

# Sammanfattning 3D-grafik

Ett sammandrag av saker jag antar att ni behärskar

- Avancerad 2D-grafik?
- Mesh-objekt, färg
- 3D-pipeline

Ingemar  
Ragnemalm  
ingis@isy.liu.se

## ”Avancerad” 2D-grafik?

- Scrollande spel
- Stora världar
- Multilayer, parallax scroll
- Pseudo-3D, isometric grid
- Palettanimation

Knappast några tekniska utmaningar idag! Vi går vidare...

Ingemar  
Ragnemalm  
ingis@isy.liu.se

(Kap 11)

# 3D pipeline overview

(kap 12)

- Datatyper, objektrepresentation
- Lite om färg
- Texturmapping
- 3D-pipelinen, 3D-motorn

(med en del praktiska kommentarer om hur man sparar plats)

Ingemar  
Ragnemalm  
ingis@isy.liu.se

## Objektrepresentation

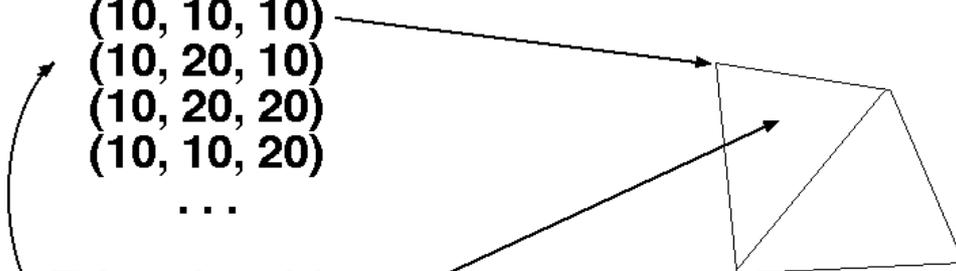
Mesh, splines mm. Vi antar mesh!  
Indexerade primitiver:

Vertex table:

(10, 10, 10)  
(10, 20, 10)  
(10, 20, 20)  
(10, 10, 20)  
...

Triangle table:

(1, 2, 3)  
(1, 2, 4)  
...



Ger färre vertexar att lagra  
och transformera

Ingemar  
Ragnemalm  
ingis@isy.liu.se

# Kvantisering

**Ett trick för att komprimera  
polyhedra-modeller**

**En vertex: x, y, z**

**Varje koordinat kräver minst 32 bitar (1+8+23)**

**Kan vi minska minnesåtgången?**

**Konventionell kompression (Huffman mm)  
lämpar sig inte för realtid.**

# Kvantisering

**Varje koordinat kräver minst 32 bitar (1+8+23)**

**MEN, om utfallsrummet är känt kan vi kvantisera  
detta till färre bitar!**

**Antar att modellen är t.ex. max 2 meter i diameter.**

**16 bitars heltal: ett fel på 0.00003 meter**

**8 bitars heltal: 0.007 meter (7 mm)**

**kvantisering:  $val = (in - min) / (max - min) * 255$   
dekvantisering:  $out = val / 255 * (max - min) + min$**

**Fundering: Borde man inte logaritmera?**

# Färgrepresentation

RGB/RGBA självklart val. Flyttal används i GPU'n.

$32 \cdot 4 = 128$  bitar per färgspecifikation!

3 eller 4 bytes bättre för masslagring

Alfakanalen för transparens, nyttig när den behövs, slöseri när den inte behövs.

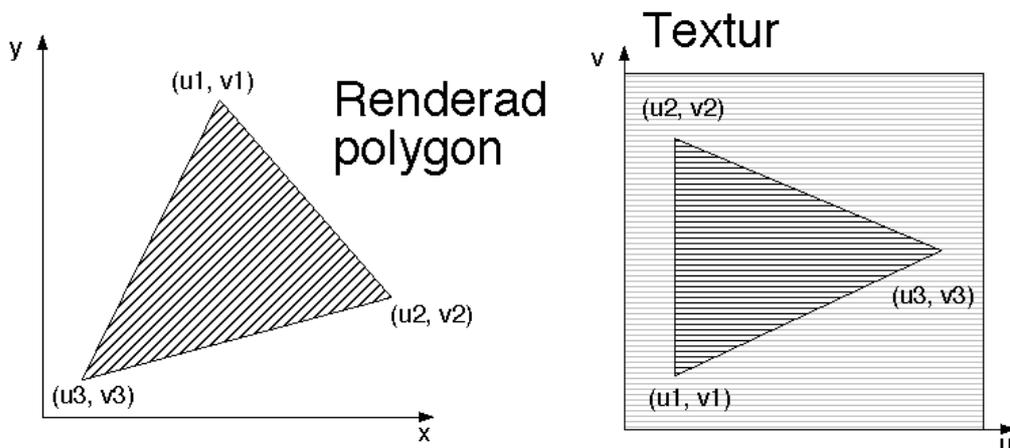
Behöver sällan 8 bitars precision heller. 4 bitar räcker långt!

Alfavärdet kan ofta specas per vertex i stället för per textel

Ingemar  
Ragnemalm  
ingis@isy.liu.se

# Texturmapning

En pixelbild sträcks över en polygon och renderas med den, för detalj och realism.



- Mappningen djupberoende för rätt perspektiv
- Texturer läses in i VRAM, på GPU
- Flera texturer kan ritas på samma gång (mer om det senare i kursen)
- Inte bara materialavbildningar, även ljus, reflektioner...

Ingemar  
Ragnemalm  
ingis@isy.liu.se

# Dataformat för polyhedra

Enklast möjliga för att lösa problemet

Vertex: 3 flyttal  
Texturkoordinater: 2 flyttal  
Normalvektor: 3 flyttal  
Färg: 3 flyttal

Snålversion:

Vertex: 3 bytes  
Texturkoordinater: 2 bytes  
Normalvektor: skippas!  
Färg: 1 byte

44 bytes blev 6!

Ingemar  
Ragnemalm  
ingis@isy.liu.se

## En grafik-pipeline

- Synlighetsberäkningar
  - Clipping
  - Culling
  - Skymda ytor
- Detaljnivå
  - LOD-analys
- Transformationer, belysning
- Rastrering, pixelrendering

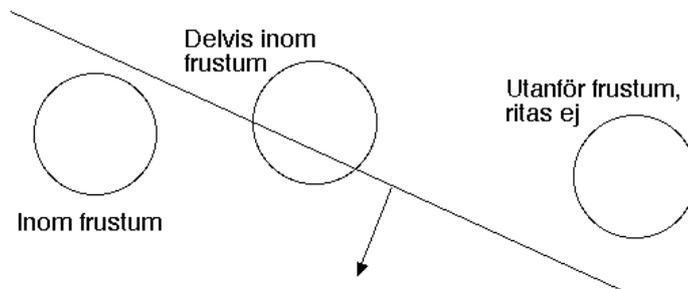
Ingemar  
Ragnemalm  
ingis@isy.liu.se

# Clipping

Klipp bort allt som faller utanför frustum

- Triangelclipping. Alla trianglar testas mot frustum före rendering
- Objektclipping. Testa objektvis, t.ex. omskrivande sfärer, mot frustum.

Behagligt lätt! Givet planets ekvation för ett frustumplan, en skalärprodukt och en jämförelse!



Ingemar  
Ragnemalm  
ingis@isy.liu.se

## Objektclipping med sfär

Behagligt lätt! Givet planets ekvation för ett frustumplan, en skalärprodukt och en jämförelse!

Planet:  $Ax + By + Cz + D = 0$ , eller  $N \cdot p + D = 0$

Sfär: radie  $S_r$ , centrum  $C$

Testet blir:

$$N \cdot C + D < S_r$$

(Lätt omskrivet från Dalmau)

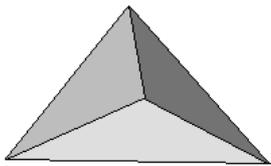
Alternativ: AABB eller OBB: testa om någon vertex är inom frustum

Ingemar  
Ragnemalm  
ingis@isy.liu.se

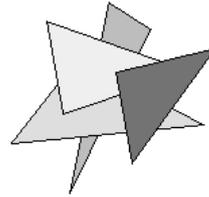
# Culling

**Backface culling, dölj polygoner med framsidan bort från kameran. Framsidan bestäms från vertexordning.**

**Fungerar på slutna polyhedra, en ytrepresentation (boundary representation).**



Sluten polyhedra, delar rymden i två delmängder



Polygon soup, inget känt system, polygonerna ligger så vitt vi vet hur som helst

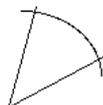
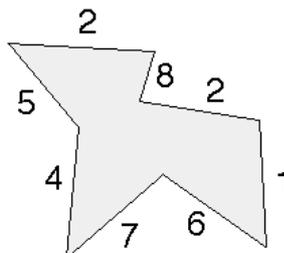
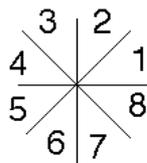
Ingemar  
Ragnemalm  
ingis@isy.liu.se

# Object culling

**Backface culling kan förbehandlas!**

**Klassificera alla polygoner i ett antal riktningsgrupper. Beroende på riktning kan många polygoner förkastas innan de skickas till rendering!**

**2D-exempel:**



Frustum i riktningszon 1 och 2.  
Normaler i zon 4-8 accepteras

Ingemar  
Ragnemalm  
ingis@isy.liu.se

# Skymda ytor

Z-buffer garanterar (nästan) att rätt yta ritas, men ger dubbelt ritande

Stort problem i inomhusmiljöer. Löses med ytterligare tester.

- PVS
- Portaler

Hårdvarustöd i moderna grafikkort (GF3+, 8500+)

Kräver ett enkelt omskrivande objekt. Grafikkortet testas mot Z-buffern.

# Detaljnivå

Varierande detaljnivå (Level-of-detail) nödvändigt för komplexa miljöer.

Detaljnivå bestäms från avståndet. Onödigt med polygoner som täcker mindre än en pixel!

Men hur gör man övergångar mellan olika detaljnivåer?

- Alfa-toning mellan LOD-nivåer
- Kontinuerlig övergång, polygoner/vertexar slås gradvis ihop med varandra. Beräkningstungt!

# Transformationer

Transformationskedjan:

Modellkoordinater-världskoordinater-  
kamerakoordinater-projicerade koordinater-  
skärmkoordinater.

I OpenGL: Matriserna modelview och projection.

Ingemar  
Ragnemalm  
ingis@isy.liu.se

# Ljussättning

Enkla metoden: Ljussätt per vertex. Gouraud shading. Bra hårdvarustöd. "Ser ut som OpenGL."

Modernare: Pixelvis ljussättning, undviker kantartefakterna.

- Programmerbara fragment shaders
- Light mapping

Vi återkommer till detta...

Ingemar  
Ragnemalm  
ingis@isy.liu.se

# Rastrering

***Rasterization***, polygoner konverteras till de pixelpositioner de upptar på skärmen.

Före GPUerna, ett svårt problem som krävde extrem optimering. Numera lämnas detta till hårdvaran. Skicka trianglarna till GPU.

Problemet är nu att skicka geometrin.

- Immediate mode
- Vertex arrays (vertex buffers)
- Server-side, skicka geometrin en gång, rita många gånger. “Compiled Vertex Arrays”