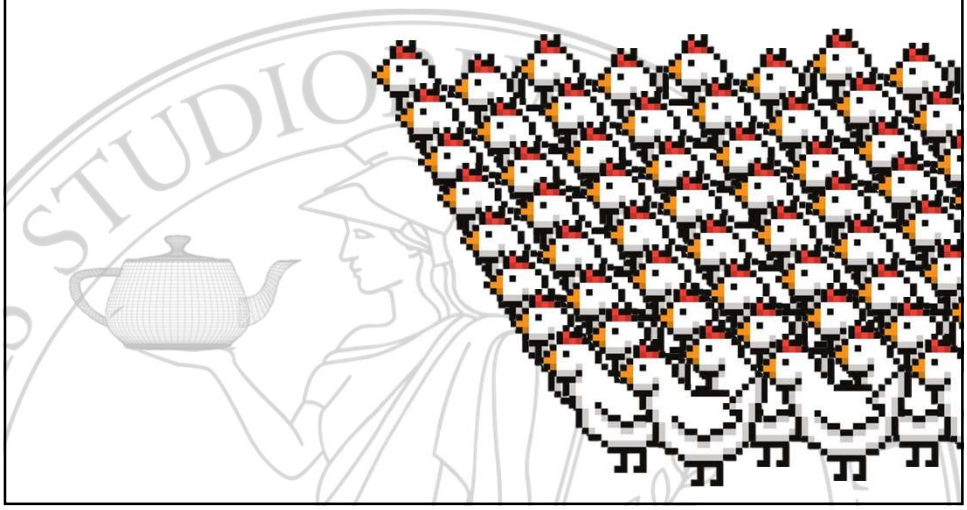
Marco Tarini - Grafica & Immagini Digitali 2019/2020
Università degli Studi di Milano

Graphic Hardware, Drivers, API e librerie (cenni)



1

Super-brief History of GPU

- ✓ dawn of times - 1995:
  - **CG is embarassing parallel**
- ✓ 1995:
  - **“let’s build a card for that”**
- ✓ 1995-2003:
  - **It’s super effective!**
  - **Cards become customizable**
- ✓ 2003-now:
  - **“let’s use it for other emb. paral. things”**
  - **GP-GPU**

2

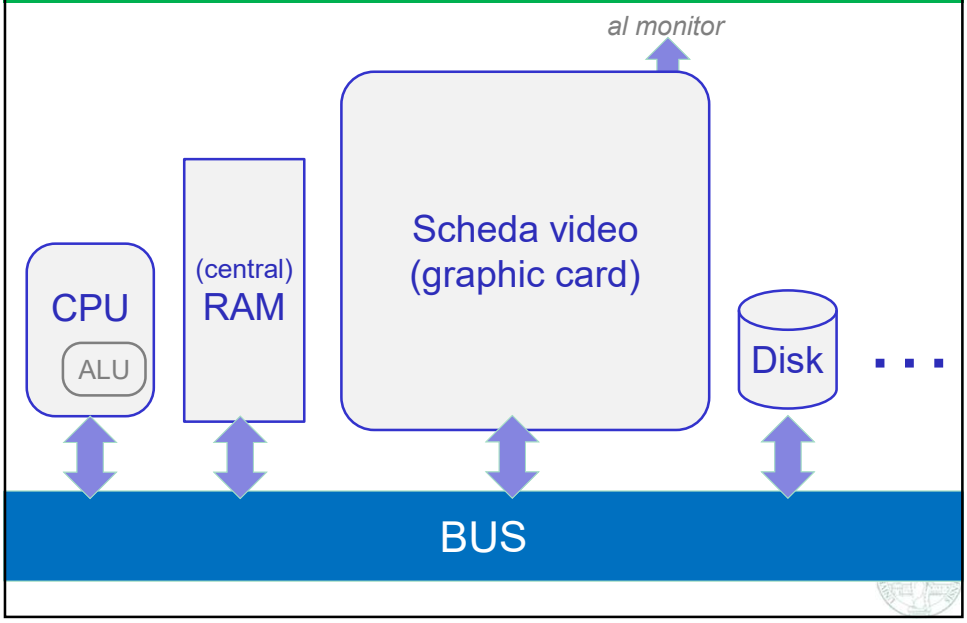
Hardware specializzato per il rendering

Visione di insieme:

- ✓ "**GPU**":
  - ⇒ Graphics Processing Unit
  - ⇒ La CPU della scheda video
  - ⇒ *Instruction Set* specializzato!
- ✓ Architettura a ***pipeline***
  - ⇒ a "catena di montaggio"
- ✓ Modello di computazione **SIMD**
  - ⇒ sfrutta l'alto grado di parallelismo insito nel problema
- ✓ Possiede la **propria** memoria RAM a bordo
  - ⇒ "RAM CPU" vs "RAM GPU"
  - ⇒ grandi copie di memoria da una all'altra dispendiose

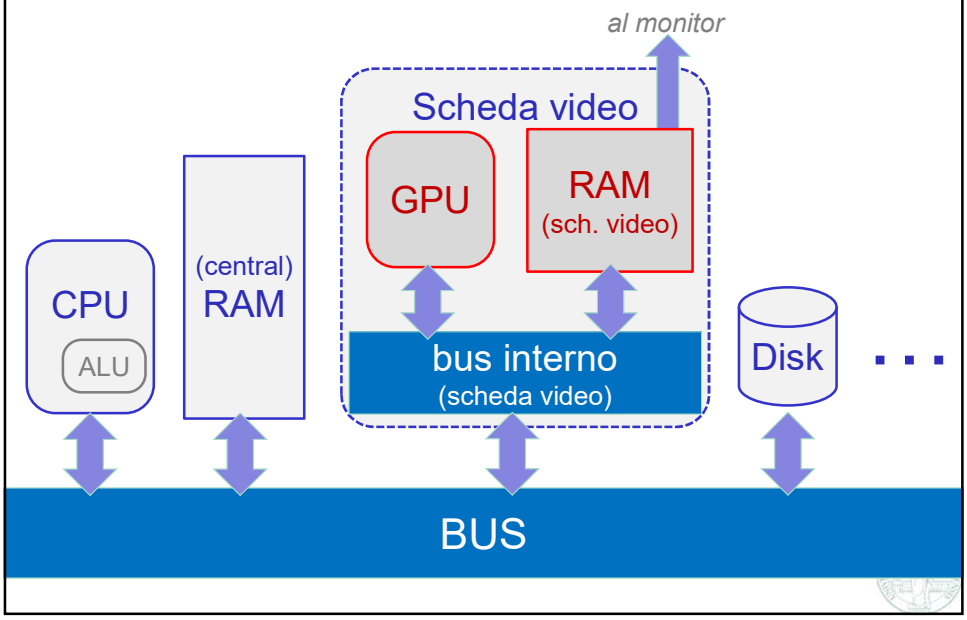
3

Schema (alto livello) di: “PC con scheda video”



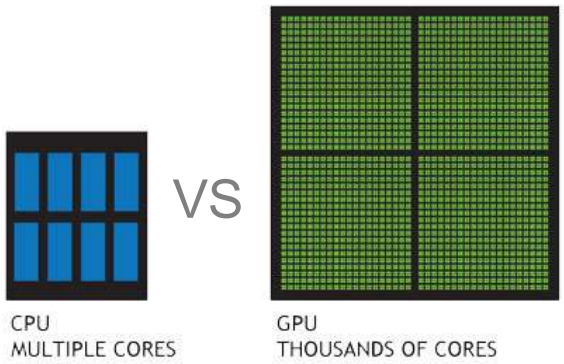
4

Schema (alto livello) di: “PC con scheda video”



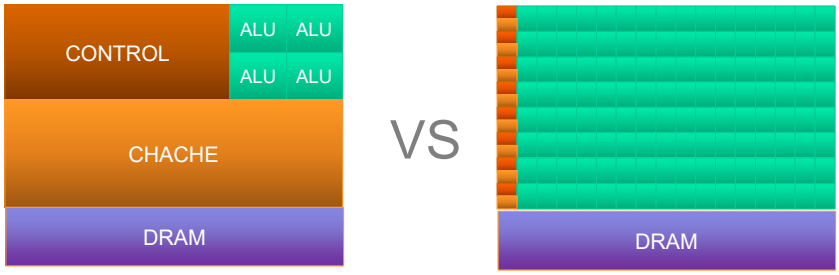
5

CPU vs GPU



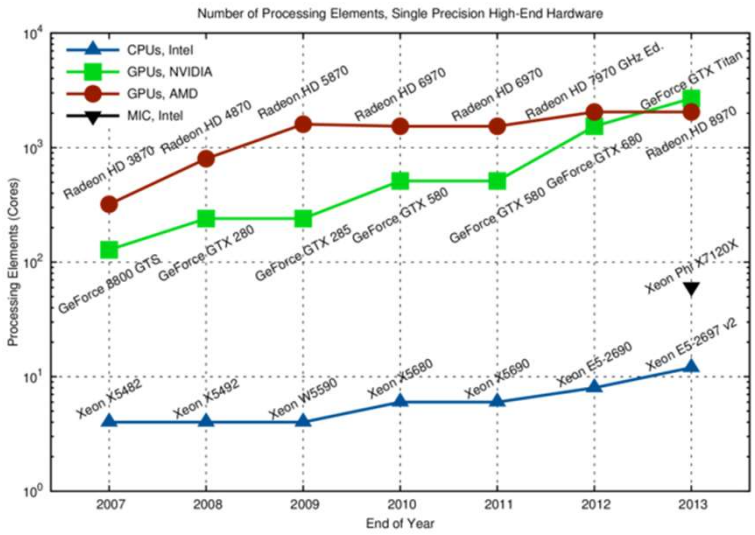
6

CPU vs GPU



7

CPU vs GPU: how many proc. elements



8

CPU vs GPU

✓Transistors:
>80%
control+cache

→ flessibilità

CPU

VS

✓Transistors:
~80% ALU

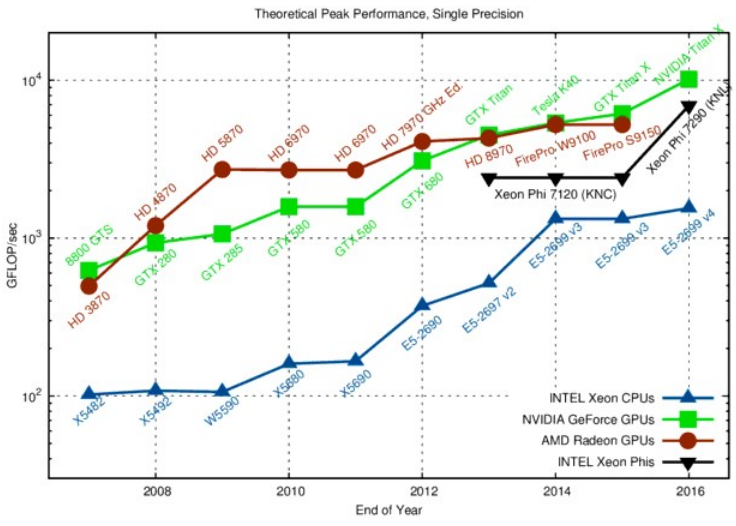
→ potenza (FLOPS)

GPU

1.4 G transistors → 1 TeraFLOP

9

CPU vs GPU



img by https://www.karlrupp.net/

10

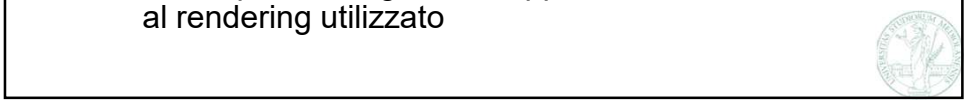
Hardware specializzato per il rendering

✓Vantaggio: **efficienza**

- instruction set specializzato
  - (computazioni più comuni sono hard-wired)
- computazioni in parallelo:
  1. fra CPU e GPU
     - » rendering demandato alla scheda grafica
     - » resto dell'applicazione libera di utilizzare la CPU e RAM base
  2. a volte: fra GPU distinte (es. più schede sullo stesso BUS)
  3. fra le fasi del pipeline (vanno tutte in parallelo)
  4. dentro *ogni fase* del pipeline (più sottoprocess. per fase)
  5. instruction level: operazioni operano su vettori di 4 operandi

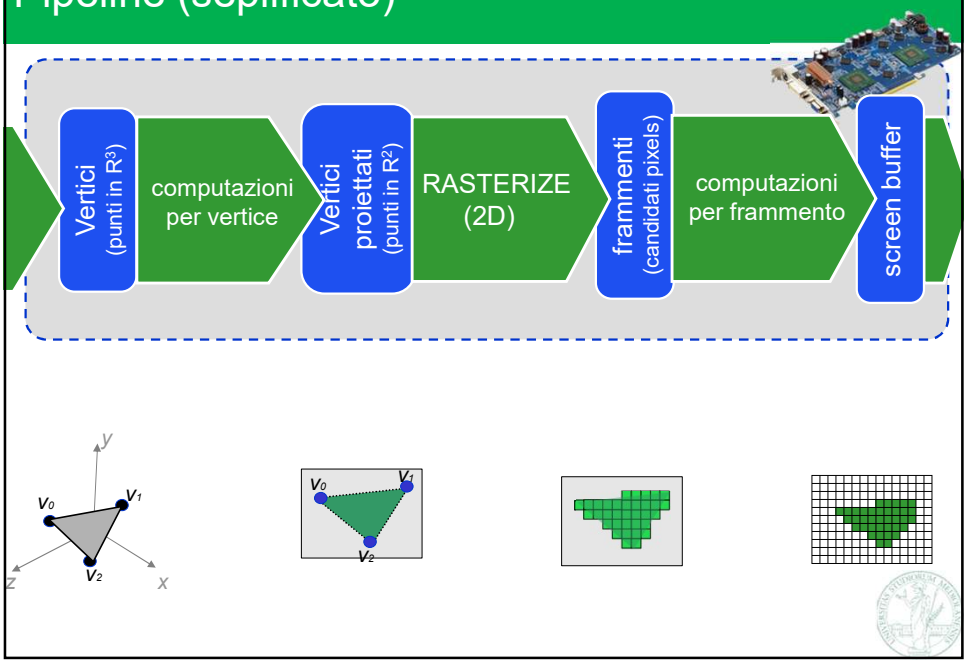
✓Svantaggio: **rigidità**

- scelta quasi obbligata dell'approccio al rendering utilizzato

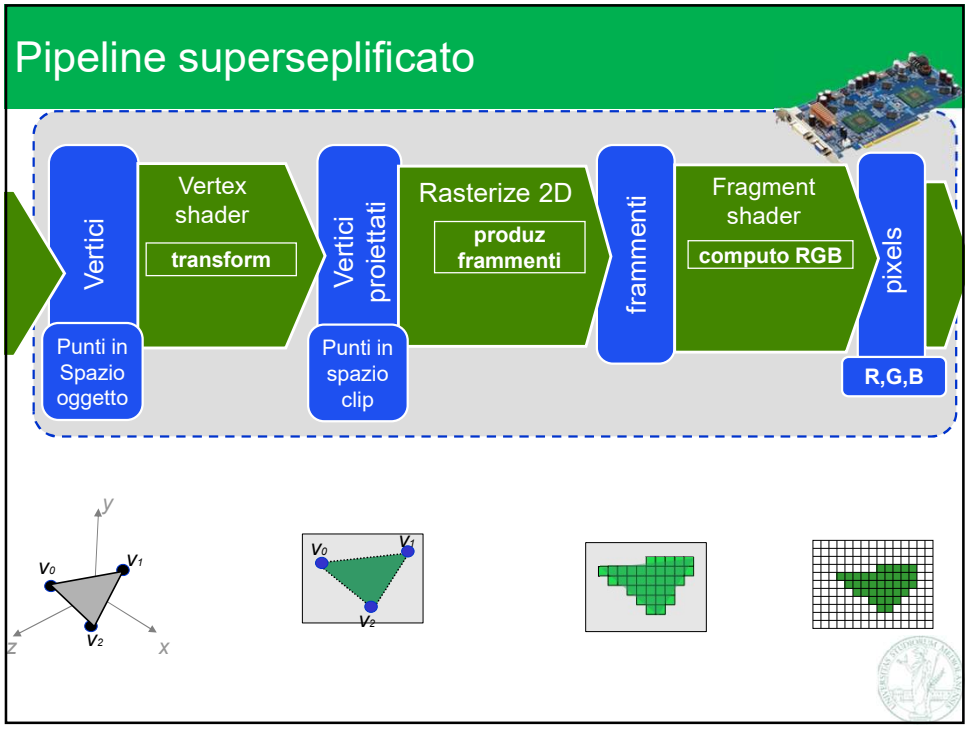


11

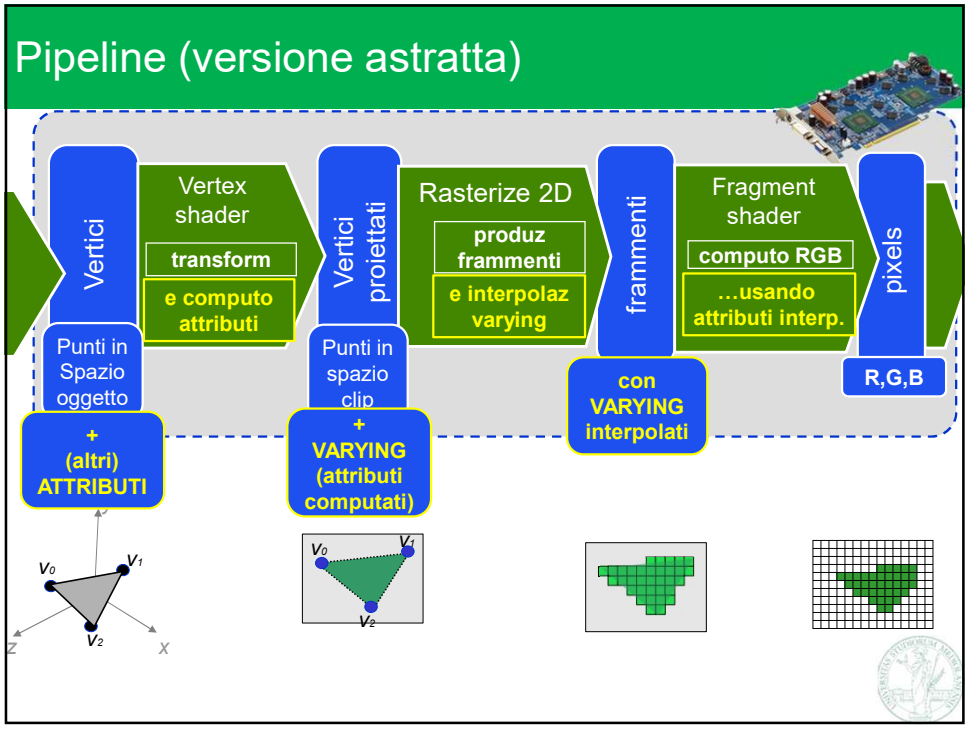
Pipeline (seplificato)



13

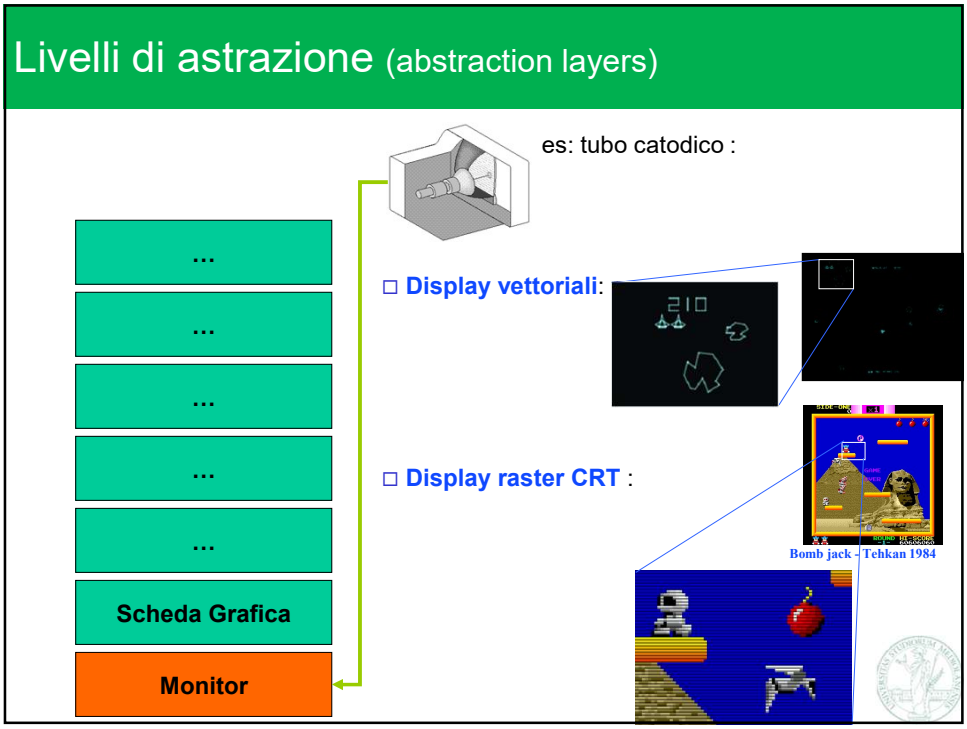

Pipeline superseplificato
Vertici
Punti in Spazio oggetto
Vertex shader
transform
Vertici proiettati
Punti in spazio clip
Rasterize 2D
produz frammenti
frammenti
Fragment shader
computo RGB
pixels
R,G,B
y
x
z
V0
V1
V2

14


Pipeline (versione astratta)
Vertici
Punti in Spazio oggetto
+ (altri) ATTRIBUTI
Vertex shader
transform
e computo attributi
Vertici proiettati
Punti in spazio clip
+ VARYING (attributi computati)
Rasterize 2D
produz frammenti
e interpolaz varying
frammenti
con VARYING interpolati
Fragment shader
computo RGB
...usando attributi interp.
pixels
R,G,B
V0
V1
V2

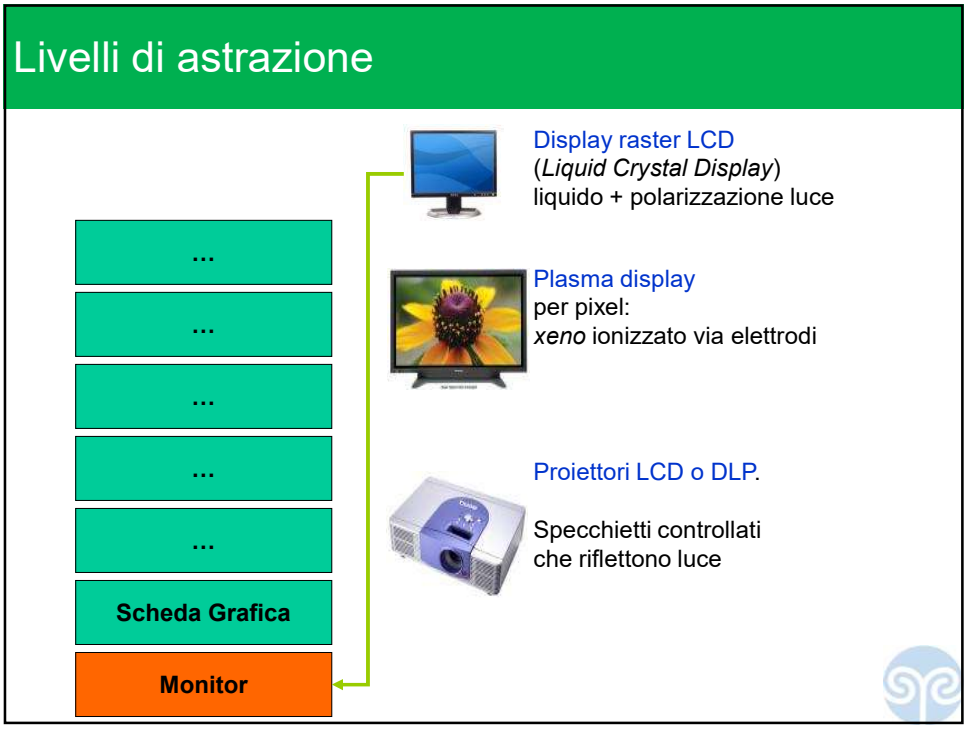
15

Livelli di astrazione (abstraction layers)



16

Livelli di astrazione



17

Livelli di astrazione

Hardware specializzato per CG

- ...
- ...
- ...
- ...
- **Driver Scheda grafica**
- **Scheda Grafica**
- **Monitor**

18

Livelli di astrazione

- ...
- ...
- ...
- **API**
- **Driver**
- **Scheda Grafica**
- **Monitor**

DirectX 12

OpenGL

Vulkan

19

API grafiche diffuse

✓ **Direct3D**

⇒Microsoft
- (proprietario, e non cross platform)

⇒Parte di DirectX

⇒Stessi scopi di OpenGL
- una API per usare le stesse GPU
- struttura non dissimile
  - di solito, meno elegante, più macchinoso
- C (e C++)

⇒E' l'alternativa più comune a OpenGL
- Grossomodo:
  - Direct3D = industrial standard (e XBOX)
  - OpenGL = industrial + academic standard

20

API grafiche diffuse

Vulkan™

⇒by Khronos (again)

⇒versione rivista e corretta delle API OpenGL
- basso livello
- "bytecode" for the shaders
- better debugging
- unified mobile / embedded / desktop

⇒Simile alla vers ≥12 di Direct3D

21

API grafiche diffuse

- ✓ Metal (Apple Inc.)
  - ⇒ Funzionalità simili a OpenGL + OpenCL
  - ⇒ Basso livello
  - ⇒ Solo per iOS, (e macOS, e tvOS)

22

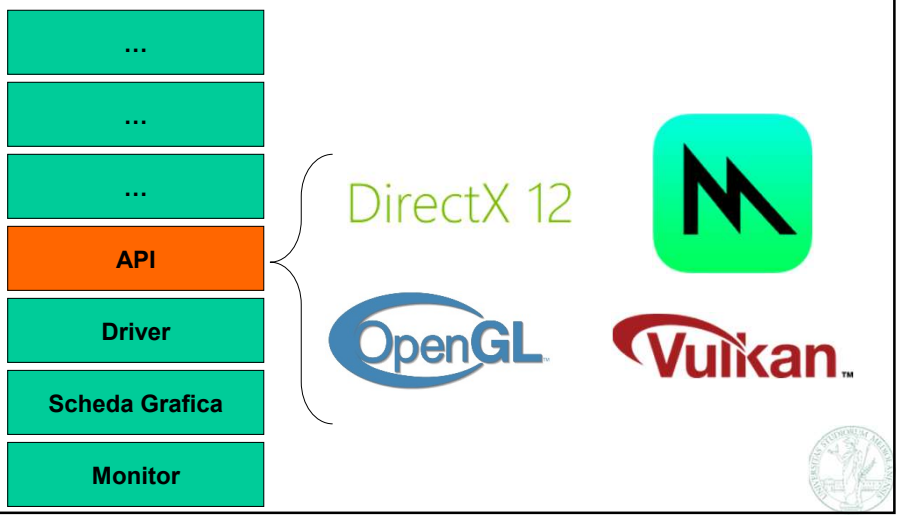
OpenGL : storia

- ✓ inizialmente sviluppato da Silicon Graphics

- ✓ dal 2002 al 2006: OpenGL **A**rchitecture **R**eview **B**oard
  - ⇒ mantiene e aggiorna le *specifiche*
  - ⇒ industria 90%, accademia 10%
  - ⇒ ogni compagnia / gruppo, un voto
- ✓ dal 2006: ARB si evolve nel Khronos Gruop

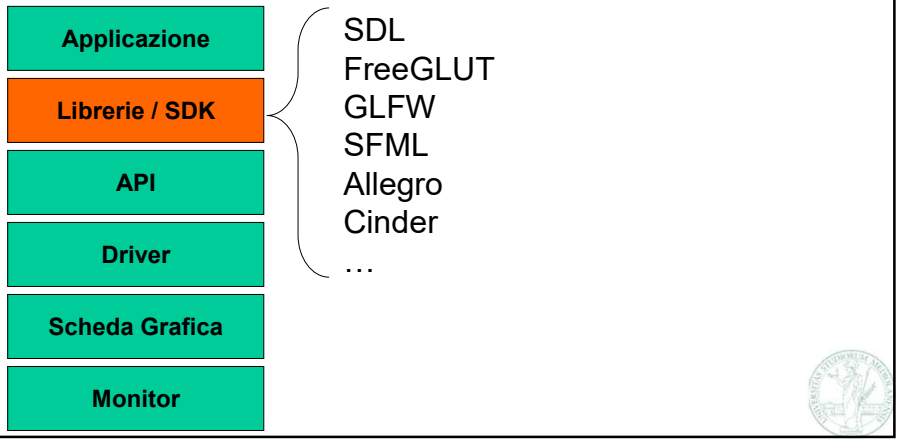
23

Livelli di astrazione



26

Livelli di astrazione



27

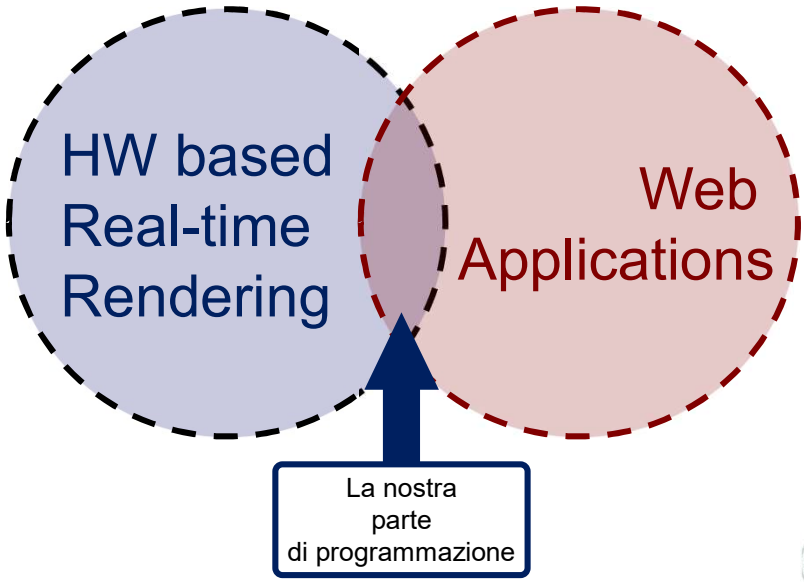
Esempio

28

Esempio

29

Programmazione



30

3D sul web: tentativi di soluzione

✓Soluzioni:

⇒Remote rendering
- Il client richiede, il server renderizza e manda immagini
- Esempio: **Stadia** by Google (per games)
- (fra l'altro: ottimo per DRM!)

⇒Rendering come preprocessing
- Esempio: **QTVR** by Apple, Inc.

⇒Web-oriented formats for 3D interactive scenes
- Esempi: **VRML**, **X3D, Unity bundles**

⇒Plugins:
- **Flash** by Adobe
- **Silverlight** by Microsoft

31

3D sul the Web: le soluzioni più usate oggi

- ✓ **WebGL** by Khronos (API)
  - ⇒ Graphics Hardware acceleration from browsers
  - ⇒ Javascript based
  - ⇒ Native! – no plugin
  - ⇒ ⊂ **GLES 2.0** ⊂ **OpenGL**
- ✓ **three.js**
  - ⇒ Higher level
  - ⇒ OpenSource, *free*, MIT licensed
- ✓ **Emscripten**
  - ⇒ transpiler: C++ + OpenGL → JavaScript + WebGL
- ✓ **Unity** (game engine)
  - ⇒ exports projects as web applications
- ✓ Ad hoc dev-tools:
  - ⇒ **3DHOP** - 3D Heritage Online Presenter
  - ⇒ **Google Earth Engine**
  - ⇒ ...

32

OpenGL : sotto API

OpenGL ES

- ⇒ per **embedded devices**
- ⇒ sottoinsieme di OpenGL (in pratica)

WebGL

- – per **web**
- – Ver 2.0: basato su ver 3.0
- – HTML5
- – un language binding in JavaScript
- – Ver 2.0
- – *soluz emergente per il 3D sul Web (senza plug-in!)*

adottiamo questo

33

I nostri progettini

34

Three.js

- ✓ Grafica su Web made easy!
- ✓ Un API a livello più alto basato su WebGL
  - ⇒ Cross platform, cross browser, cross vendror (davvero)
  - ⇒ Moltissime utili funzioni grafiche, supportano
    - Mesh poligonali
    - Telecamere
    - Luci, materiali, e lighting
    - Animazioni
    - Shaders (in GLSL)
    - Importers
    - Virtual reality
  - ⇒ Supporto allo sviluppo:
    - debuggers, esempi, documentazione…

35

Primo microprogetto – plan of attack
File nella pagina del corso `index00.html`

1. Costruiamo una paginetta per il nostro programma ← HTML
2. Perpariamo three.js
3. Prepariamo una mesh in memoria di un cubo
4. Instanziamo un rendering
5. Disegnamo

← JavaScript + three.js

36

HTML Page

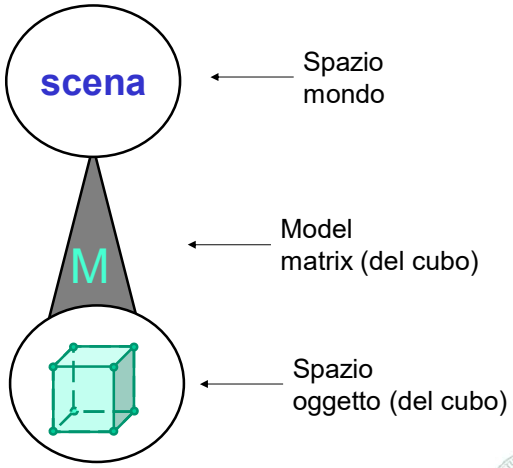
```
<html>
 <head>
  <script>
      … /* qui il JavaScript */
  </script>
 </head>
 <body>
  <canvas
        id     = “mioCanvas"
        width  = 500
        height = 500
        style  = "border: 1px solid black"
  ></canvas>
 </body>
</html>
```

head

body

37

Three.js: scena
struttura ad albero (per ora: di due soli nodi)



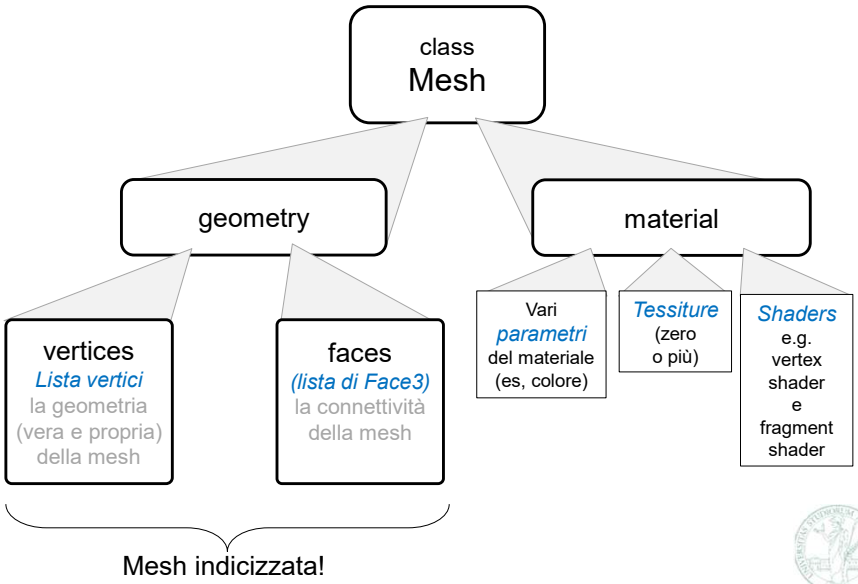
40

Nodi nell'albero della scena
(come il nodo `unCubo` nel codice)

- ✓ Il suo campo `matrix` contiene la model-matrix dell'oggetto in questione
  - ⇒ altri campi utile: matrice `modelViewMatrix` automaticamente aggiornato
- ✓ Per manipolare la matrice, è possibile specificare separatamente:
  - ⇒ Rotazione (`rotation`)
  - ⇒ Scalatura (`scale`)
  - ⇒ Traslazione (`position`)
- ✓ Il campo `matrix` viene settato a: $T \cdot S \cdot R$

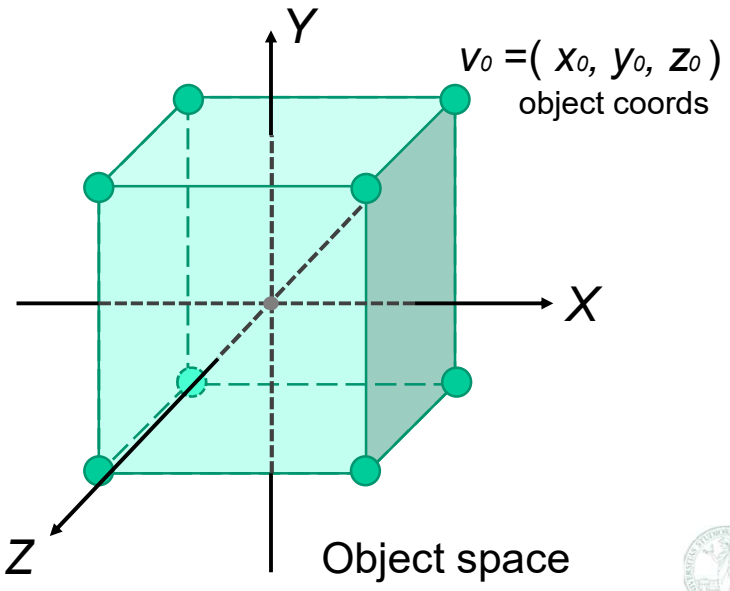
42

Struttura per le Mesh su Three.js



43

Geometria (e connettività) del cubo



44

Camera in three.js

- ✓La variabile `camera` rappresenta la macchina fotografica virtuale
- ✓Contiene: (1) la matrice di proiezione `projectionMatrix`
  - ⇒Dipende (ed è settata automaticametne) dai parametri intrinseci, come `fov`, `aspect`
  - ⇒Nota: sono passati al costruttore
  - ⇒La matrice di vista `matrixWroldInverse` (perché si chiama così?)
    - Rotazione (`rotation`)
    - Scalatura (`scale`)
    - Traslazione (`position`)

45

Camera in three.js

- ✓La variabile `camera` rappresenta la macchina fotografica virtuale
- ✓Contiene: (2) La matrice di vista `matrixWroldInverse`
  - ⇒(perché si chiama così?)
  - ⇒Può essere settata attaverso i parametri estrinseci
    - Rotazione (`rotation`)
    - Traslazione (`position`)
  - ⇒Oppure, come facciamo, attraverso un comando «`lookAt`» (vedi codice)
    - Risolve l'esercizio che abbiamo svolto in classe

46