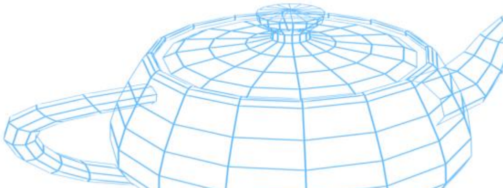
SUPSI

Computer Graphics

Introduction

Achille Peternier, lecturer

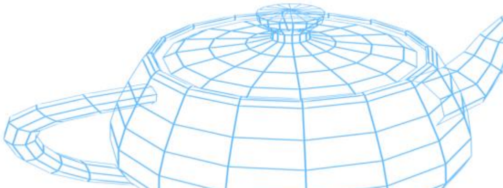


Team

- Achille Peternier, lecturer
  - achille.peternier@supsi.ch
  - contact: forum (preferred, generic questions) or email

- Marco Paoliello, assistant
  - marco.paoliello@supsi.ch
  - contact: practical sessions

- Diego Del Ponte, assistant
  - diego.delponte@supsi.ch
  - contact: email

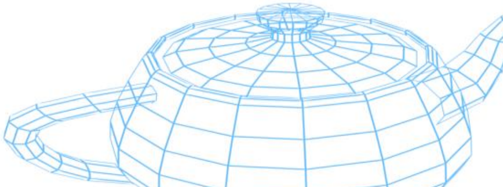


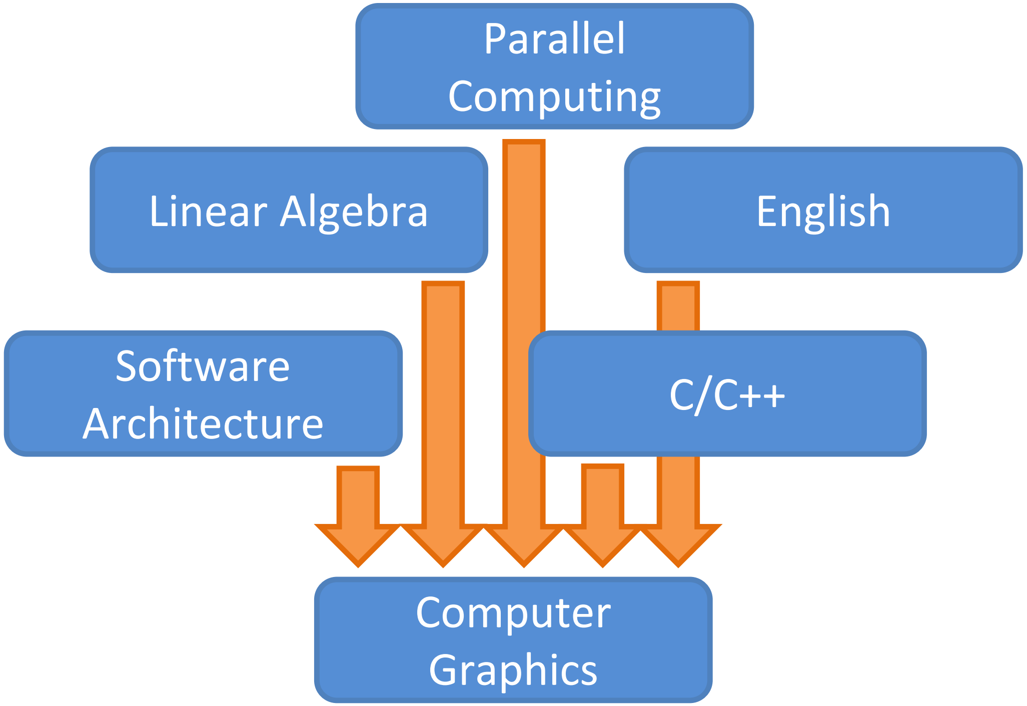
Course material

- iCorsi:
  - M02039 - C02050 Grafica
  - Password: **cg2018$**

- Slides are available (as .pdf) at the beginning of each lesson:
  - Feel free to annotate them using Adobe Reader or similar tools.
  - You can use a laptop/tablet during the class ***exclusively*** for this purpose.

- Exercises (series) are expected to be completed at home:
  - Exercises are corrected in class.
  - Exercises are not graded nor mandatory but **critically** important.
  - Detailed solutions are always provided.





Parallel Computing

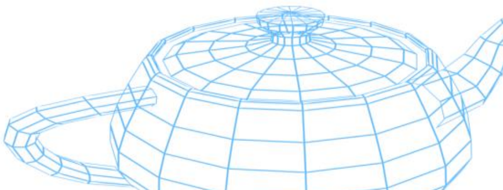
Linear Algebra

English

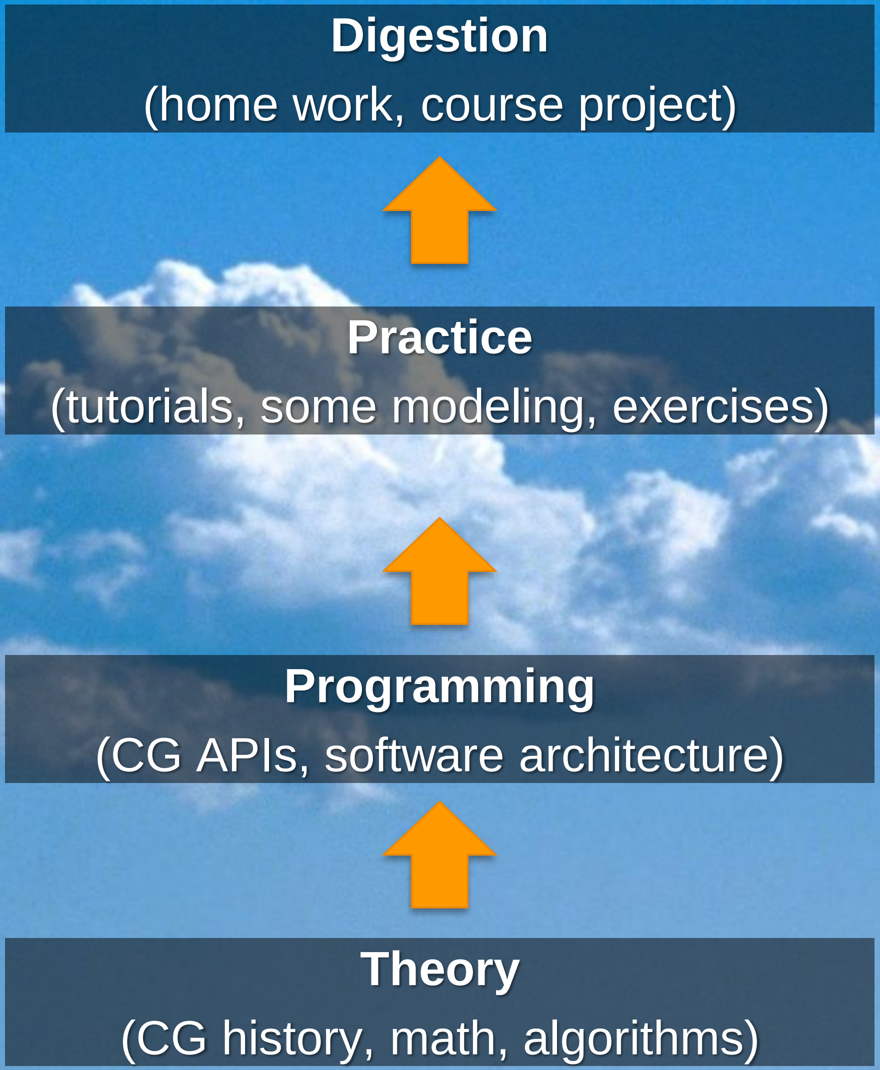
Software Architecture

C/C++

Computer Graphics



Bottom-up approach



Digestion

(home work, course project)

Practice

(tutorials, some modeling, exercises)

Programming

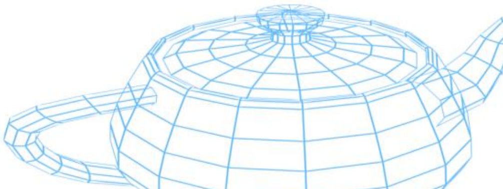
(CG APIs, software architecture)

Theory

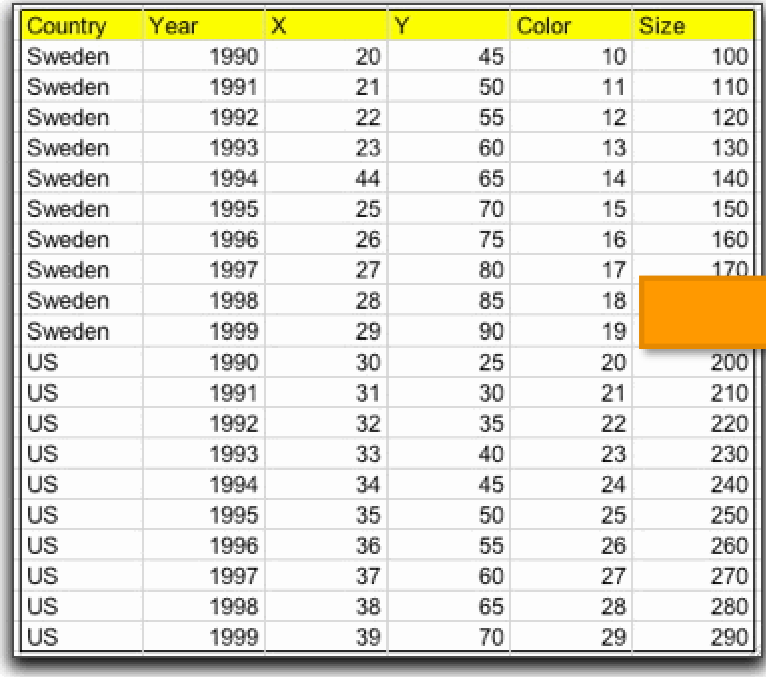
(CG history, math, algorithms)

Course overview

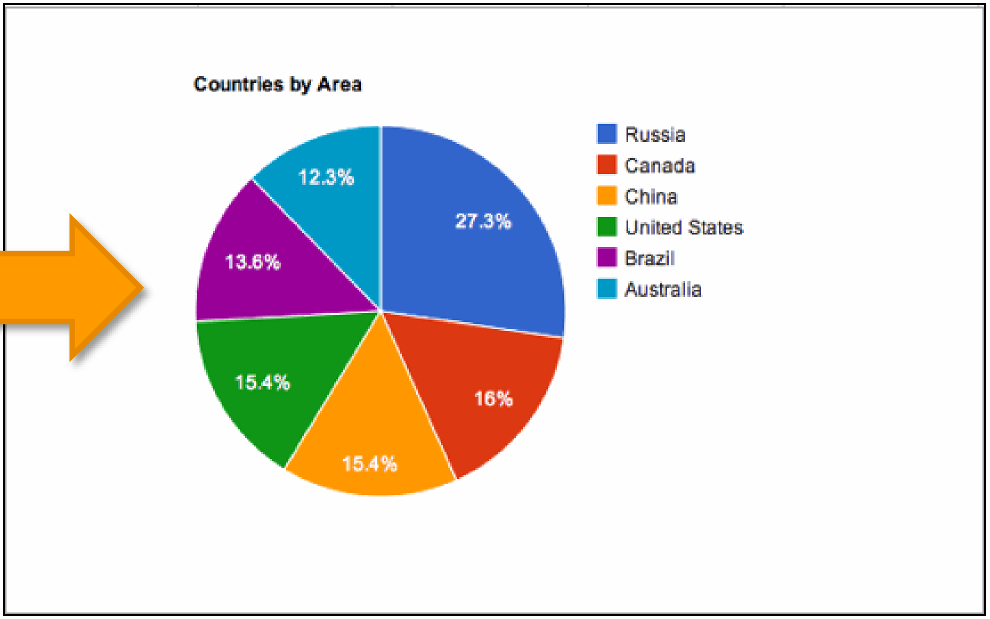
- Introduction:
  - A brief history of CG.
  - Ray tracing vs. rasterization, rendering pipeline.
- Math for CG:
  - Vectors, matrices, transformations, code and libs for 3D math, …
- Modeling and rendering:
  - OpenGL, lighting, materials, cameras, texture mapping, …
  - Scene-graph, graphics engine architecture.
- 2D/3D file formats:
  - .tga, .jpeg, .png, Collada, .fbx, .obj, .3ds, …
- Advanced topics:
  - Transparency, shaders, …



Why Computer Graphics?

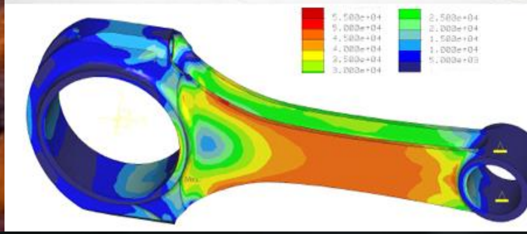


Country	Year	X	Y	Color	Size
Sweden	1990	20	45	10	100
Sweden	1991	21	50	11	110
Sweden	1992	22	55	12	120
Sweden	1993	23	60	13	130
Sweden	1994	44	65	14	140
Sweden	1995	25	70	15	150
Sweden	1996	26	75	16	160
Sweden	1997	27	80	17	170
Sweden	1998	28	85	18	
Sweden	1999	29	90	19	
US	1990	30	25	20	200
US	1991	31	30	21	210
US	1992	32	35	22	220
US	1993	33	40	23	230
US	1994	34	45	24	240
US	1995	35	50	25	250
US	1996	36	55	26	260
US	1997	37	60	27	270
US	1998	38	65	28	280
US	1999	39	70	29	290

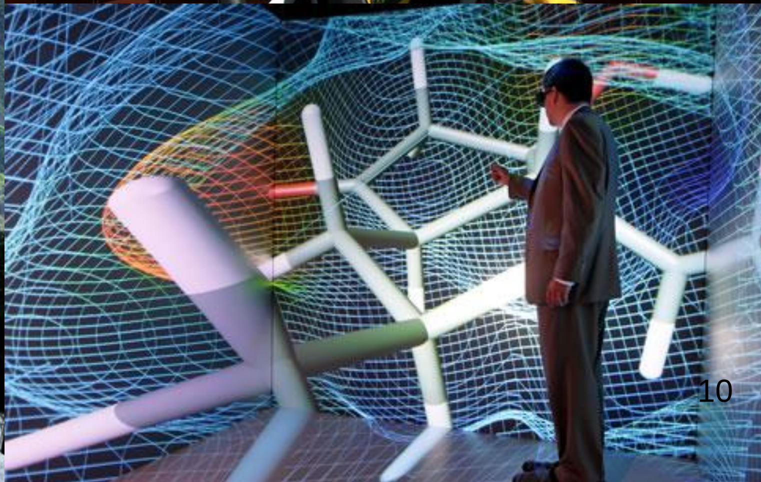


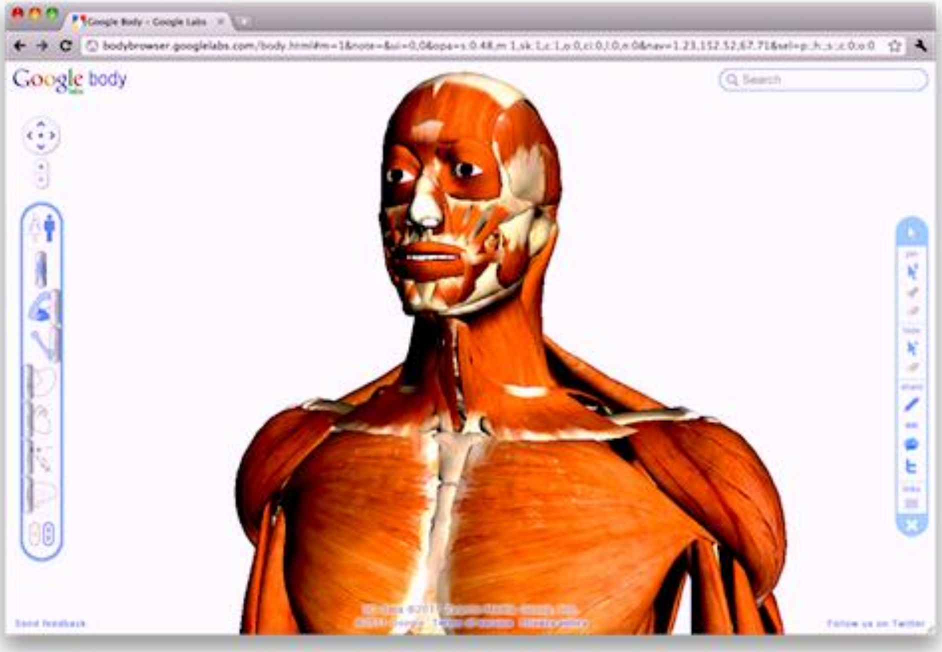
YOURWALLPAPER.COM

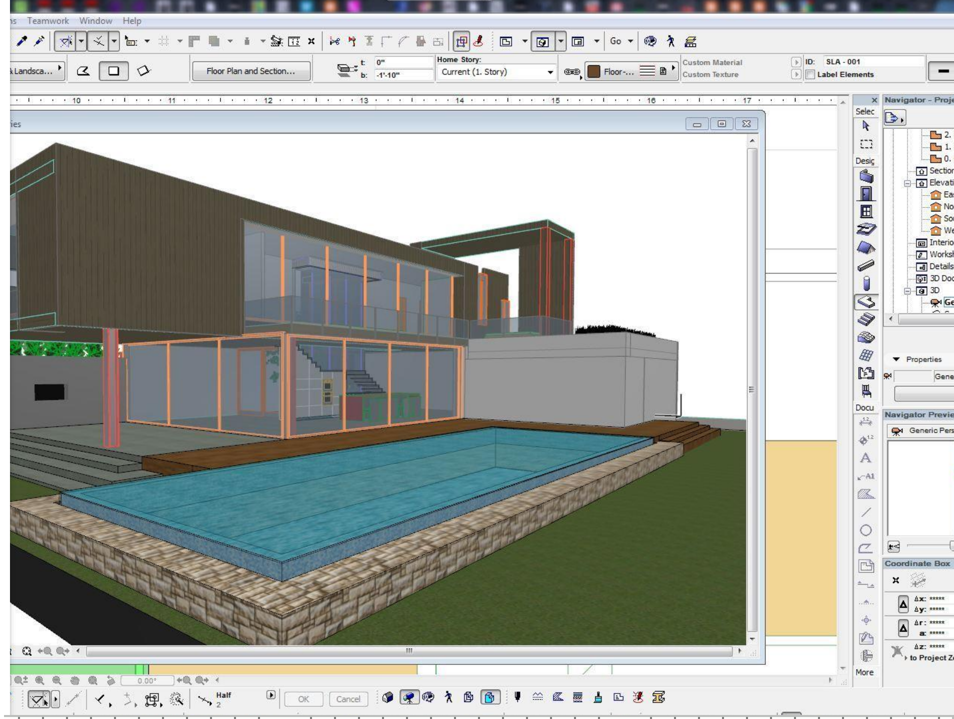
DAS 302
LNAV 232
00804
T 6
29
302
1.545
UPUPUP
04:30


5.500e+04
5.000e+04
4.500e+04
4.000e+04
3.500e+04
3.000e+04
2.500e+04
2.000e+04
1.500e+04
1.000e+04
5.000e+03

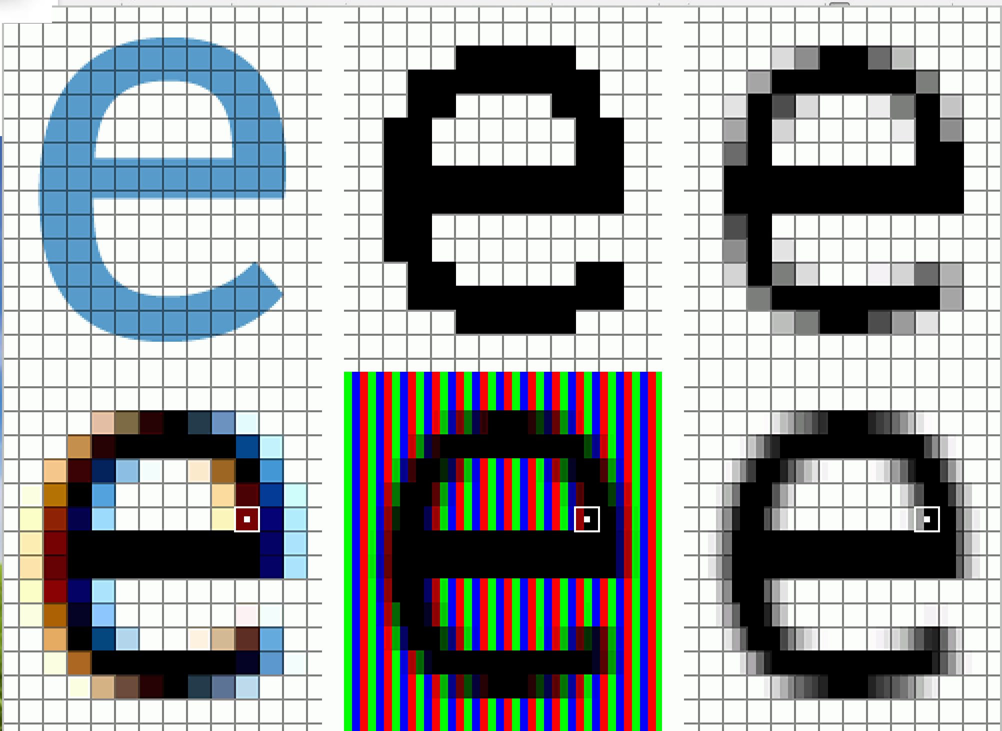
Position
9/16
Lap
2/5
Best Lap
2'35.202
Last Lap
2'35.202
1'12.780
CASIO
DUNLOP
32
km/h
1
sparco




Google Body - Google Labs
bodybrowser.googlelabs.com/body.html#m=1&note=&ui=0,0&opa=s:0.48,m:1,sk:1,c:1,o:0,ci:0,l:0,n:0&nav=1.23,152.52,67.71&sel=p:,h:,s:,c:0,o:0
Google body
Search
Send feedback
Follow us on Twitter

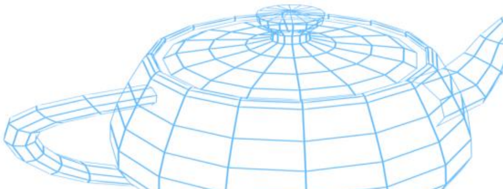

Teamwork
Window
Help
Go
Floor Plan and Section...
Home Story:
Current (1. Story)
Floor-...
Custom Material
Custom Texture
ID:
SLA - 001
Label Elements
Navigator - Proje
Properties
Navigator Previe
Generic Pers
Coordinate Box
to Project Z
More
Half
OK
Cancel

MENU
INFO.
JUMP
ON
OFF



What will you learn?

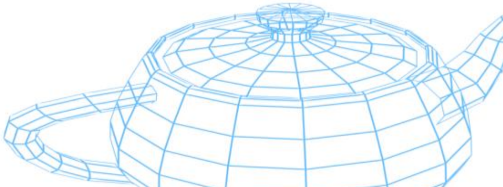
- Primary objectives:
  - Fundamentals of 3D CG.
  - How graphics APIs and dedicated hardware work.
  - How to design and implement a simple 3D graphics engine to render a real-time 3D scene.
  - How to deal with CG biodiversity: standards, file formats, devices, performance, etc.



What will you learn?

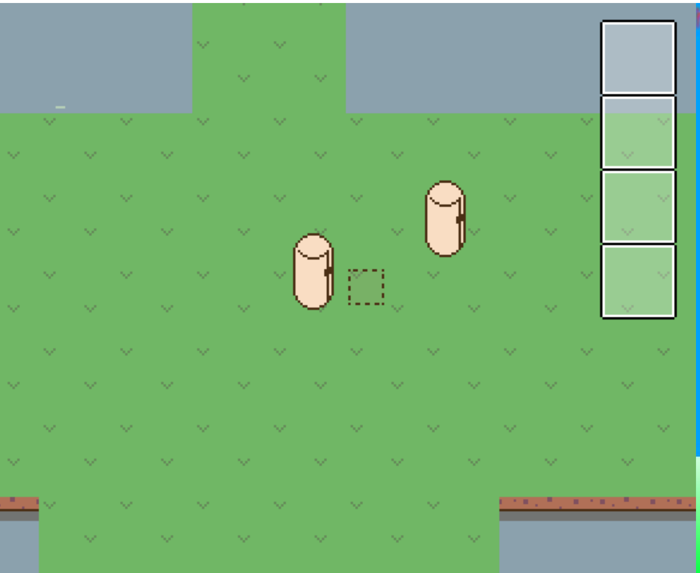
- Secondary objectives:
  - Improve your C/C++ fluency and object-oriented programming skills.
  - Improve your experience in software design.
  - Improve cross-platform project management.
  - Improve teamwork skills.
  - Learn how to optimize resources.
  - Write clean, professional and documented code.

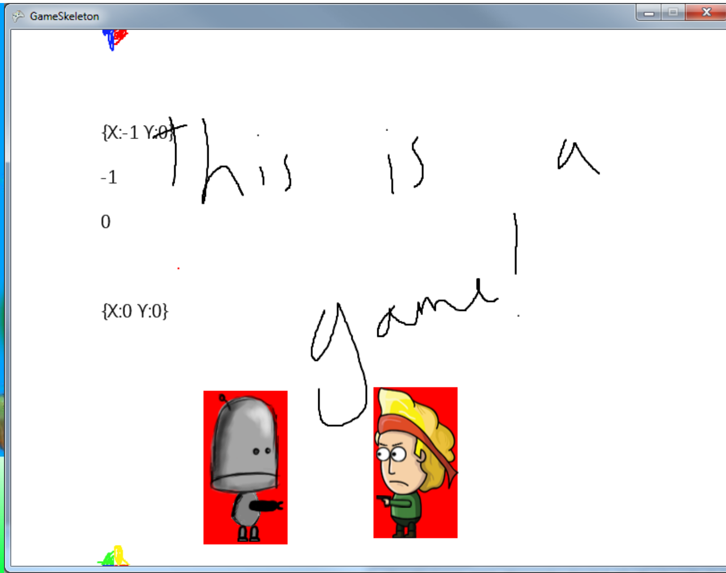
- In collaboration with Laboratorio di Ingegneria del Software II.



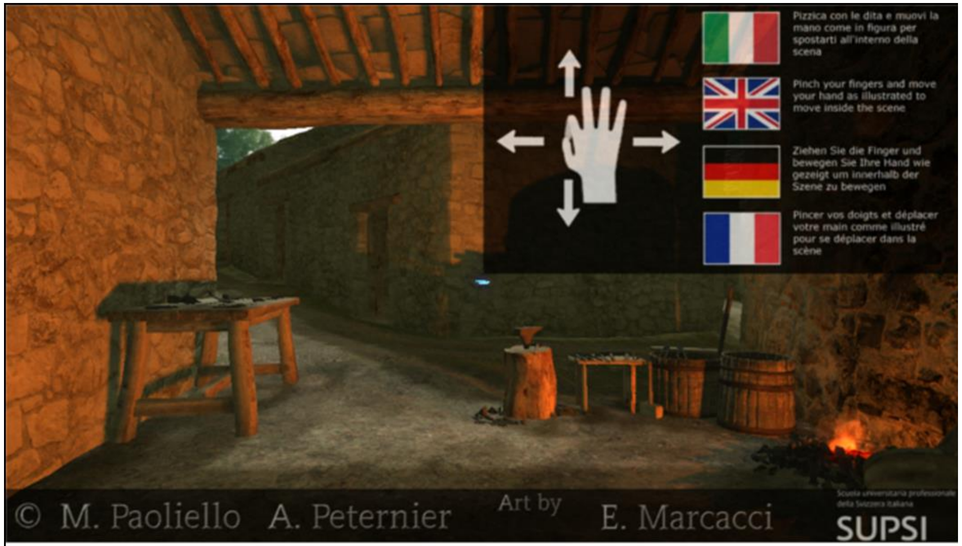
What you will NOT learn

- How to become a 3D graphics designer:
  - …but you will get a crash course on ***programmer art*** ☺
- How to design videogames:
  - …but this topic is covered by other courses (like VR and Game Design).
- Graphics APIs other than OpenGL 1.*:
  - …although you will be familiar with WebGL, OpenGL|ES, more recent versions of OpenGL and DirectX (to some extent).





Bachelor projects



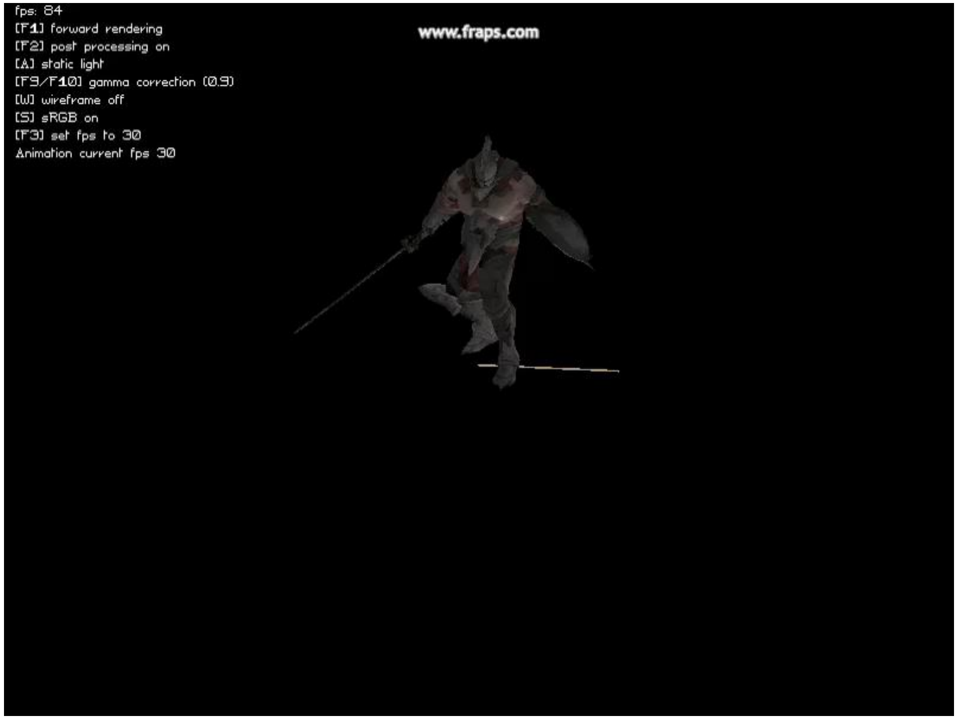
Pizzica con le dita e muovi la mano come in figura per spostarti all'interno della scena

Pinch your fingers and move your hand as illustrated to move inside the scene

Ziehen Sie die Finger und bewegen Sie Ihre Hand wie gezeigt um innerhalb der Szene zu bewegen

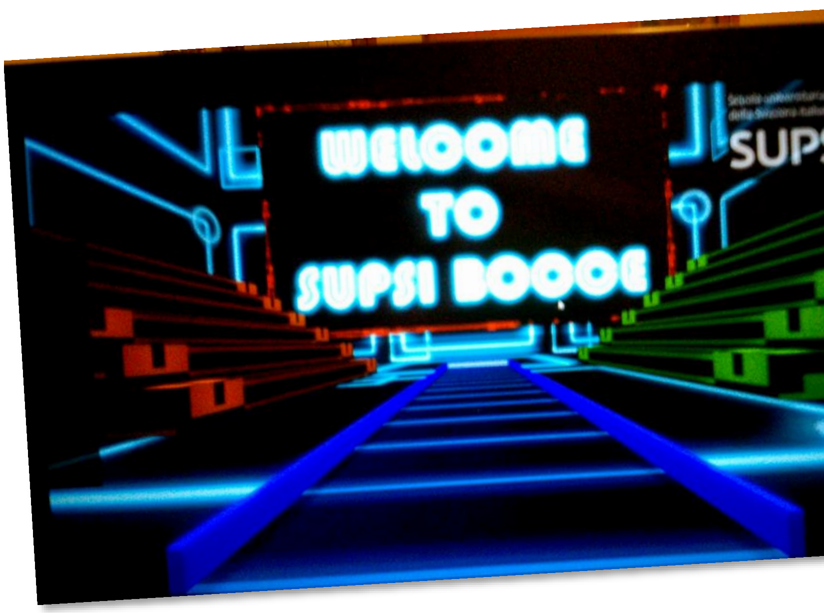
Pincer vos doigts et déplacer votre main comme illustré pour se déplacer dans la scène

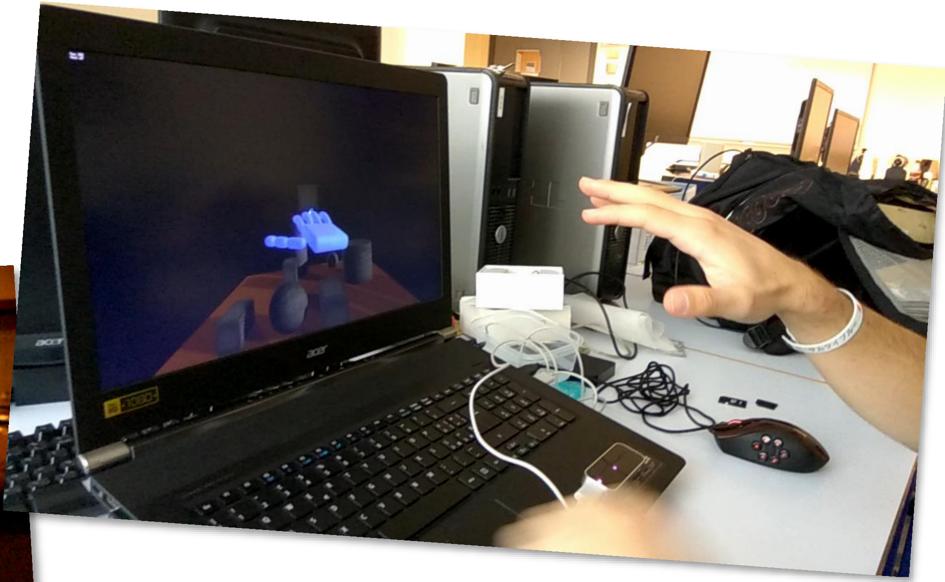
© M. Paoliello A. Peternier Art by E. Marcacci

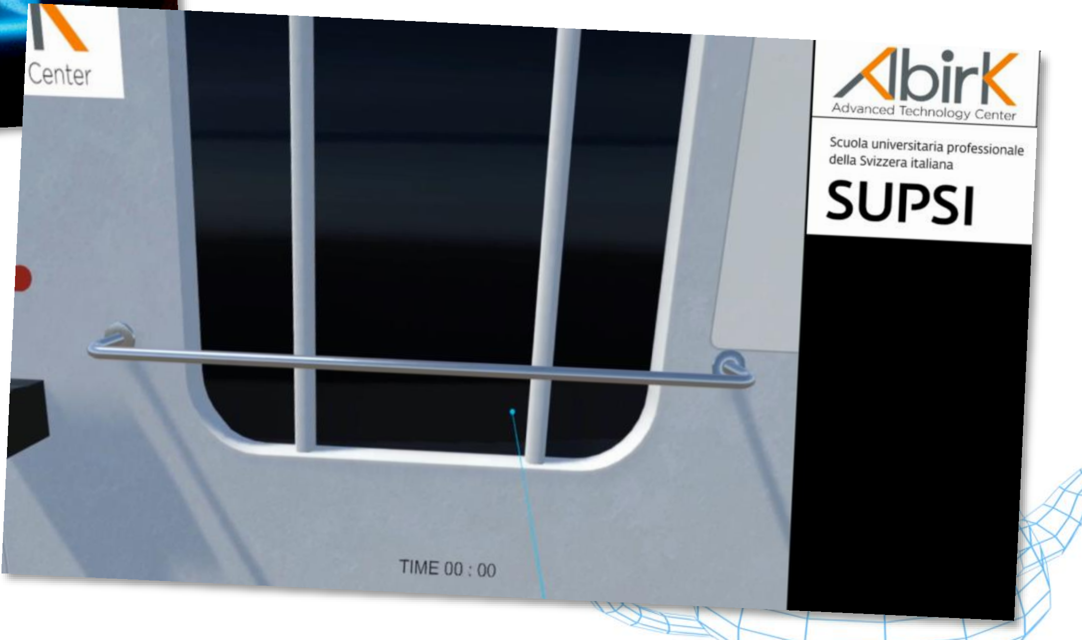


```
fps: 84
[F1] forward rendering
[F2] post processing on
[A] static light
[F9/F10] gamma correction (0.9)
[W] wireframe off
[S] sRGB on
[F3] set fps to 30
Animation current fps 30
```

Master projects



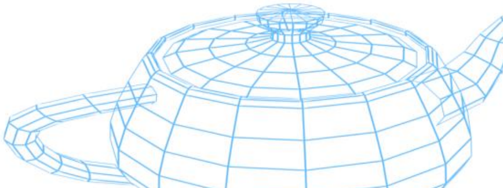



Center
AbirK
Advanced Technology Center
Scuola universitaria professionale
della Svizzera italiana
SUPSI
TIME 00 : 00

Exercises

- About one series each week/two weeks:
  - Started during the practical work sessions, shall be completed on your own time.
  - Teamwork is allowed.
  - Series are not evaluated but are critical to understand and fulfill the course objectives.

- Complete solutions will be provided (Windows/Linux):
  - ***Always*** take some time to go through the solutions and compare them to your work. Ask when something is not clear.
  - Feel free to reuse the code provided in the solutions for your project.
  - I do my best to provide clean solutions (feedback is always welcome): please do the same.



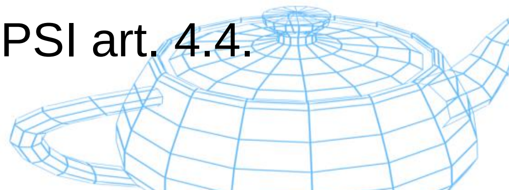
Evaluation

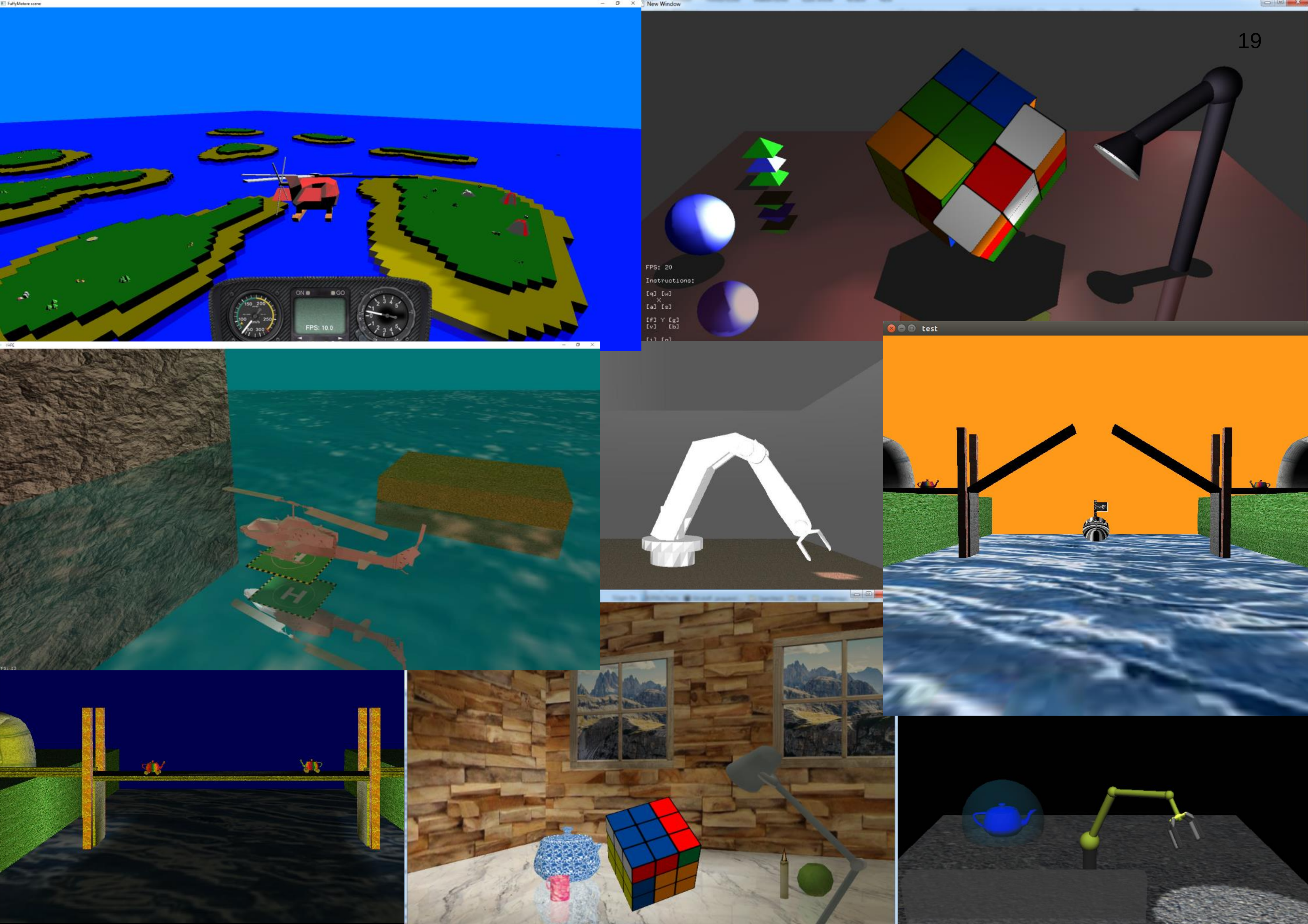
- **20%** written exam:
  - 1h30, near the middle of the semester.

- **40%** course project:
  - Done by teams of 3 students.
  - One ***same grade*** for ***all the members*** of the same team.
  - Will be also evaluated in Laboratorio di Ingegneria del Software II (using different criteria).

- **40%** oral exam:
  - Individual, random questions.

- Intermediate grades range from 1 to 6.
- Final grade also ranges from 1 to 6, refined according to SUPSI art. 4.4.





```
FPS: 10.0
```

```
FPS: 20

Instructions:

[q] [w]
   X
[a] [s]

[f] Y [g]
[v]   [b]

[i] [o]
```

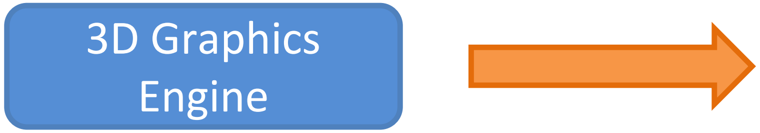
```
FPS: 308.0
```

Previous projects

This year’s project

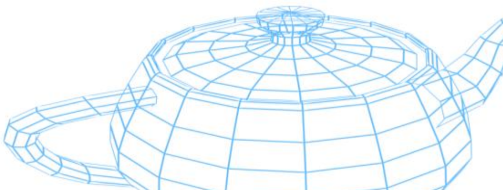
Course project

- Consists of two distinct components: the 3D graphics engine (as a library) and the animated gauntlet (as an application that uses the 3D graphics engine).



- Is a **library** (.dll, .so).
- Has an API (designed by you).
- Provides a series of high-level objects for managing 3D models, light sources, materials, etc.
- Wraps the lower-level components that interact with OpenGL and other internal libraries.

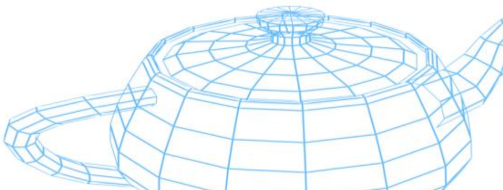
- Is an **application** (.exe).
- Uses the graphics engine to implement the required scenario.
- Integrates the simulation logic.
- Manages the user interaction (via keyboard/mouse).



Course project

3D Graphics Engine

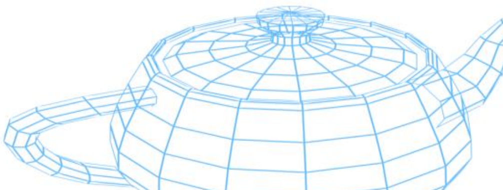
- Design and implement a basic real-time 3D graphics engine featuring:
  - 3D model loading from files.
  - Scene-graph manipulation.
  - Dynamic light sources and cameras.
  - Texture mapping and loading.
  - Transparency.
  - …



Course project

Animated Gauntlet

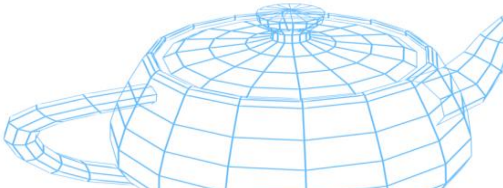
- Use your graphics engine to setup a 3D interactive scene with:
  - A gauntlet with user-controlled fingers (via keyboard, mouse, whatever).
  - A basic, textured environment including a pedestal holding the gauntlet, a polished marble floor and a few additional (static) objects.
  - At least two light sources (one static and one dynamic).
  - An option to allow users to cycle through different cameras showing the scene from different viewpoints (including a dynamic camera).
  - A framerate constantly above 24 FPS.
  - …



www.d20monkey.com
©2016 Brian Patterson

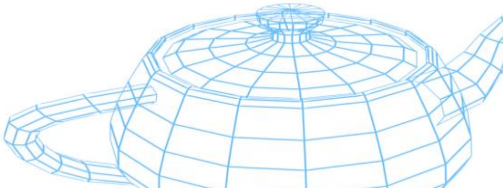
Course project

- Entirely written in C/C++.
- Windows (mandatory):
  - Visual Studio 2017 Community Edition.
  - 3D Studio Max 2018.
- Linux (Ubuntu 18.04 LTS recommended):
  - CodeBlocks:
    - http://www.codeblocks.org/
- The project must compile and run on both platforms (will be checked on my machine).
- MacOS:
  - …at your own risk!
  - In case, MacOS will replace Linux.



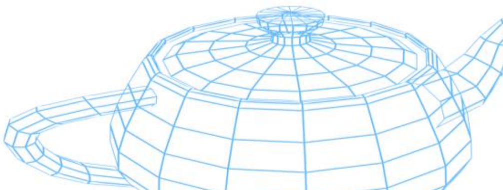
Course project

- External libraries and APIs:
  - OpenGL:
    - http://www.opengl.org/
  - FreeGLUT:
    - http://freeglut.sourceforge.net/
  - GLM:
    - http://glm.g-truc.net/
  - FreeImage:
    - http://freeimage.sourceforge.net/
  - …



Course project

- Work on the project during the practical work sessions (in parallel with the exercises), during Laboratorio di Ingegneria del Software II, and on your own.
- Coding style ***does*** matter:
  - Sound object-oriented implementation.
  - Comments, variable names, English, …
  - Adopt a programming style and stick to it (e.g., Google style): http://google-styleguide.googlecode.com/svn/trunk/cppguide.html
- Usage of automatic documentation tools is strongly advised (e.g., Doxygen).
- Write a report:
  - min. 8, max. 12 pages.
  - written in ***English***.
- Dump everything to CD/DVD (including a runnable demo).



Course project

- Work ***with*** your teammates and share information: each group member is supposed to know ***every*** line of the code, including contributions written by other members.

- You can copy/paste code from the web, from my solutions and from other projects but you have to ***completely*** understand it:
  - Projects will be evaluated through individual questions during the final presentation.

- Start working on the project ***regularly*** and progressively: do not wait until the last minute!

- Already keep in mind that the 3D graphics engine will be further extended in the optional VR course.