Game Development

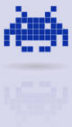
Modelli 3D

Marco Tarini

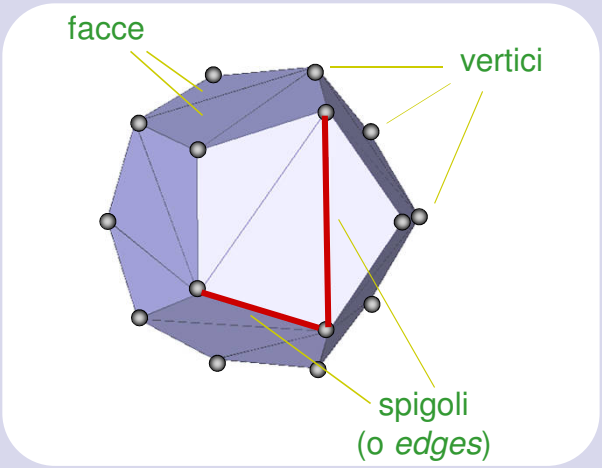
Modelli 3D

- Modello 3D = Mesh poligonale
  - 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)



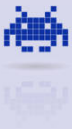
- Un insieme di triangoli adiacenti



Mesh di triangoli

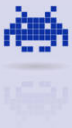
- discretizzazione lineare a tratti di una superfice continua (un “2 manifold”) immersa in R3
- Componenti:
  - geometria
    - i vertici, ciascuno con pos (x,y,z)
    - un campionamento della superficie!
  - connettività
    - come sono connessi i vertici
    - (es.: in una tri-mesh, i triangoli)
  - attributi
    - es: colore, materiali, normali, UV, temperatura

Mesh: geometria



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

Mesh: connettività

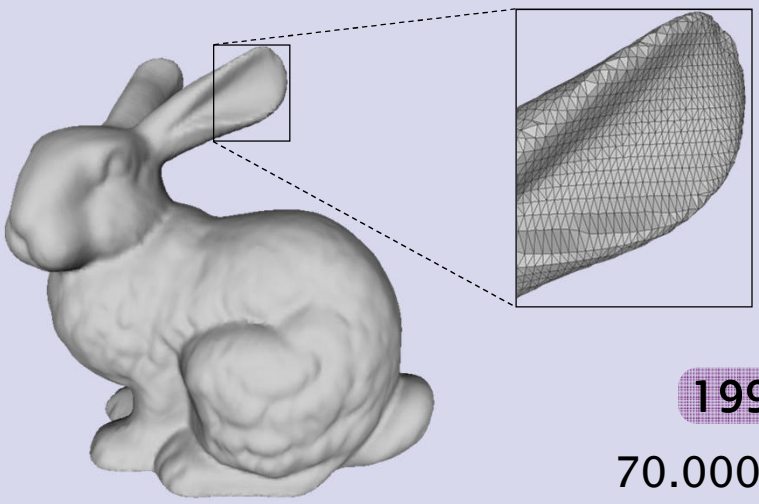


- Triangoli (e edges) che connettono fra loro i vertici
  - Come nodi in un grafo

Mesh: attributi

- Proprietà che variano sulla superficie
- Di solito memorizzati per vertice
  - (almeno, nei games)
- Attributi più diffusi nei games:
  - Normale
    - per: re-lighting dinamico
  - Colore
    - per: baked lighting (ambient occlusion)
    - per: aggingere varietà (RGB)
  - Coordinate tessitura (“uv mapping”)
    - per: texture mapping
  - Direzioni tangenti
    - per: bump mapping
  - Bone assignment (“rigging” o “skinning”)
    - per: animazioni skeletali

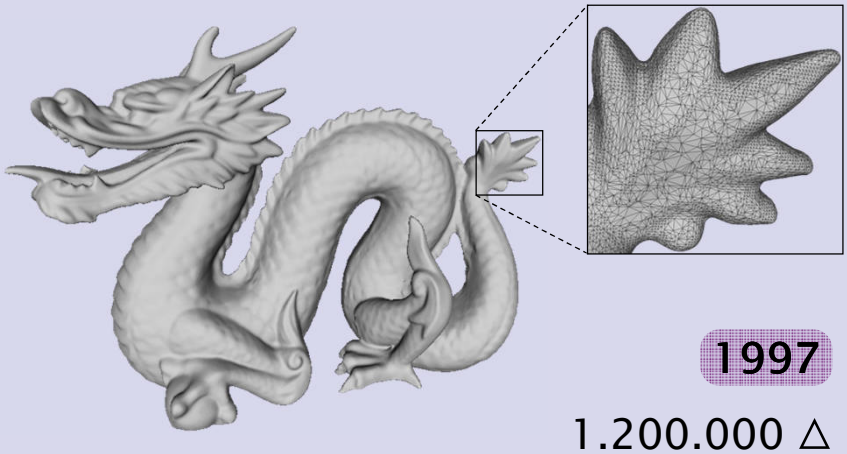
Meshes: complessità crescente



1994

70.000 △

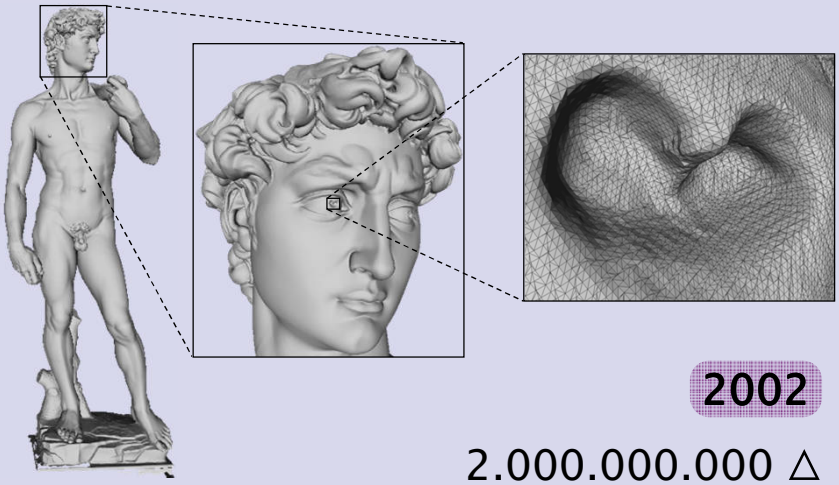
Meshes: complessità crescente



1997

1.200.000 △

Meshes: complessità crescente

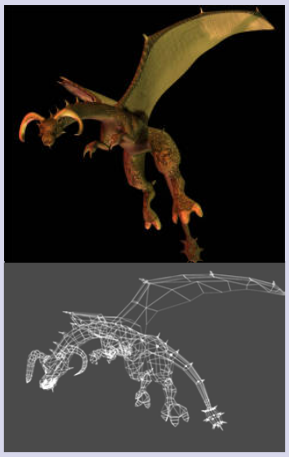


2002

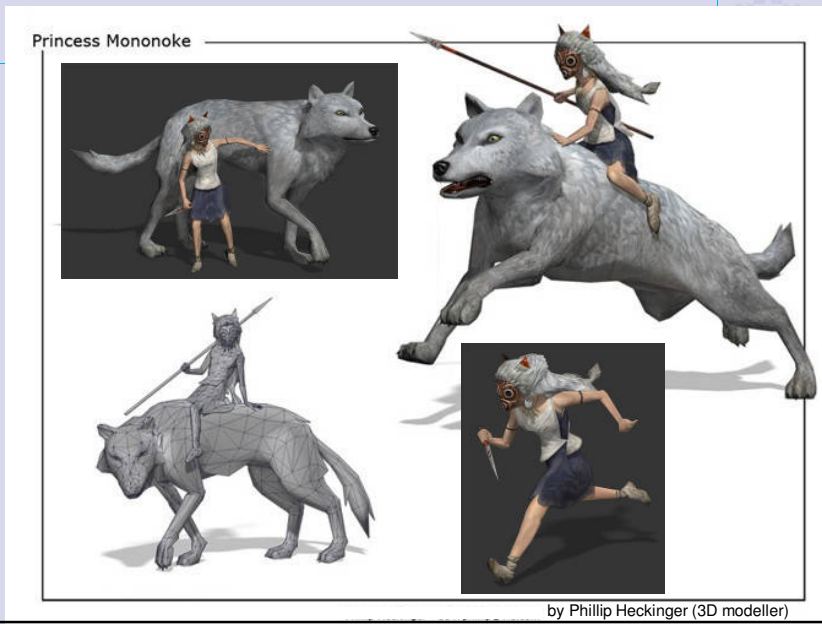
2.000.000.000 △

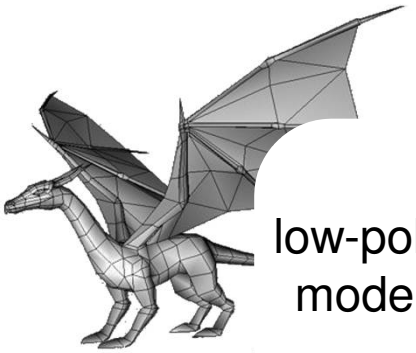
Ma… in ambiente games

- LOW POLY MODELLING!



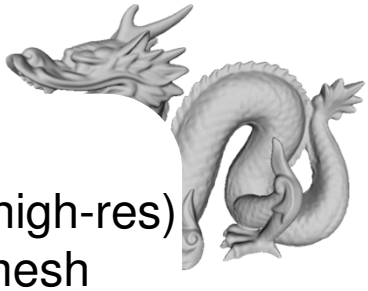
Low-poly modelling

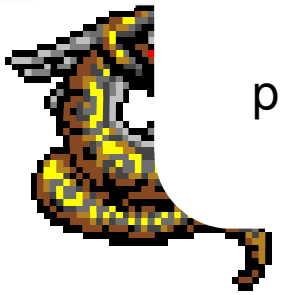




low-poly models : (high-res) mesh

=





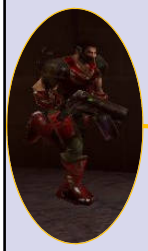
pixel art : (high res) images

Solomons's key (1986, Temco) on Z80

reminder:
per tutti gli '80,
il principale **asset** dei games
è consistito da
sprites/tiles in **pixel art** ...

Metal Slug (1996, Nazca Copr), on Neo Geo (SNK)

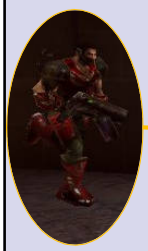
Anche nei games



800 △

Unreal Torunement (1999)

Anche nei games



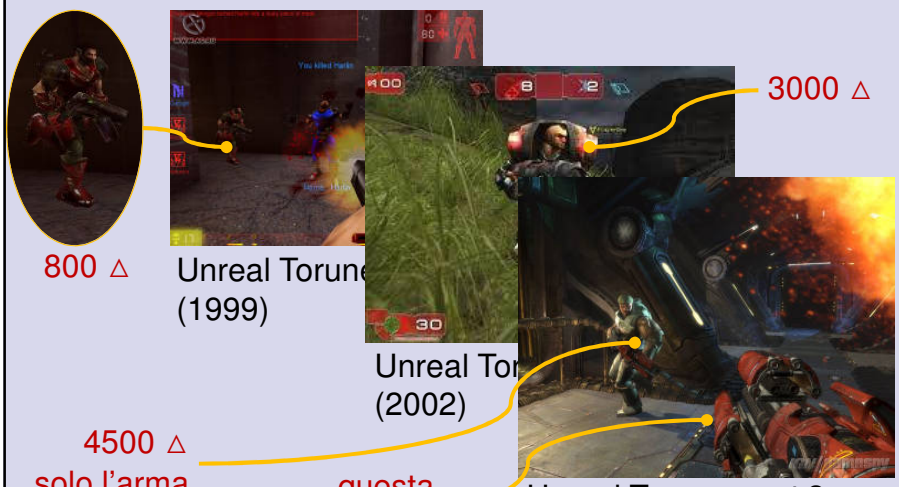
800 △

Unreal Torune (1999)

3000 △

Unreal Torunement 2K3 (2002)

Anche nei games



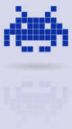
Anche nei games

Come rappresento una mesh?
(quali strutture dati)

- Una tri-mesh è un insieme di triangoli adiacenti
- Modo **diretto**:
  - un vettore di triangoli
  - e per ogni triangolo tre vertici
  - e per ogni vertice tre coordinate

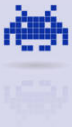
  - Ma: replicazione dati
    - poco efficiente in spazio
    - oneroso fare updates

Come rappresento una mesh? (quali strutture dati)



- Modo **indexed**:
  - Geometria: array di vertici
    - in ogni vertice, posizione e attributi
  - Attributi:
    - coi vertici
      - (e.g. campi della classe “vertice”)
  - Connettività:
    - Array di triangoli
    - Per ogni triangolo:
      - tripletta di **indici** a vertice

Come rappresento una mesh? (quali strutture dati)



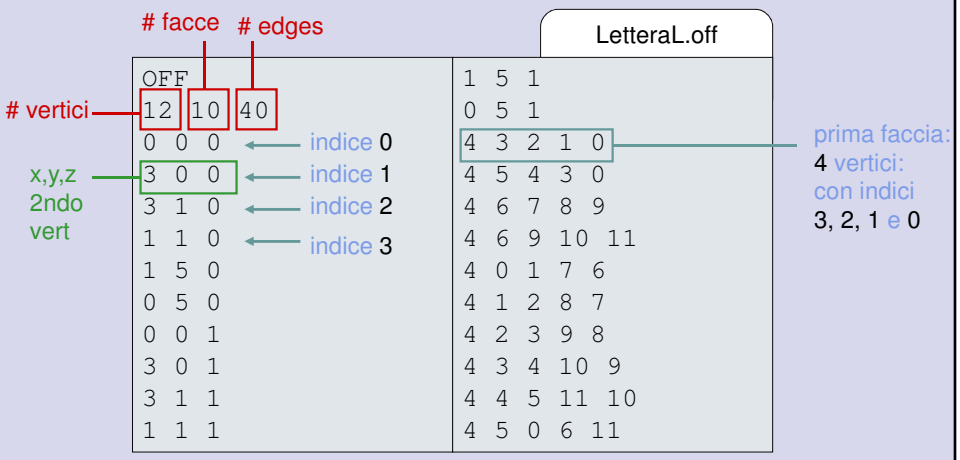
- Modo **indexed**:

```
class Vertex {
  vector3 pos;
  rgb color;   /* attribute 1 */
  vector3 normal; /* attribute 2 */
};

class Face{
   int vertexIndex[3]
};

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

Esempio di mesh indexed:
guardiamo dentro un file in formato OFF

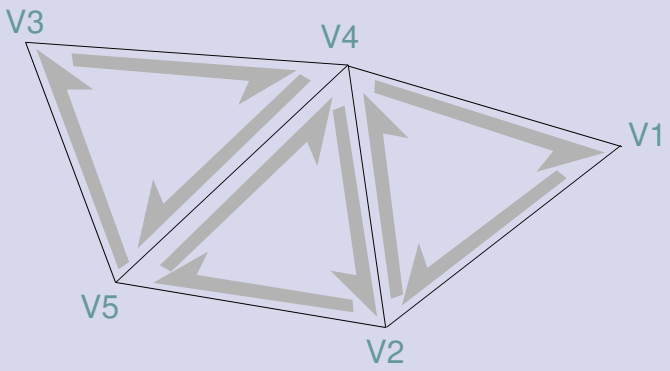



```
OFF
12 10 40
0 0 0
3 0 0
3 1 0
1 1 0
1 5 0
0 5 0
0 0 1
3 0 1
3 1 1
1 1 1
```

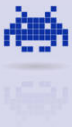
```
1 5 1
0 5 1
4 3 2 1 0
4 5 4 3 0
4 6 7 8 9
4 6 9 10 11
4 0 1 7 6
4 1 2 8 7
4 2 3 9 8
4 3 4 10 9
4 4 5 11 10
4 5 0 6 11
```

Come rappresento una mesh?
(quali strutture dati)

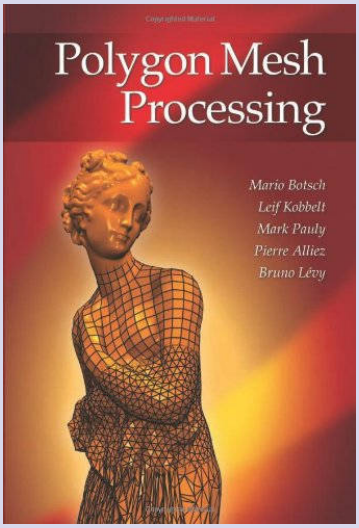
- Alternativa x storare la *connettività* : come vettore di **half-edges**



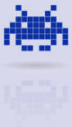
Mesh processing aka Geometry Processing



- Un buon manuale x programmare mesh procssing:



Mesh processing aka Geometry Processing



Librerie:

- VCG-Lib *(CNR,*  *)*
  - Vision and Computer Graphic Lib

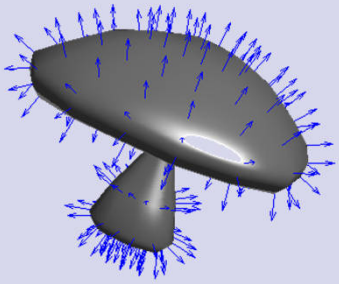
- OpenMesh *(RWTH,*  *)*
  - + open flipper

- CGAL *(INRIA,*  *)*
  - Computational Geometry Algorithms Library

(tutte e tre: C++, open-suorce.)

Attributi comuni: normale

- Vettore direzione unitario
- Orientamento della superficie
- Usato per il lighting
- Di solito, calcolato automaticamente dalla geometria
  …
- Ma l'artista decide decido quali edges sono *soft* e quali *hard*

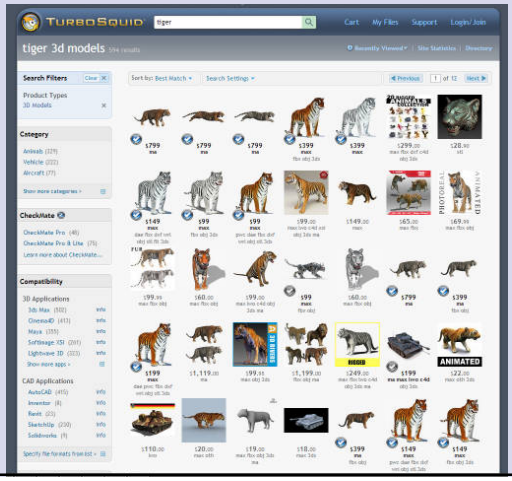


Attributi Comuni: colore

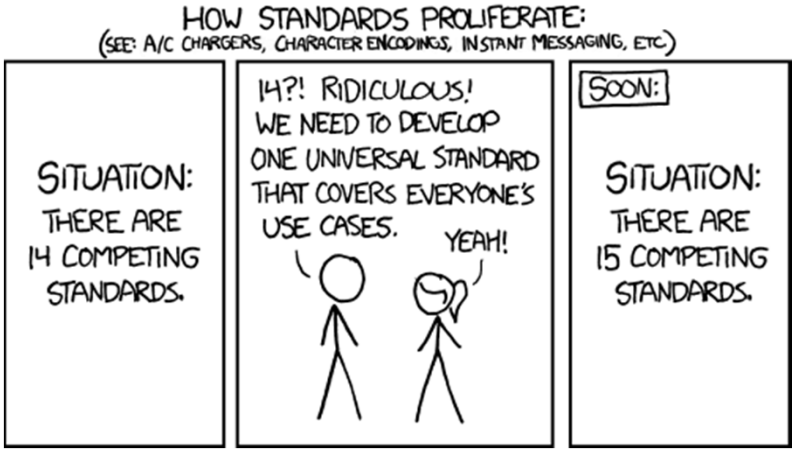
- Utile per:
  - Differenziare modelli
  - Baked global lighting
    (per vertex amient occlusion)

Modelli 3D: come otterenrli

- Come tutti gli asset, possible aquistarli

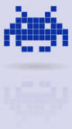


Formati files per mesh ...



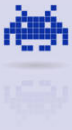
(da **xkcd.com**)

Formati files per mesh
(una Torre di Babele!)



- 3DS - 3D Studio Max file format
- OBJ - Another file format for 3D objects
- MA, MB - Maya file formats
- 3DX - Rinoceros file format
- BLEND - Blender file format
- DAE - COLLADA file format (Khornos)
- FBX - Autodesk interchange file format
- X - Direct X object
- SMD - good for animations (by Valve)
- MD3 - quake 3 vertex animations
- DEM - Digital Elevation Models
- DXF - (exchange format, Autodesk's AutoCAD)
- FIG - Used by REND386/AVRIL
- FLT - MulitGen Inc.'s OpenFlight format
- HDF - Hierarchical Data Format
- IGES - Initial Graphics Exchange Specification
- IV - Open Inventor File Format Info
- LWO, LWB & LWS - Lightwave 3D file formats
- MAZ - Used by Division's dVS/dVISE
- MGF - Materials and Geometry Format
- MSDL - Manchester Scene Description Language
- 3DML - by Flatland inc.
- C4D – Cinema 4D file format
- SLDPTR - SolidWork "part"
- WINGS - Wings3D object
- NFF - Used by Sense8's WorldToolKit
- SKP - Google sketch up
- KMZ - Google Earth model
- OFF - A general 3D mesh Object File Format
- OOGL - Object Oriented Graphics Library
- PLG - Used by REND386/AVRIL
- POV - "persistence of vision" ray-tracer
- QD3D - Apple's QuickDraw 3D Metafile format
- TDDD - for Imagine & Turbo Silver ray-tracers
- NFF & ENFF - (Extended) Neutral File Format
- VIZ - Used by Division's dVS/dVISE
- VRML, VRML97 - Virtual Reality Modeling Language (RIP)
- X3D - tentato successore di VRML
- PLY - introdotto by Cyberware – tipic. dati range scan
- DICOM - Dalla casa omonima – tipic. dati CAT scan
- Renderman - per l'omonimo visualizzatore
- RWX - RenderWare Object
- Z3D - ZModeler File format
- etc, etc, etc...

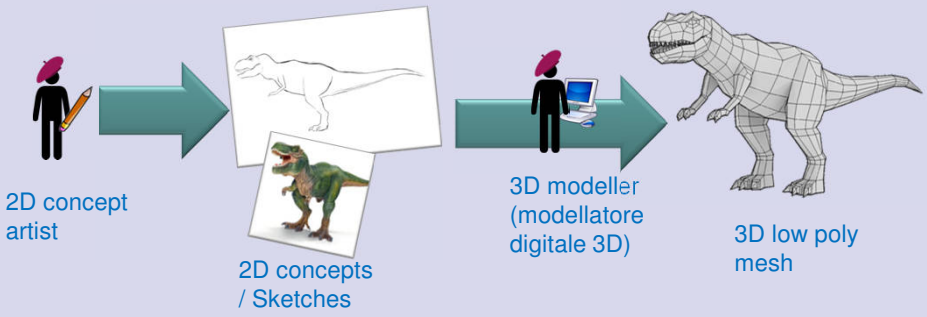
Formati per mesh più diffusi nei games



- .OBJ (wavefront)
  - ☺ indexed, normali, uv-mapping
  - ☺ semplice da parsare, compreso da tutti
  - ☹ no colori (solo indice materiale x faccia), no rigging
- .SMD (VALVE)
  - ☺ animazioni scheletali, normali, skinning
  - ☹ no indexed, no colori
- .MD3 (Quake, IDsoft)
  - ☺ animaz vertx animations, normali
  - ☹ non diffusissimo, no supporto colori
- .3DS (AUTODESK.)
  - ☺ abbatanza completo (colori, uv-mapping, indexed, matriali, tessiture)
  - ☹ no: normali, limite al numero di vert (64K)
  - ☹ abbastanza difficile da parsare, dunque non tanto supportato
- .COLLADA (KHRONOS GROUP)
  - ☺ completissimo, nato apposta per essere interscambio
  - ☹ quasi impossiile da parsare completamente
- .PLY (cyberware)
  - ☺ customizzabile, facile da parsare
  - "accademico", non diffusissimo

Modelli 3D: come ottenerli

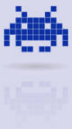
- Modellazione digitale manuale
  - Lavoro dei modellatori digitali



Tecniche di modellazione digitale di modelli 3D

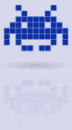
- Tecniche:
  - Low poly diretta
    - e.g. wings3D
  - Subdivision surfaces
    - e.g. con blender
  - Digital sculpting
    - e.g. con Z-brush

Mesh editing: applicativi generici

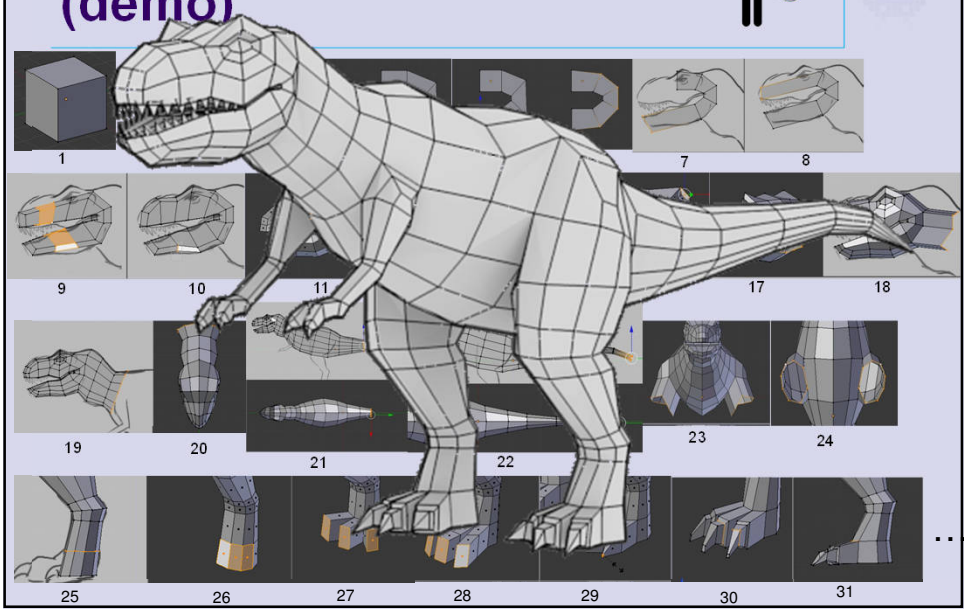


- 3D Studio Max (autodesk) , Maya (alias) , Cinema4D (maxon)
  - generici, potenti, completi
- Blender
  - idem, ma open-source e freeware (simile a: Gimp VS. Adobe Photoshop per 2D images)
- MeshLab
  - open-source, grande collezione algoritmi di geometry processing …
- AutoCAD (autodesk), SolidWorks (SolidThinking)
  - per CAD
- ZBrush (pixologic), + Sculptris , Mudbox (autodesk)
  - metafora scultura virtuale, specializzato in ritocco manuale dettagli hi-freq, bumpmapping, normalmaps…
- Wings3D
  - open-source, piccolo, specializzato in low-poly editing, subdivision surfaces
- [Rhinoceros]
  - parametric surfaces (NURBS)
- FragMotion
  - specializzato per mesh animate
- …
- + moltissimi strumenti per contesti specifici
  - (editing di umani, di interni architetturali, di paesaggi, o editor specifici per game-engines, etc...)

Low poly diretta (demo)

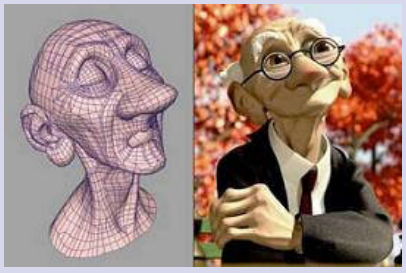


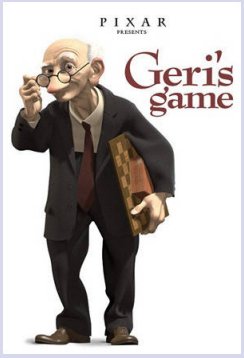
Low poly diretta (demo)

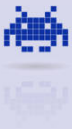


Tecniche di modellazione digitale di modelli 3D

- Subdivision surfaces
  - Raffinamento progressivo della mesh da lowest res ➔ hi res
  - Ottimo per oggetti dall'aspetto smooth, organico e "pulito"

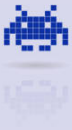






Superfici di suddivisione

- Modo molto diffuso per costruire mesh
  - 1: fare mesh di controllo
    - a bassa risoluzione
    - "a mano"
  - 2: raffinarla automaticamente
    - iterativamente
    - (ad ogni interazione si aggiungono facce e vertici)
- molti schemi matematici differenti
  - con diverse peoprietà



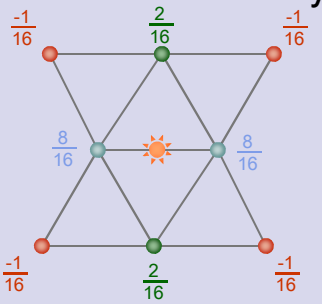
Superfici di suddivisione

- Esempio: schema butterfly (per mesh triangolari)
  - e' uno degli schemi 1=>4
    (in un passo di suddivisione, da ogni triangolo se ne ottengono 4)
    (aggiunta di un vertice per ogni edge)

- MA... quali coordinate assegnare al nuovo vertice?
  Ogni schema di suddivisone ha la sua formula. Ad esempio...

Superfici di suddivisione

- Esempio: schema butterfly



$$POS(\odot) = \frac{8}{16}(POS(\bigcirc) + POS(\bigcirc)) + \frac{2}{16}(POS(\bigcirc) + POS(\bigcirc)) + \frac{-1}{16}(POS(\bigcirc) + POS(\bigcirc) + POS(\bigcirc) + POS(\bigcirc))$$

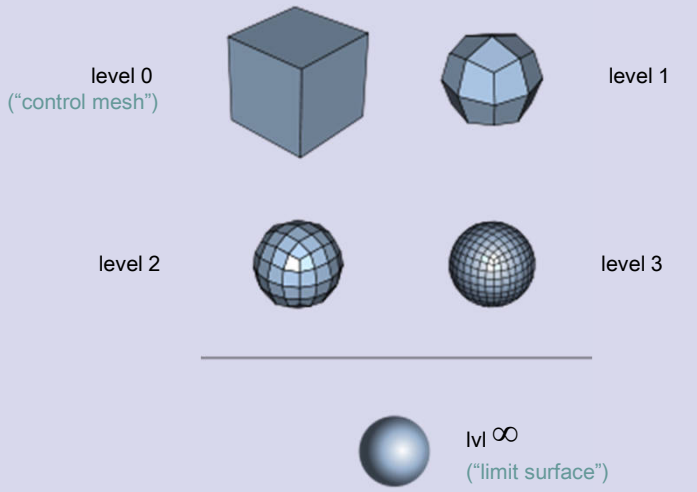
Superfici di suddivisione

Ad ogni passo di suddivisione

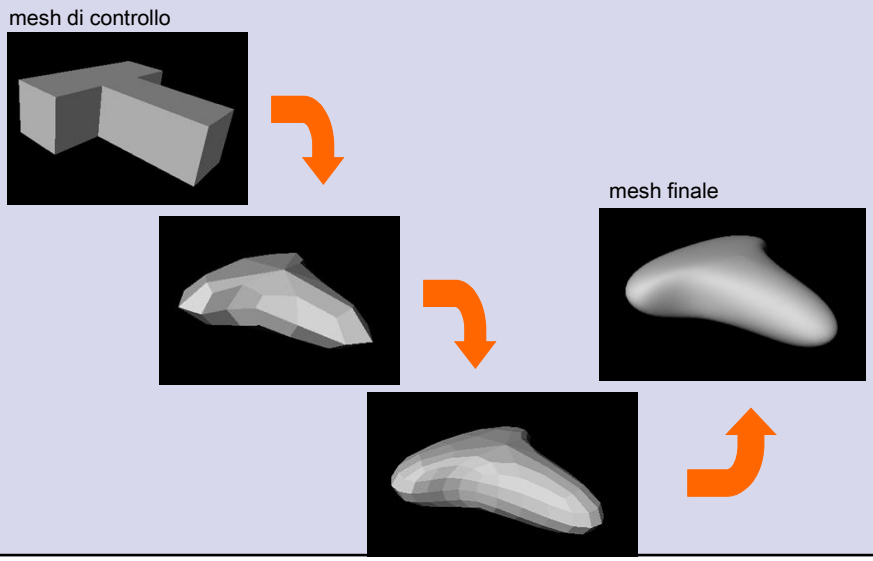
- (x,y,z) dei nuovi vertici inseriti
  - formula (estrapolazione)
- (x,y,z) dei vecchi vertici
  - si tiene la vecchia pos (schemi "interpolativi")

  oppure
  - formula (estrapolazione) (schemi "approssimativi")

Esempio: con schema Catmull-Clark



Superfici di suddivisione



Superfici di suddivisione

Anche iterativamente:

1- Modellare “control mesh”

(editing manuale)

2- Suddivisione

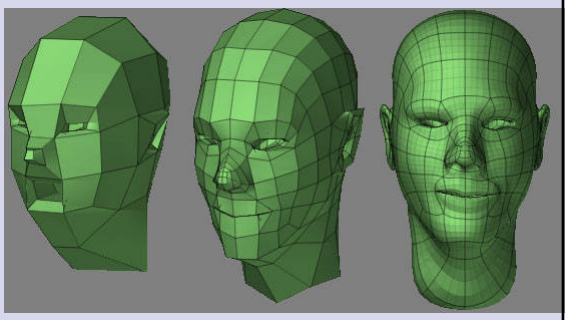
(un passo)

3- Ritocco!

(editing manuale)

4- Goto 2

(fino a raggiungimento risultato voluto alla risuolzione voluta)

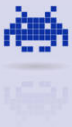


Molti schemi…

- Catmull-Clark
- Doo-Sabin
- Loop
- sqrt(3)
- Butterfly
- Mid-edge
- ….

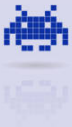
recente aumento di popolarità (GPU friendliness)

Differenze fra gli schemi di suddivisione



- interpolativi VS approssimativi
- solo triangoli, solo quads, qualunque cosa
- incremento complessità
  - (per ogni passo di suddivisione)
- proprietà della limit surface
  - (esistenza, smoothness)
- esistenza forma chiusa per la limit surface
  - (esatta o approssimata)
- …

Tecniche di modellazione digitale di modelli 3D



- Tecniche:
  - Low poly diretta
    - e.g. wings3D
  - Subdivision surfaces
    - e.g. con blender
  - Digital sculpting
    - e.g. con Z-brush

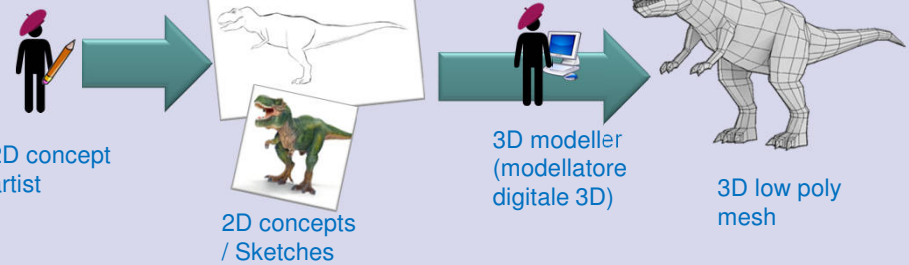
DEMO

Digital Sculpting

cisel
(scalpello)

Modelli 3D: come ottenerli

- Modellazione digitale manuale
  - Lavoro dei modellatori digitali



2D concept artist

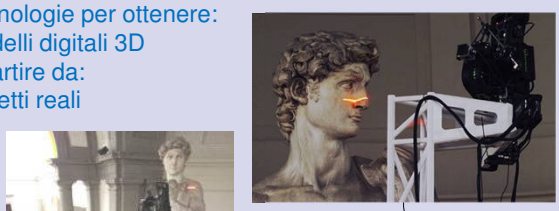
2D concepts / Sketches

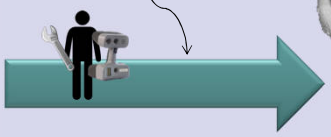
3D modeller (modellatore digitale 3D)

3D low poly mesh

Modelli 3D: come ottenerli

- Attraverso 3D scanning
  - Tecnologie per ottenere: modelli digitali 3D a partire da: oggetti reali





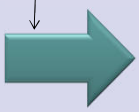
3D scanning

Modelli 3D: come ottenerli

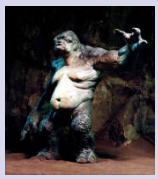
- 3D scanning
  - A.k.a. *automatic 3D model acquisition*
  - Molte tecnologie diverse
    - Laser scanners
    - Time of flight
    - Structured light (kinect)
    - …
  - Caratteristiche diverse
    - Qualità risultati
      - Rumore / risoluzione
    - Automatismo
    - Invasività
      - Markers? Powder?
    - Real time? (kinect)
    - Costo
    - Dimensione massima oggetti
      - (full body scanner?)

Modelli 3D: come ottenerli

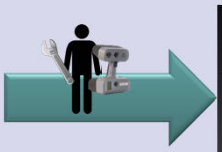
- Attraverso 3D scanning



Scultore (fisico)



Modello reale

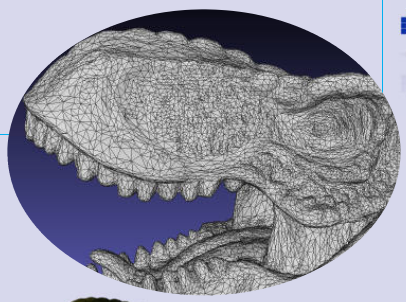


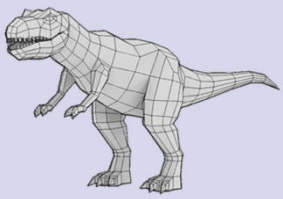
3D scanning

Hi res model

Modelli 3D: come ottenerli

PERFECT for games!
(much easier to: animate, re-edit, uvmap, ...)





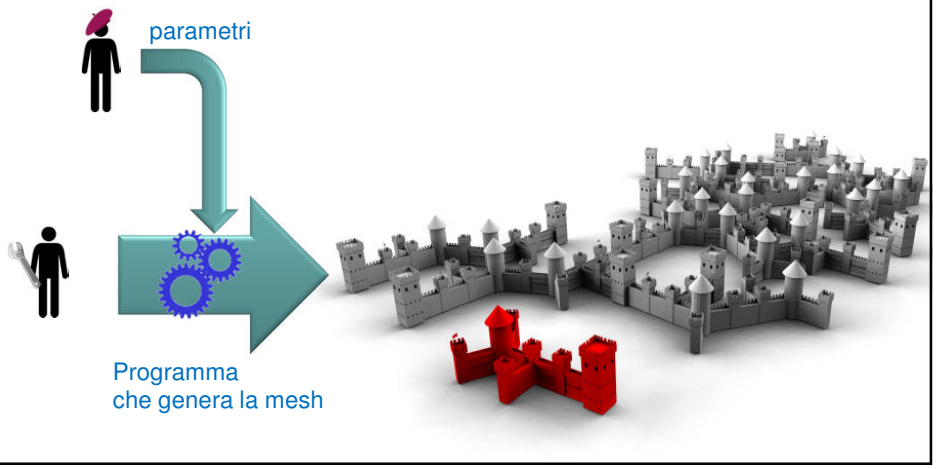
vs

(artistic)
manually edited
3D low poly mesh

Hi res model
30000 tri
(by artec3d)

Modelli 3D: come ottenerli

- Modellazione procedurale



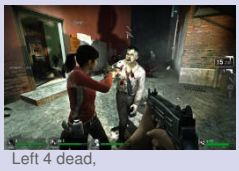
Parentesi:

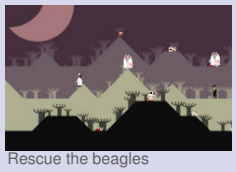
Procedural generation: ottimo per games

- Concetto: invece di avere un asset, avere un programma che lo crea dinamicamente
  - Modellazione procedurale
  - AI procedurali, boss procedurali…
  - Livelli procedurali
  - Terreni procedurali
  - Musica procedurale
  - Scene procedurali

Minecraft, Mojang, 2009


Elite, Acornsoft, 1984


Left 4 dead, Valve, 2008

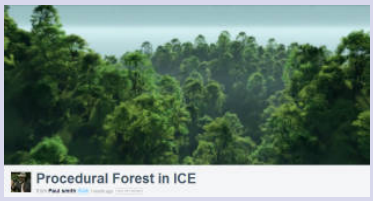

Rescue the beagles 16x16, 2008

- Vantaggi: varietà, no RAM, …

Parentesi:

Procedural generation: ottimo per games

- Concetto: invece di avere un asset, avere un programma che lo crea dinamicamente
  - Modellazione procedurale
  - AI procedurali, boss procedurali…
  - Livelli procedurali
  - Terreni procedurali
  - Musica procedurale
  - Scene procedurali

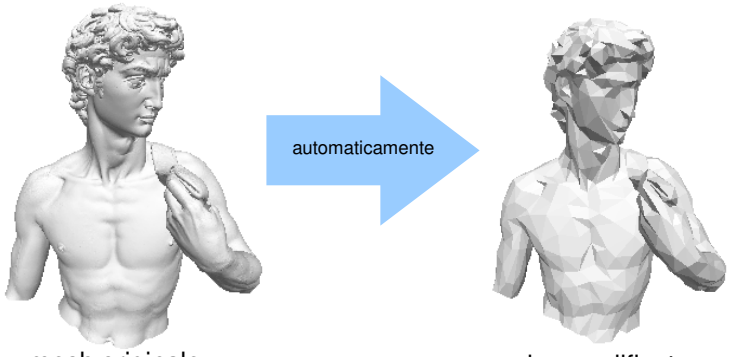


Elite, 1984

- Vantaggi: varietà, no RAM, …

Semplificazione automatica di modelli 3D

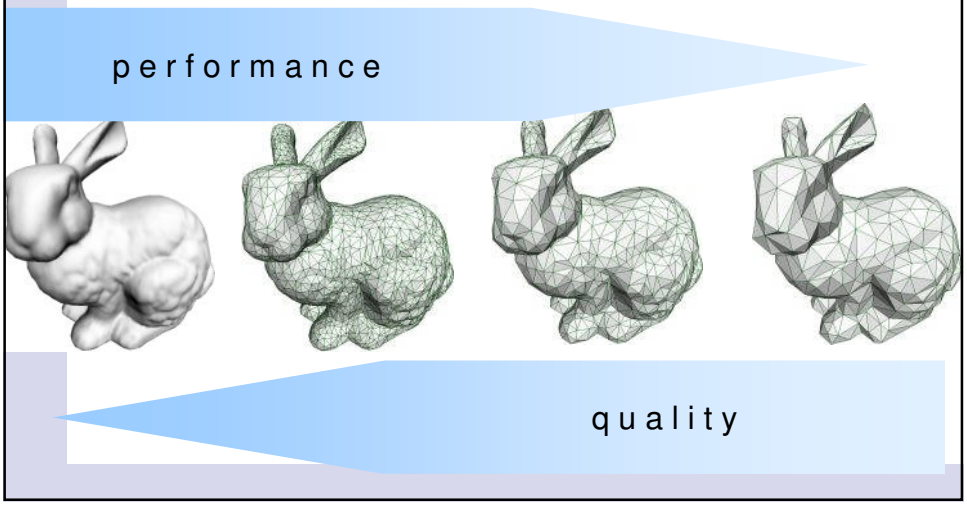
- parametri:
  - un errore massimo
  - o un numero di facce obiettivo



mesh originale
500K triangoli

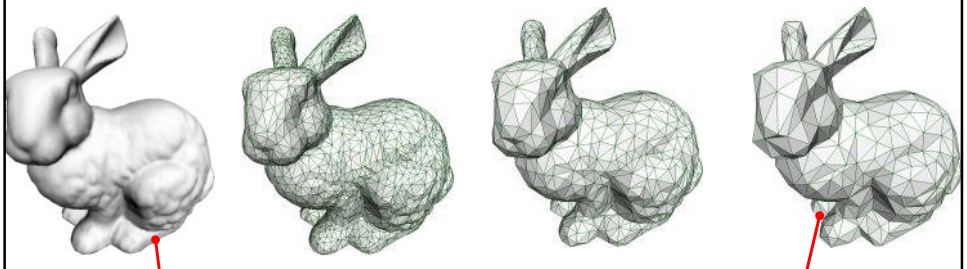
mesh semplificata
2K triangles

Semplificazione automatica di modelli 3D



Semplificazione automatica

Una piramide di Livelli di Dettaglio



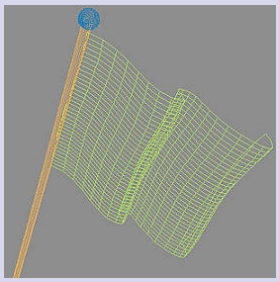
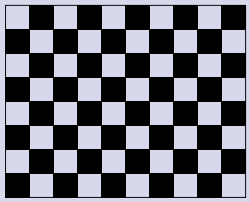
LoD pyramid

Demo

Semplificazione automatica

- Molte tecniche diverse
  - Adattive oppure no
    - usare piu' triangoli dove c'e' bisogno (es non nelle zone cmq piatte)
    - oppure no
  - Errore massimo introdotto:
    - misurato e/o limitato
    - oppure no
  - Topologia:
    - mantenuta
    - oppure no
  - Streaming
    - Possibile
    - Oppure no
  - ...

Texture mapping


+

=


geometria 3D
(insieme di quadrilateri)

RGB texture 2D
(color-map)

Altro esempio di color-map

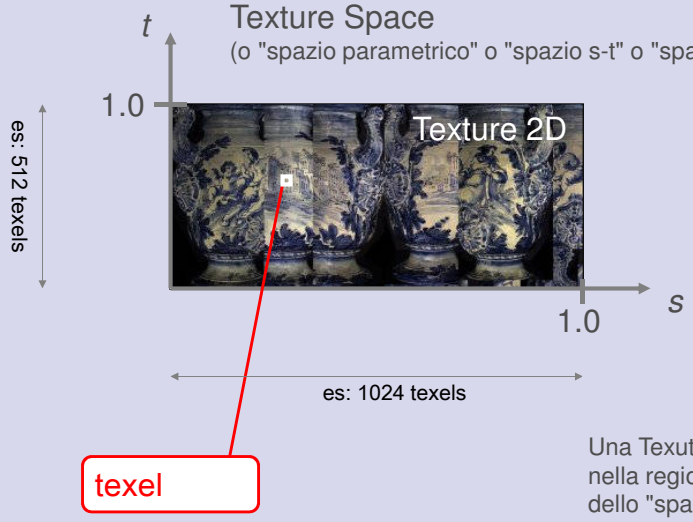

+

=


Altri esempi di color-map



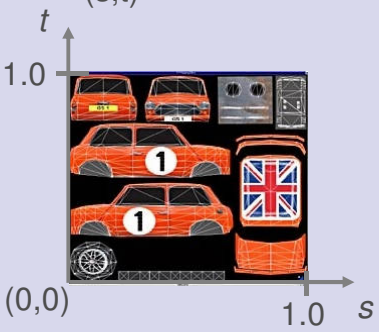

Notazione



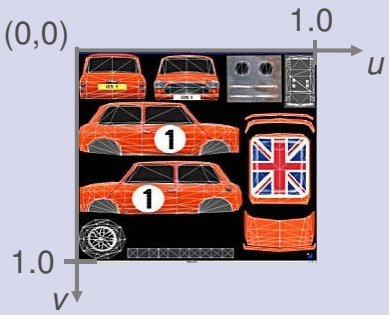
Una Texutre e' definita
nella regione [0,1] x [0,1]
dello "spazio parametrico"

Due notazioni

Texture Space in OpenGL (s,t)

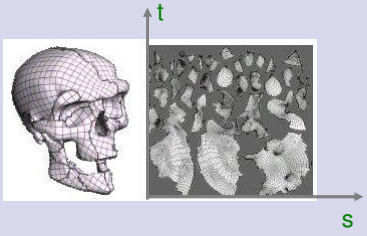


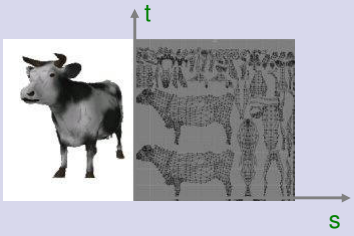
Texture Space in DirectX (u,v)



Task: "u-v mapping" di una mesh ("u-v" == "s-t")

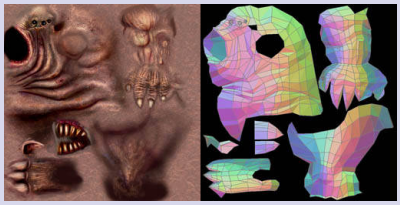
- Assegnare una coppia di coordinate texutres ad ogni vertice della mesh
  - In preprocessing





Problema difficile: “u-v mapping” (“u-v” == “s-t”)

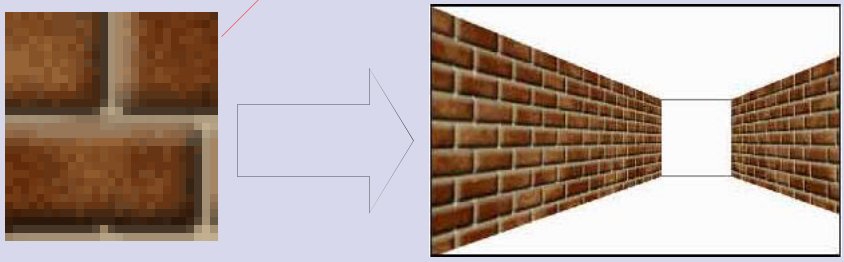
fatto a mano,
oppure automatizzato



Tessiture ripetute (tiled textures)

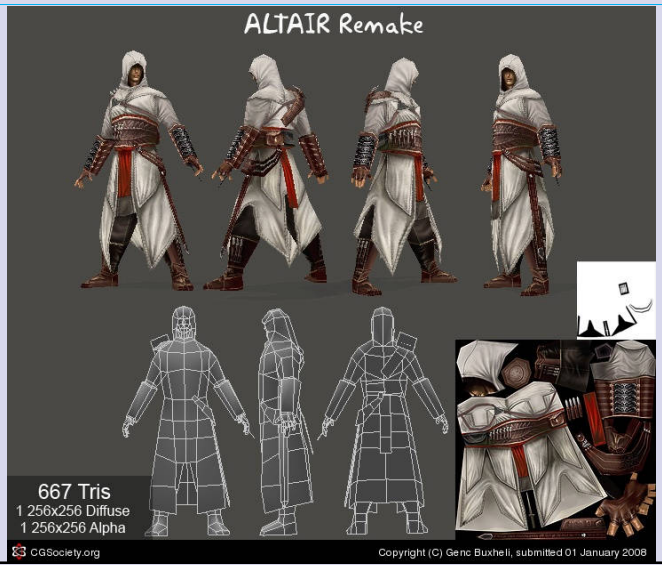
- Tipico utilizzo:

Nota: deve essere un’immagine “TILEABLE”



Molto efficiente in spazio!

Alpha mapping (texels = lvl trasparenza)

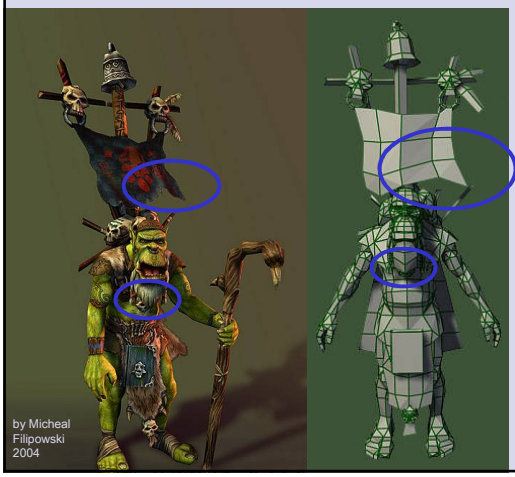


Alpha map

RGB map

Alpha mapping (texels = lvl trasparenza)

- es: drappi, barba...



tessitura

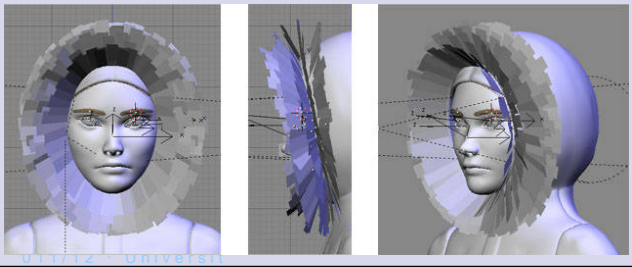
Texture mapping e Alpha Test

- es: alberi, foliage

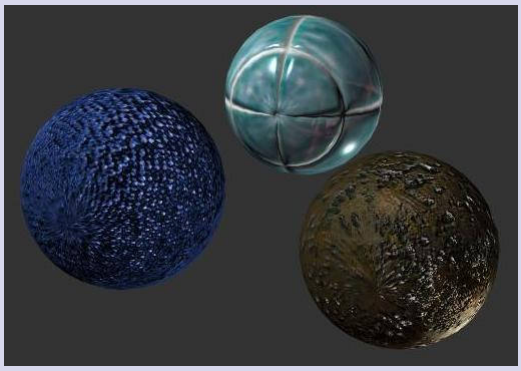
Texture mapping e Alpha Test

- Es: pelo, pellicce

tessitura (ripetuta)



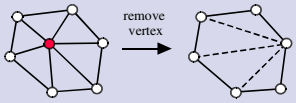
Bump-Mapping (see demo)

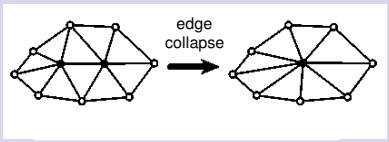


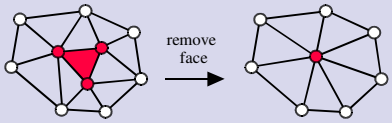
stessa geometria (una sfera)
bumpmaps diverse

Semplificazione automatica

- Strategie completamente diverse
  - Approcci iterativi
    - repeat
      - compi l'azione di semplificazione atomica meno costosa (in termini di errore aggiunto)
      - aggiorna costi
    - until (obiettivo raggiunto)
      - es: numero faccie, errore

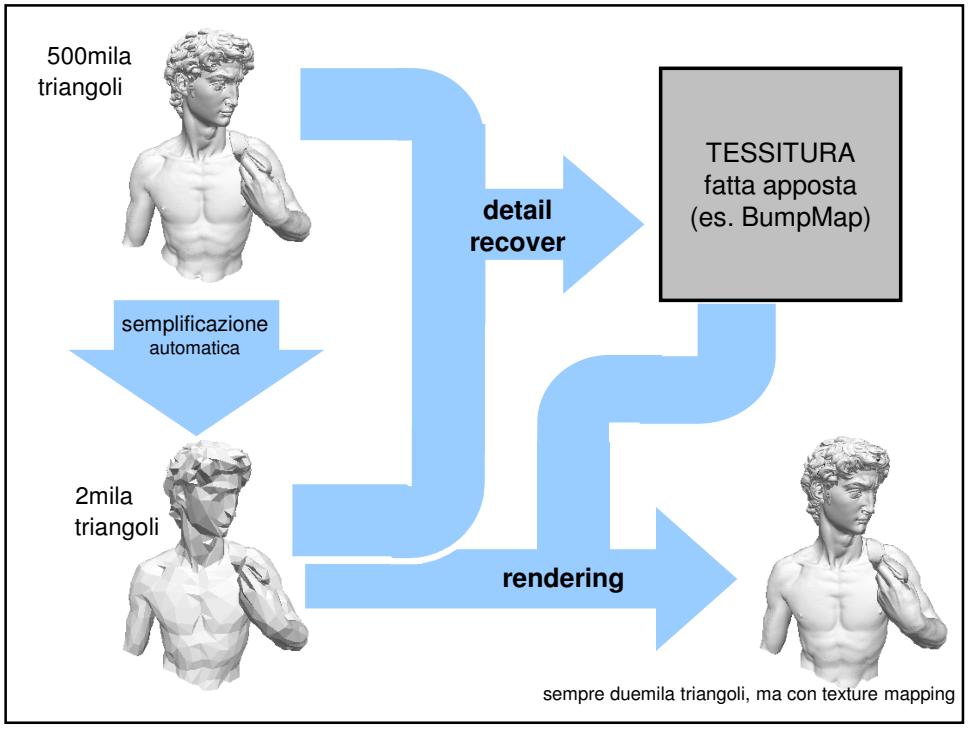


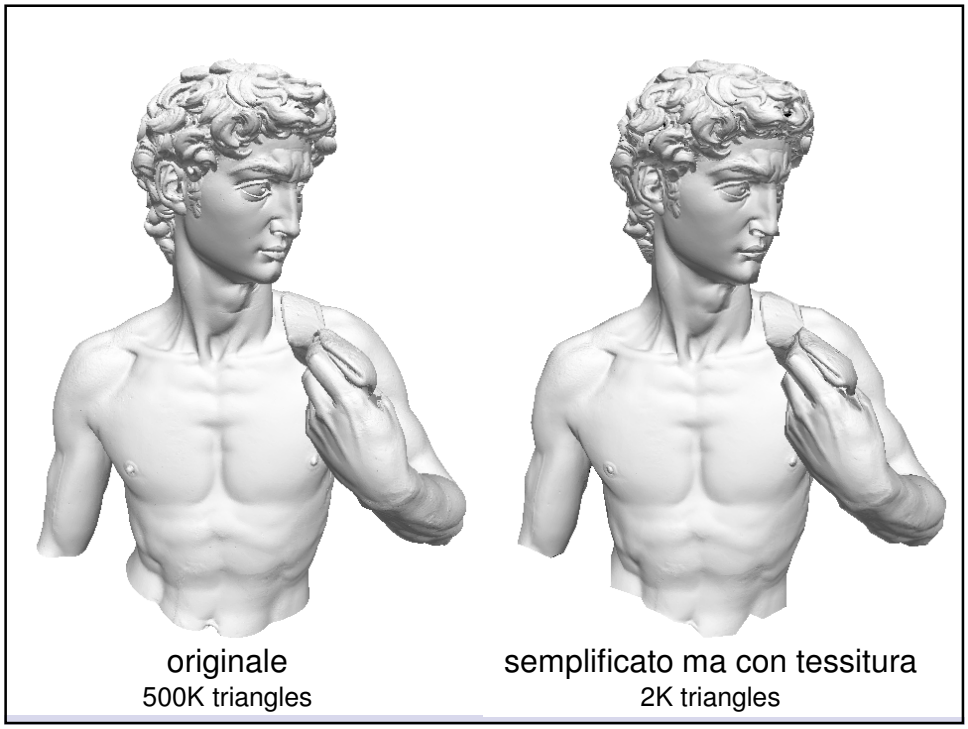




Detail preservation (o "texture for geometry")

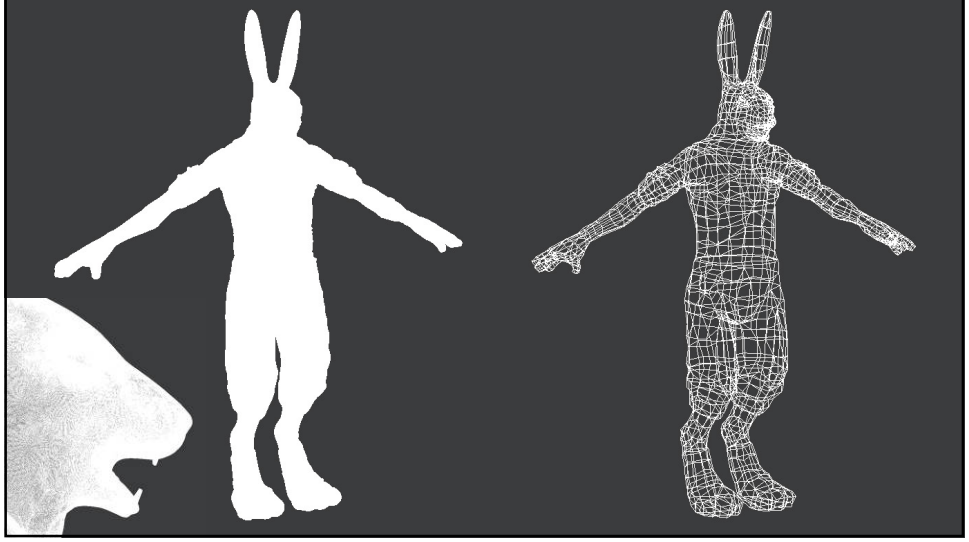
- Idea:
  - semplificare una mesh
  - sintetizzare una tessitura
  - per ripristinare il dettaglio perso durante la semplificazione


500mila triangoli
detail recover
TESSITURA fatta apposta (es. BumpMap)
semplificazione automatica
2mila triangoli
rendering
sempre duemila triangoli, ma con texture mapping

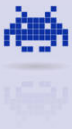

originale
500K triangles
semplificato ma con tessitura
2K triangles

Esempio

Esempio



Mesh: task tipici nella game industry



- Semplificazione automatica
  - LOD construction
- Light baking
  - Precomputazione Luce
  - Tipico esempio: Ambient Occlusion
- U-V mapping
  - parametrizzazione
- Texturing
  - creazione tessiture
- Rigging / Animation
  - linear blend skinning