

Autodesk Maya
modeling, animation, scripting
and C++ programming
2016-17

lionel.reveret@inria.fr

Cours ENSIMAG, Ingénierie de l'Animation 3D

Goals

- Discover a professional tool in 3D production
 - Practical implementation of theoretical concept
 - cf: “Synthèse d’images”, “Visualisation scientifique 3D »
 - Gain experience on a software that is a reference in the digital media industry
- Learn the role of programmers in 3D workflows
 - To cooperate with artists and engine programmers
 - Developing tools
 - Scripts (MEL / Python)
 - Plug-ins C++ (hot reload system)

Organization & Evaluation

- Introduction to Maya (6h)
- Development project (9h)
- Evaluation (3h)
 - Attendance
 - Results
 - + clean source code for bonus points ?

3D Programming

- Different software categories
 - Libraries
 - Low level: OpenGL, DirectX, CUDA, OpenCL
 - Higher level: Qt, OpenACC, Boost.Compute
 - Engines
 - Rendering, Animation, Physics, All-in-One..
 - Artist software
 - General purpose: 3ds Max, Maya, Blender
 - Rendering: Mental Ray, Mitsuba, Lightwave
 - Animation: MotionBuilder, Houdini, Cinema 4D,
 - Modeling: Rhinoceros, ZBrush, Mudbox

3D Programming

- Different language levels

- CPU

- APIs C/C++ dedicated to 3D or computation: OpenGL, DirectX
 - Delegation to GPU though drivers or emulation on CPU depending on the hardware.
 - Libraries, engines, software APIs

- GPGPU (GPU for Computation)

- Low level (specific language): Compute Shaders
 - Mid level (C++ & specific language): CUDA, OpenCL
 - High level (C++): Nvidia PhysX, Havok Game Dynamics

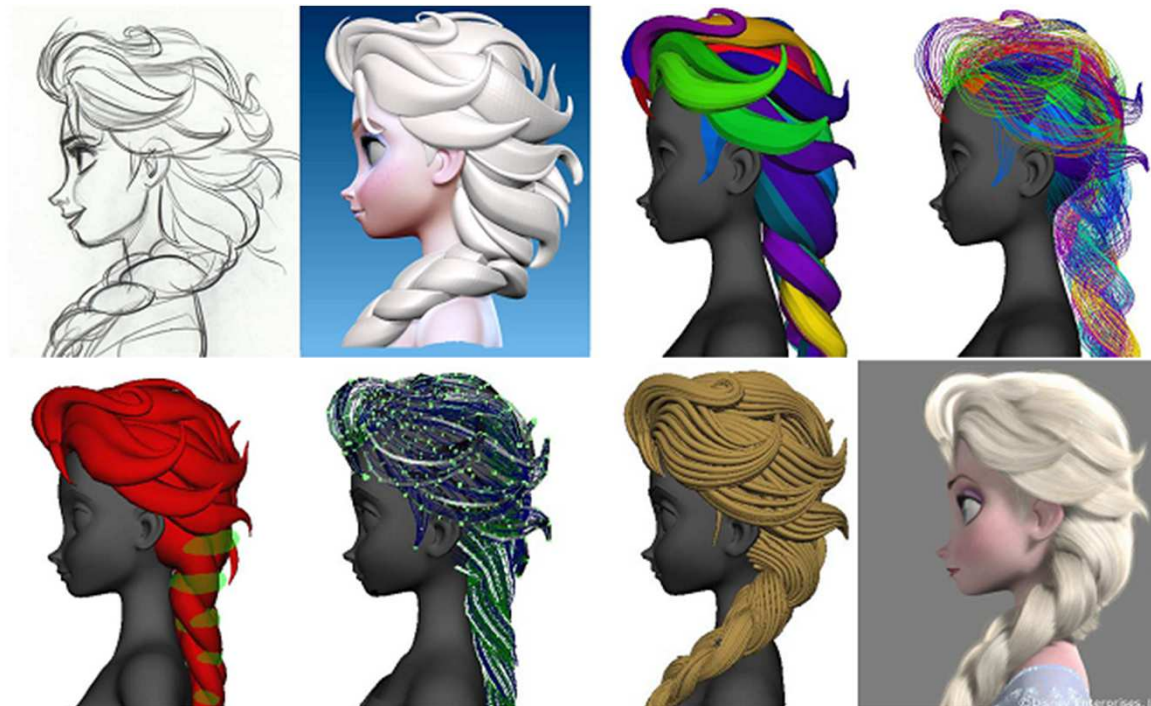
- GPU (for Graphics)

- Specific language (between C and assembly): shaders
 - Depends on target platform: NVIDIA, ATI now AMD, PlayStation, Xbox
 - ex: GLSL, HLSL, CG, PSSL

3D Programming in the Digital Media Industry

- Animation Studios

- Maya and 3ds Max are the reference
- Proprietary suites are used in the biggest studios (Disney, DreamWorks, Pixar)
 - ex: Disney's Tonic tool for hair modeling & simulation



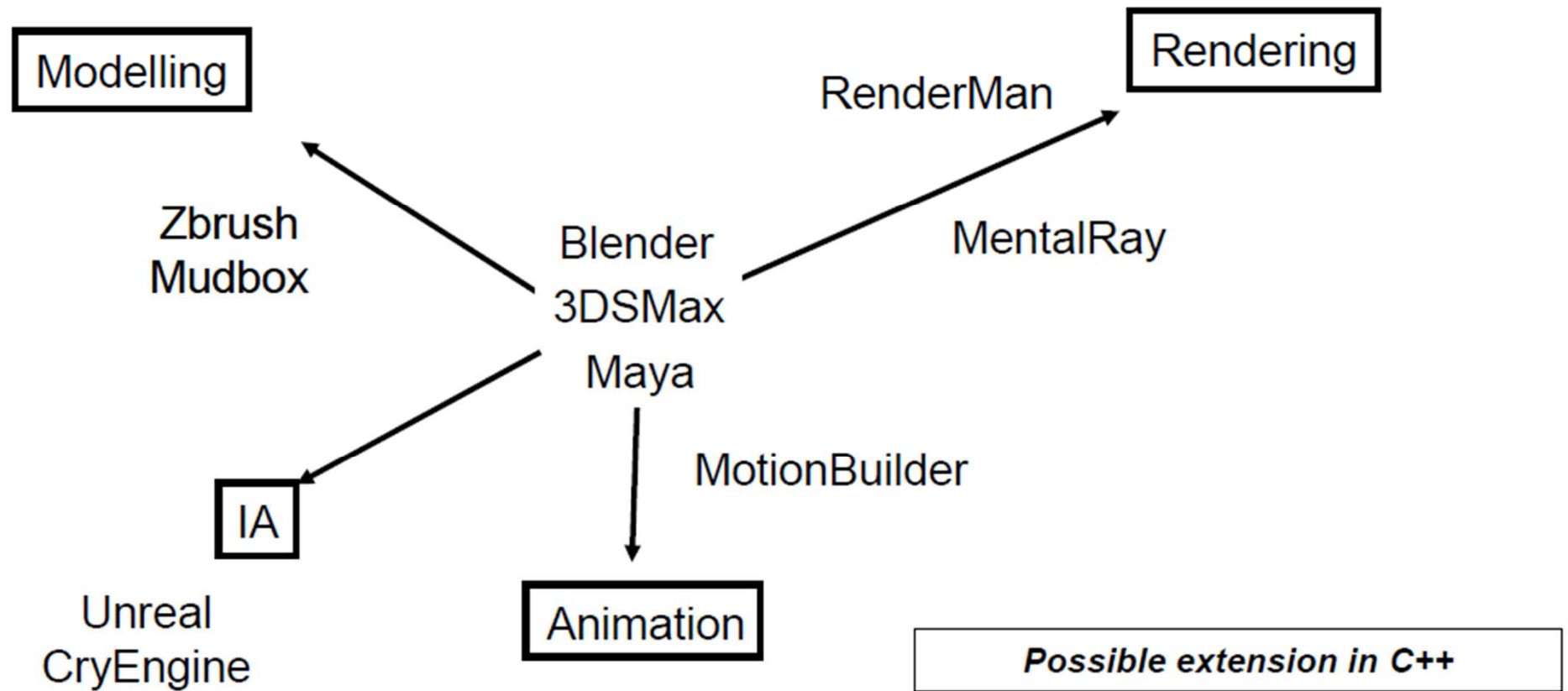
3D Programming in the Digital Media Industry

- Game Studios

- Most studios use Maya or 3ds Max
- Tools are often proprietary as well as file formats (to protect assets)
 - ex: EA's, Blizzard's and Riot's file formats
- Assets are used by the 3D engine of an end-user software
 - Implies the use of proprietary solutions to store geometries, animations, particle emitters...
 - ex: Valve provides the workflow standards of their game Dota2 to freelancers
<http://www.dota2.com/workshop/>

3D Programming in the Digital Media Industry

- Extended workflow



Autodesk Maya

- Interactive application for:
 - Modeling
 - Polygons, NURBS, Subdivision, Texture placement
 - Animation
 - keyframe, skeleton, physics
 - Rendering
 - Interactive (real-time visu.), off-line (ray-tracing)
- Open software architecture (script and C++)
- Multi-platform (formats, script and code)
- Nice Documentation support (F1)

Project summary

- Development of a tool related to animation
 - Specific deformation technique: Green Coordinates

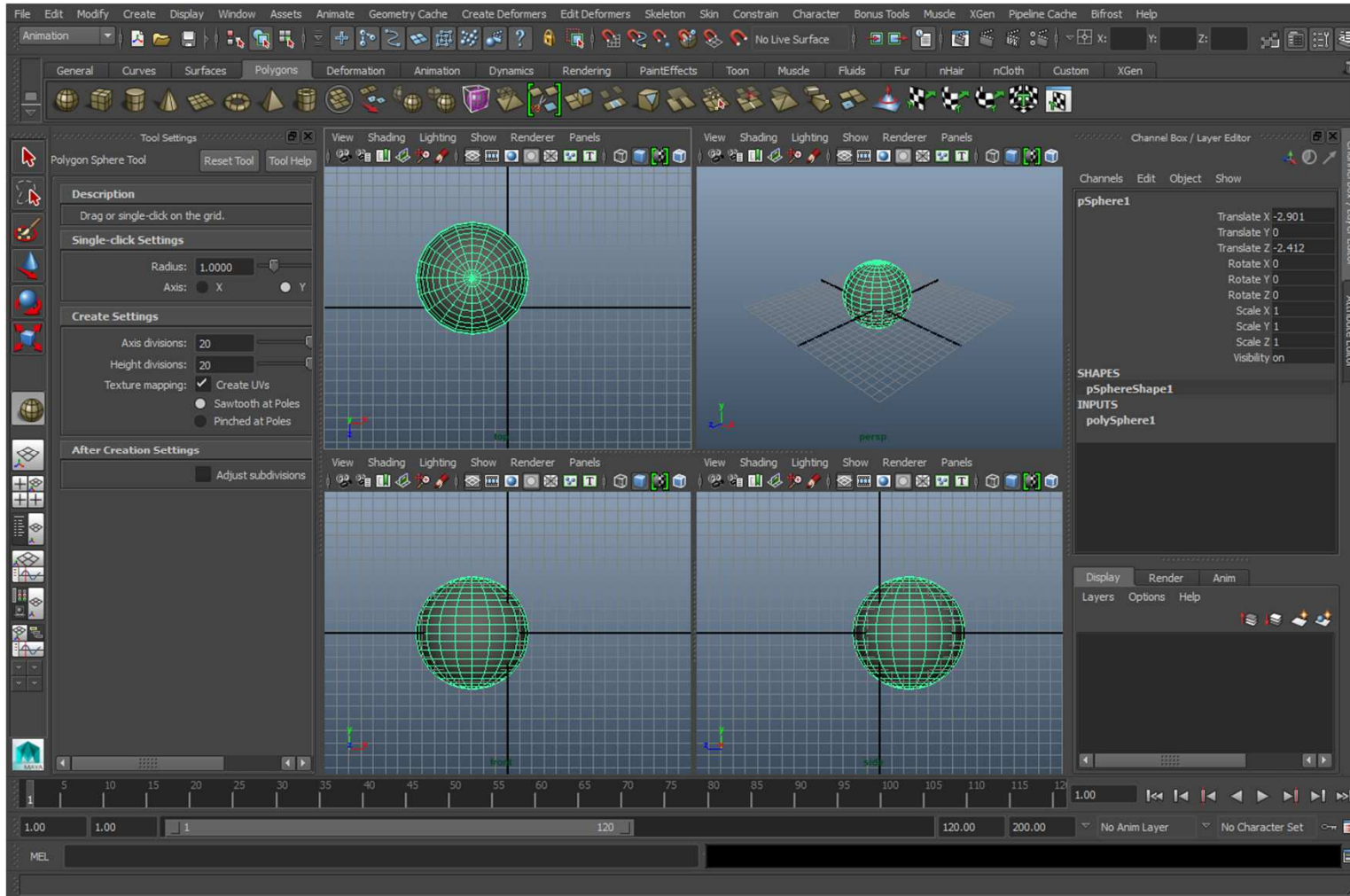
- 4 languages will be addressed
 - Maya Embedded Language (MEL) Script
 - Maya Python Script
 - Python using Maya API
 - C++ using Maya's C++ API
 - Integrating the algorithm into Maya's core mechanics

But first !

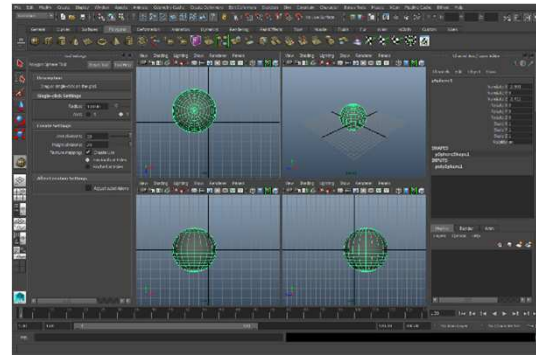
You will discover and manipulate this software a bit

(Artists have around 6 months of full-time training to use it properly,
so don't expect to be experts at the end of this course)

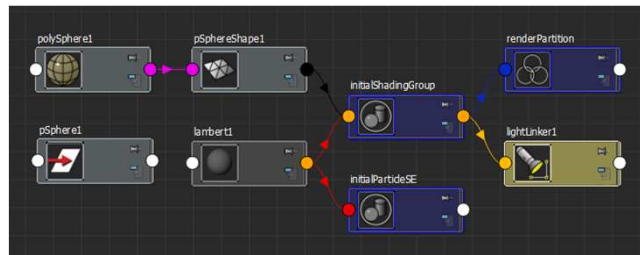
Maya Interface



Maya's software architecture



Interface



DG (Nodes & Attributes)
DAG (Scene Graph)

```
MEL Python
1 updateRendererUI;
2 CreatePolygonSphere;
3 setToolTo CreatePolySphereCtx;
4 polySphere -ch on -o on -r 4.602758 ;|
5 select -cl ;
6 select -r pSphere1 ;
```

MEL / Python

```
MStatus status;
MDagPath dag_path;
MFnIKJoint fn_joint;

// MITDag::kDepthFirst used to assure hierarchical order :)
MITDag it_dag(MITDag::kDepthFirst, MFn::kJoint, &status);
if (status != MStatus::kSuccess)
    FAILURE("SkIWriter: MITDag::MITDag()");

int num_joints = 0;
for (; !it_dag.isDone(); it_dag.next())
{
    SkIBone bone;
    num_joints++;
    it_dag.getPath(dag_path);
    data_.joints.append(dag_path);
    fn_joint.setObject(dag_path);
    MString jointName = fn_joint.name();
    strcpy_s(bone.name, SkIBone::kNameLen, jointName.asChar());
    MQuaternion rotation, axe;
    axe = fn_joint.rotateOrientation(MSpace::kTransform);
    fn_joint.getRotation(rotation, MSpace::kWorld); // since it's kWorld it
```

C++ API

MEL (Maya Embedded Language)

- create/edit objects, query/set attributes
 - geometry, animation, computation nodes
- Algorithm
 - control flow, data structures, procedures
- GUI (Maya own programming widgets)
 - input handling

Maya API C++

- Programmable nodes
 - input: geometry[], float[], time, etc
 - output: geometry[], float[], color[], etcExemples : shaders, skinning
- Custom File I/O
- Custom tools (3D HCI widgets)
 - access to OpenGL context of interactive view
- Stand-alone application
 - use Maya API but no need for Maya Interface
 - open Maya binary files

Maya programming books

- Complete Maya Programming, D. Gould, Morgan Kaufmann
 - for MEL script and C++
 - two volumes

Maya Learning tools

- A nice set of tutorials
 - See Maya Documentation “Getting Started” through Help
- Additional tutorials and content from Autodesk Creative Suite
 - Should be installed on your computer

Maya and 3D on the web

- www.autodesk.com
 - FREE version for **student**
- www.creativecrash.com
 - great source of scripts and plug-ins (mostly free), for Maya and others
- www.turbosquid.com
 - A lot of models (some free ones)