

Marco Tarini - Grafica & Immagini Digitali 2024/2025
Università degli Studi di Milano

GPU, Drivers, API e librerie (in breve)

1

Hardware specializzato per il rendering

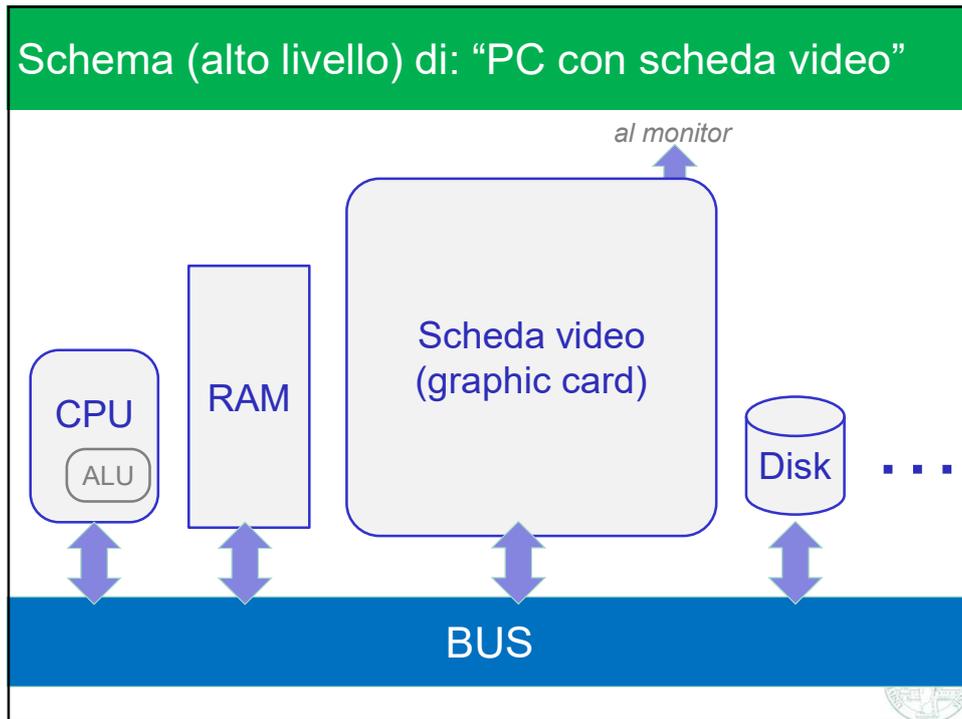
Visione di insieme:

- ✓ **"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

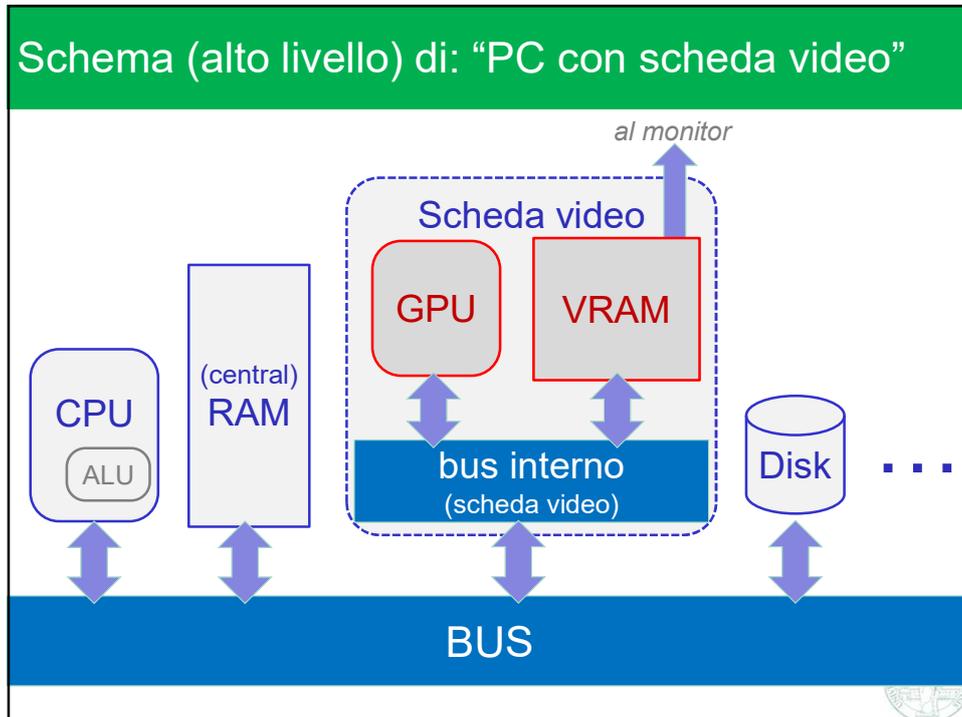
al monitor

dalla scheda madre

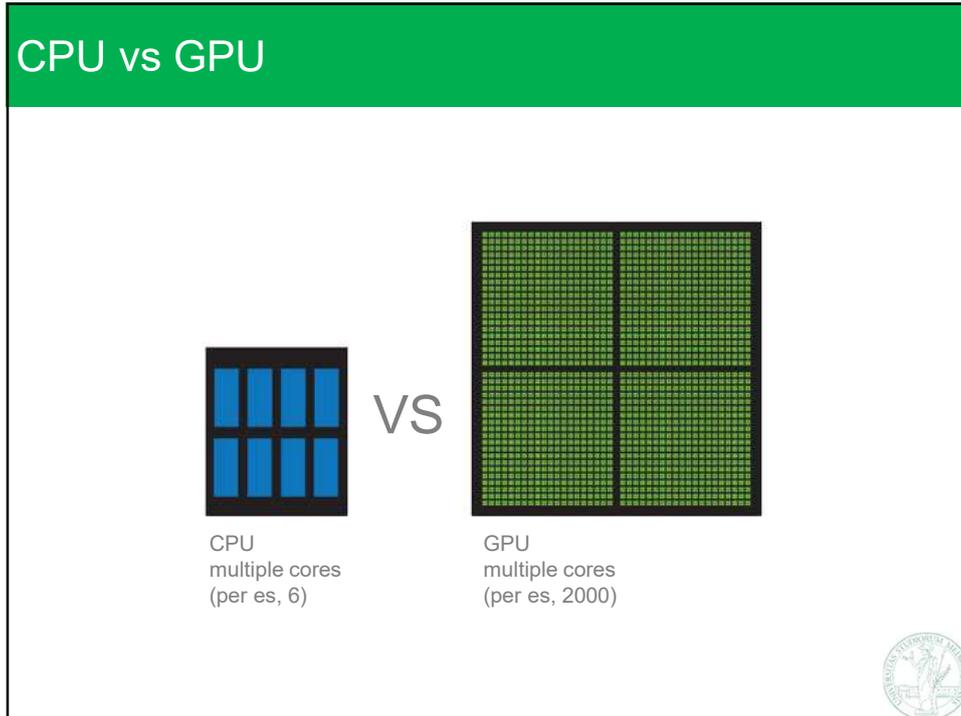
3



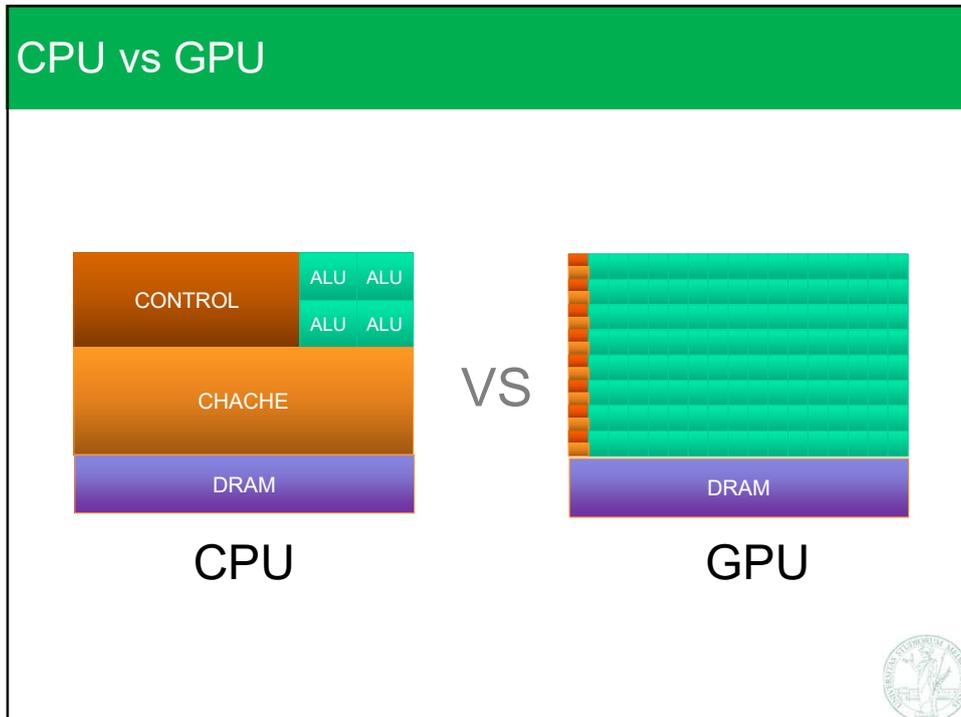
4



5



6



7

CPU vs GPU

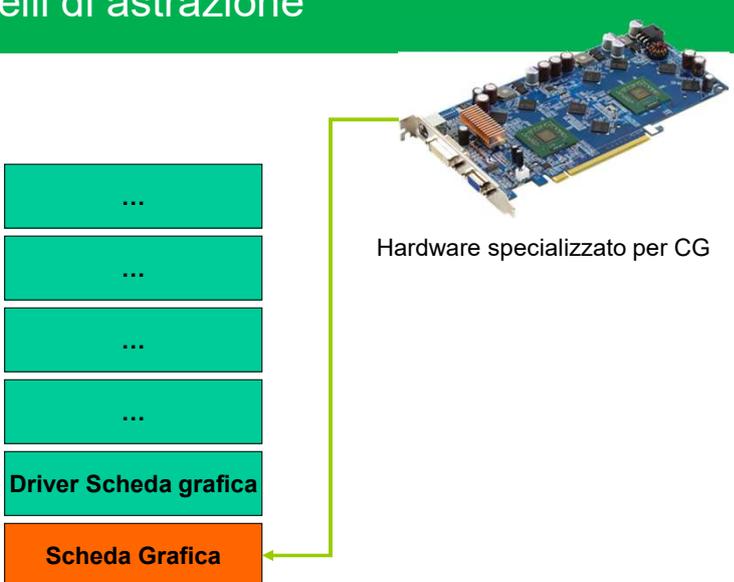
✓ Transistors: >80% control+cache → flessibilità	VS	✓ Transistors: ~80% ALU → potenza (FLOPS)
CPU		GPU

Ad esempio, attualmente (metà 2020):
Una moderna GPU può avere una potenza di calcolo misurabile in centinaia di TERAFLUPS (cioè circa 10^{14} FLOPS)
(FLOPS = Floating-point Operation per Second)
Una moderna CPU: «solo» qualche TERAFLUPS (cioè circa 10^{12} FLOPS, uno o due ordini di grandezza inferiore!)
Questo divario si è mantenuto nelle generazioni



9

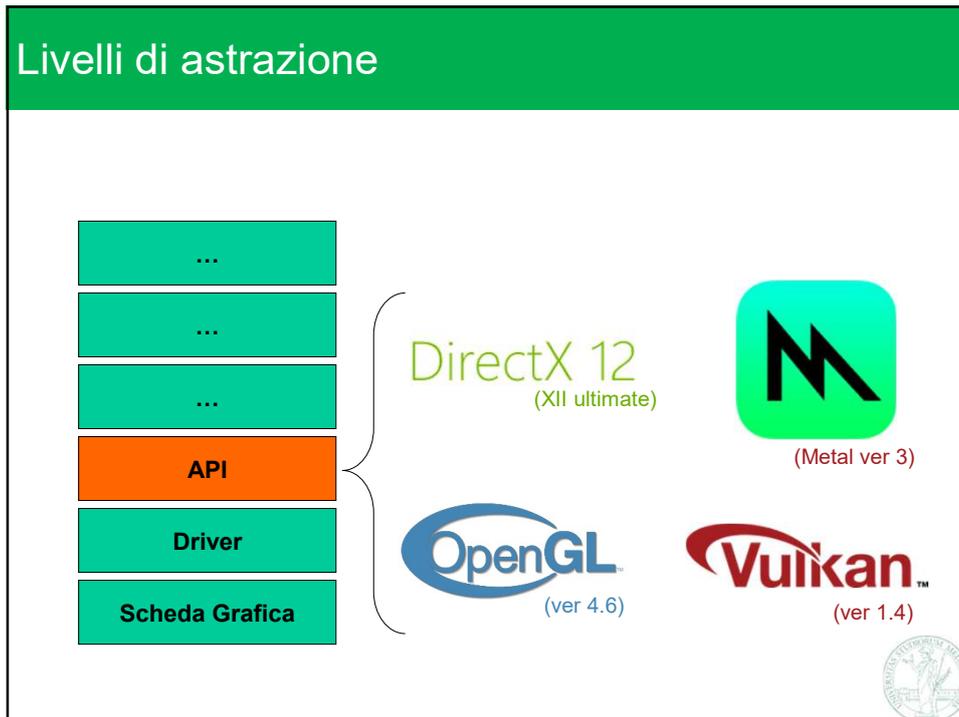
Livelli di astrazione



Hardware specializzato per CG



21



22

OpenGL : storia

- ✓ inizialmente sviluppato da Silicon Graphics 
- ✓ dal 2002 al 2006:
OpenGL Architecture Review Board
 - ⇒ mantiene e aggiorna le *specifiche*
 - ⇒ industria 90%, accademia 10%
 - ⇒ ogni compagnia / gruppo, un voto
- ✓ dal 2006: ARB evolve nel Khronos Group



23

OpenGL : varianti

 ←

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



- per **web**
- Ver 2.0: basato su ver 3.0
- **HTML5**
- un language binding in **JavaScript**
- **Ver 2.0**
- *soluzione stabile per il 3D sul Web*



24

API grafiche diffuse

✓ **Direct3D**

- ⇒ Microsoft
 - proprietario, e **non cross platform**
- ⇒ Parte di DirectX
- ⇒ Stessi scopi di OpenGL
 - una API per usare la GPU
 - struttura non dissimile
 - C (e C++)
- ⇒ E' l'alternativa più comune a OpenGL
 - Semplificando:
 - Direct3D = industrial standard (e )
 - OpenGL = industrial + academic standard
 - Metal = proprietaria Apple



25

API grafiche diffuse



- ⇒ by Khronos (again)
- ⇒ versione più a basso livello delle API OpenGL
 - "bytecode" for the shaders
 - better debugging
 - unified mobile / embedded / desktop
- ⇒ Simile alla vers ≥ 12 di Direct3D



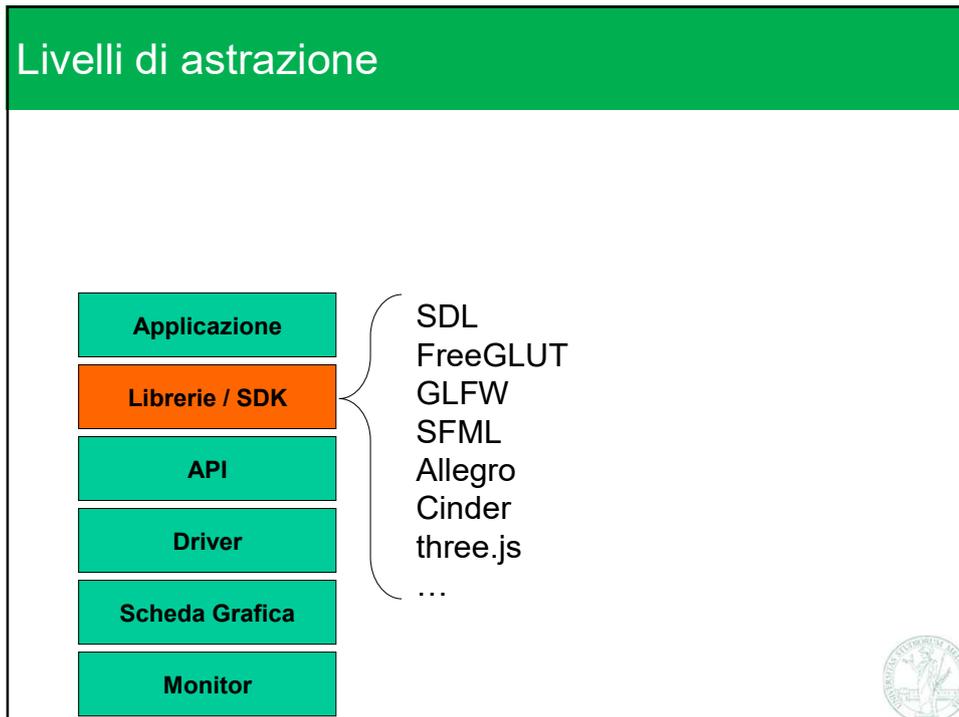
26

API grafiche diffuse

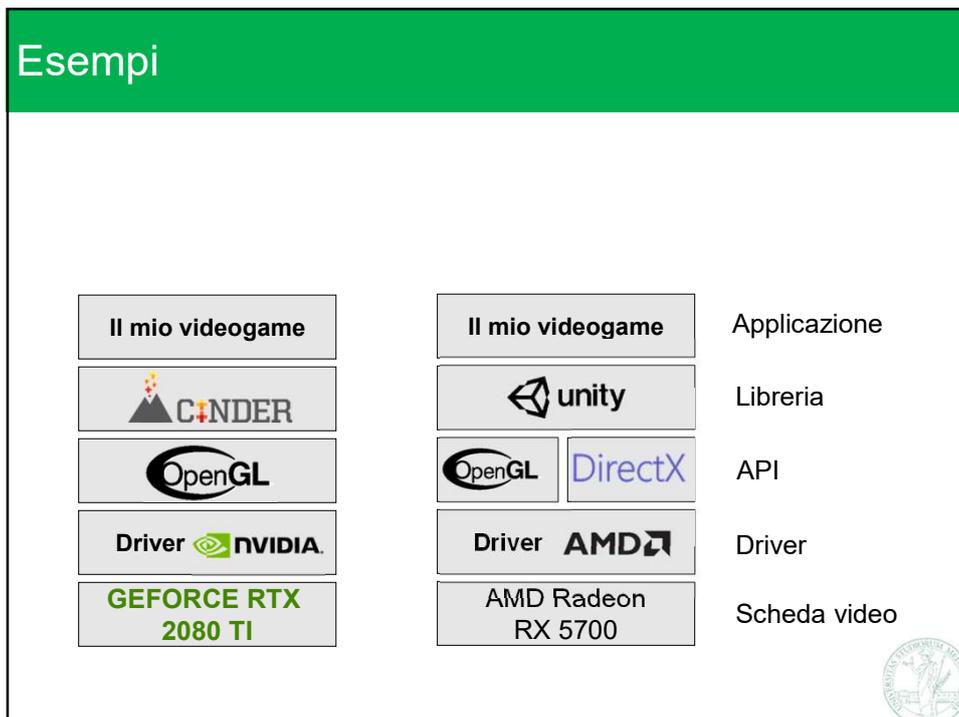
- ✓ Metal (Apple Inc.)
 - ⇒ Funzionalità simili a OpenGL + OpenCL
 - ⇒ Basso livello
 - ⇒ Solo per iOS, (e macOS, e tvOS)



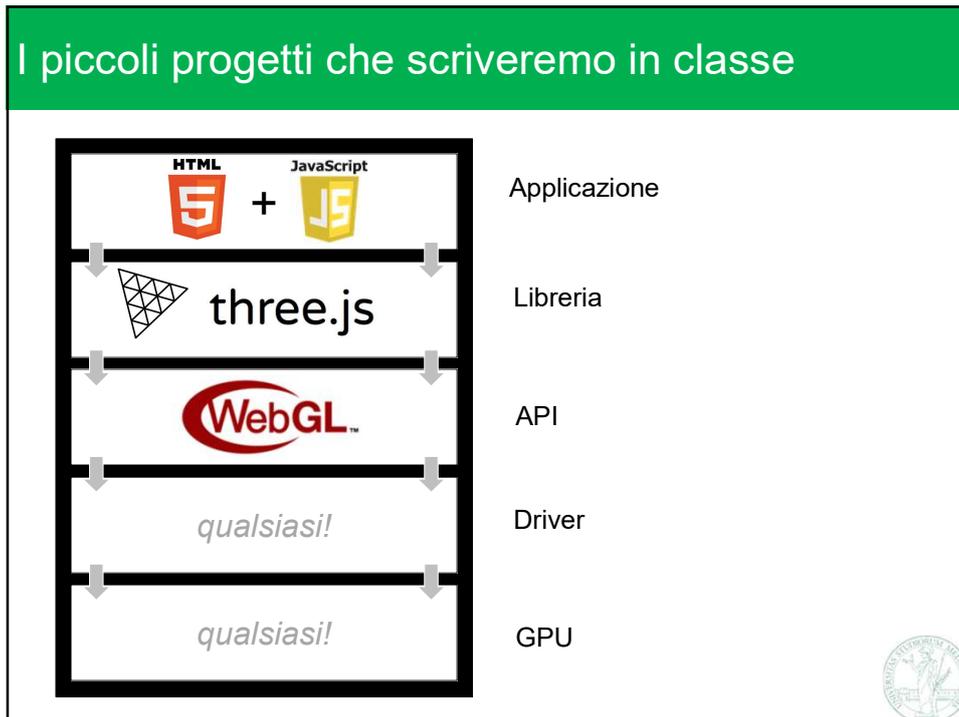
27



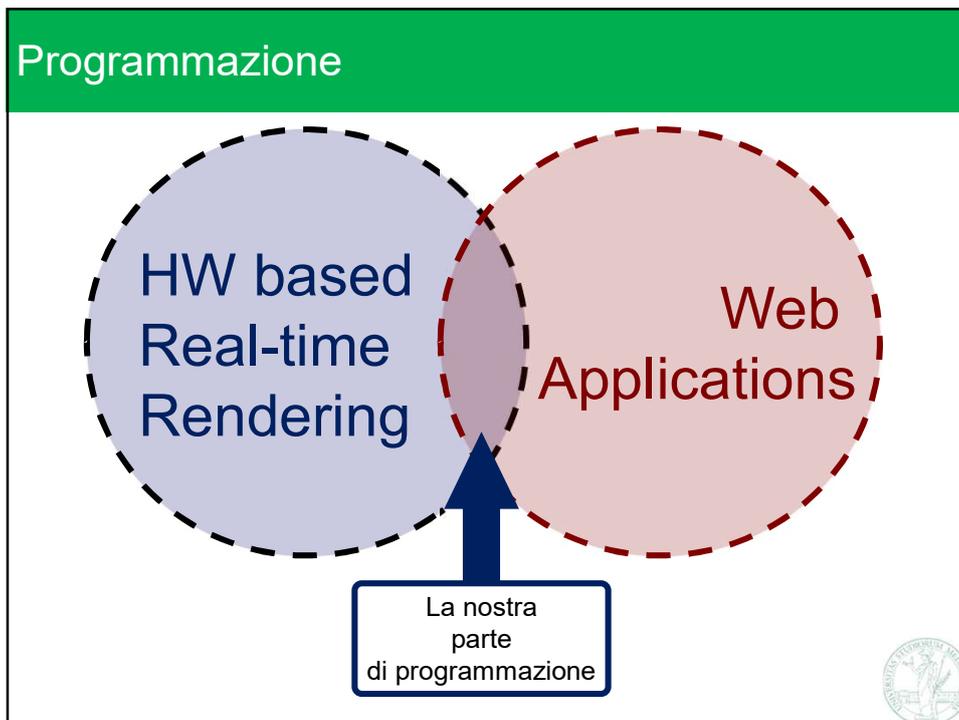
31



34



35



36

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 → WASM + WebGL
- ✓ **Unity** (game engine)
 - ⇒ exports projects as web applications
- ✓ Ad hoc dev-tools:
 - ⇒ **3DHOP** - 3D Heritage Online Presenter
 - ⇒ **Google Earth Engine**
 - ⇒ ...



37

Three.js: CG su Web

- ✓ Una lib ad alto livello basato su WebGL
 - ⇒ Cross platform, cross browser, cross vendor (al 100%)
 - ⇒ Moltissime utili funzioni grafiche: support di
 - Mesh poligonali (quad, tri), splines, subdivision surfaces, voxel models, etc (compreso, lettura da formati file standard)
 - Tessiture
 - Transformazion spaziali (modellazione, vista, proiezione...)
 - Luci, materiali, e lighting
 - Modelli animati ←
 - Shaders (in linguaggio GLSL)
 - Virtual Reality
 - E molto altro

Argomento che non abbiamo trattato. Vedi così di Real-time Graphics Programming e/o 3D Videogames

- ⇒ Supporto allo sviluppo:
 - debuggers, molti esempi, documentazione...



38