

Game Engines
Master Game Dev
Verona, 2014-2015
Assets: 3D Mesh

Marco Tarini

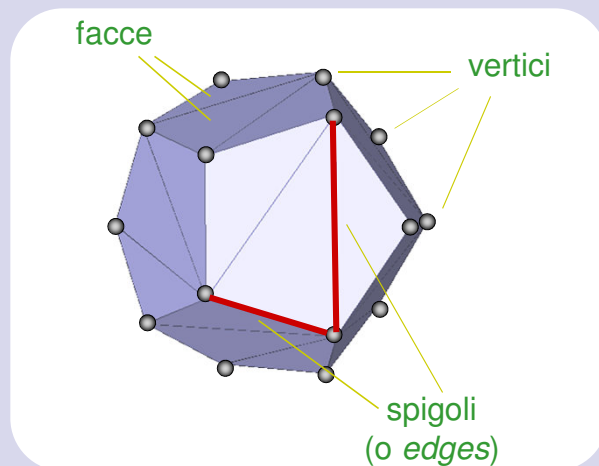


**I Modelli 3D più comuni:
Mesh poligonali**

- Di triangoli, o mista (quadrilateri + triangoli)
- Struttura dati per modellare oggetti 3D
 - GPU friendly
 - Risoluzione (potenzialmente) adattiva
 - “Complessità” = numero facce

Mesh triangolare (o mesh simpliciale)

- Un insieme di triangoli adiacenti



Mesh di triangoli

- discretizzazione lineare a tratti di una superficie continua (un “2 manifold”) immersa in R^3
- Componenti:
 - **geometria**
 - i vertici, ciascuno con pos (x,y,z)
 - un campionamento della superficie!
 - **connettività** (a volte: “topologia”)
 - come sono connessi i vertici
 - (es.: in una tri-mesh, i triangoli)
 - **attributi**
 - es: colore, materiali, normali, UV, ...

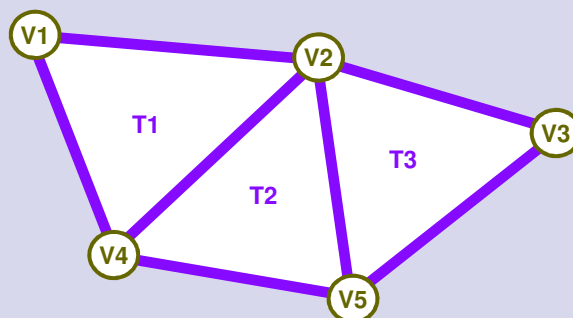
Mesh: geometria

- Insieme di posizioni dei vertici
 - Un vettore posizione (x,y,z) per ogni vertice
 - (Spazio Oggetto)



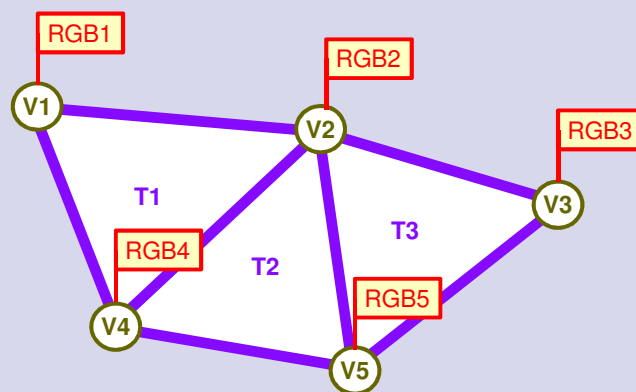
Mesh: connettività (o topologia)

- Triangoli (o quads, o edges...)
 - che connettono fra loro i vertici
 - Come nodi connessi da archi, in un grafo



Mesh: attributi

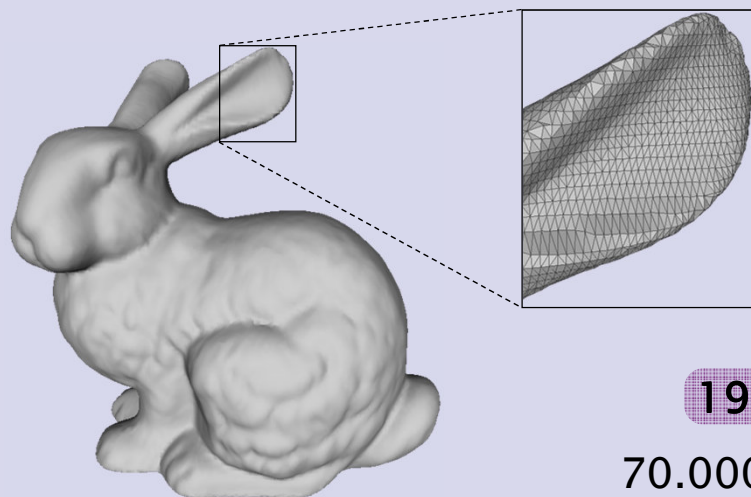
- Quantità che variano sulla superficie
 - Campionati per vertice, interpolati nei poly



Mesh: attributi

- Proprietà che variano sulla superficie
- Memorizzati per vertice
 - (almeno, nei games)
- Attributi più diffusi nei games:
 - Normale
 - per: re-lighting dinamico
 - Colore
 - per: baked lighting (ambient occlusion)
 - per: aggiungere varietà (RGB)
 - Coordinate tessitura ("uv mapping")
 - per: texture mapping
 - Direzioni tangenti
 - per: bump mapping
 - Bone assignment ("rigging" o "skinning")
 - per: animazioni scheletrali

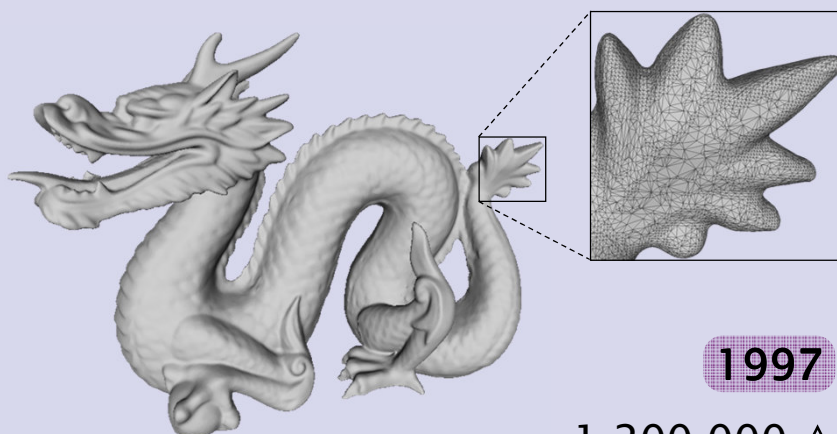
Meshes: complessità crescente



1994

70.000 Δ

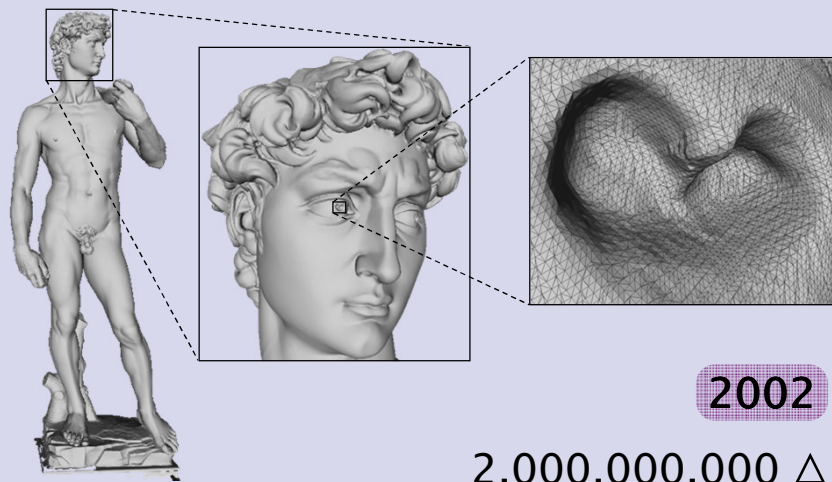
Meshes: complessità crescente



1997

1.200.000 Δ

Meshes: complessità crescente



Low Poly Meshes



Come rappresento una mesh? (quali strutture dati)

- Modo **indexed** in C++ :

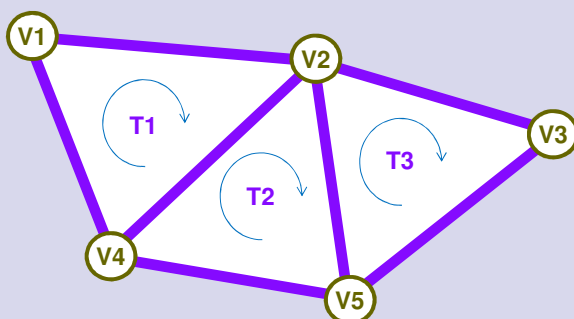
```
class Vertex {
    vec3 pos;
    rgb color; /* attribute 1 */
    vec3 normal; /* attribute 2 */
};

class Face{
    int vertexIndex[3];
};

class Mesh{
    vector<Vertex> vert; /* geom + attr */
    vector<Face> tris; /* connettivita' */
};
```

Come rappresento una mesh? (quali strutture dati)

- **indexed mesh**



vert	X	Y	Z	R	G	B
V1	x1	y1	z1	r1	g1	b1
V2	x2	y2	z2	r2	g2	b2
V3	x3	y3	z3	r3	g3	b3
V4	x4	y4	z4	r4	g4	b4
V5	x5	y5	z5	r5	g5	b5

GEOMETRIA + ATTRIBUTI

Tri:	Wedge 1:	Wedge 2:	Wedge 3:
T1	V4	V1	V2
T2	V4	V2	V5
T3	V5	V2	V3

CONNETTIVITA'

