SI4 - CRÉATION DE MONDES VIRTUELS

INTRODUCTION TO 3D WORLDS

Hui-Yin (Helen) Wu

Chargée de recherche (ISFP), Centre Inria d'Université Côte d'Azur

PLAN

1. Course overview
2. Fundamentals in computer graphics
3. 3D worlds and tools to create them
4. 3D modeling with Blender

COURSE OVERVIEW

By the end of this course, you will have:

1. Global understanding of the processes that allow one to go from idea to fully fledged 3D worlds
2. Synergy with domains of design, user experience, and interaction
3. Created an interactive 3D demonstration

In short: the knowledge that will enable you to pursue further advanced study and a professional research or industriel career in domains relevant to 3D virtual worlds.

COURSE OVERVIEW

What this course will not teach:

1. Virtual and augmented reality (SI5-IHM)
2. How to make video games
3. How to become a 3D animator

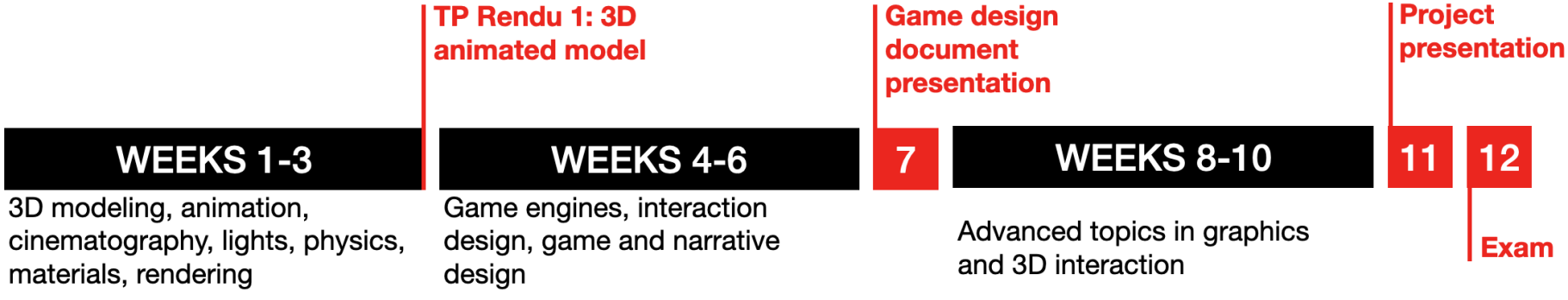
COURSE OVERVIEW

In this course, you need to:

1. Submit two practicals (rendus TD) - 20%
2. Complete a team project - 40%
3. Pass an exam - 40%

COURSE OVERVIEW

Planning:



COURSE OVERVIEW

Key rules:

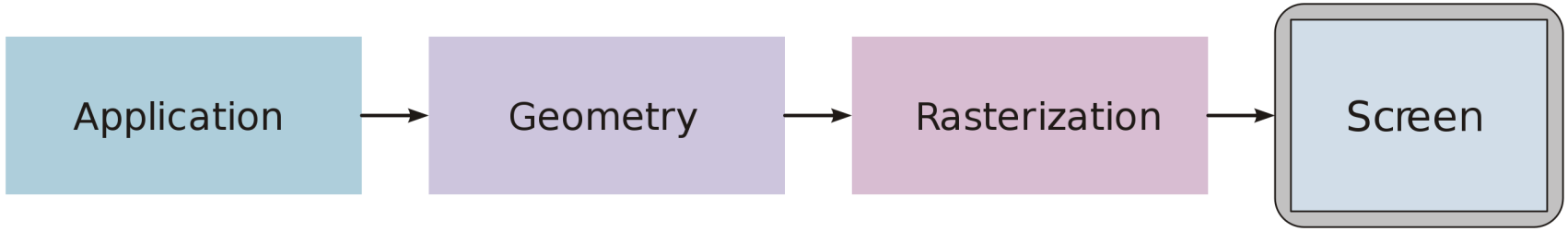
1. Objective score: 0-15, subjective score: 15+
2. Plagiarism = 0 (1st time the assignment, 2nd time the course)
3. Copying = score / N (where N = number of people involved)

PLAN

1. Course overview
2. Fundamentals in computer graphics
3. 3D worlds and tools to create them
4. 3D modeling with Blender

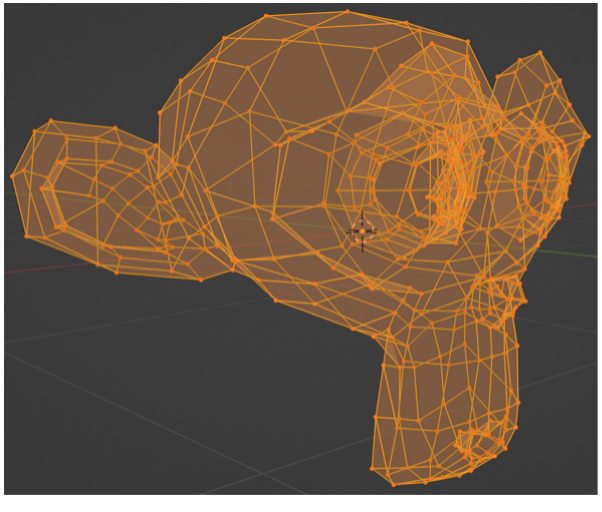
FUNDAMENTALS IN COMPUTER GRAPHICS

The **graphics pipeline** concerns rendering a 3D scene onto a 2D screen. It is composed of four steps:

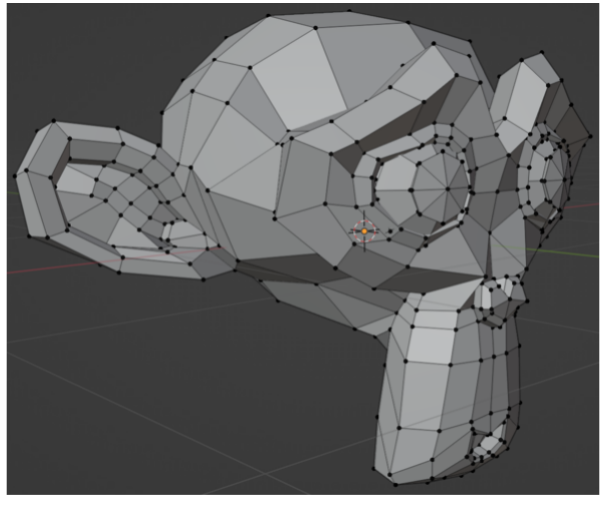


FUNDAMENTALS IN COMPUTER GRAPHICS

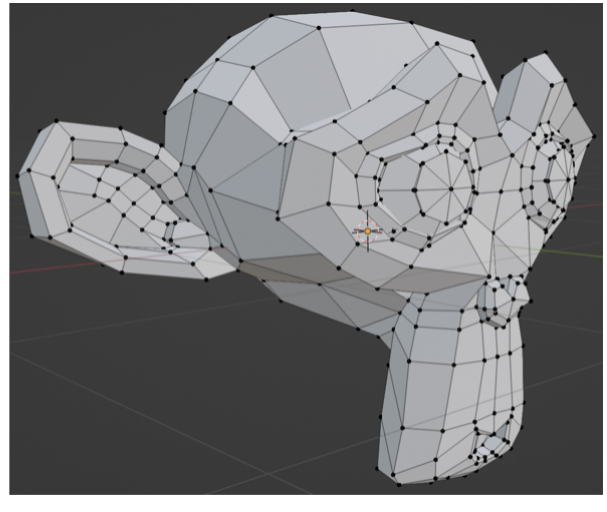
The **application step** concerns all that happend in the 3D world: user inputs, collisions, animations, physics, etc. and calculates the geometric primitives (triangles, vertices, edges ...) for the geometry step.



GEOMETRY



LIGHTING



VIEWPORT

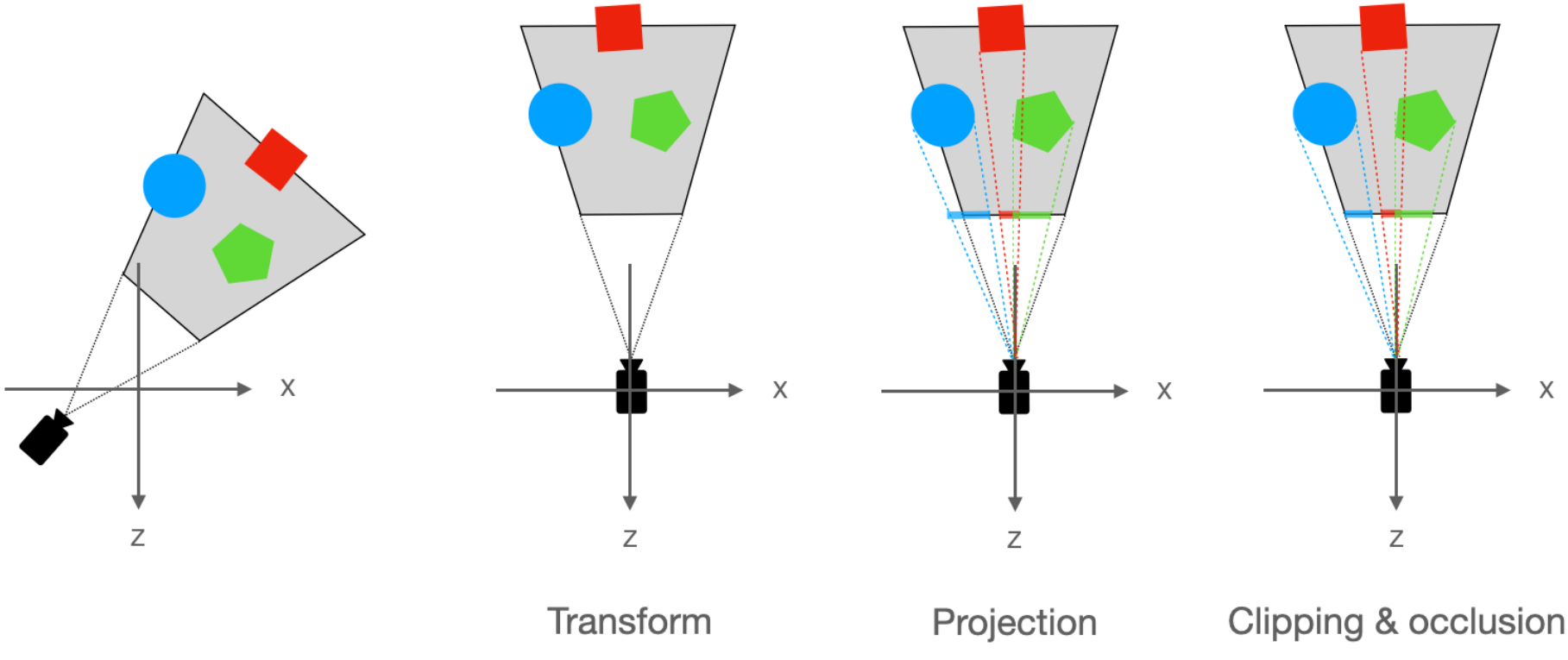
FUNDAMENTALS IN COMPUTER GRAPHICS

The **geometry step** prepares the primitives from the previous step by:

- Transforming the 3D model into world coordinates
- Calculating the lighting (with raytracing for example)
- Projecting the 3D model into the visual volume (frustrum)
- Clipping forms outside of the frustrum
- Transforming the frustrum into the 2D window (viewport)

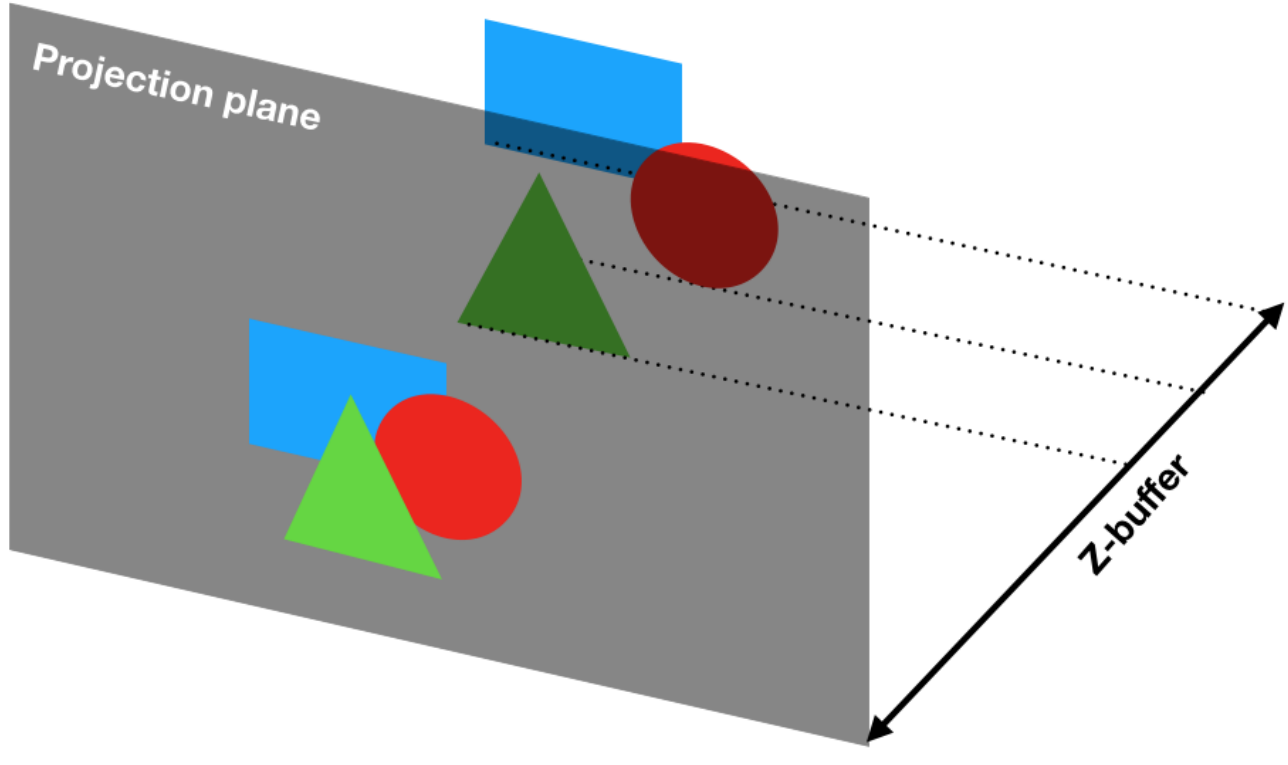
FUNDAMENTALS IN COMPUTER GRAPHICS

The **geometry step** visualized:



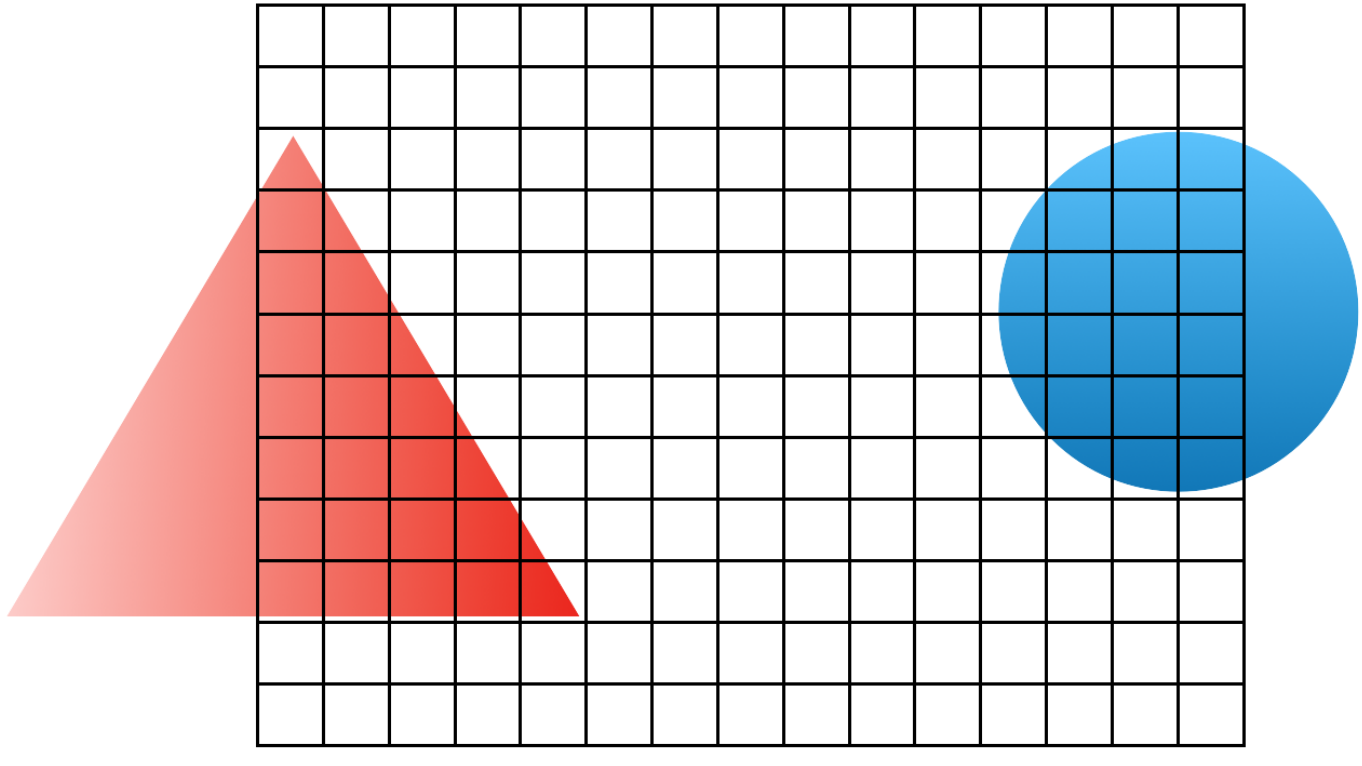
FUNDAMENTALS IN COMPUTER GRAPHICS

The **rasterization step** then calculates for each primitive the colors on the 2D viewport, and uses the Z-buffer to determine the occlusion.



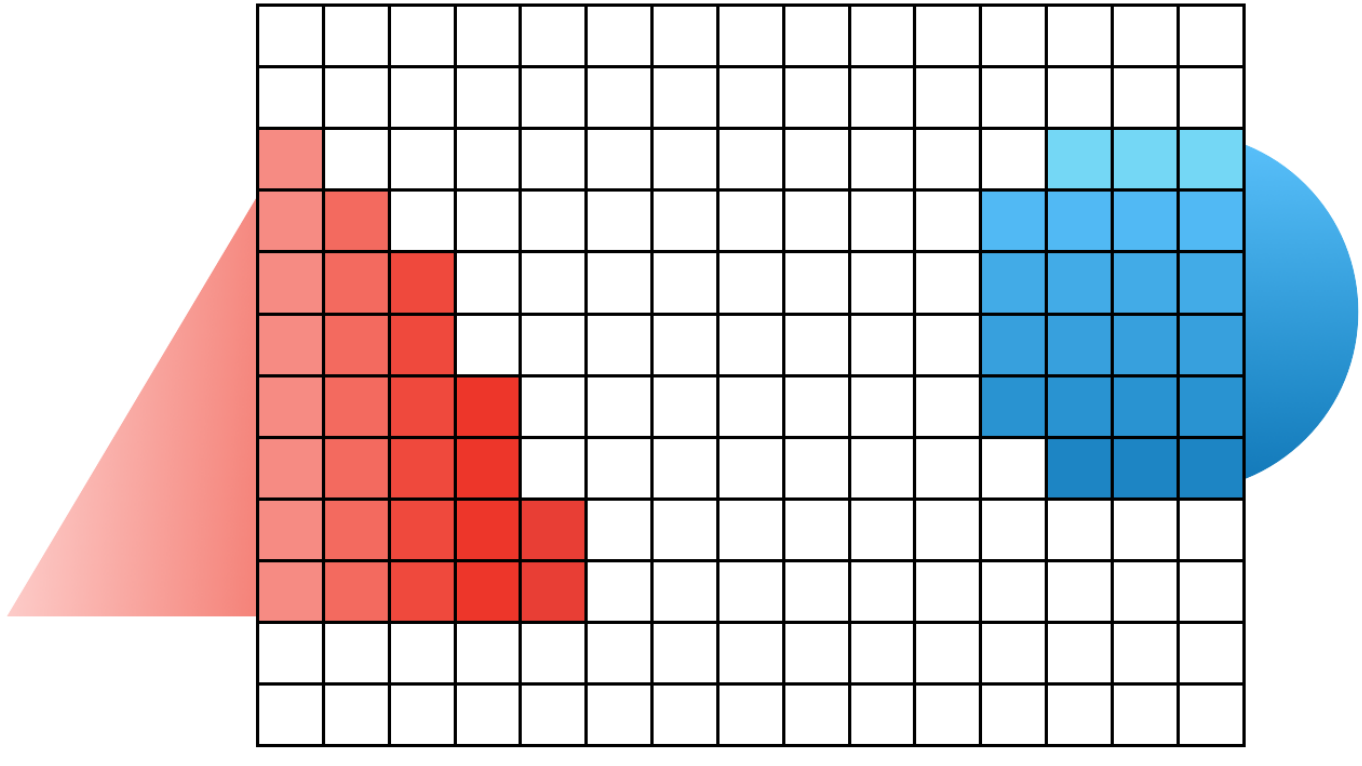
FUNDAMENTALS IN COMPUTER GRAPHICS

The **rasterization step** then calculates for each primitive the colors on the 2D viewport, and uses the Z-buffer to determine the occlusion.



FUNDAMENTALS IN COMPUTER GRAPHICS

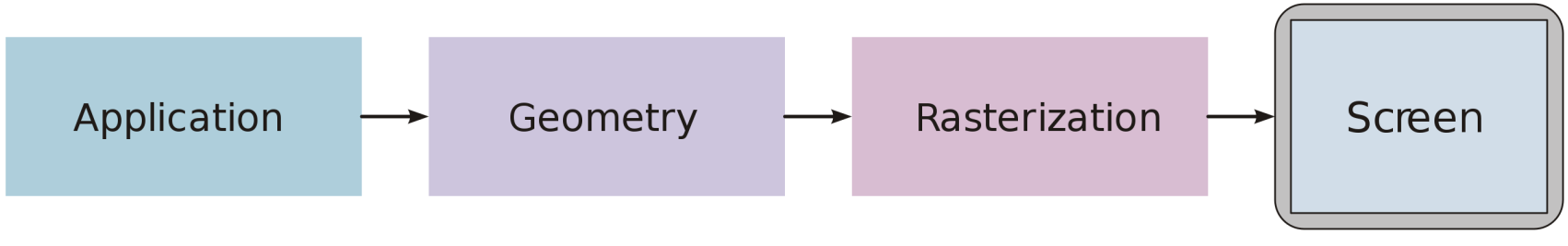
The **rasterization step** then calculates for each primitive the colors on the 2D viewport, and uses the Z-buffer to determine the occlusion.



FUNDAMENTALS IN COMPUTER GRAPHICS

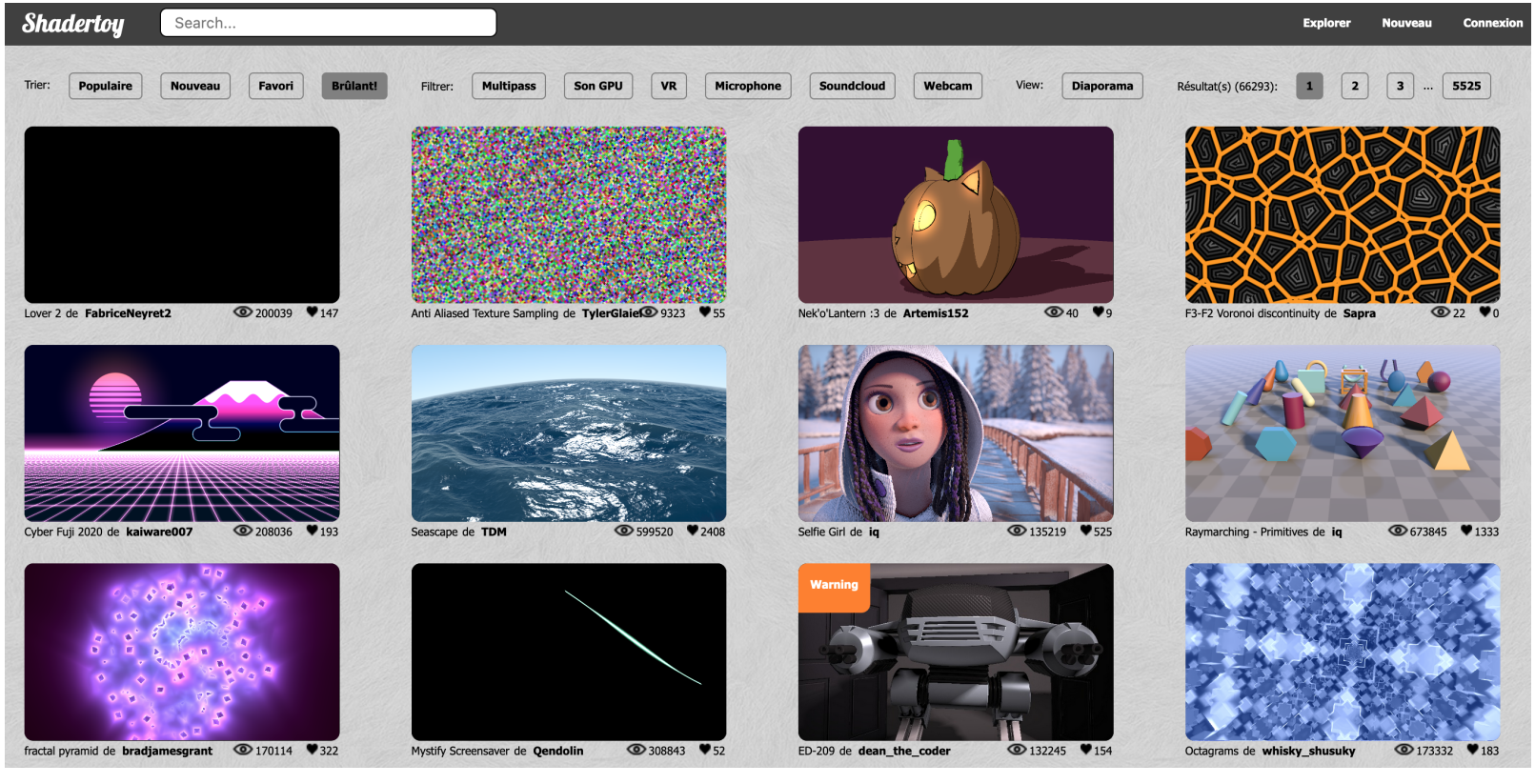
This course only deals with the application step.

So why talk about mention rendering in detail? Art, realism, efficiency -- the gap between the worlds we imagine and the final result



FUNDAMENTALS IN COMPUTER GRAPHICS

Shadertoy: cool stuff that the community is doing



FUNDAMENTALS IN COMPUTER GRAPHICS

Important topics in the region

- Photorealistic rendering, animation, simulation (Inria GRAPHDECO)
- Geometric modeling, physics (Inria TITANE)
- Extended Reality, user experience, accessibility (Inria BIOVISION)
- XR2C2, MAJIC

PLAN

1. Course overview
2. Fundamentals in computer graphics
3. 3D worlds and tools to create them
4. 3D modeling with Blender

3D WORLDS AND TOOLS TO CREATE THEM

Application	Graphics

Modeling and animation | **Game engines**

Media & Entertainment

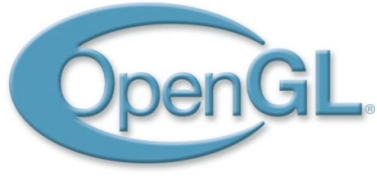
Blender

Maya

GODOT Game engine

unity

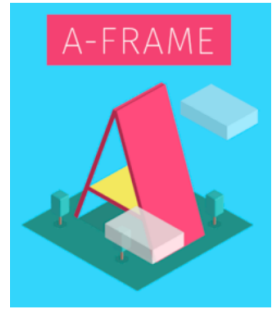
UNREAL ENGINE

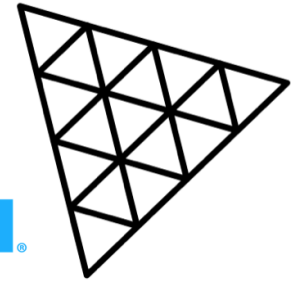


OpenGL

Vulkan

RENDERMAN





Three.js

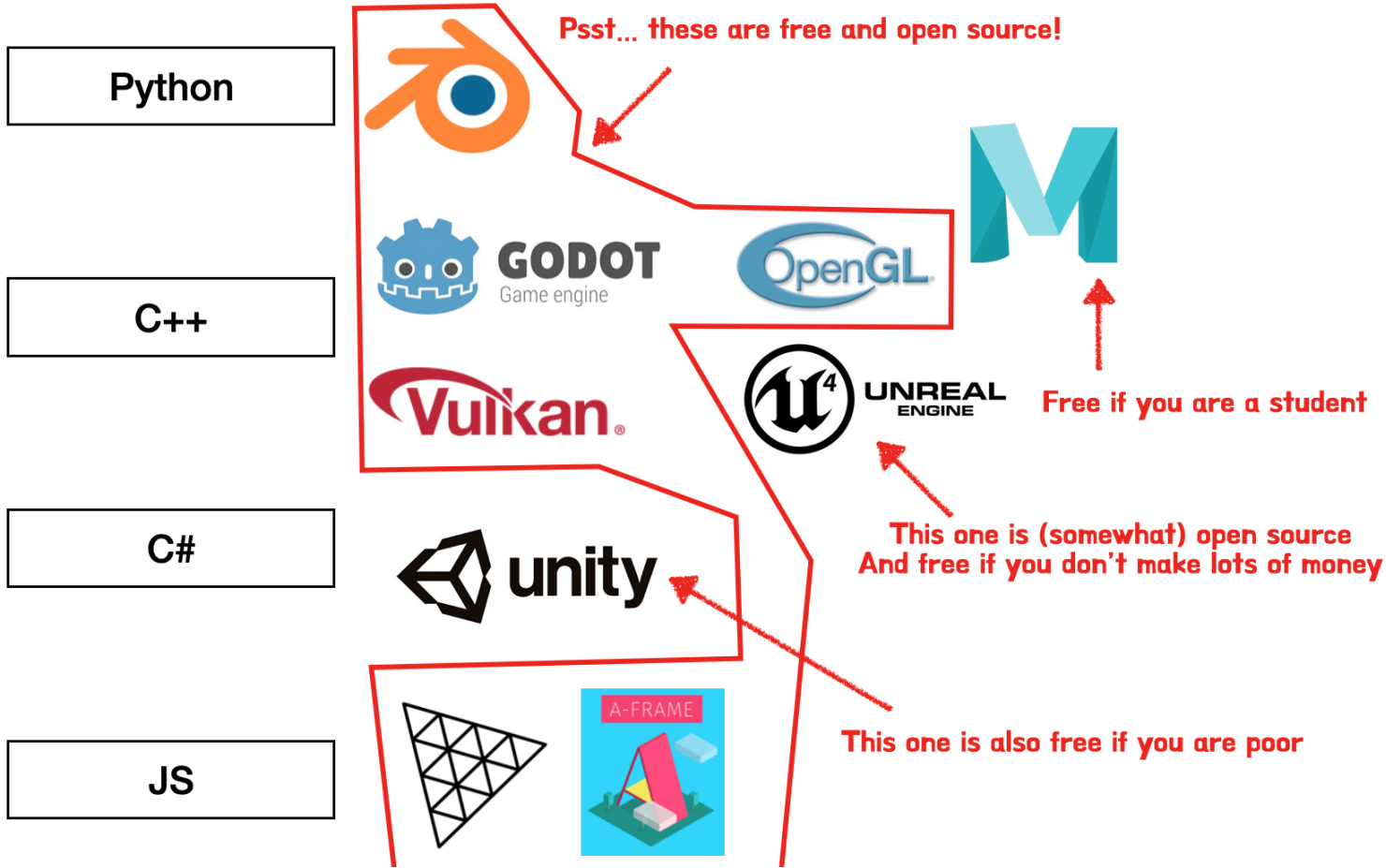
Architecture & Engineering

3DS Max

Freecad

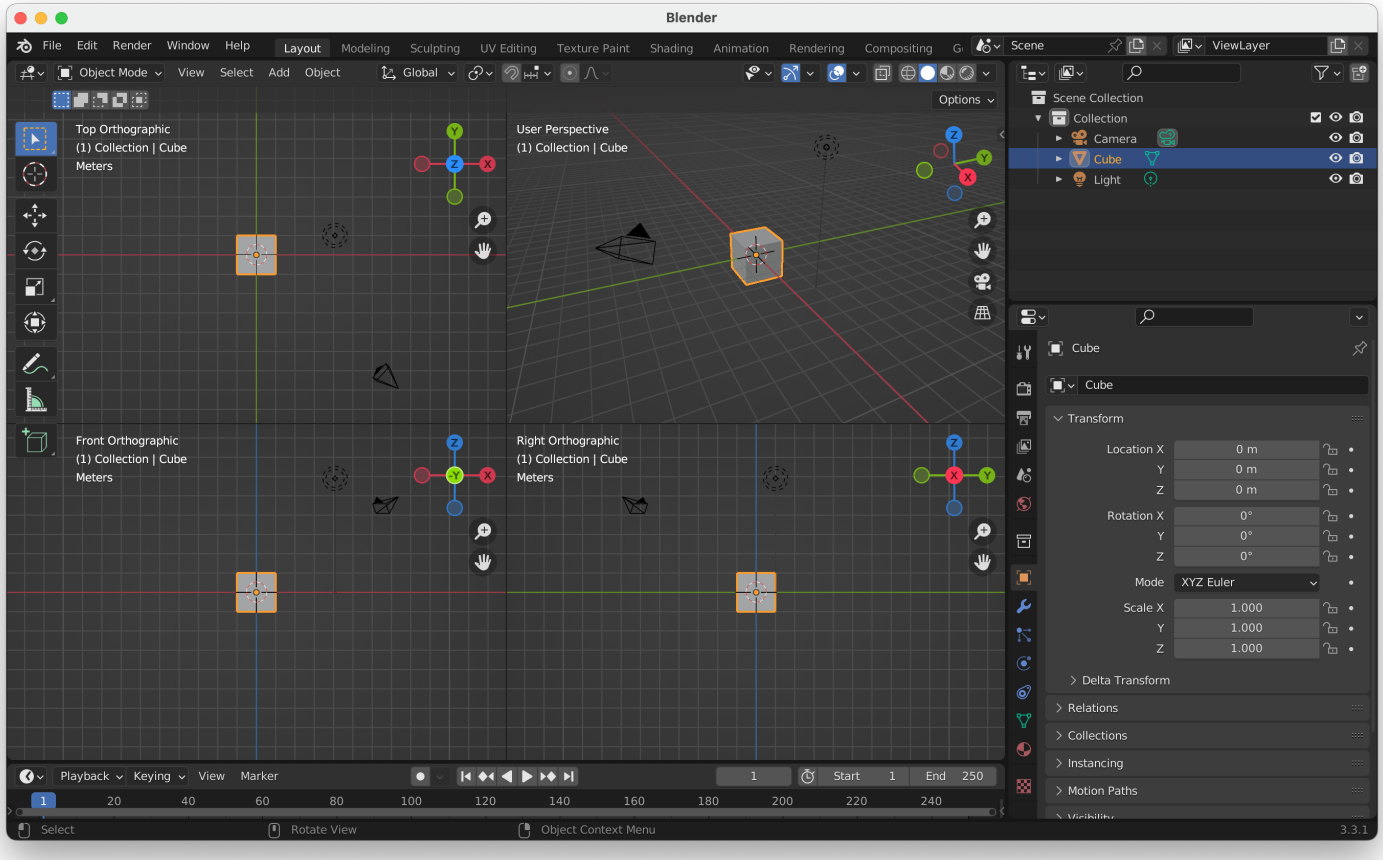
3D WORLDS AND TOOLS TO CREATE THEM

FOSS (Free Open Source Software)



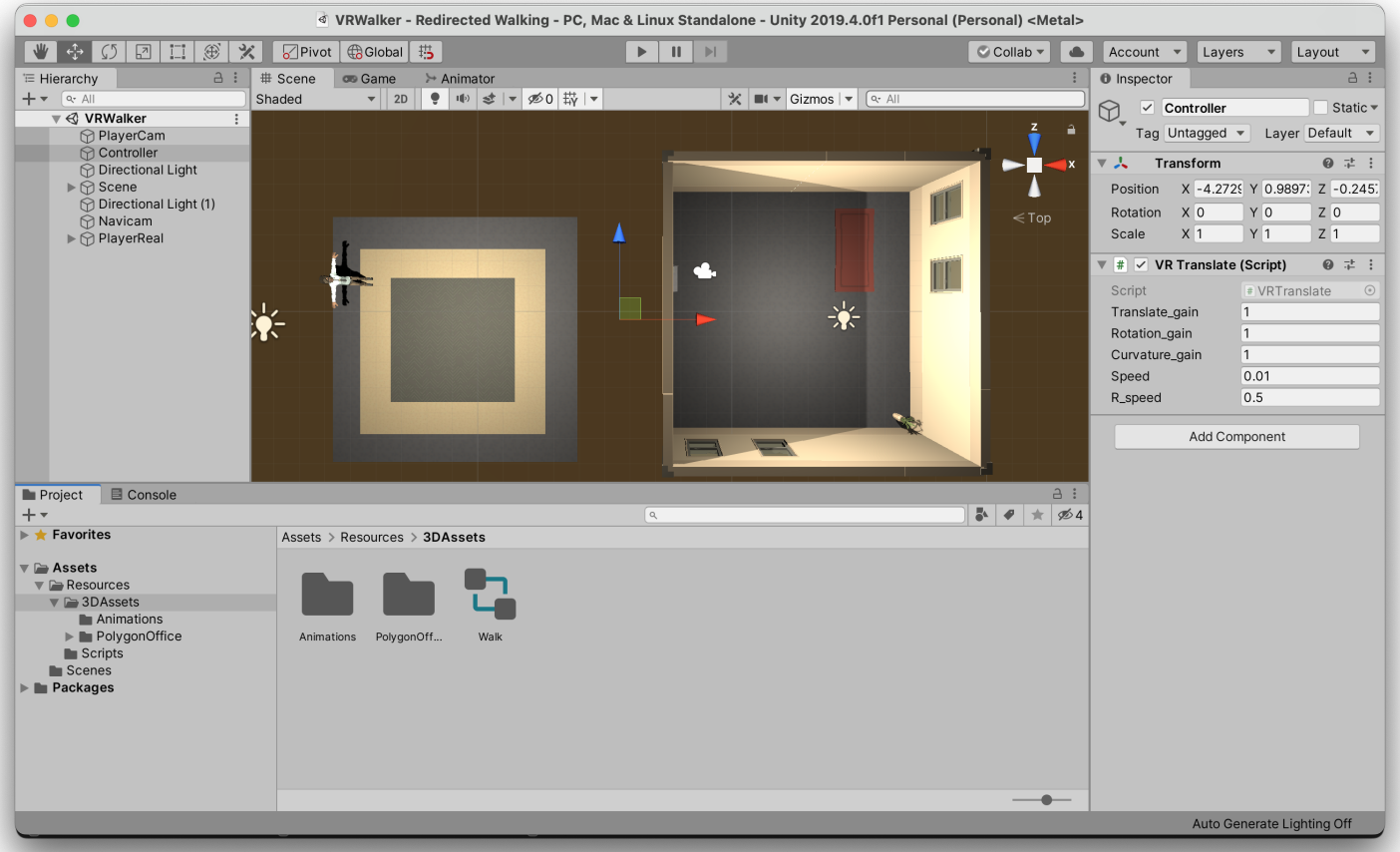
3D WORLDS AND TOOLS TO CREATE THEM

First look at 3D software



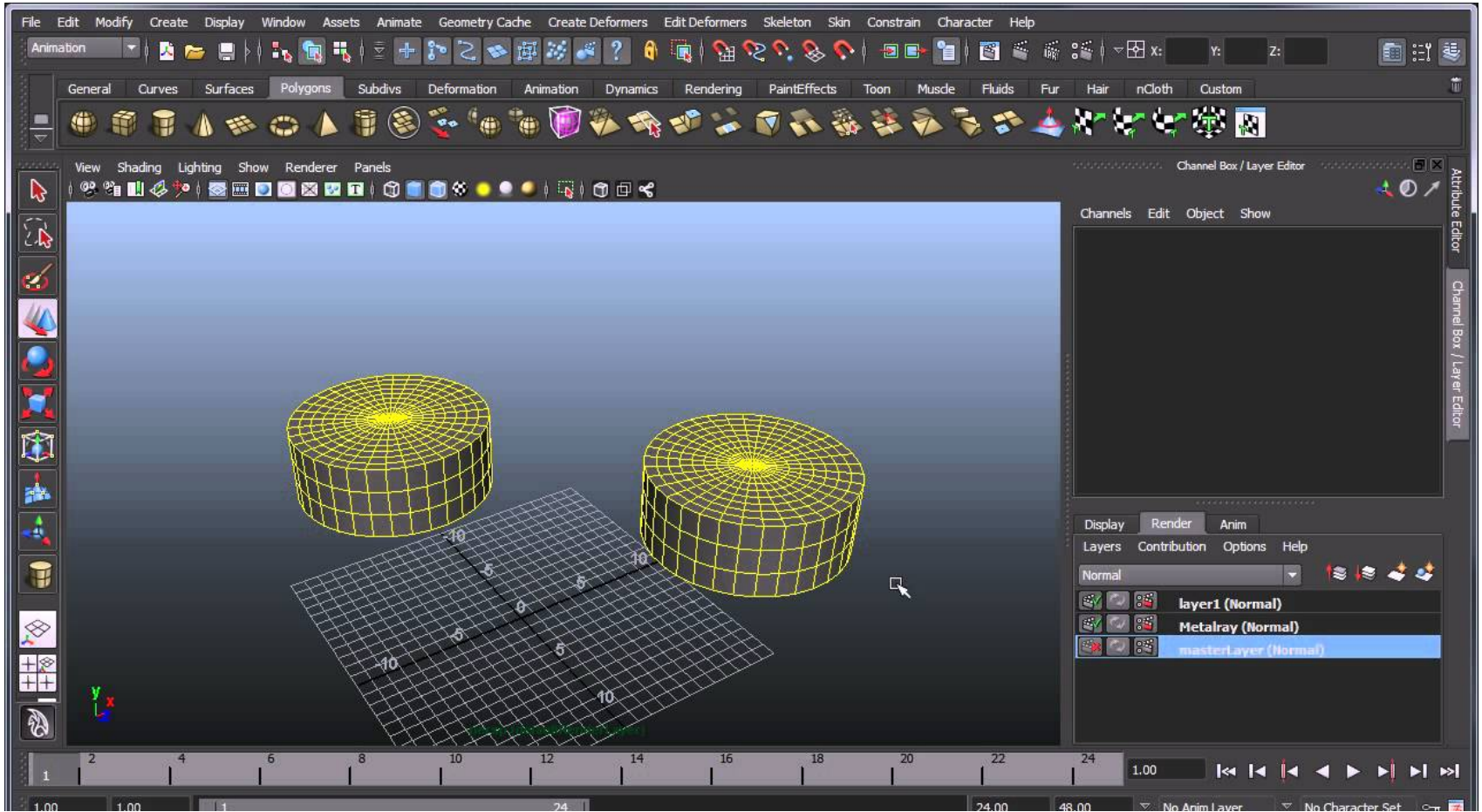
3D WORLDS AND TOOLS TO CREATE THEM

First look at 3D software



3D WORLDS AND TOOLS TO CREATE THEM

First look at 3D software



3D WORLDS AND TOOLS TO CREATE THEM

First look at 3D software. Features:

- View for 3D environment
- Lots and lots of buttons
  - Timeline
  - Play / Pause
  - Manipulations
  - ...
- Lots and lots of properties
  - Coordinates
  - Materials
  - Scripts
  - ...

3D WORLDS AND TOOLS TO CREATE THEM

Blender

- Goal: create content
- Tools: modeling, lighting, texturing, physics, rigging, animation...
- Scripting: Python
- Output: image, video, 3D models, 3D animations (secondary: interactive experiences)

3D WORLDS AND TOOLS TO CREATE THEM

Unity

- Goal: create (interactive / dynamic) applications
- Tools: asset management, event handling, i/o handling, scene and game views
- Scripting: C#, javascript
- Output: image, video, 3D models, 3D animations (secondary: interactive experiences)

3D WORLDS AND TOOLS TO CREATE THEM

Keys to getting started with 3D environments

- Understanding the purpose of software
- Knowing how to move around in the 3D environment (using keyboard and mouse)
- Familiarizing with the functionalities of the software
- Putting in place or remembering commong quick keys
- Lots and lots of trial and error

PLAN

1. Course overview
2. Fundamentals in computer graphics
3. 3D worlds and tools to create them
4. 3D modeling with Blender

TD1 -- DEMO OF BLENDER