

**Università degli studi di Roma “Tor Vergata”**



**Facoltà di Ingegneria**

**Laurea in Ingegneria Informatica**

***Creazione e animazione interattiva di grafica tridimensionale***

**Relatore**

**Ing. Francesco Martinelli**

**Candidato**

**Michele Martone**

**Sviluppo, a partire da librerie grafiche OpenGL, di un software (UGP) con le seguenti finalità nell'ambito dell' automazione:**

- Rappresentazione grafica 3D di tessuti organici e degli effetti di sollecitazioni sugli stessi.
- Visualizzazione da remoto di manipolatori robotici.
- Rappresentazione di ambienti aperti 3D e navigazione di robot mobili.

# Campo d'applicazione : medico e automazione medica

Nel campo della chirurgia guidata visualmente, è possibile visualizzare tessuti e manipolatori meccanici per guidare l'intervento con precisione laddove la visione da video camera fosse impossibilitata o fuorviante.

- Si sta diffondendo l'uso di rappresentazioni tridimensionali (endoscopie 3D, didattica medica assistita al computer, simulazioni).

```
tissue  
0.000  
4.764  
5.386
```

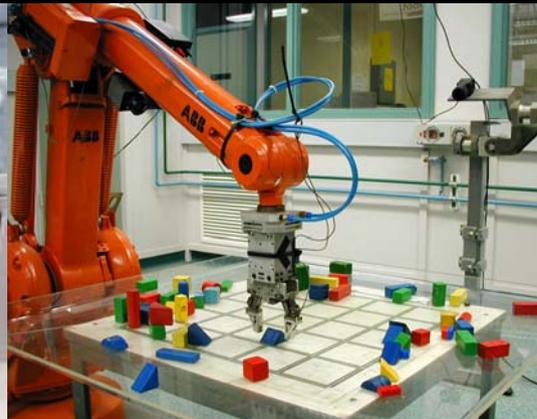


# Campo d'applicazione : robotica

Per vari motivi, in diversi ambiti il monitoraggio/la visualizzazione del robot è fuorviante/non disponibile. Una rappresentazione snella della sua configurazione permetterebbe controllo remoto.

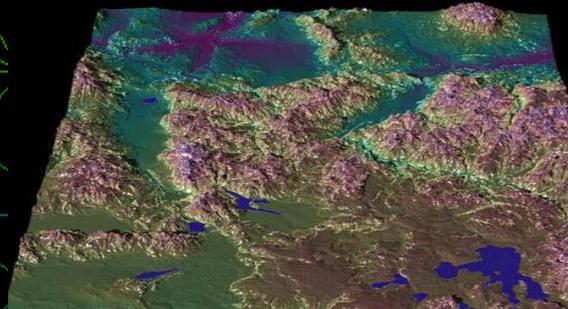
Rappresentazione Denavit-Hartenberg descrive la configurazione di un manipolatore con pochi dati.

```
link#1: alpha, theta, d  
0.000  
0.000  
3.000  
link#2: alpha, theta, d  
3.141  
7.299  
0.000  
0.000  
link#3: alpha, theta, d  
0.000  
1.000  
0.000
```



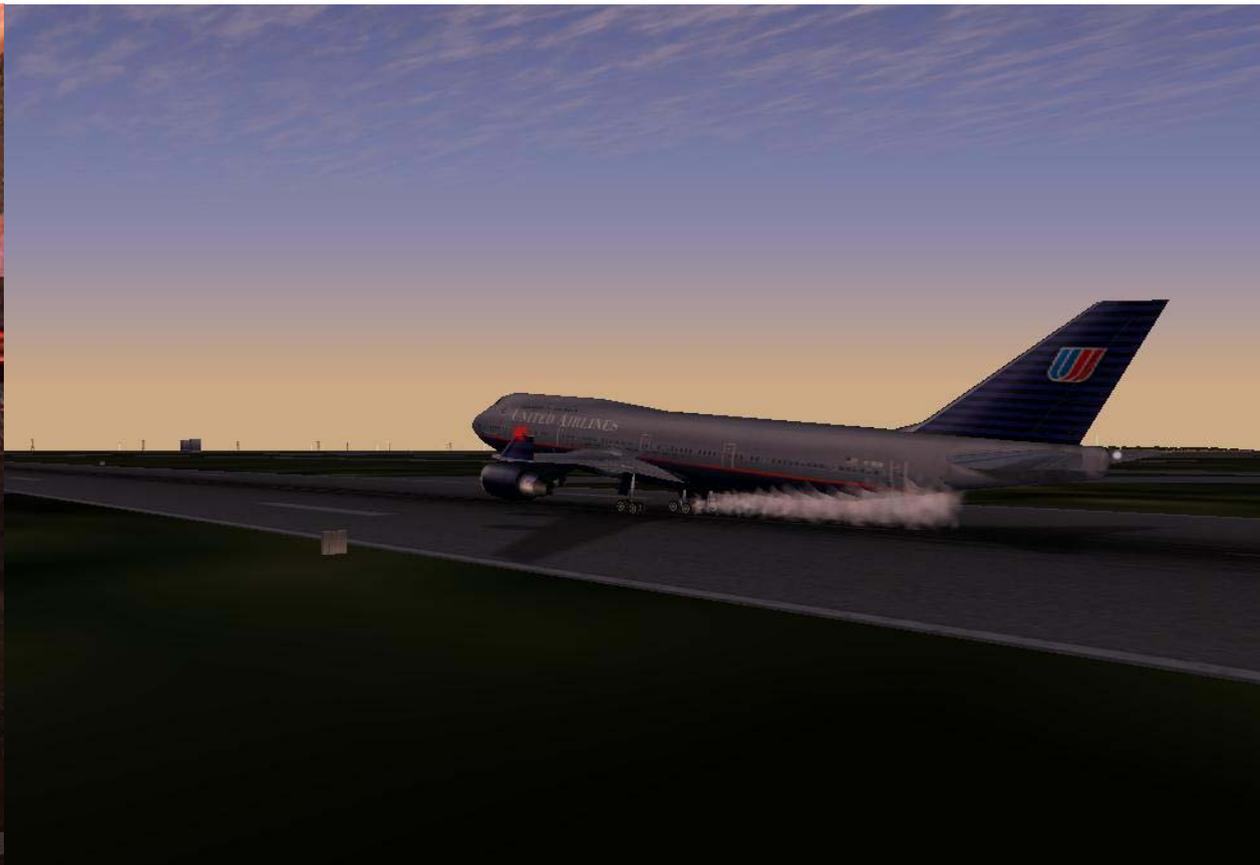
# Campo d'applicazione : visualizzazione terreni 3D

La navigazione di robot mobili in ambienti ostili (es.: rover) può trarre ausilio da una rappresentazione tridimensionale del terreno, laddove fossero noti dati sulla sua geometria (es.: altimetrie).



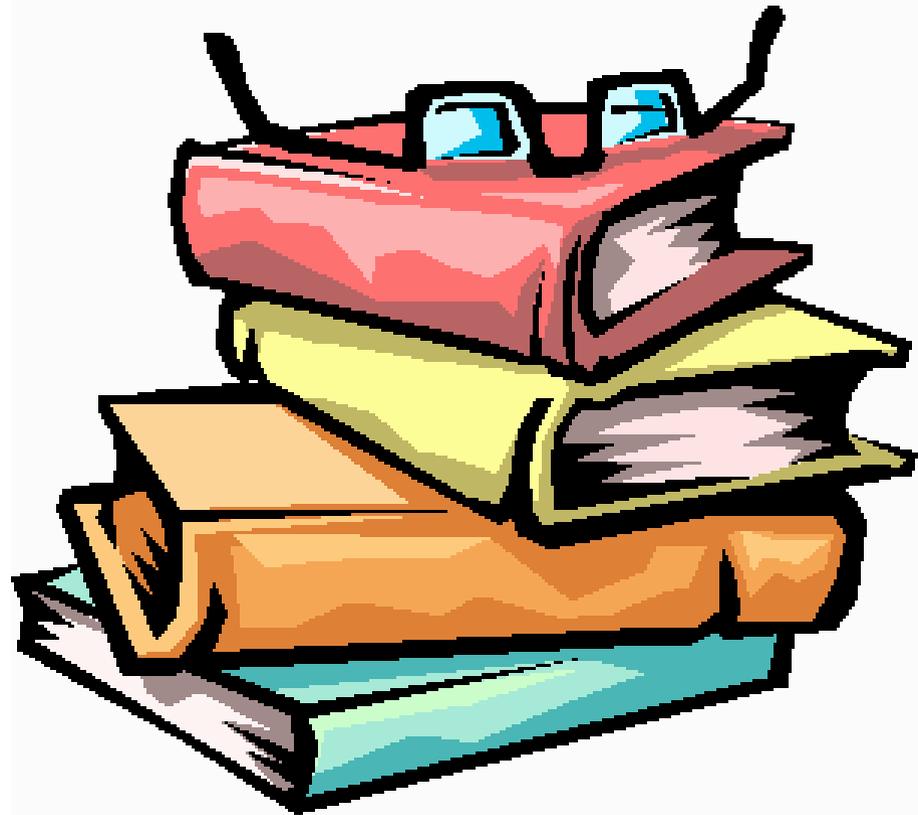
# Campo d'applicazione : videogiochi e simulatori di volo

- E, inevitabilmente,... i videogames....



# Discipline correlate alla computer grafica:

- algebra lineare, geometria proiettiva
- teoria dei segnali
- algoritmi e strutture dati
- geometria computazionale



# Pixel e Colore

- L'elemento di base di ogni rappresentazione grafica è il *pixel*, o *P*icture *E*lement, ossia la macchietta di colore di dimensione minima rappresentabile su un display.
- Il colore è una sensazione, quindi la sua misura è soggettiva e la sua corretta rappresentazione è un problema.

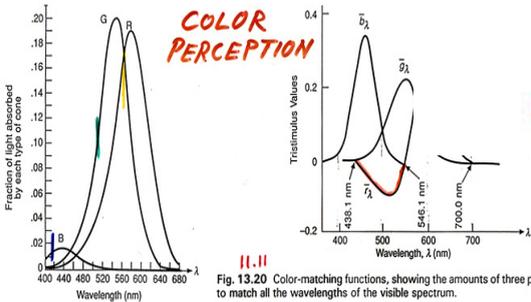


Fig. 13.20 Color-matching functions, showing the amounts of three primaries to match all the wavelengths of the visible spectrum.

Spectral-response functions of each of the three types of cones on the eye.

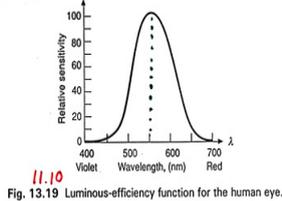
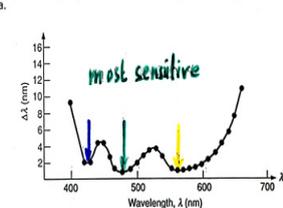
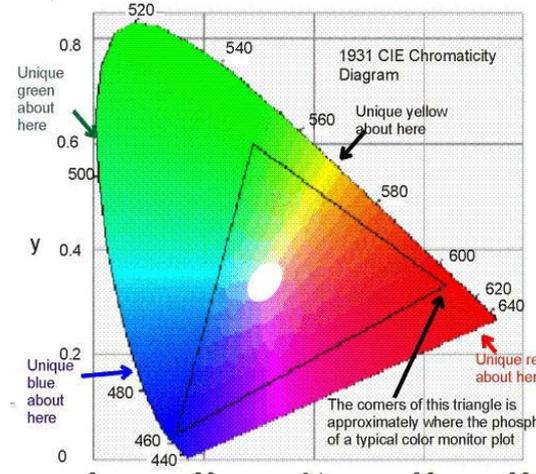


Fig. 13.19 Luminous-efficiency function for the human eye.



# Filtri, pixel & co.

- Grazie alla rappresentazione formalizzata in forma di segnale discreto bidimensionale, un'immagine può essere sottoposta a trasformazioni e analisi tipiche della teoria dei segnali (DFT, convoluzioni, correlazioni, etc)



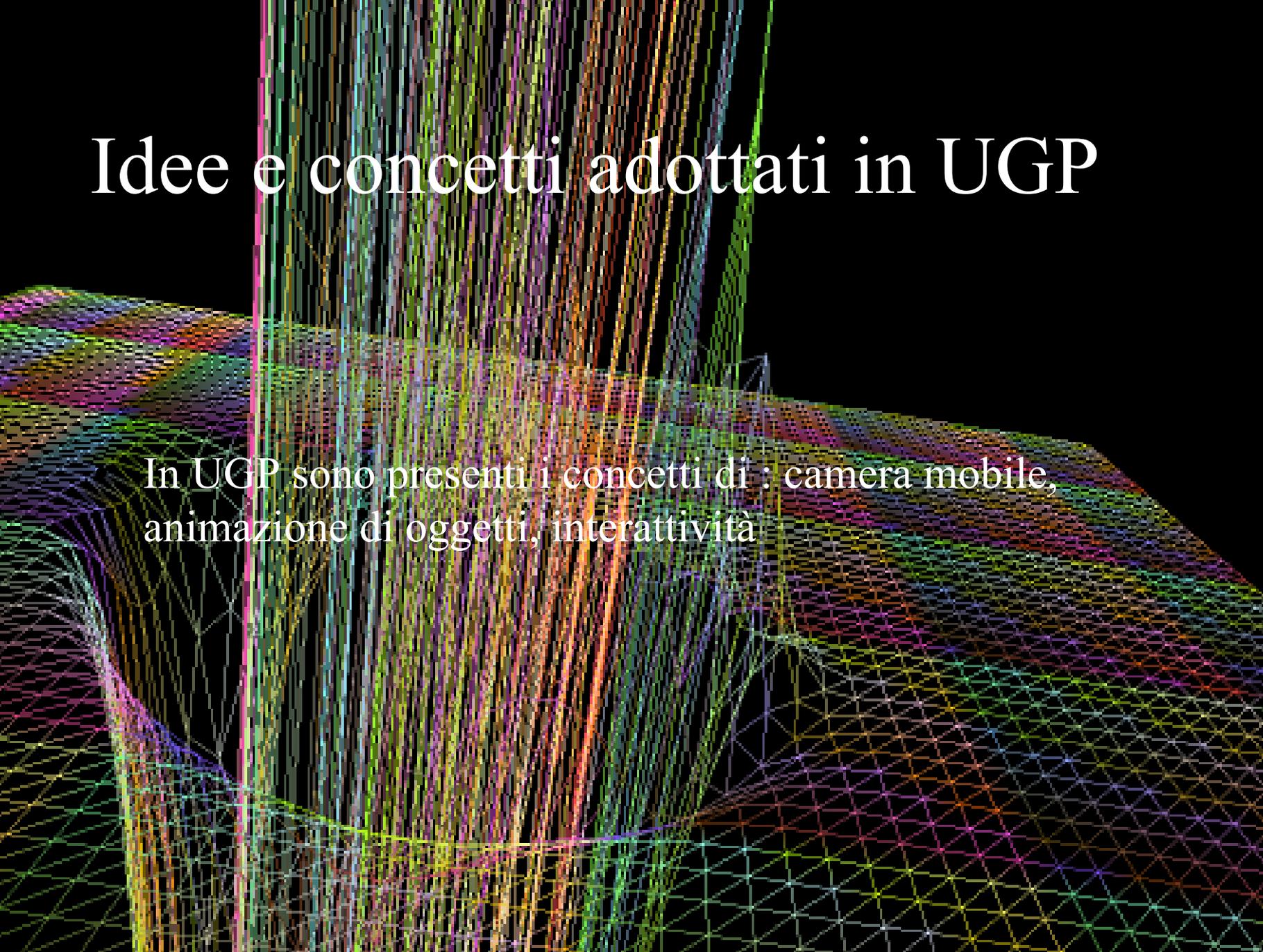
# 3D : meshes, triangoli, pro/contro

Attualmente le rappresentazioni tridimensionali si avvalgono di approssimazioni in forma di composizioni di poligoni (*polygon meshes*).

Esistono altre rappresentazioni, ma per la grafica *real-time* la rappresentazione poligonale è lo standard.

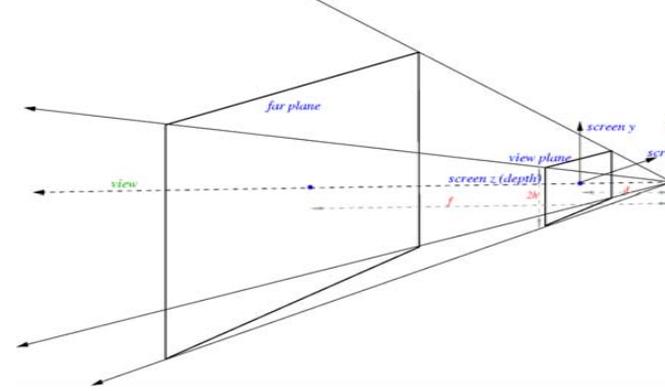


# Idee e concetti adottati in UGP

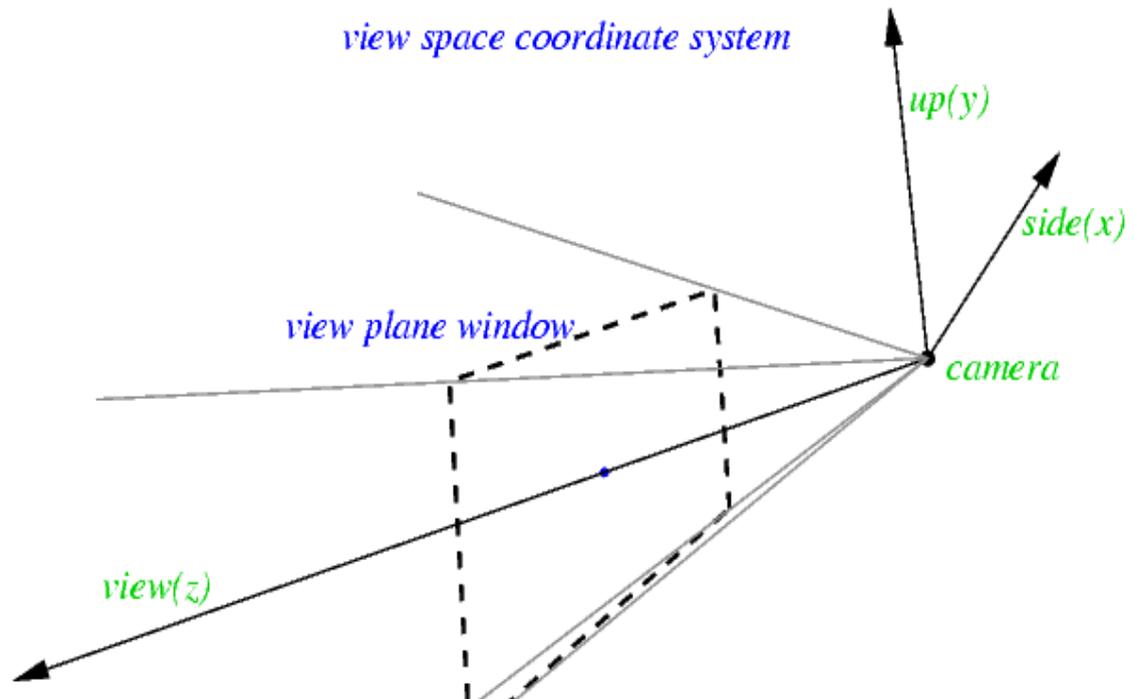
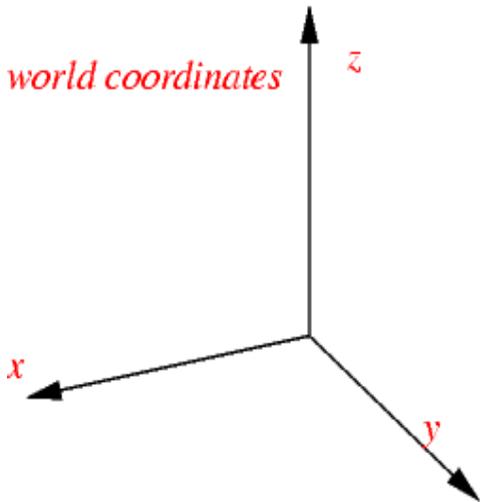


In UGP sono presenti i concetti di : camera mobile,  
animazione di oggetti, interattività

# Camera & World

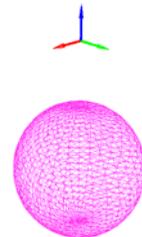
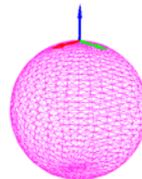
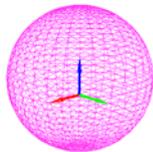
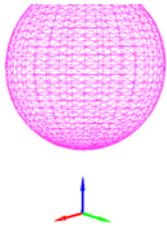


- La scena è proiettata sul piano corrispondente a quello del display.



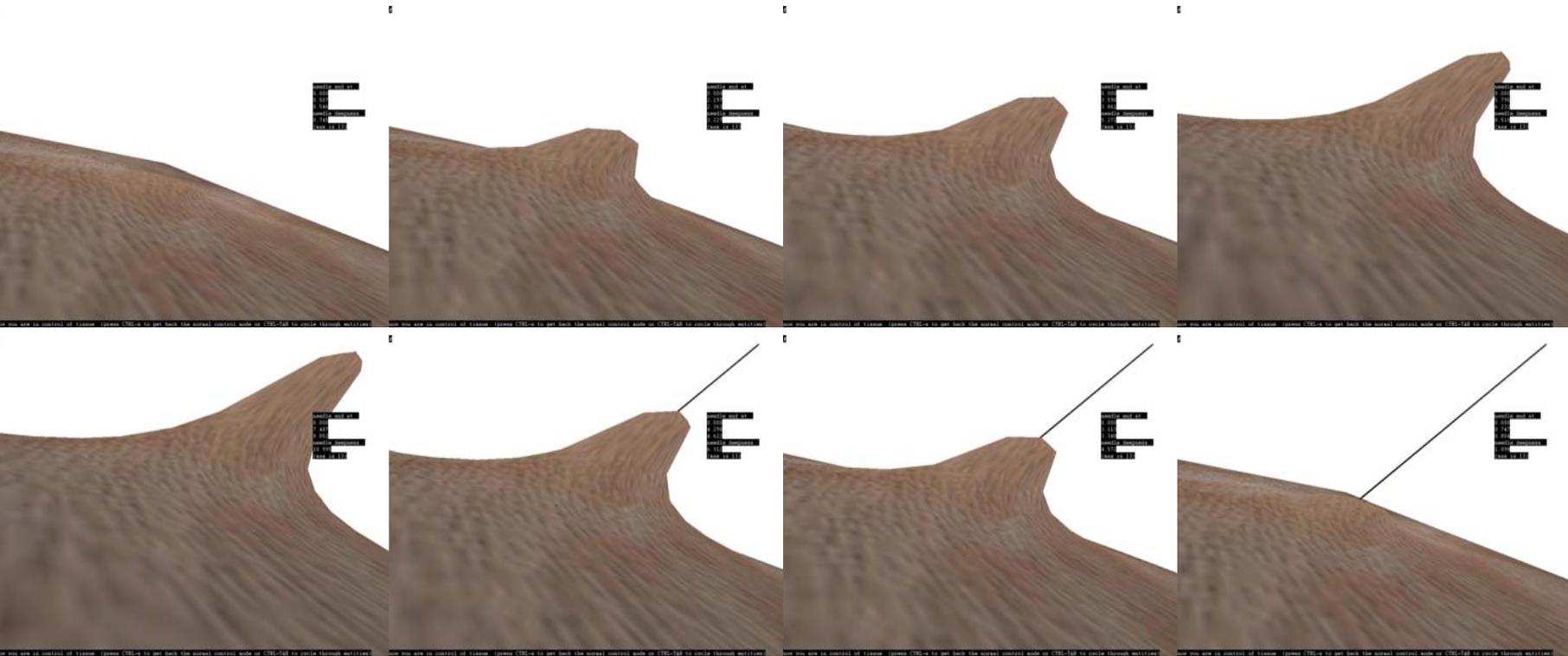
# Animazione

- In UGP diversi agenti possono avere un'evoluzione propria, e la loro geometria essere dinamica.
- es.: palla che rimbalza



# Interattività

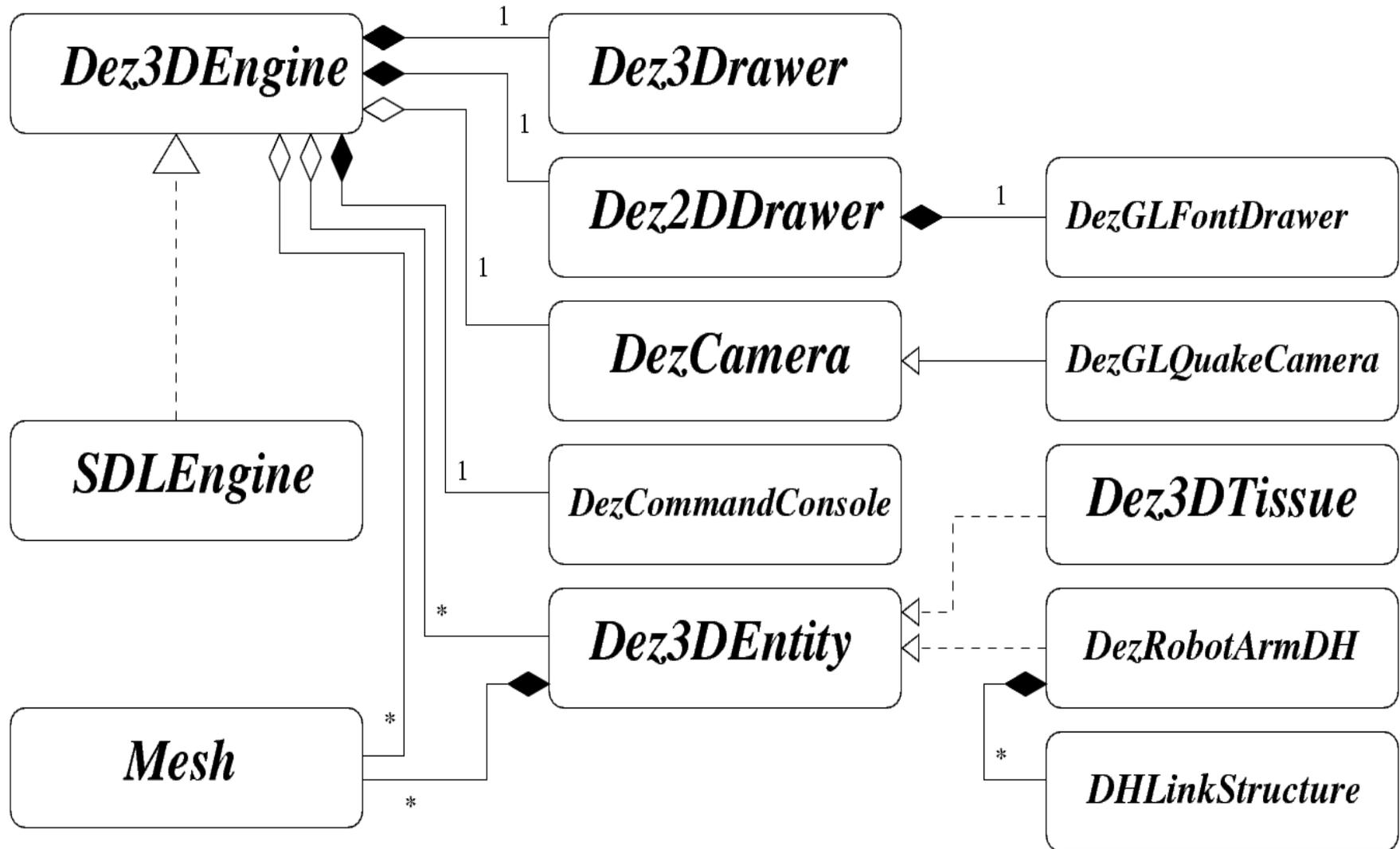
- Diversi oggetti, possono essere 'posseduti' dall'utente e guidati dall'input di tastiera e mouse.
- es.:tessuto penetrato da ago



# Comandi testuali

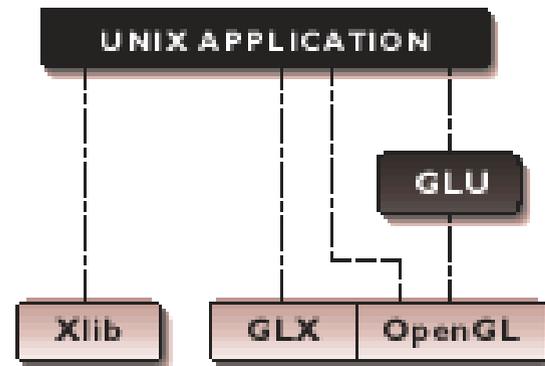
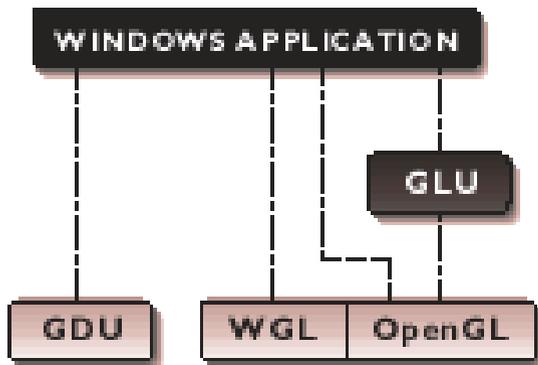
- C'è la possibilità di immettere comandi testuali, che permettono di cambiare opzioni dell'ambiente o dare messaggi diretti agli agenti presenti.
- #esempi di comandi:
- `drawer set drawtextures 0`
- `meshcreator cube bounded 0 0 0 1 1 1 texture data/textures/smiley.bmp`
- `meshcreator cylinder radius 1`
- `meshcreator flatSystolicMesh cellSize 1 xCells 16 yCells 16 in 0 0 -10 color 255 0 0 texture 3`
- `del *`
- `camera set theta 3.98`

# Architettura del software UGP



# Software libero & librerie

- UGP è stato creato appoggiandosi a *free software* (le API *Mesa GL* e *SDL*) ed è compilabile con gli strumenti GNU (*gcc, make*).
- La documentazione del codice è stata creata automaticamente con il software *doxygen*.



# Possibili sviluppi futuri di UGP :

- didattica medica e simulazioni chirurgiche: rappresentazione di fenomeni più complessi (es.:tagli,sezioni) e supporto di interfacce aptiche.
- manipolazione remota: integrazione di un'interfaccia per l'acquisizione dati di configurazioni di robot tramite network (es.:su TCP/IP).
- sperimentazione di algoritmi di navigazione per robot e visualizzazione ambienti 3D per rover.

Altri possibili sviluppi:

Semplici simulazioni con agenti; implementazione di algoritmi di *surface removal*; esperimenti di algoritmi AI