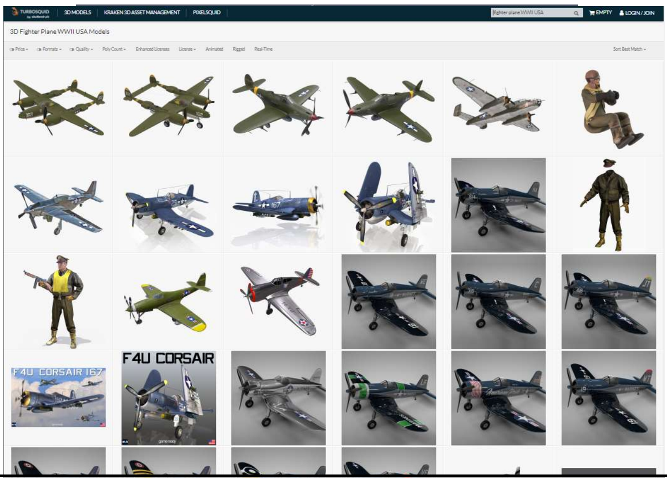
Course Plan

lec. 1: Introduction ●
lec. 2: Mathematics for 3D Games ●●●●●●
lec. 3: Scene Graph ●
lec. 4: Game 3D Physics ●●●◖+ ◗●
lec. 5: Game Particle Systems ◖
lec. 6: Game 3D Models ◗●
lec. 7: Game Textures ●●
lec. 9: Game Materials ◖
lec. 8: Game 3D Animations ◗●●
lec. 10: Networking for 3D Games ●
lec. 11: 3D Audio for 3D Games ●
lec. 12: Rendering Techniques for 3D Games ●
lec. 13: Artificial Intelligence for 3D Games ●

63

3D models: suorces

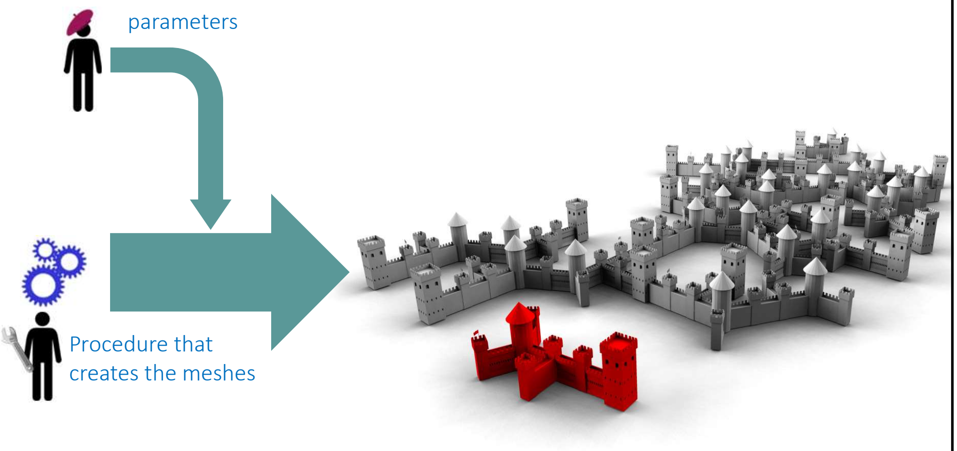
- Like any asset, often just bought / off-sourced



3D Fighter Plane WWII USA Models

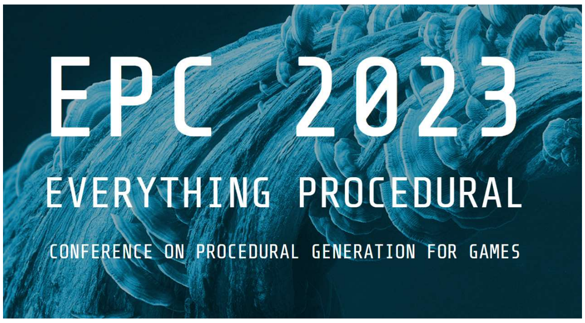
67

Sources for 3D models: procedural modelling



68

Procedural modelling – see also…

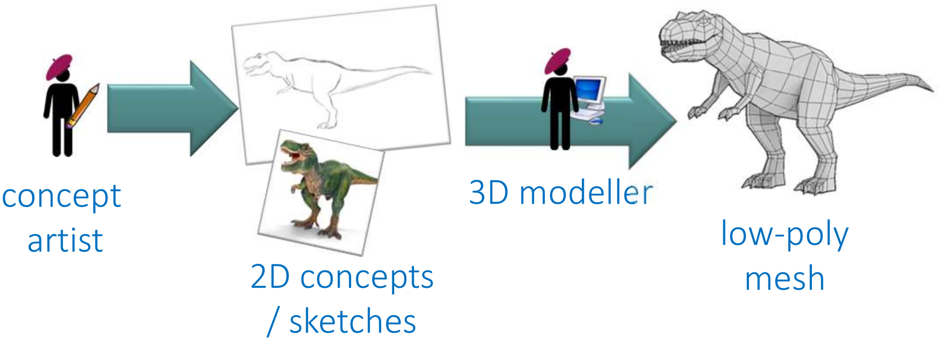


http://everythingprocedural.com/

this week
Game-of-the-Week

69

3D models: authoring pipeline



70

Mesh: authoring

- Task of the 3D modeller
  - A type of digital artist
- Popular 3D modeling approaches:
  - Manual low-poly modelling
    - e.g. with wings3D
  - Subdivision surfaces
    - e.g. with blender
  - Digital sculpting
    - e.g. with Z-brush

71

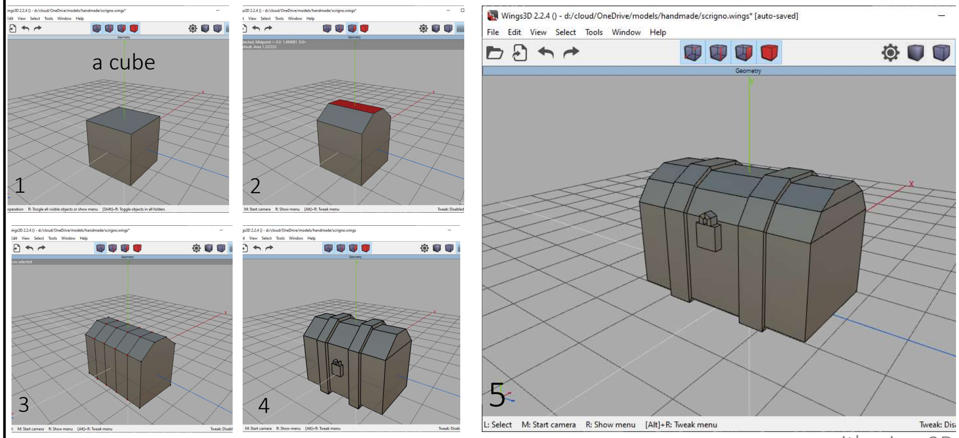
Mesh authoring (aka 3D modelling): a few applications

used in classroom demos

- 3D Studio Max (autodesk) ,
  Maya (autodesk) ,
  Cinema4D (maxon)
  Lightweight 3D (NewTek),
  Modo (The Foundry) , ...
  - all-purpose, powerful, complete
- Blender
  - the same, plus open-source and freeware (compare: Gimp VS. Adobe Photoshop for 2D images)
- MeshLab
  - open-source, big collection of geometry processing algorithms ...
- AutoCAD (autodesk),
  SolidWorks (SolidThinking)
  - for CAD
- ZBrush (pixologic)
  (+ Sculptris alpha),
  Mudbox (autodesk)
  - Sculpting (inclusing texturing)
- Wings3D
  - low-poly modelling (& subdivision surfaces)
    open-source, small, specialized
- [Rhinoceros]
  - parametric surfaces (NURBS)
- FragMotion
  - small, specialized on animated meshes
- + a many more for specific contexts
  - editing of human models, of architectural interiors, environments, or specific editors for game-engines, etc...

72

Low-poly modelling (demo)

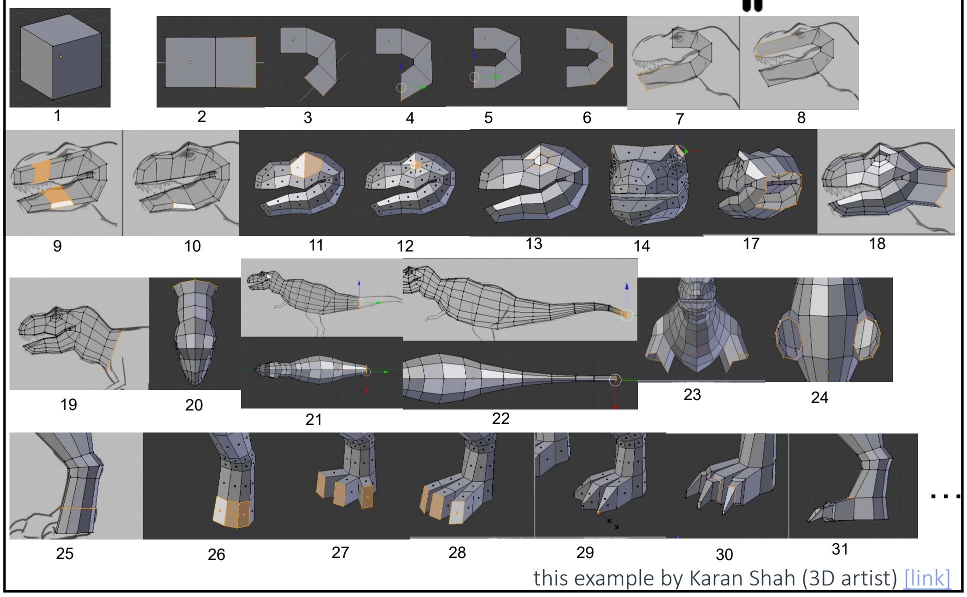

a cube
1
2
3
4
5
Wings3D 2.2.4 () - d:/cloud/OneDrive/models/handmade/scrigno.wings* [auto-saved]
File Edit View Select Tools Window Help
Geometry
L: Select M: Start camera R: Show menu [Alt]+R: Tweak menu

with wings3D

Note: during creation, the meshes can be polygonal instead of triangle based, but is simple to decompose any polygon into triangles
E.g. this can be done by the game engine as a simple preprocessing.

73

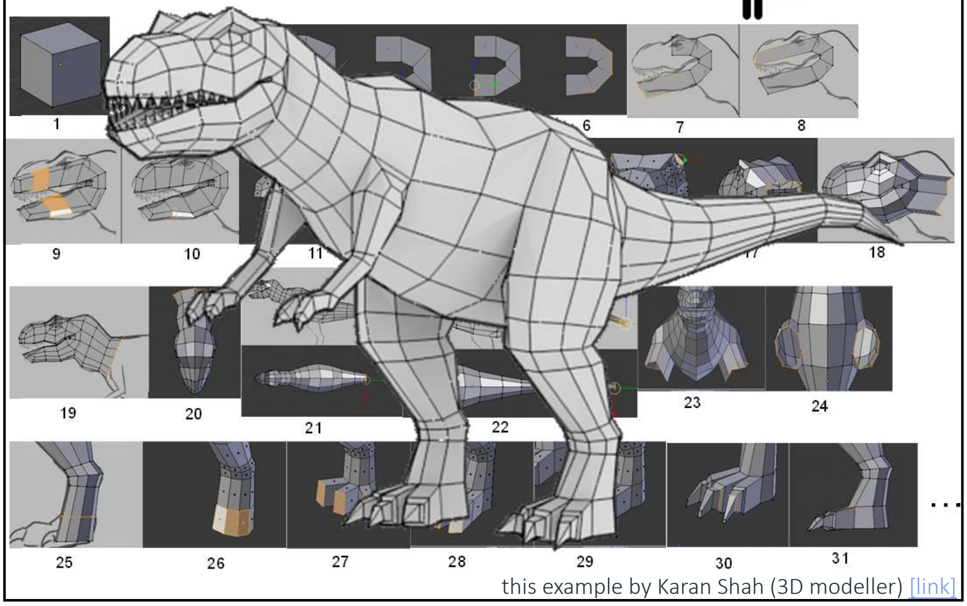
Low-poly modelling



this example by Karan Shah (3D artist) [link]

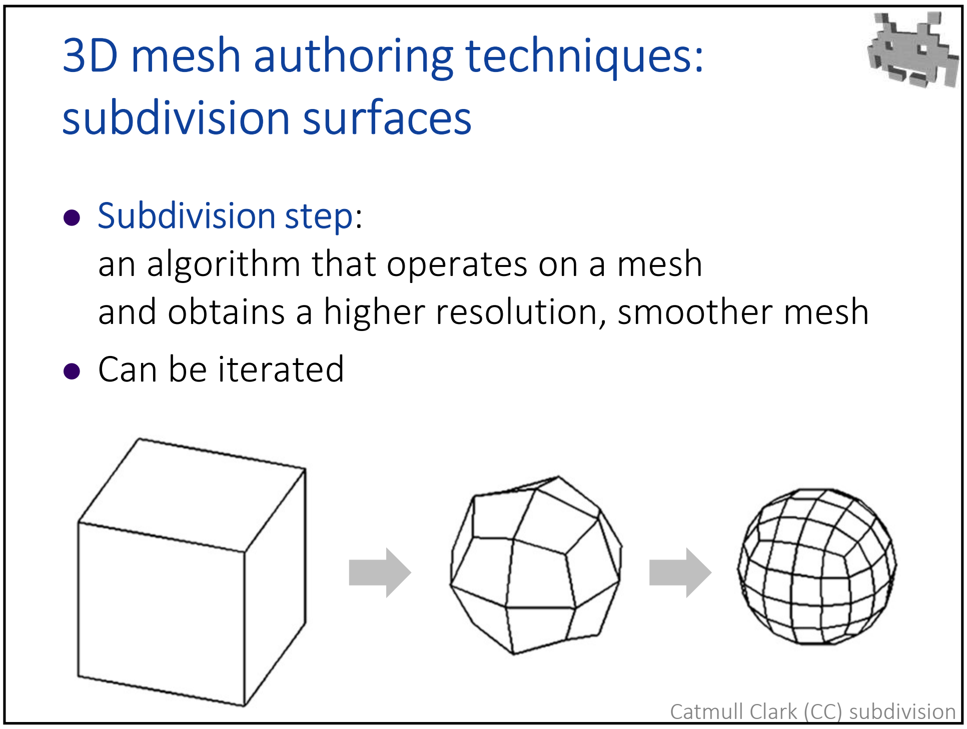
75

Low-poly modelling



this example by Karan Shah (3D modeller) [link]

76

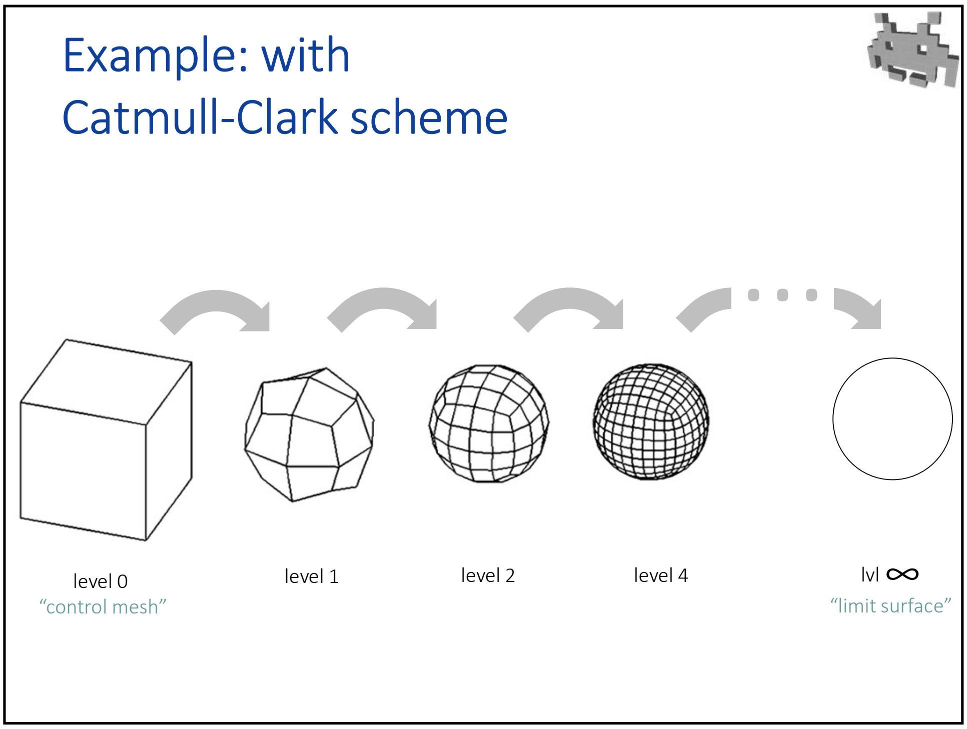


3D mesh authoring techniques: subdivision surfaces

- Subdivision step:
  an algorithm that operates on a mesh
  and obtains a higher resolution, smoother mesh
- Can be iterated

Catmull Clark (CC) subdivision

77



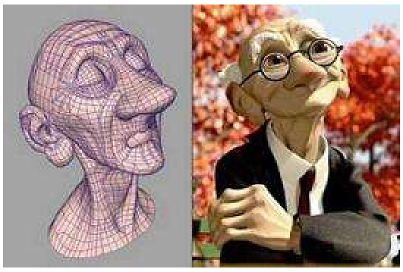
Example: with Catmull-Clark scheme

level 0
“control mesh”
level 1
level 2
level 4
lvl ∞
“limit surface”

78

3D mesh authoring techniques: subdivision surfaces

- Many subdivision algorithms (schemas) exists
  - each with its own properties
- Produces clean, regular meshes
- Excellent for smooth, curved, organic looking objects



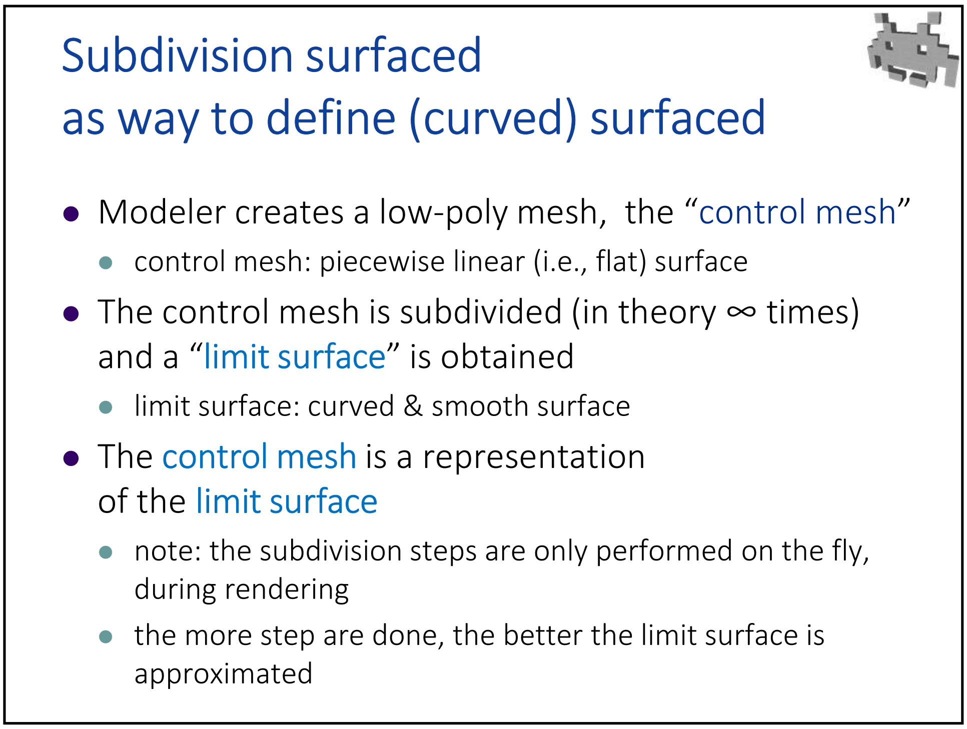
famously pioneered by movie industry (not games):

79

Subdivision surfaces as a tool…

- …to encode smooth surfaces
  - Idea: we encode the control mesh to represent the limit surface
  - use in games: rendering (now, rare – but popular around 2015)
    1. keep control mesh in GPU ram
    2. let 1-3 subdivision steps happen during rendering
- …to author 3D meshes
  - idea: alternate (low-poly) editing and subdivisions steps
  - at first steps: edit global shape
  - at last steps: edit minute details
  - use in games: during asset creation, by artists

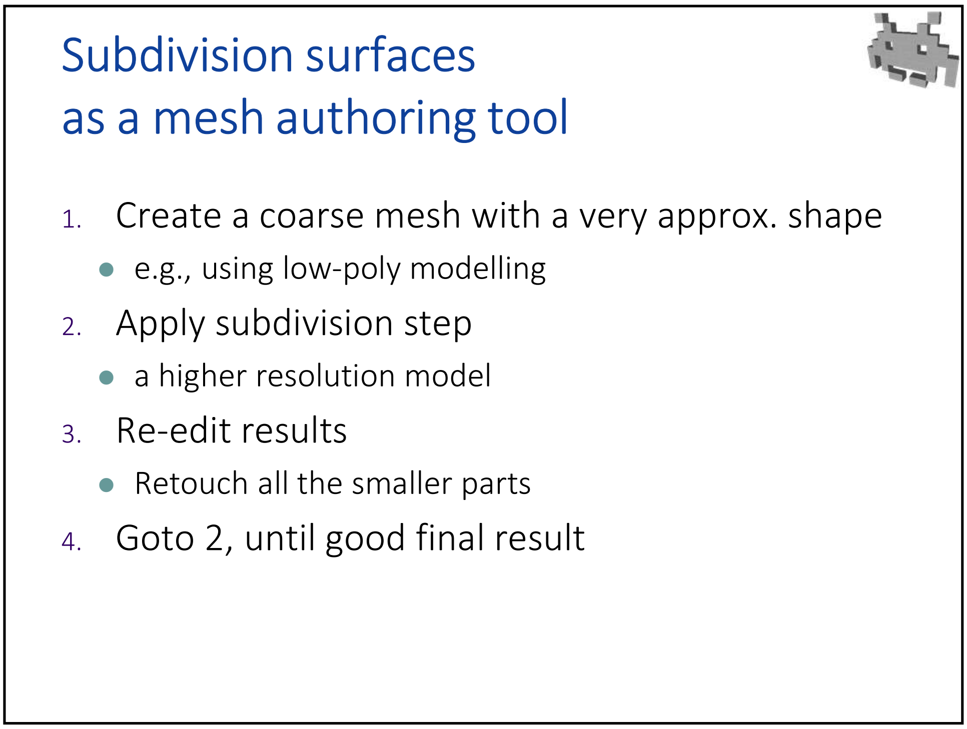
80



Subdivision surfaced as way to define (curved) surfaced

- Modeler creates a low-poly mesh, the “control mesh”
  - control mesh: piecewise linear (i.e., flat) surface
- The control mesh is subdivided (in theory ∞ times) and a “limit surface” is obtained
  - limit surface: curved & smooth surface
- The control mesh is a representation of the limit surface
  - note: the subdivision steps are only performed on the fly, during rendering
  - the more step are done, the better the limit surface is approximated

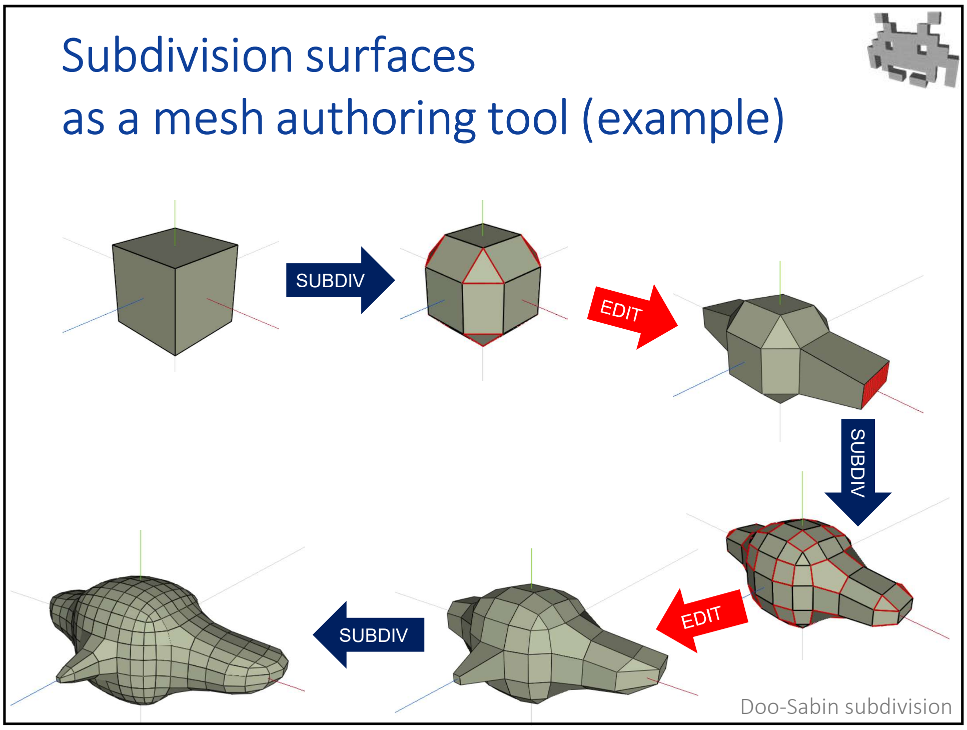
81



Subdivision surfaces as a mesh authoring tool

1. Create a coarse mesh with a very approx. shape
   - e.g., using low-poly modelling
2. Apply subdivision step
   - a higher resolution model
3. Re-edit results
   - Retouch all the smaller parts
4. Goto 2, until good final result

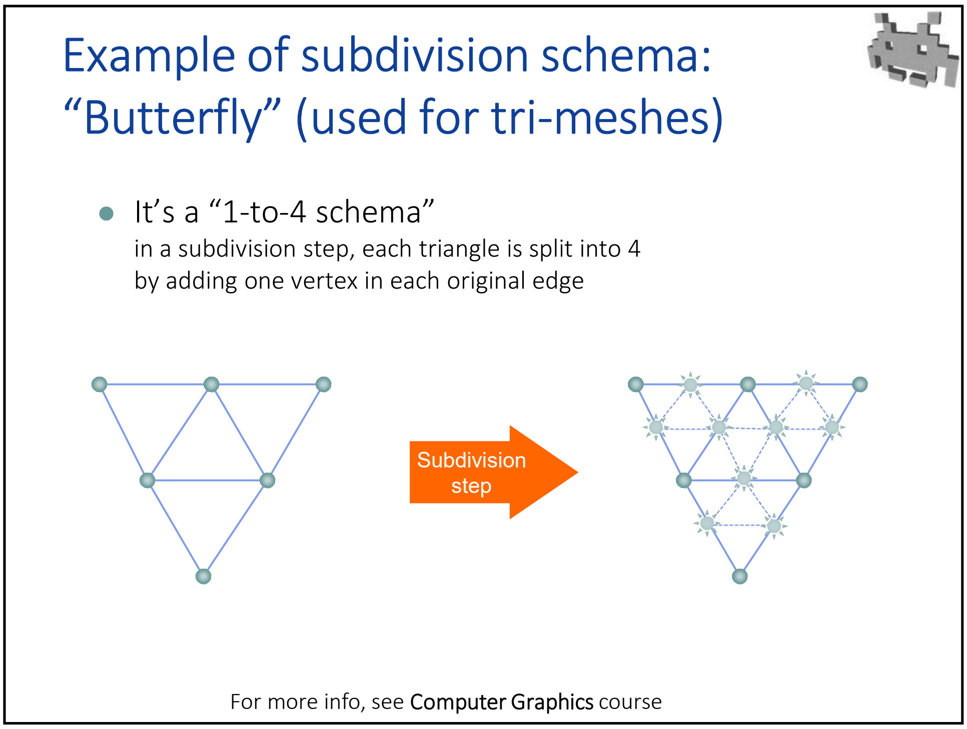
82



Subdivision surfaces as a mesh authoring tool (example)

Doo-Sabin subdivision

84



Example of subdivision schema: "Butterfly" (used for tri-meshes)

- It's a "1-to-4 schema"
  in a subdivision step, each triangle is split into 4
  by adding one vertex in each original edge

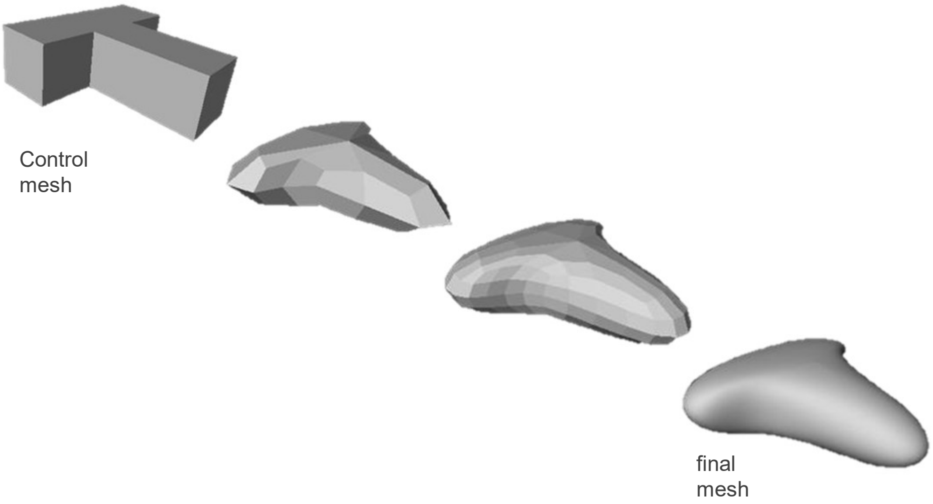
For more info, see **Computer Graphics** course

88

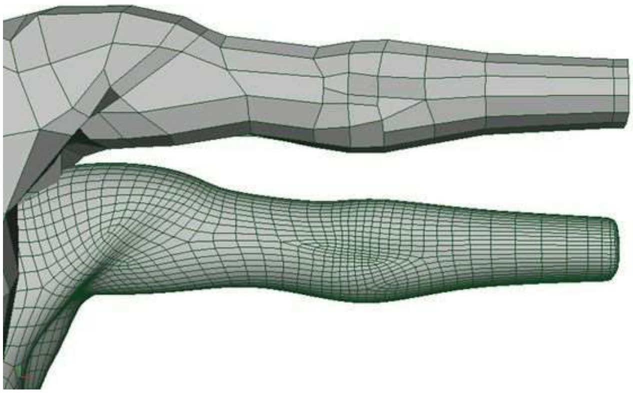
Subdivision surfaces in general

- A step typically increases resolution by a factor x4
- The geometry of the subidvided mesh (3D points) is computed according to a formula of the pos of their neighbors.
  - In some schemas (called interpolative), the old vertices are kept at the same positions
  - In other schemas (called approximative), old vertices are kept but moved into a new position
  - In other schemas (called dual) older vertices aren't kept
- Most created vertices are *regular*

An example with Catmull Clark



An example with Catmull Clark



Some common subdivision schemas

- Doo-Sabin
  - operates on any polygonal mesh
  - produces polygonal meshes
- Loop
  - 1-to-4 scheme for triangle meshes (only)
- Butterfly
  - 1-to-4 scheme for triangle meshes (only)
- Catmull-Clark
  - operates on any polygonal mesh
  - produces quad-meshes
  - traditionally, movie-industry favorite
  - a recent trend in games: use during mesh rendering

3D Mesh authoring: approaches

- Popular 3D modeling approaches:
  - Direct low-poly modelling
    - e.g. with wings3D
  - Subdivision surfaces
    - e.g. with blender
  - Digital sculpting
    - e.g. with Z-brush,
      (or Sculptris Alpha)

Digital Sculpting

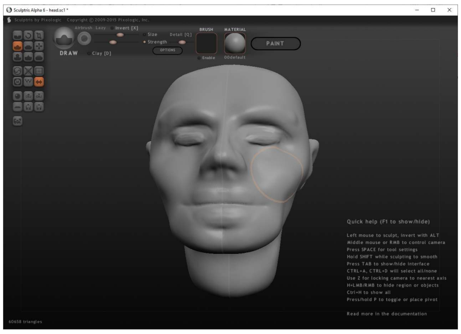
mouse
(or stylus)

=

chisel

Digital Sculpting

- demo



Quick help (F1 to show/hide)

Left mouse to sculpt, invert with ALT
Middle mouse or RMB to control camera
Press SPACE for tool settings
Hold SHIFT while sculpting to smooth
Press TAB to show/hide interface
CTRL+A, CTRL+D will select all/none
Use Z for locking camera to nearest axis
H+LMB/RMB to hide region or objects
Ctrl+H to show all
Press/hold P to toggle or place pivot

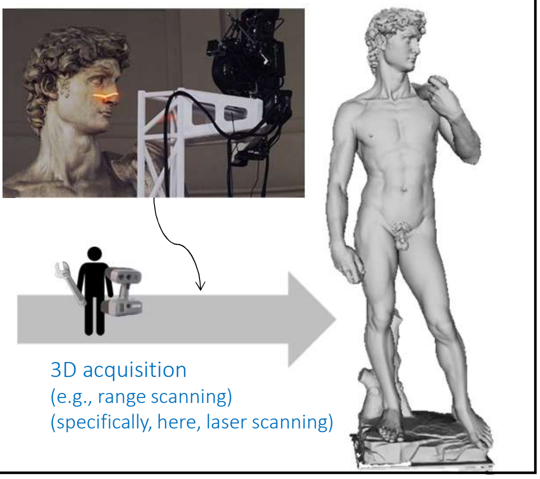
Read more in the documentation

with wings3D

Sources for 3D models: 3D acquisition

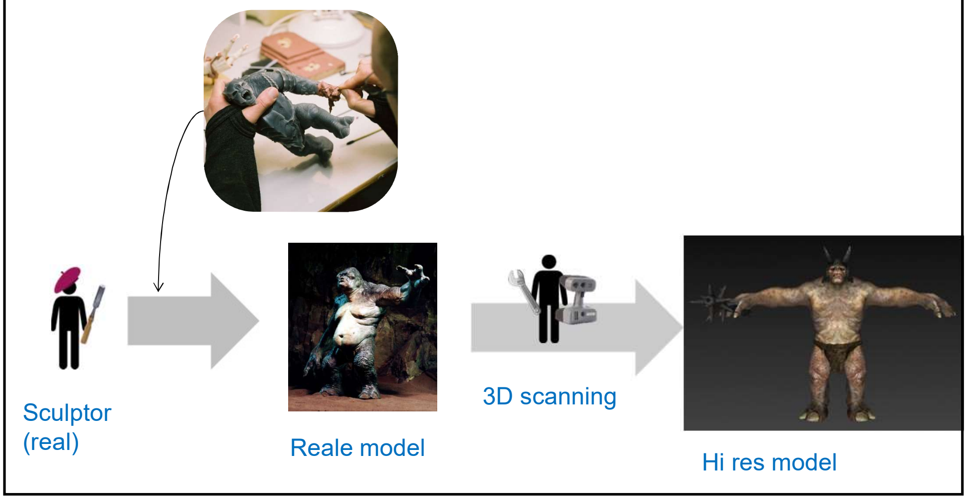
For more info, see **Computer Graphics** course

- 3D acquisition / 3D scanning
  - Technologies for obtaining 3D digital models from real-world objects



3D acquisition
(e.g., range scanning)
(specifically, here, laser scanning)

Sources for 3D models: 3D acquisition

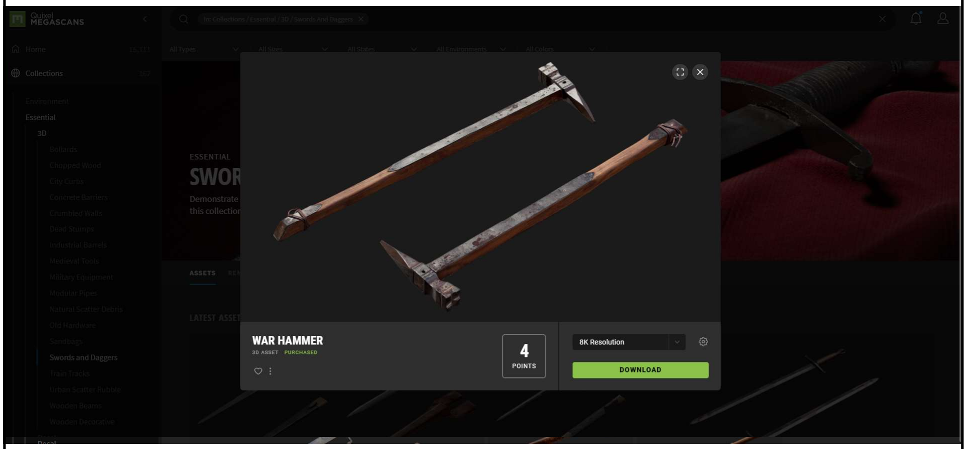


Sculptor (real)

Reale model

3D scanning

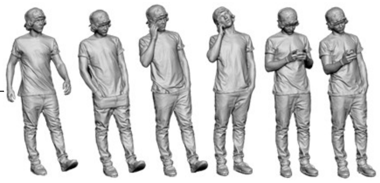
Hi res model



https://quixel.com/

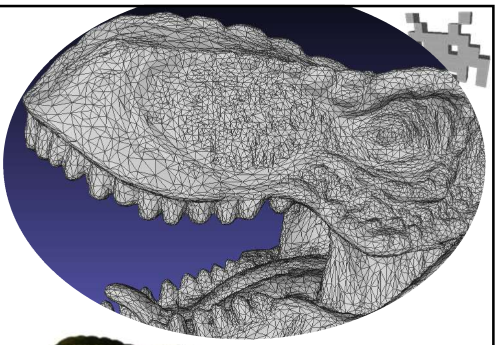
Sources for 3D models: 3D acquisition

- 3D scanning
  - A.k.a. *automatic 3D model acquisition*
  - Lot of different technologies
    - Laser scanners
    - Time of flight
    - Structured light (kinect)
    - …
  - Different characteristics
    - Results quality
      - Noise / resolution
    - Automatism
    - Invasiveness
      - Markers? Powder?
    - Real time? (kinect)
    - Price
    - Max object dimension
      - (full body scanner?)

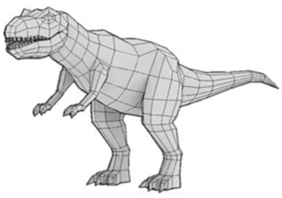


103

3D models sources: comparison



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



manually edited
low-poly mesh
(2K triangles)

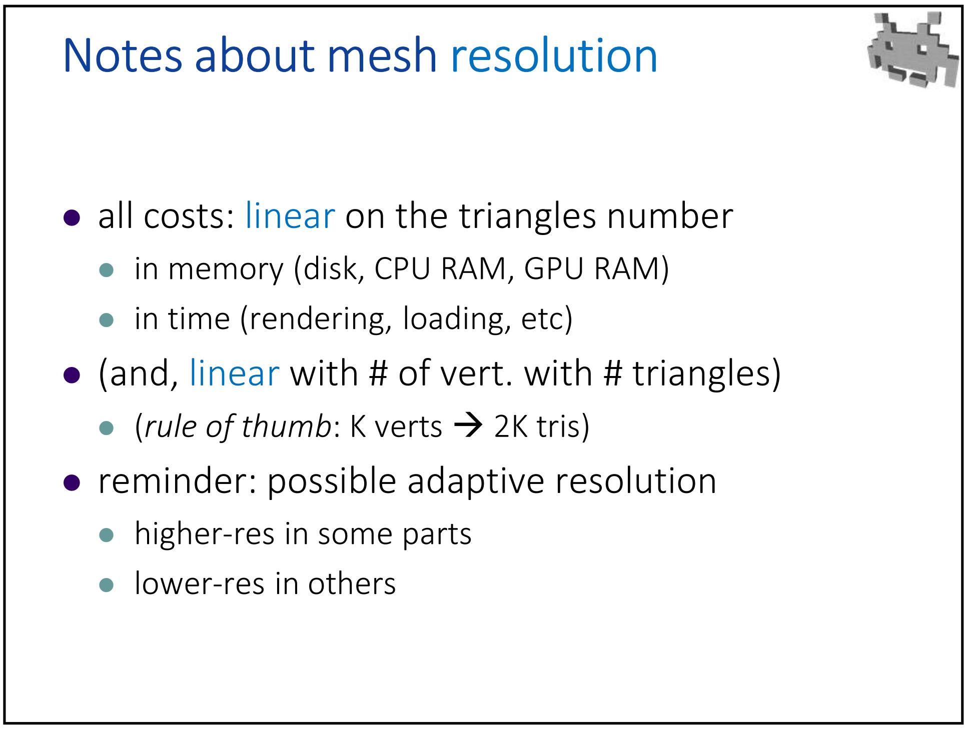
VS

Dino, scanned by artec3d

scanned & cleaned
hi res mes
(30K triangles)

(sculpted meshes are similar)

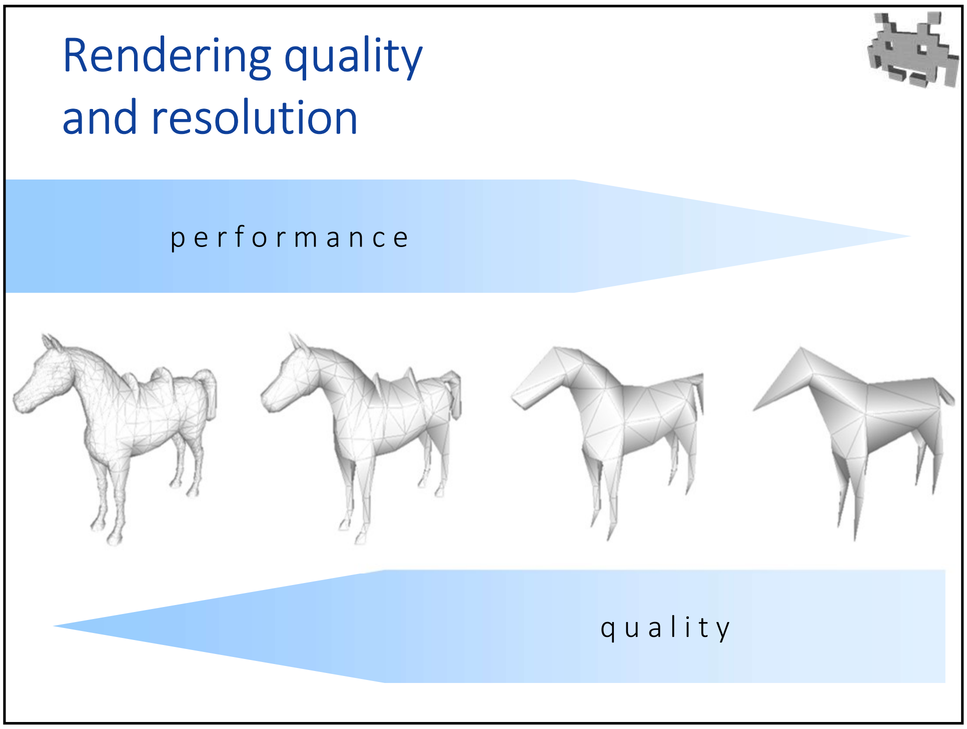
104



Notes about mesh resolution

- all costs: linear on the triangles number
  - in memory (disk, CPU RAM, GPU RAM)
  - in time (rendering, loading, etc)
- (and, linear with # of vert. with # triangles)
  - (*rule of thumb*: K verts → 2K tris)
- reminder: possible adaptive resolution
  - higher-res in some parts
  - lower-res in others

105

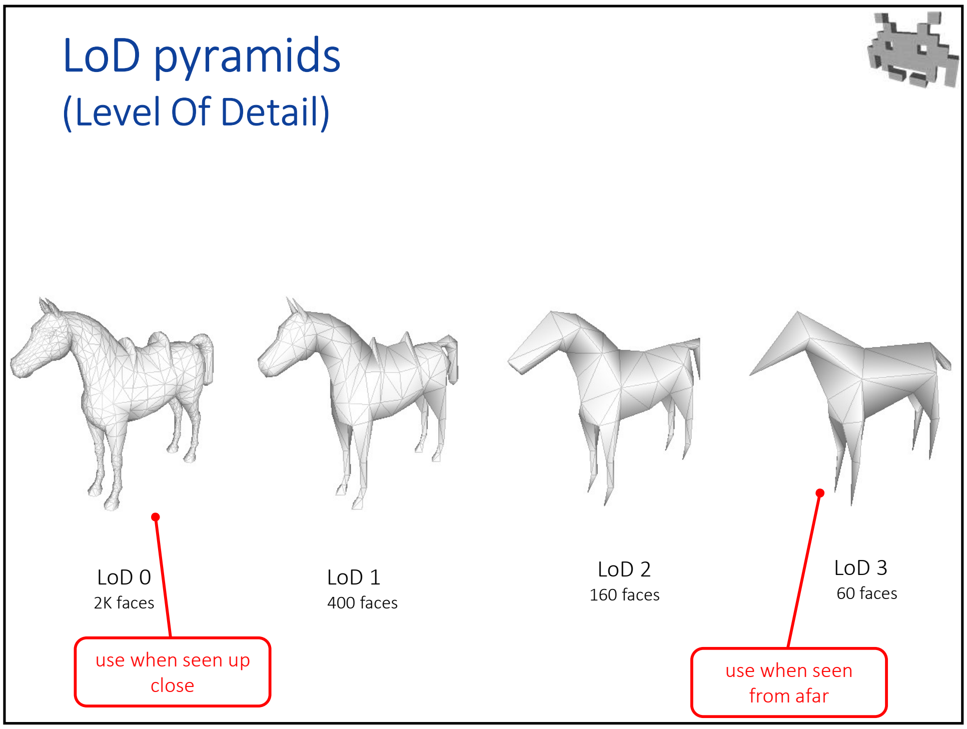


Rendering quality and resolution

performance

quality

106



LoD pyramids
(Level Of Detail)

LoD 0
2K faces

LoD 1
400 faces

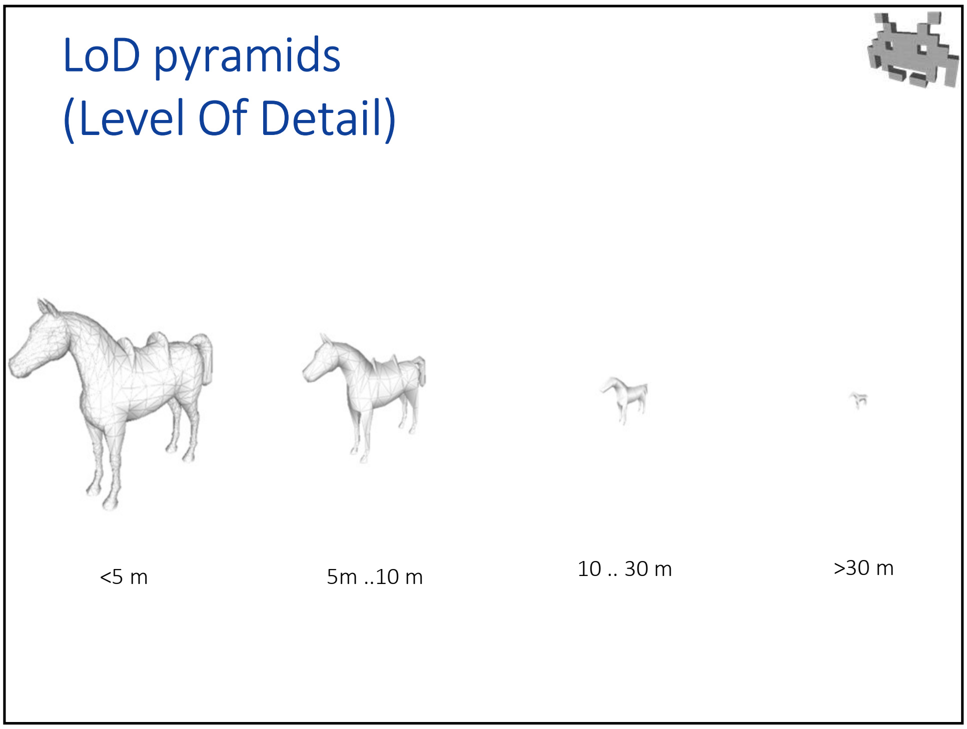
LoD 2
160 faces

LoD 3
60 faces

use when seen up close

use when seen from afar

107



LoD pyramids
(Level Of Detail)

<5 m

5m ..10 m

10 .. 30 m

>30 m

108

LoD pyramids (Level Of Detail)

- Goal:
  - decrease the geometry budget (total number of vertices)
  - ideal: size of triangles in screen space (in pixel): constant
    - importance / geometrical complexity being the same
- Task: determining the level to use (dynamically, at runtime)
  - depending on observer distance
  - and/or, depending on rendering workload
    - e.g.: rendering is lagging ⇒ decrease LoD
  - this is task of the rendering engine)
- Task: LOD creation or “LOD-ding” (during asset creation)
  - starting from LOD-0 (higher-res)
  - manual, or automatic *(see later on),* or assisted (mixed)
    - often manual, for very coarse LODs
  - note: sometimes “LoD 0” is used only in special cases
    - e.g., for cut-scenes

computed from
scene graph
(how?)

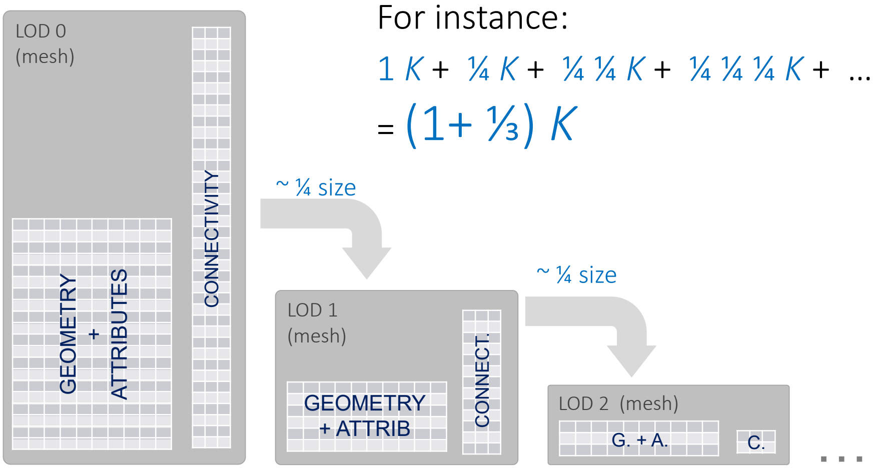
109

LoD pyramids (Level Of Detail)

Total memory usage: limited
For instance:

$1\ K + \tfrac{1}{4}\ K + \tfrac{1}{4}\tfrac{1}{4}\ K + \tfrac{1}{4}\tfrac{1}{4}\tfrac{1}{4}\ K + \ldots$

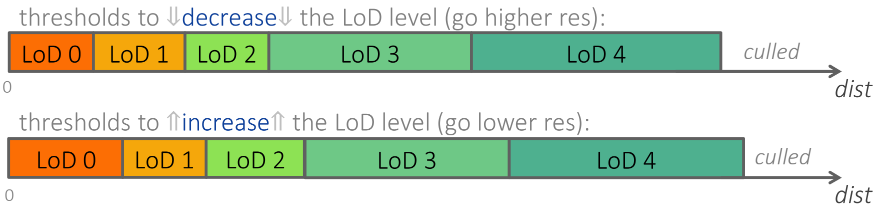
$= (1 + \tfrac{1}{3})\ K$



110

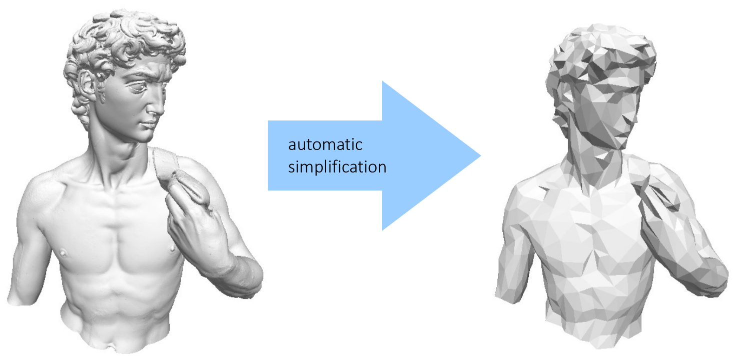
LoD pyramids: which level to use

- Basic strategy: use a fixed LoD for each interval of distance (from camera)
- ⚠ popping artefacts!
  - to mitigate it: used different thresholds to increase and to decrease the LoD


thresholds to ⇓decrease⇓ the LoD level (go higher res):
LoD 0
LoD 1
LoD 2
LoD 3
LoD 4
culled
0
dist
thresholds to ⇑increase⇑ the LoD level (go lower res):
LoD 0
LoD 1
LoD 2
LoD 3
LoD 4
culled
0
dist

Poly-reduction (aka Mesh “simplification” / “decimation”)

- parameters:
  - a maximum error
  - or number of faces objective



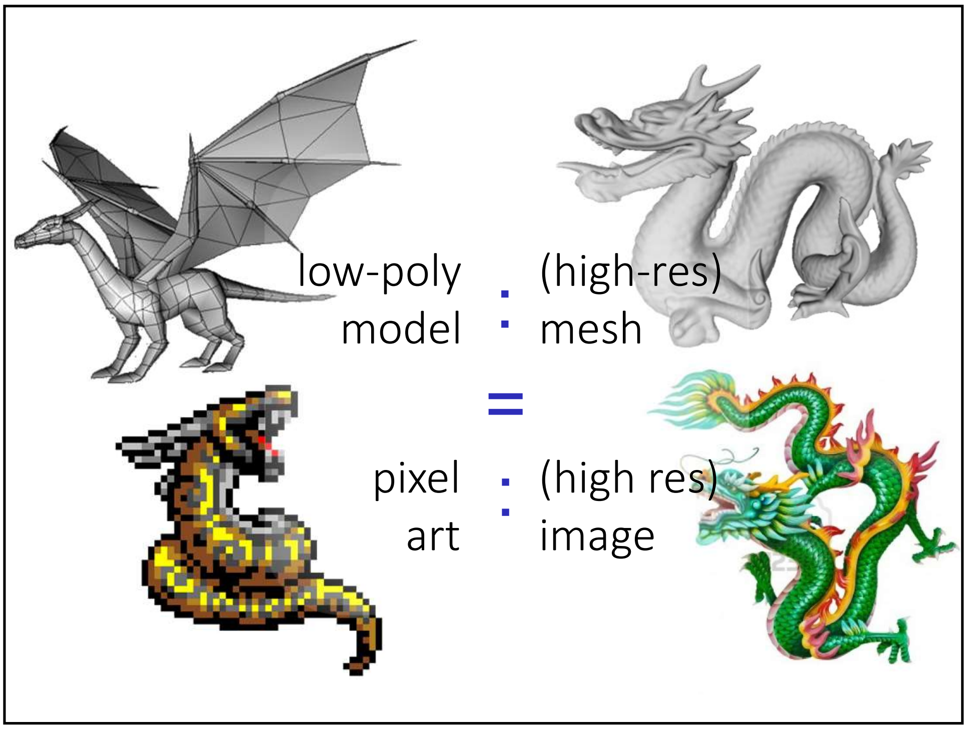
Original mesh
500K triangles

Simplified mesh
2K triangles

Poly-reduction
(aka mesh simplification, mesh coarsening)

- Different approaches are studied in Geometry Processing.
  - Adaptive or not
    - use more triangles where needed (ex. not in flat parts)
    - or not
  - Maximum error introduced:
    - measured and/or limited
    - or not
  - Topology:
    - kept
    - or not
  - Streamable
    - Possible
    - or not

114



low-poly model : (high-res) mesh

=

pixel art : (high res) image

115

New hi-res mesh formats

- Nanite (EPIC GAMES)
- Micro-Meshes (NVIDIA)

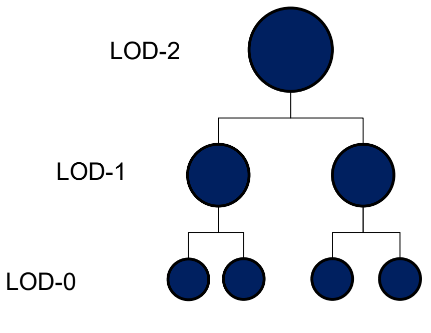
Very different internal structures, common features:

- Cheaper per-triangle VRAM cost
  - Compressed, but
  - on-the-fly decompression during rendering ("geometry augmentation")
- Cheaper per-triangle rendering cost
  - Micro-Meshes: intended for ray-tracing too
- Multiresolution, i.e., intrinsic LODs
  - Can decide on the fly which level of detail show
  - Nanite: LOD level varies across mesh
- Reduced need for UV-maps (see next lecture)

116

New hi-res mesh format 1/2: NANITE

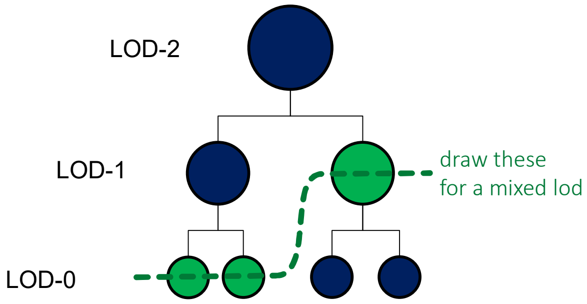
- A tree of *patches*
  - 1 patch = small *optimized* mesh of 128 tris



117

New hi-res mesh format 1/2: NANITE

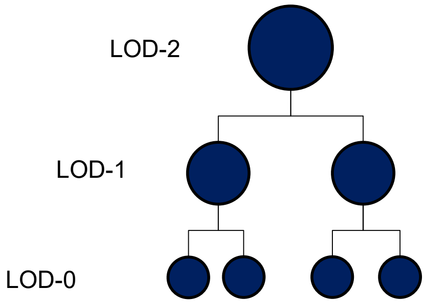
- A tree of *patches*
  - 1 patch = small *optimized* mesh of 128 tris



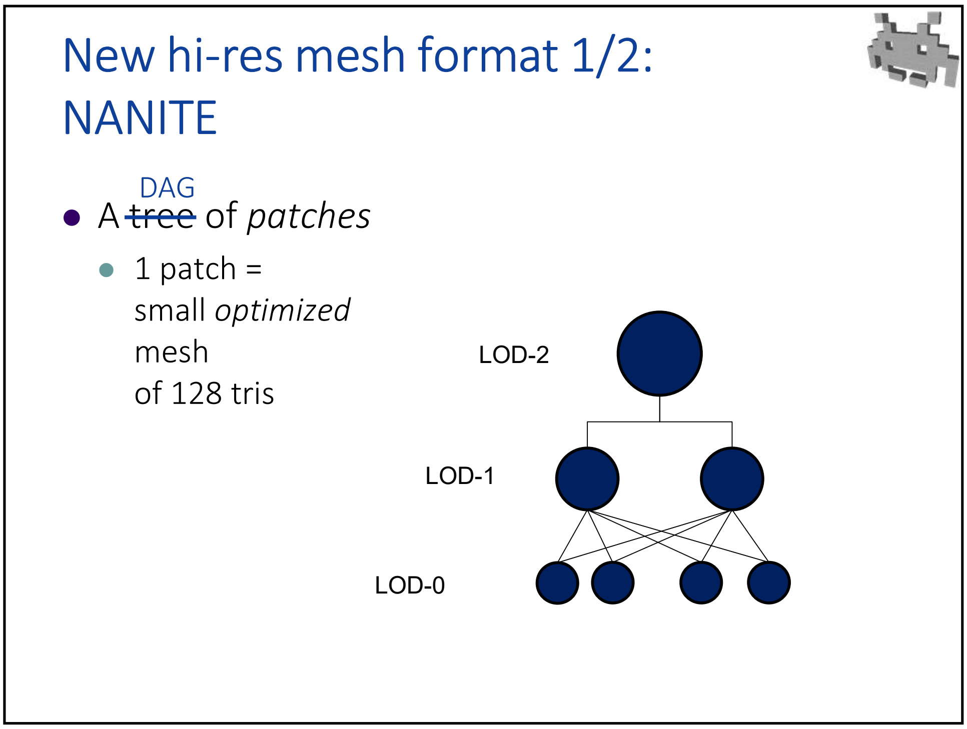
118

New hi-res mesh format 1/2: NANITE

- A tree of *patches*
  - 1 patch = small *optimized* mesh of 128 tris



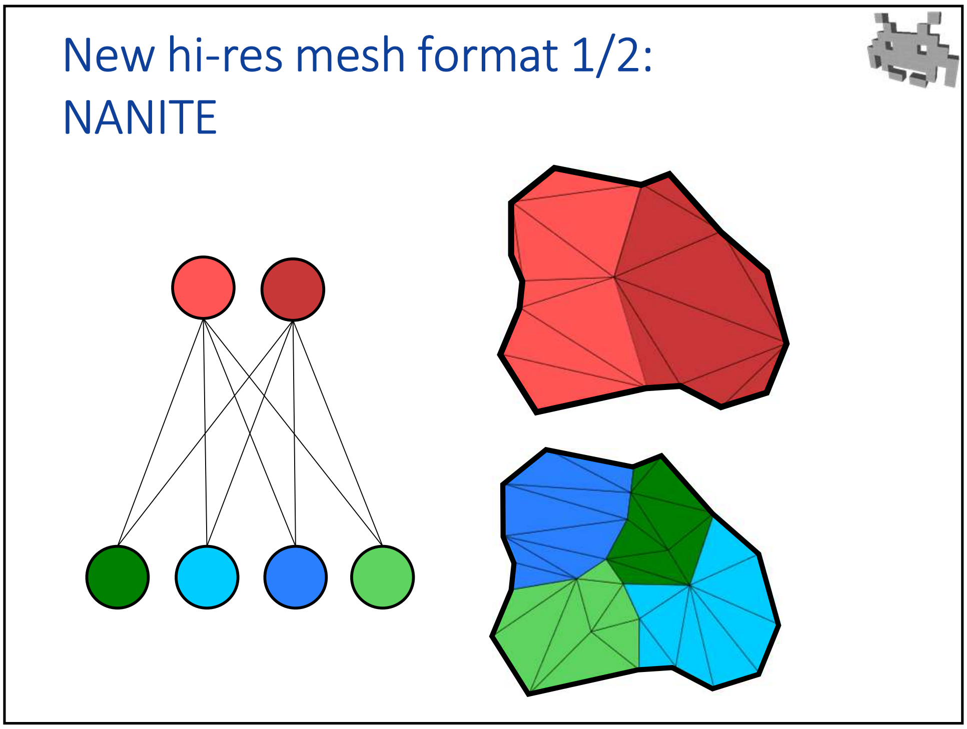
119



New hi-res mesh format 1/2: NANITE

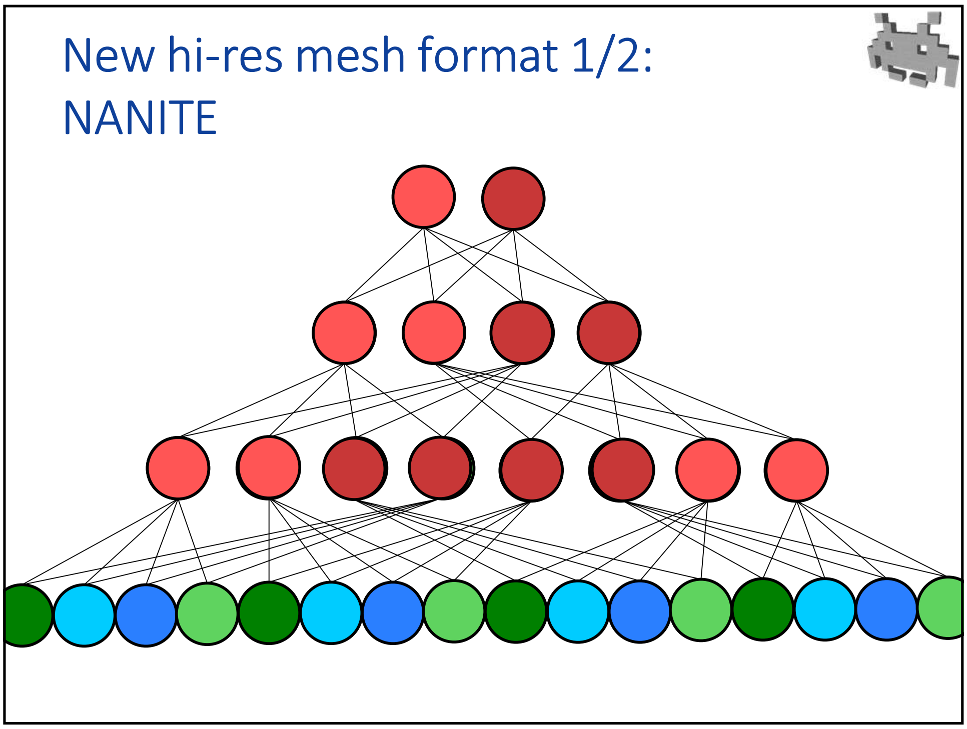
- A ~~tree~~ DAG of *patches*
  - 1 patch = small *optimized* mesh of 128 tris

120



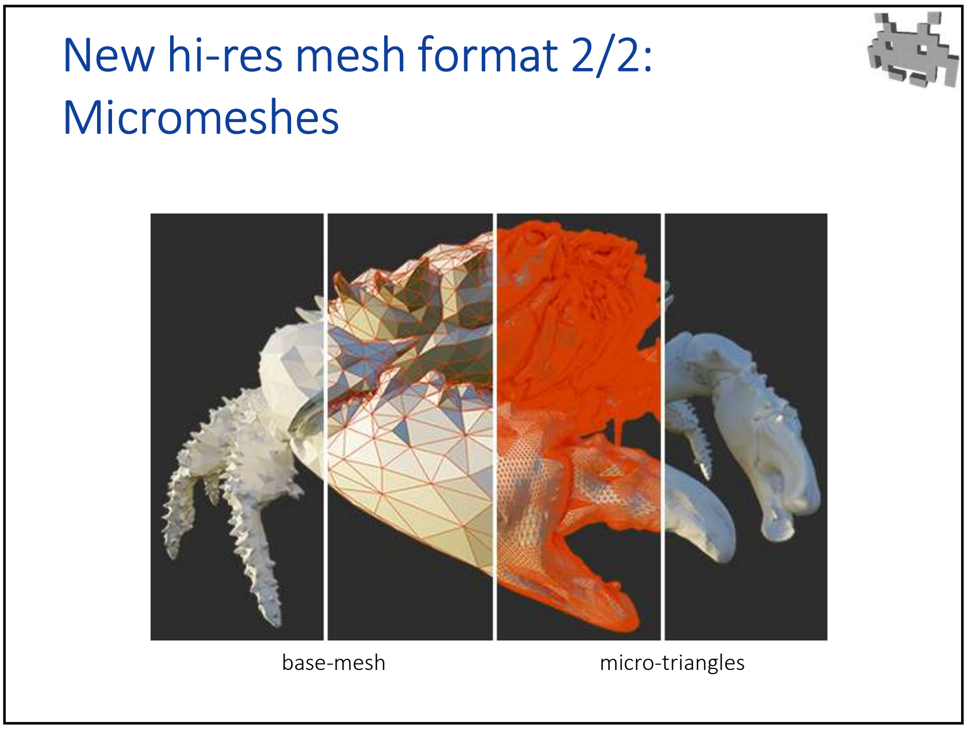
New hi-res mesh format 1/2: NANITE

121



New hi-res mesh format 1/2: NANITE

123

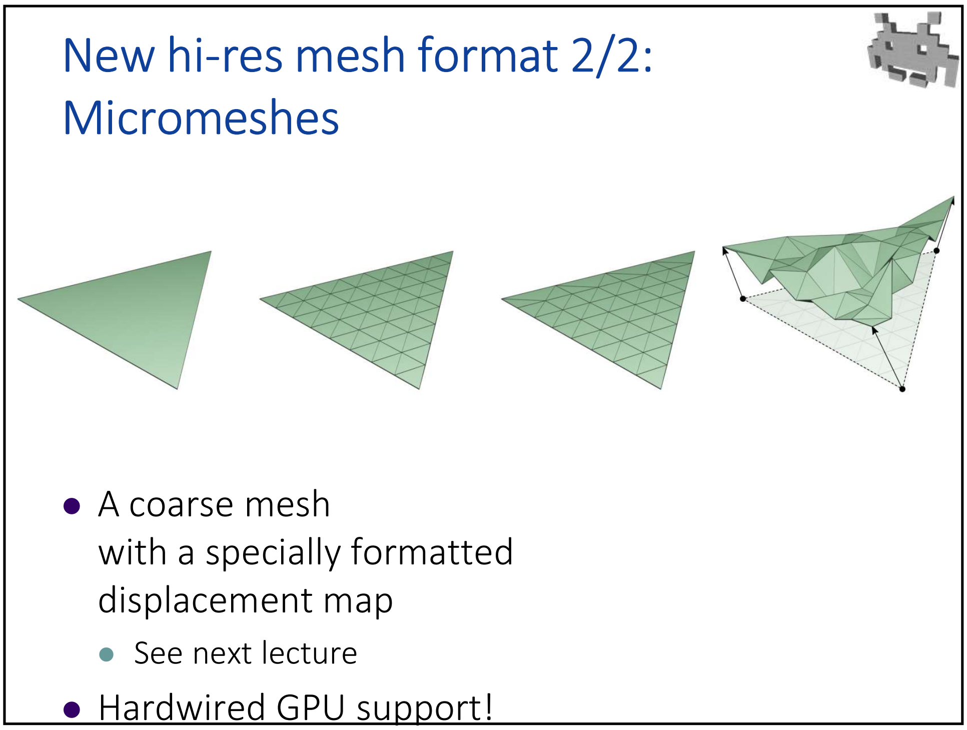


New hi-res mesh format 2/2: Micromeshes

base-mesh

micro-triangles

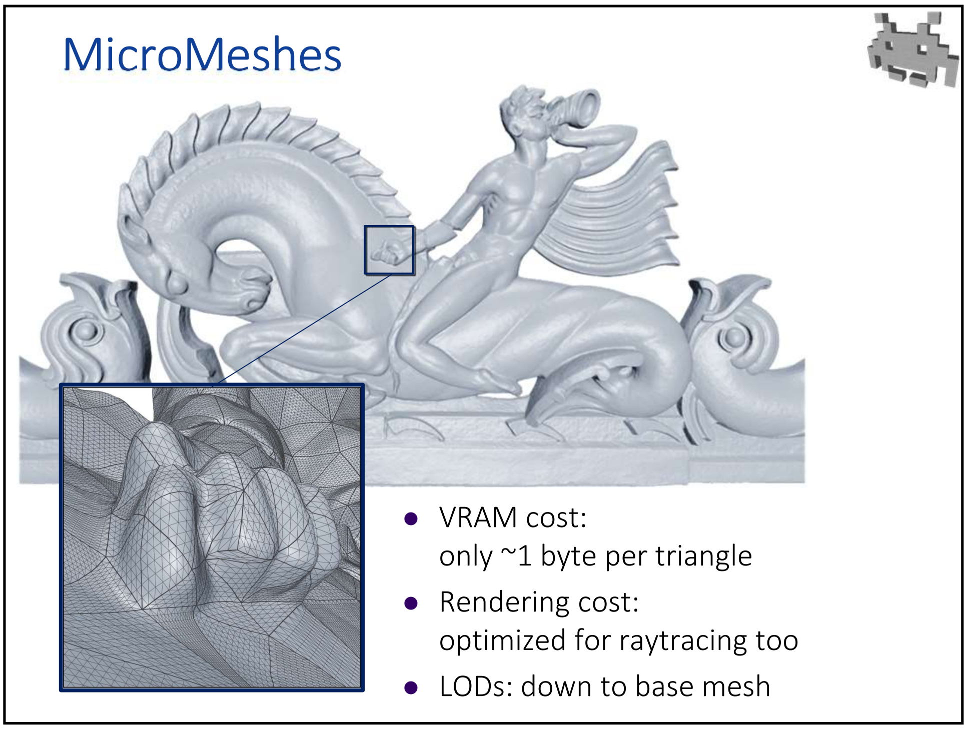
126



New hi-res mesh format 2/2: Micromeshes

- A coarse mesh with a specially formatted displacement map
  - See next lecture
- Hardwired GPU support!

127



MicroMeshes

- VRAM cost: only ~1 byte per triangle
- Rendering cost: optimized for raytracing too
- LODs: down to base mesh

128