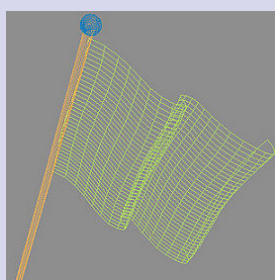


Game Engines Master Game Dev Verona, 2014-2015 Texture Mapping

Marco Tarini

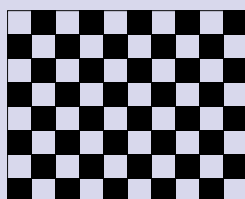


Texture mapping



geometria 3D
(insieme di quadrilateri)

+



RGB texture 2D
(color-map)

=

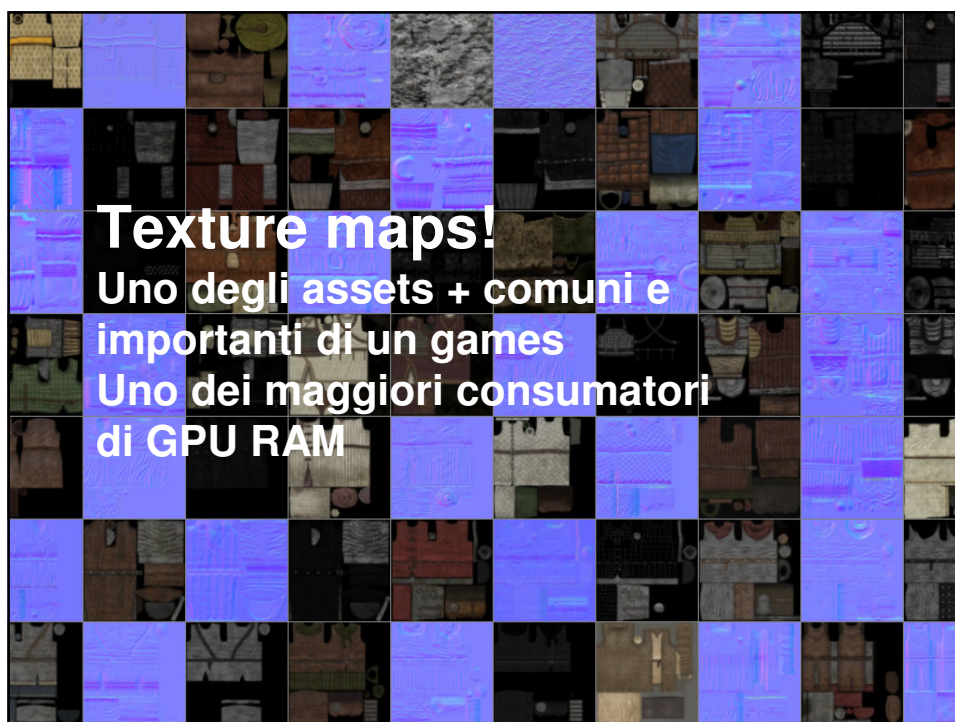


Altro esempio di color-map



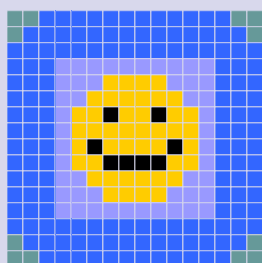
Altri esempi di color-map





Texture maps: strutture dati

- Sostanzialmente, un'immagine rasterizzata?



«Texel»

Texture maps: strutture dati

- Ogni texel è...
 - Un colore RGB (*color map*, *RGB map*, *diffuse map*)
 - Una normale (*bump map*, o *normal map*)
 - Il coefficiente speculare (*specular map*)
 - Un fattore di trasparenza (*alpha map*, o *cutout texture*)
 - Una scavo dalla superficie (*displacement map*, o *height texture*)
 - Un lighting precalcolato (*light map*, *baked light texture*)

MIP map levels

- Pre-filtering delle tessiture
- "LOD levels per immagini!"
- Hardware sceglie livello giusto per pixel
- Evita artefatti di sottocampionamento



Texture maps come assets

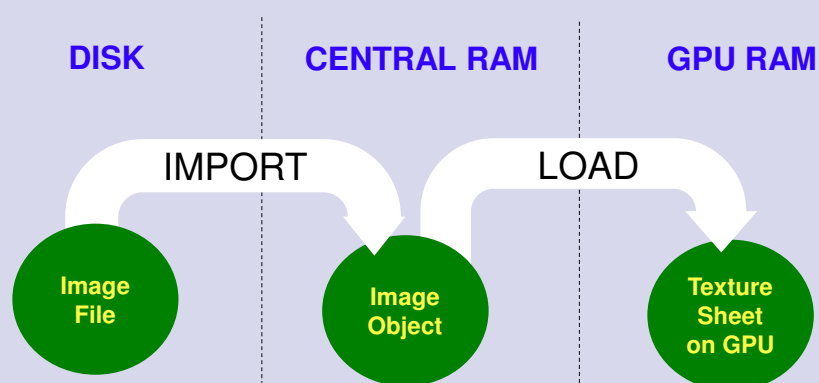
- Caratteristiche:

- Size:
 - risoluzione
 - canali (es: alpha?)
- MIP-map (presenti o no?)
- Compressione?

Vincoli ricorrenti:

- res come **potenze di 2**
 - (richiesto da sempre meno engines)
 - sia in U che in V
 - (es: 256x256 o 1024x512)
- res per lato < max
 - (es: max = 4096 o 2048)

Life of a Texture *(and, basically, any 3D assets)* in a Game Engine



Texture maps come assets: formati files

- | | |
|---|--|
| <ul style="list-style-type: none"> • x immagini generiche
(decomprimere tutta immagine
prima di accedere ai pixel) 😊 compressione: ottima ☹ loading: pesante: <ul style="list-style-type: none"> • decomprimere da RAM,
(forse) ricomprimere in
GPU-RAM ☹ MIP-map lvls etc:
controllato dall'engine | <ul style="list-style-type: none"> • x textures
(random accessibility ai texels
senza decomprimere) ☹ compressione: cattiva 😊 loading: leggero <ul style="list-style-type: none"> • direct copy
Disco => RAM =>
GPU RAM 😊 MIP-map lvls etc:
controllato dall'artista |
|---|--|

asset on
RAM / disk
(maybe)

once in
GPU RAM
(for sure)

Texture maps come assets: formati files

- | | |
|--|---|
| <ul style="list-style-type: none"> • x immagini generiche: <ul style="list-style-type: none"> • .JPG / .JPEG <ul style="list-style-type: none"> ☹ lossy, 😊 > compressione, 😊 immagini "fotografiche": best ☹ solo 3 canali (no choice) ☹ 8 bit x canale (no choice) • .PNG <ul style="list-style-type: none"> 😊 lossless ☹ < compressione 😊 disegni: best 😊 anche canale alpha possibile 😊 anche 16 bits possibile • .TIFF e .RAW (rari) <ul style="list-style-type: none"> 😊 lossless ☹ ☹ compressione: vabbè' 😊 max flessibilità canali etc • .PNM (davvero raro) <ul style="list-style-type: none"> ☹ ☹ ☹ compressione: ahah 😊 ma parsing facilissimo! (no lib) | <ul style="list-style-type: none"> • x textures: <ul style="list-style-type: none"> • .DDS <ul style="list-style-type: none"> (direct draw surface) ☹ lossy ☹ compressione: poco (a rate fisso) 😊 GPU ready 😊 include MIPmap levels (volendo) <p>con compressione a scelta
(S3TC):</p> <ul style="list-style-type: none"> • DXT1 → "no / 1-bit alpha" • DXT2 } "rough alpha" • DXT3 } • DXT4 } "smooth alpha" • DXT5 } <p>(diversi compromessi fra qualità,
costo, canali...)</p> |
|--|---|

Come si legano le tessiture alle mesh?

Parte 3D dei game assets

