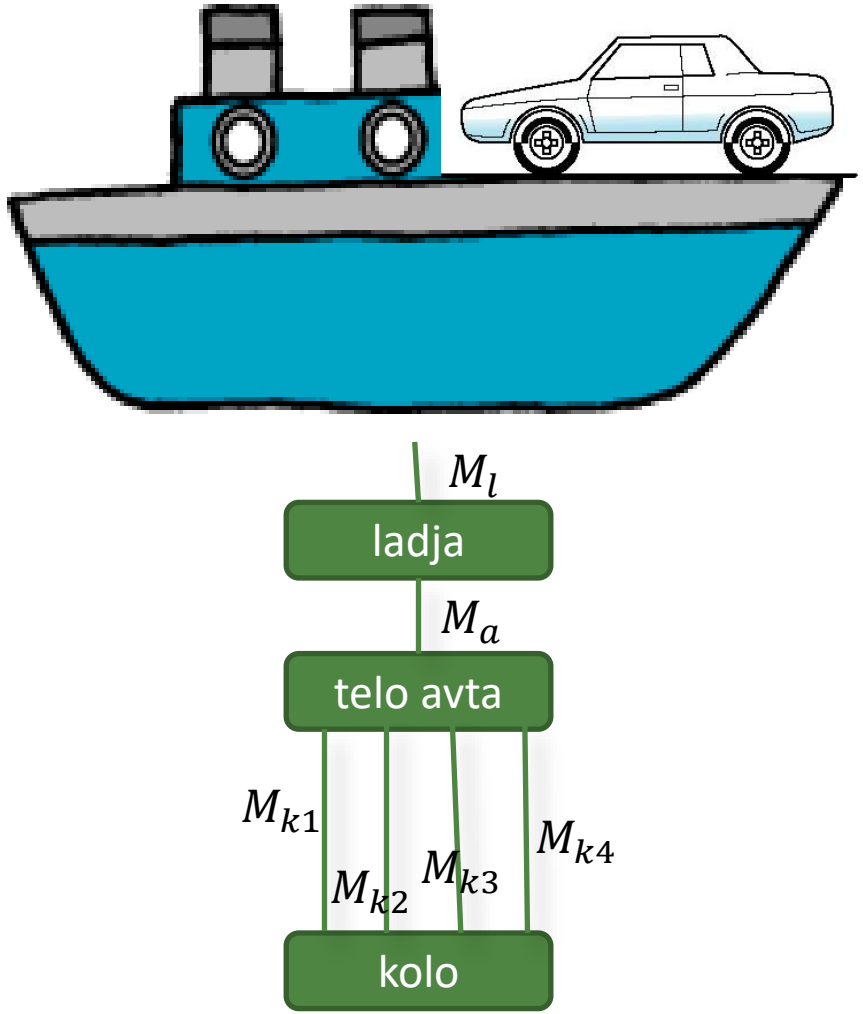




Graf scene

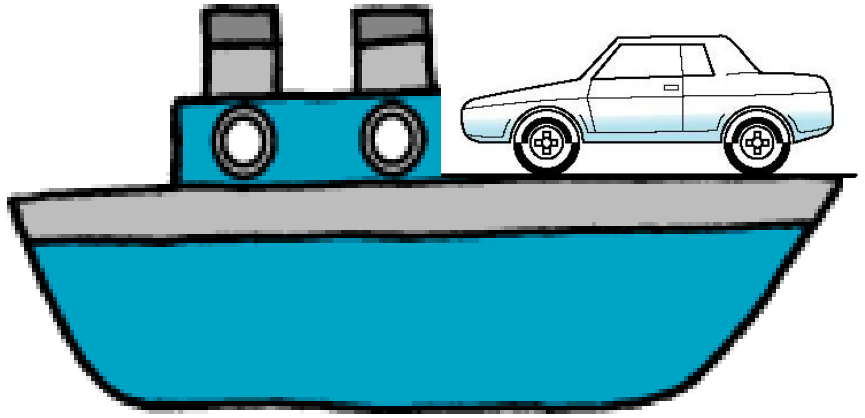
- Za izris grafa scene lahko vmesne produkte transformacijskih matrik hranimo na **skladu**
- Izris je rekurziven sprehod skozi graf:

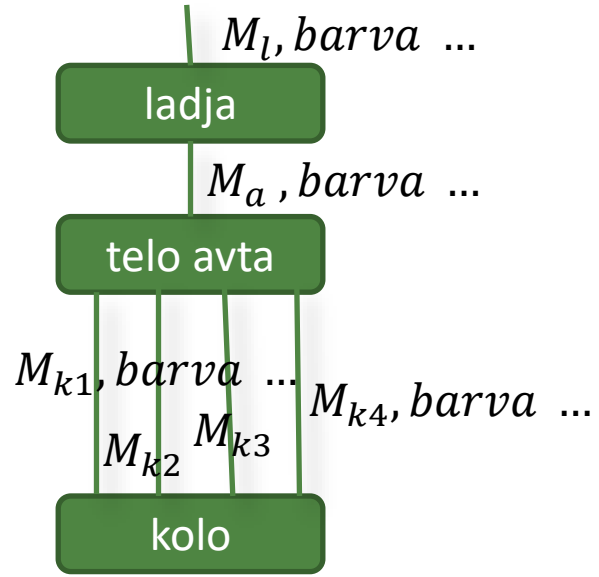
```
// W is the model matrix
// (initially identity)
void drawItem(i)
  W=Mi*W
  stack.push(W)
  render(i)
  foreach child c
    drawItem(c)
  stack.pop()
```



Graf scene – dodatne lastnosti

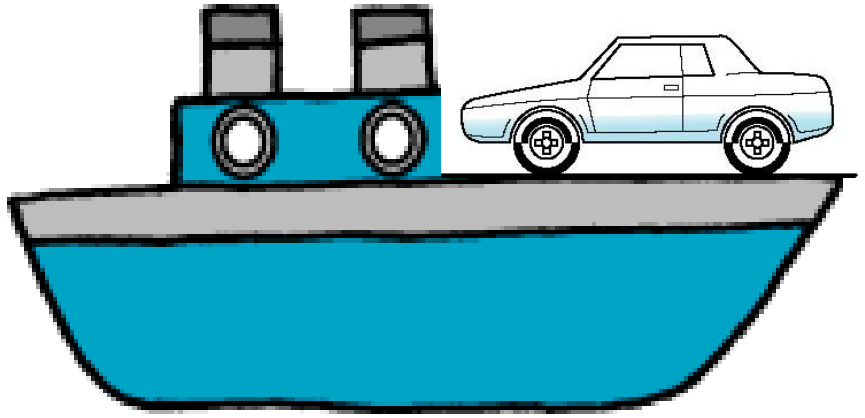
- Predmeti imajo poleg položaja lahko še druge lastnosti
  - barva
  - material
  - animacijski parametri…
- Tudi te lastnosti lahko dodamo v graf
  - in jih upoštevamo pri izrisu

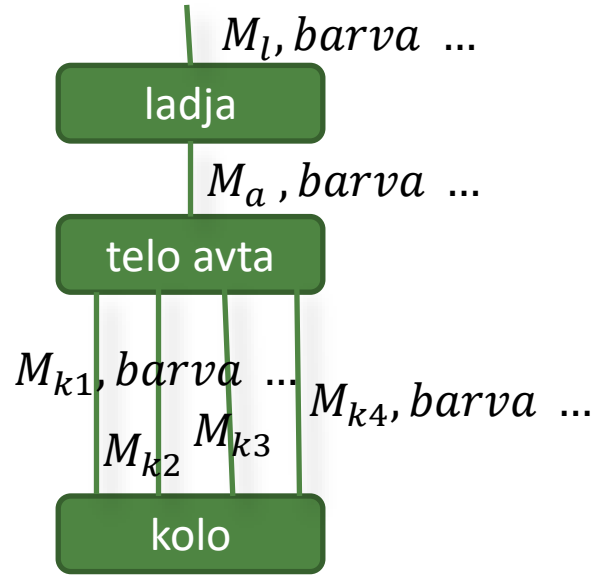




Graf scene – dodatni plusi

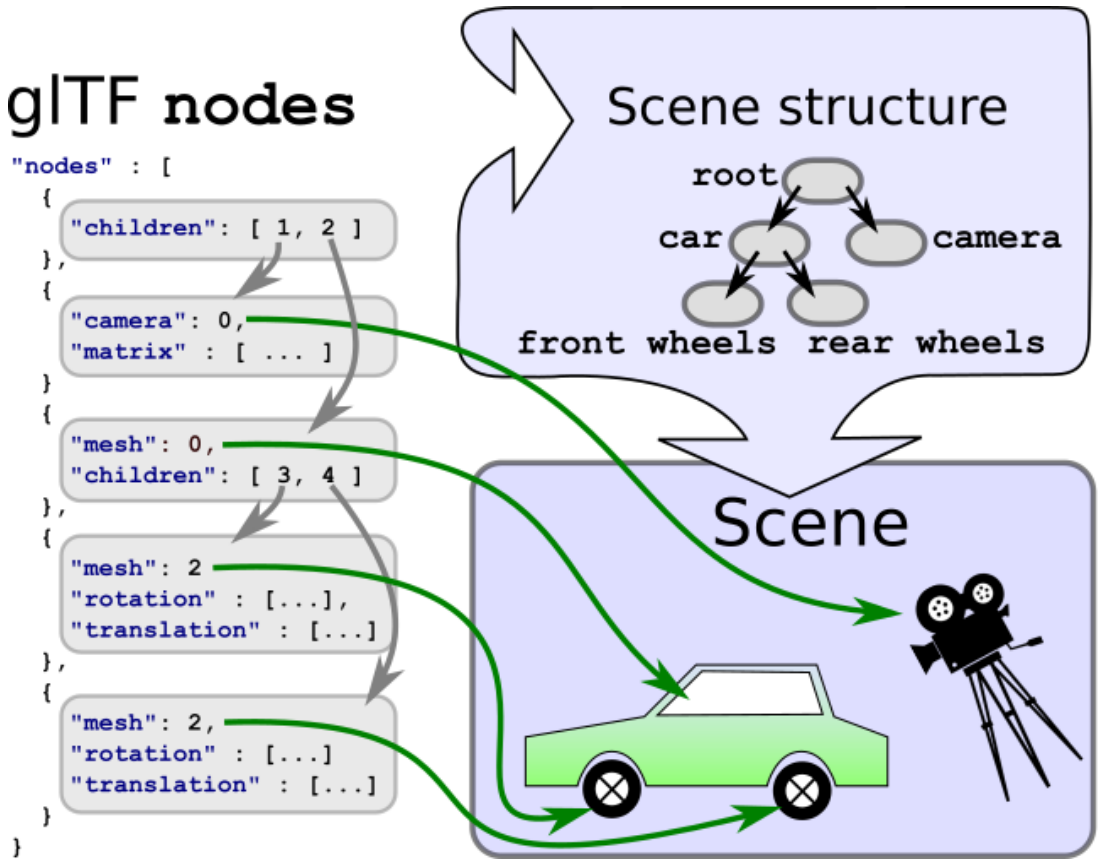
- Hierarhija predmetov omogoča tudi lažje
  - hierarhično deljenje prostora
  - izločanje predmetov, ki jih ne vidimo še pred izrisom
  - izračun nivojev podrobnosti
  - urejanje poligonov od zadnjega proti prvem
  - itn.





Graf scene – uporaba

- Programiranje
  - 3D Scene Graph, ThreeJS , BabylonJS… implementirajo grafe scene



glTF Scenes and Graphs

REFERENCE

- N. Guid: *Računalniška grafika*, FERI Maribor
- J.D. Foley, A. Van Dam et al.: *Computer Graphics: Principles and Practice in C*, Addison Wesley