



# Grafica al calcolatore - Computer Graphics

Laboratorio 1 - Modellazione



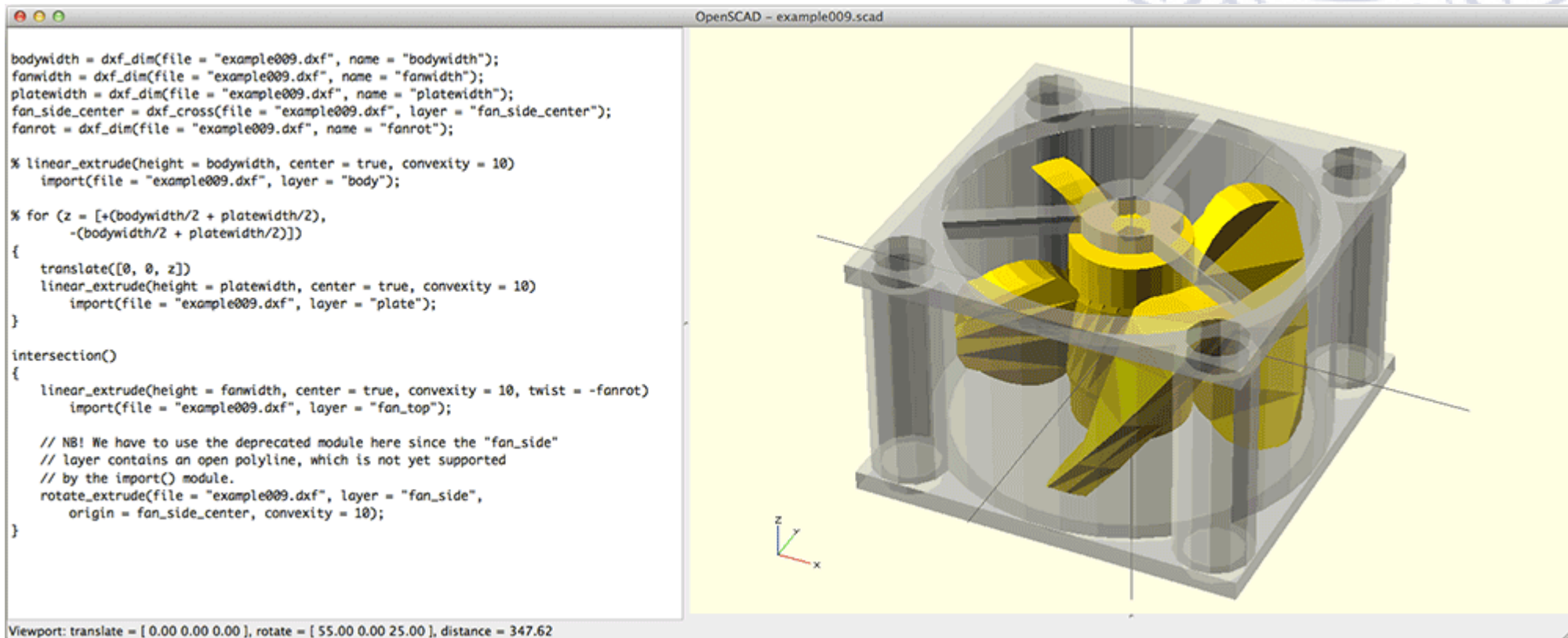
# Software per modellazione

- Vari tipi
  - CAD (computer aided design) per progettare manufatti
  - Constructive Solid Geometry
  - Editor per modellazione di personaggi (rigging, animazione)
  - Sistemi per acquisire modelli 3D con scanner
  - Sistemi spesso sofisticati e con caratteristiche peculiari
  - Non possiamo dedicarci a imparare questi, ma vogliamo acquisire “computational geometry fluency”
- Vedremo un paio di cose semplici e gratuite
  - OpenSCAD per CSG
  - Meshlab per processing nuvole di punti, elaborazione di maglie triangolate, semplificazione e molto altro



# OpenSCAD

- Semplice tool per creare modelli procedurali e esportarli come mesh triangolate



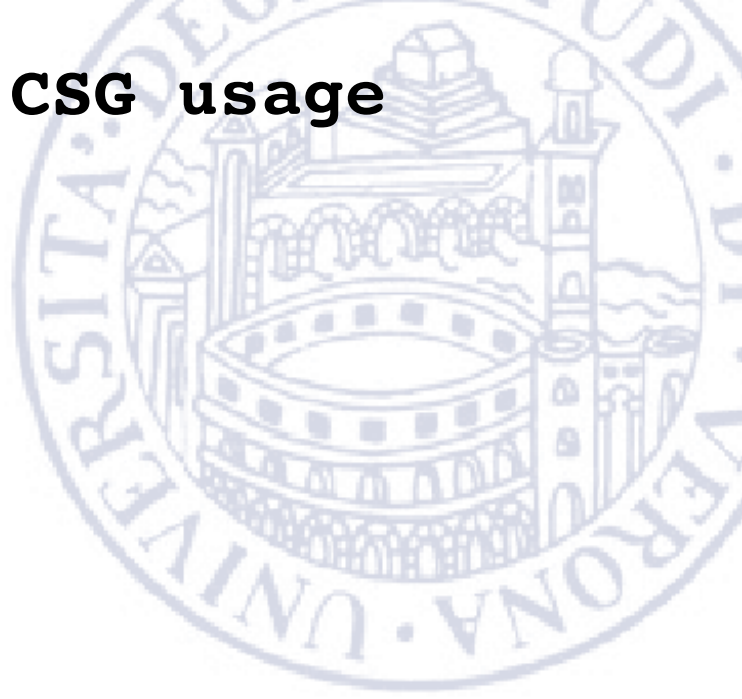
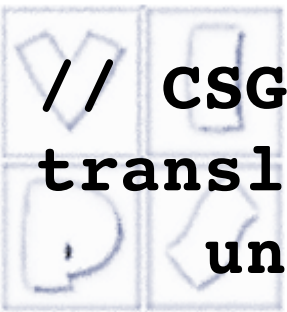


# Sintassi

- Semplice.



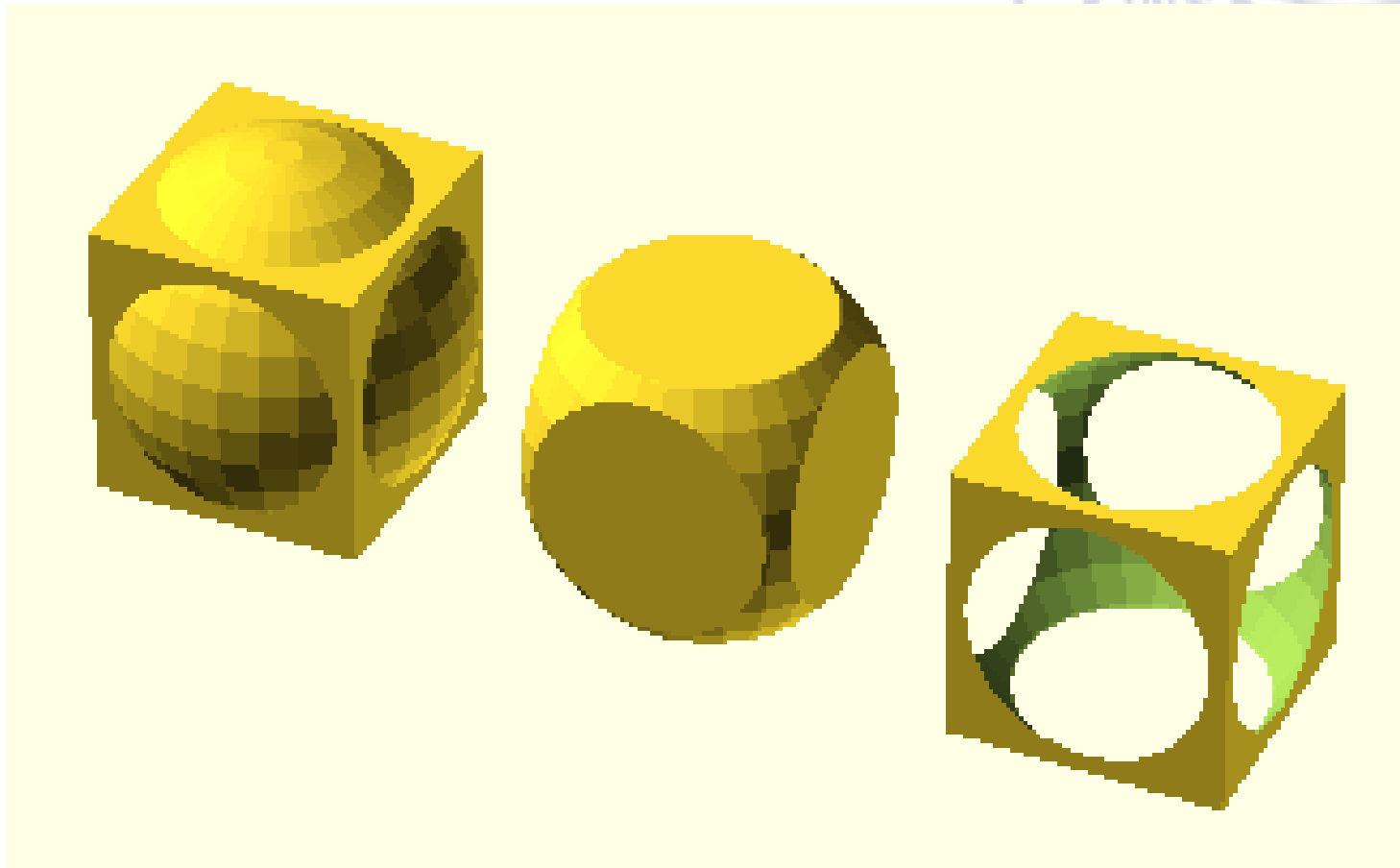
```
module example004()  
{  
    difference() {  
        cube(30, center = true);  
        sphere(20);  
    }  
}  
  
example004();
```



```
// CSG.scad - Basic example of CSG usage
translate([-24,0,0]) {
    union() {
        cube(15, center=true);
        sphere(10);
    }
}
intersection() {
    cube(15, center=true);
    sphere(10);
}
translate([24,0,0]) {
    difference() {
        cube(15, center=true);
        sphere(10);
    }
}
}
```



# Esempio







# Output

- Naturalmente la struttura dati procedurale non è ideale per il rendering (come vedremo)
- In realtà il programma converte a mesh poligonale per il rendering
- E poi ci interessa salvare su mesh poligonale per riutilizzare
- Sfere, cilindri(coni) etc. possono essere definiti coi parametri che ne determinano la discretizzazione
- Es. sfera: posso usare non solo `sphere(r)`
- Ma anche aggiungere parametri `$fa` dimensione dell'elemento di superficie, `$fs` angolo sotteso dal frammento di superficie
- Alternativa uso di `$fn` con specifica del numero esatto di elementi

```
sphere(2, $fa=5, $fs=0.1);
```



# Molte altre cose

- Linguaggio che comprende iteratori, ricerche, ecc
  - Modelli che possono includere parti estruse
  - Uso del colore
  - Animazione
- 
- Documentazione:
    - <http://www.openscad.org/documentation.html>





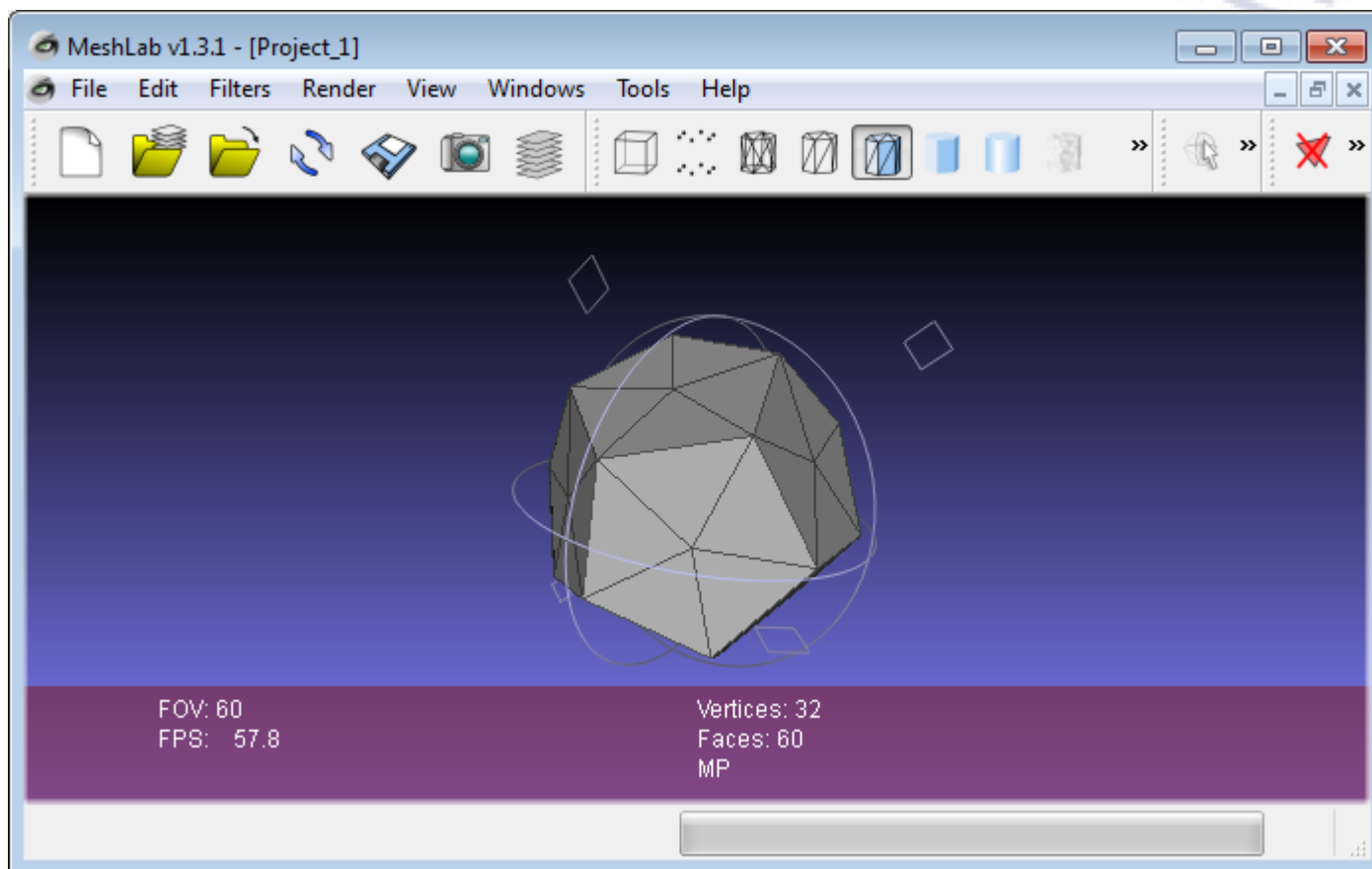
# Esercizio

- Creare un modello di astronave usando CSG con openSCAD
  - Usare varie primitive traslate e ruotate
  - Allineare l'astronave lungo l'asse z
- Salvarlo in formato .obj



# Meshlab

- Strumento open source ottimo per il processing di modelli con algoritmi moderni e aggiornato
  - <http://meshlab.sourceforge.net/>
- Sviluppato in Italia (ISTI-CNR Pisa)





# Molto materiale

- Non abbiamo tempo/possibilità di vederne i vari usi. Ma si trova documentazione ampia, ad esempio
- MeshLab online help (devoted to programmers)
  - [http://meshlab.sourceforge.net/wiki/index.php/MeshLab\\_Documentation](http://meshlab.sourceforge.net/wiki/index.php/MeshLab_Documentation)
- MeshLab blog
  - <http://meshlabstuff.blogspot.com/>
- MeshLab forum
  - <http://sourceforge.net/projects/meshlab/forums/forum/499533>
- Video tutorials
  - <http://www.youtube.com/user/MrPMeshLabTutorials>



# Formati file supportati

- Input
  - PLY, STL, OFF, OBJ, 3DS, COLLADA, PTX, V3D, PTS,
  - APTS, XYZ, GTS, PDB, TRI, ASC, X3D, X3DV, WRL,
  - ALN...
- Export:
  - PLY, STL, OFF, OBJ, 3DS, COLLADA, VRML, DXF, U3D, GTS, IDTF, X3D...



# Mesh processing

- Meshlab non è un programma di editing manuale di modelli (anche se consente di farlo), ma un ottimo programma di processing:
  - Creare triangolazioni
  - Cambiare risoluzione
  - Gestire attributi, colore
  - Regularizzare
  - E molto altro
  - Ottimo per processare i dati acquisiti da scanner



# Operazioni e feature

- Si possono creare vari layer gestibili con il menu apposito
- Si possono registrare modelli tra loro sulla base di punti di riferimento o con algoritmi appositi
- Si possono realizzare serie di operazioni e memorizzarli in script richiamabili da console usando l'applicazione meshlabserver



# Esercizio 1

- Carico una nuvola di punti e la trasformo in un modello triangolato
  - Caricare la nuvola “gufo.ply”
  - Applicare
    - Compute Normals from a point set
    - Surface Reconstruction: Poisson
    - Vertex attribute transfer per passare il colore alla mesh
    - Esportare in formato obj.





## Esercizio 2

- Creare una sfera
- Applicare del rumore ai vertici
- Colorare secondo la curvatura
- Salvare in formato obj





# Esercizio 3

- Caricare il modello face.obj
- Tappare i buchi
- Salvare in formato .obj





## Esercizio 4

- Caricare il modello di astronave creato prima
- Ricavare una versione semplificata ed una con un numero maggiore di triangoli
- Salvare i modelli in formato .obj