Simulation et Applications Interactives

Introduction

Caroline Larboulette

Fall 2020

Contact Information

- Email: caroline.Larboulette@univ-ubs.fr
- Office: C106 (Bâtiment ENSIBS)

Always send an email before coming to my office !

Fraude

- La copie directe d'information sans citation de sources constitue un plagiat et est par conséquent considérée comme une fraude par l'Université.
- Le partage de code ou de TP, ou la préparation en groupe d'un TP individuel est considérée comme une fraude par l'Université.

Toute fraude (plagiat, copie de TP, tricherie pendant un contrôle…) entraine un rapport suivi d'un conseil de discipline.

Fraude

Une fraude (plagiat ou triche) peut entraîner les conséquences suivantes:

- **Interdiction de se présenter à l'examen**
- **Interdiction de se présenter à tout examen universitaire pendant X années (à vie…)**
- **Interdiction d'exercer toute profession du secteur public**
- **Suppression du permis de conduire pendant 5 ans**
- **…**

Lecture Webpage

ENT

INF3001 - Simulation et Applications Interactives (SAI)

Le premier cours aura lieu jeudi 27 septembre 2018 à 10h30.

Volume horaire
CM : 20H, TP : 22H

Crédits
ECTS : 5

Objectifs
Ce cours est un cours avancé d'informatique graphique qui se focalise sur l'animation et la simulation de phénomènes 3D en temps-réel. Ces simulations sont couplées à des techniques de capture du mouvement (UE Mouvement et Interaction Gestuelle) pour créer des applications interactives contrôlées par l'humain. Nous verrons aussi comment valider / améliorer les couplages et les applications réalisées à l'aide d'études perceptuelles.

- Techniques de déformation géométriques
- Techniques de déformation basées physique (simulation): masses-ressorts, particules, fluides
- Déformation de systèmes complexes et multi-couches: personnages (facial, cheveux, muscles), végétation

Application Domains

Movies / Commercials

[Evian Baby Bay 2016]

- Realism or Cartoon exaggerations
- As fast as possible but far from real-time (time is money !)
- High quality

Application Domains

Special Effects (FX)

Gollum
[The Lord of the Rings 2001]

- Integrating virtual objects into real footage
  - Lighting / shadows
  - Interaction with real actors
  - Realism

Application Domains

Special Effects (FX)

- Simulation of explosions
  - Realism (realism /= reality)
  - Example of DMM: Avatar [2009], Star Wars: The Force Unleashed [2008]

Application Domains

Videos Games

[World of Warcraft 2006]

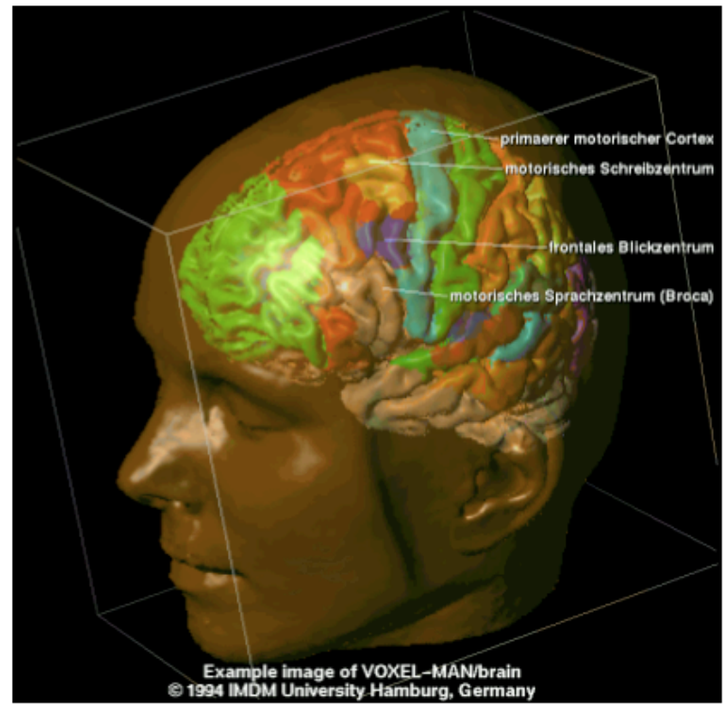
[Star Wars: The Force Unleashed 2008]

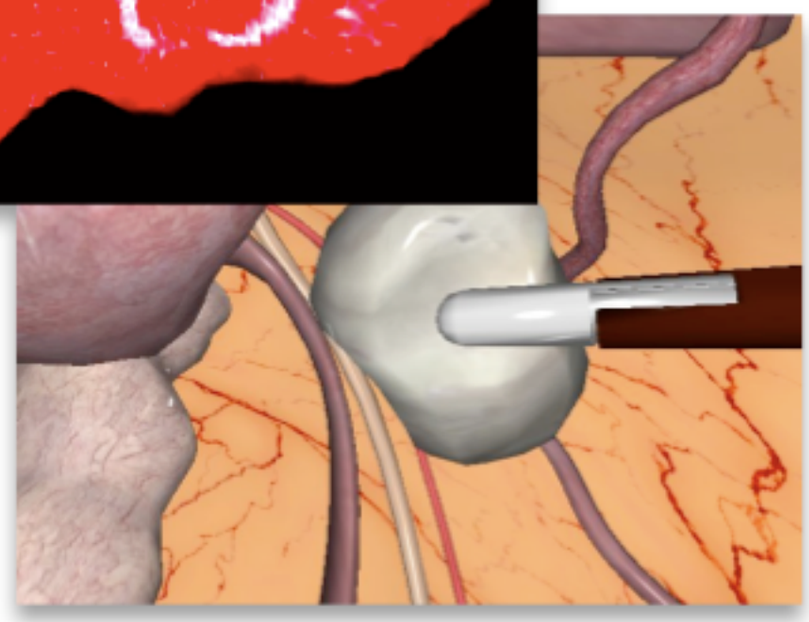
- Real-Time for one or a few characters
- Realism as much as possible (but not a priority)
- Intensive use of GPU (skinning and shaders)

Application Domains

Medicine

- Surgery simulator
  - Accurate deformation of surfaces
- Surgery planning
- 3D visualisation
  - Useful rendering

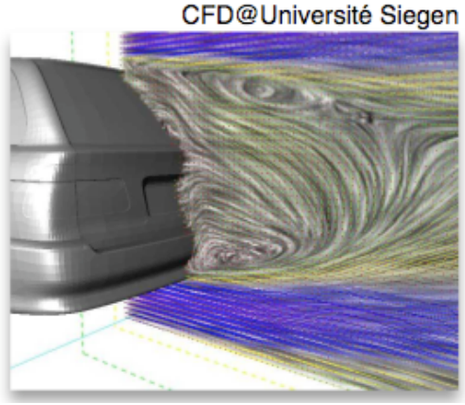




Application Domains

Scientific Visualization

Visible human



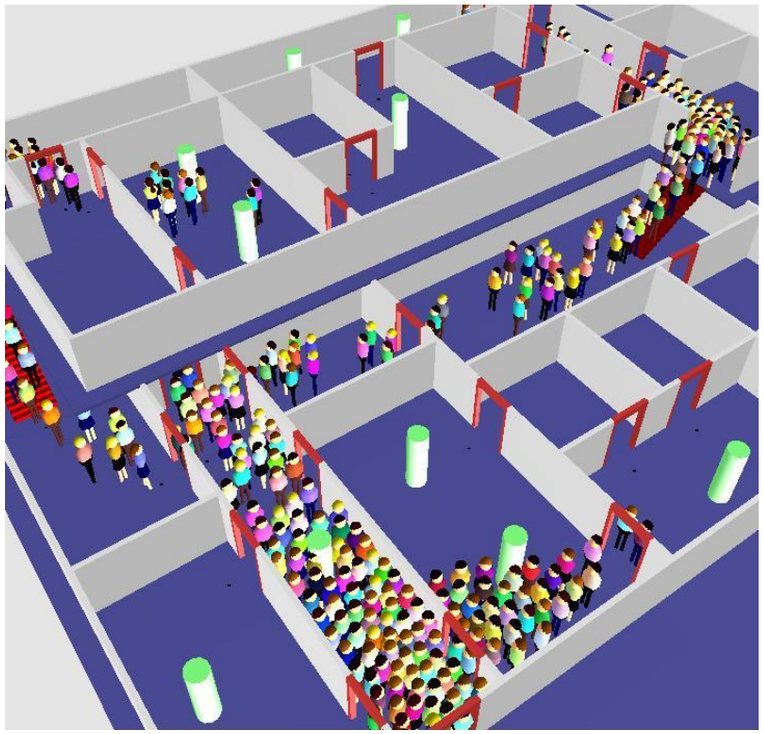
- Visualize complex data
- Structured or unstructured
- Show hidden information (modeling is important)
- Labels
- Meaningful rendering

Application Domains
Computer Aided Design (CAD)

- Impact Assessment
  - Simulation of natural phenomena
- Various domains
  - Architecture
  - Mechanics
  - Electrical Engineering
  - Car / plane / boat design

Application Domains

Virtual Environments



- Crowd (VEs need to be populated) : AI for behaviour
- Real-time for a lot of characters
- Realism definitely not a priority

Application Domains

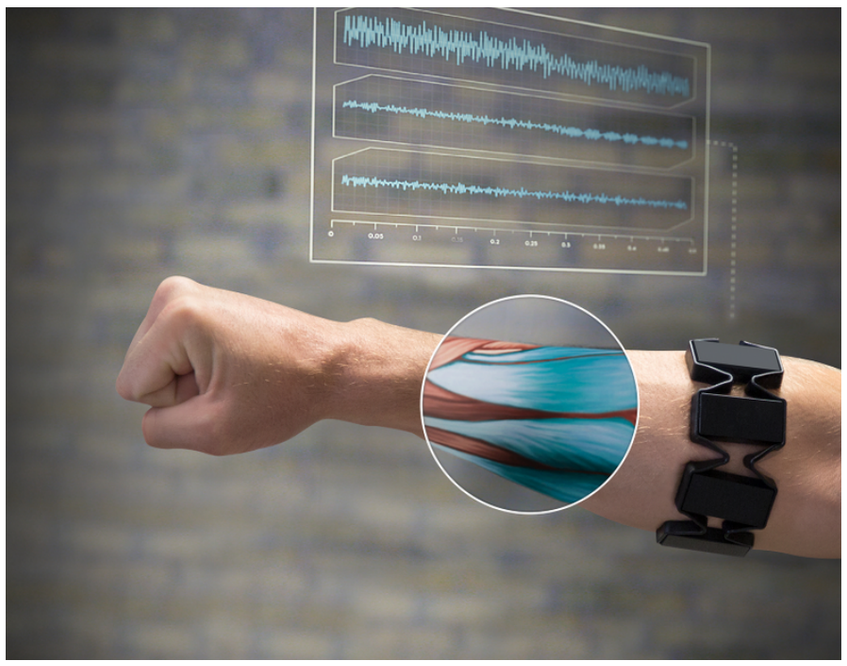
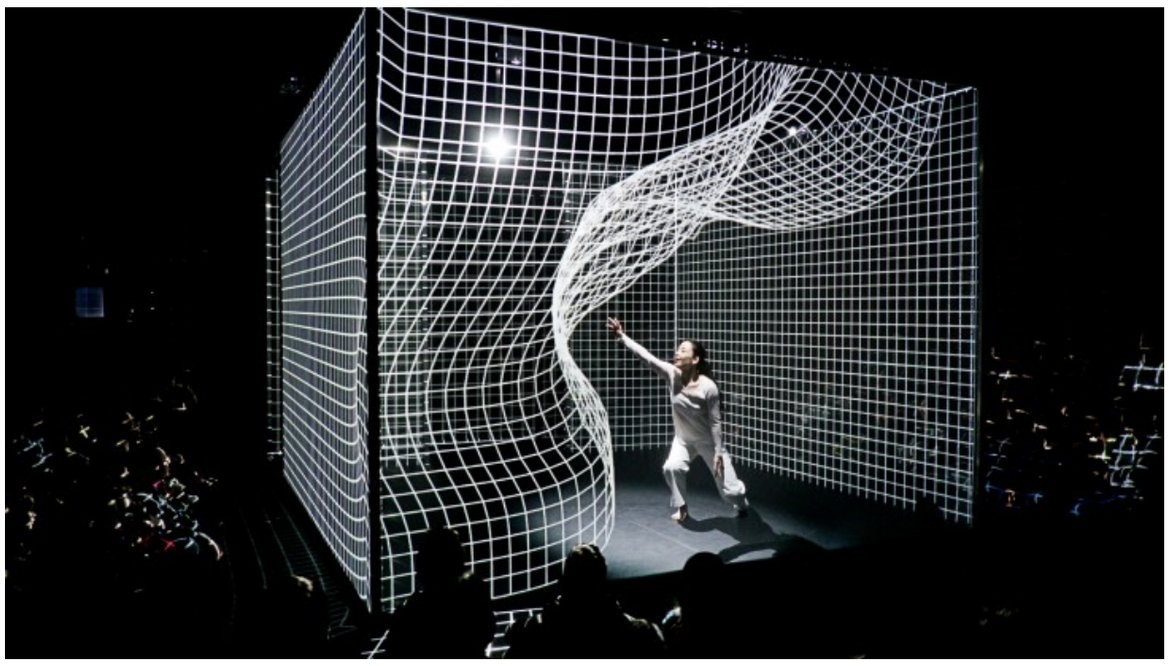
Digital Art

- Representation of emotions
- Movement and colors

Application Domains

Multi-Media Systems

- Systems that involve more than one combination medium: graphics, text, sound
  - Interactive applications
  - Animation / rendering software (Autodesk Maya)

Application Domains

Interactive Applications

- Data-based
- Guided by human
  - Movement (full-body, gesture)
  - Brain (EEG technologies)
- Real-time applications

Interactive Computer Graphics

- Control of content, structure and appearance of objects via rapid visual feedback
- Components
  - Input / Capture (mouse, multi-touch, body…)
  - Processing (movement + rendering + data transmission, analysis and storage)
  - Output / Display (screen, VR/AR systems…)

Modern Graphics

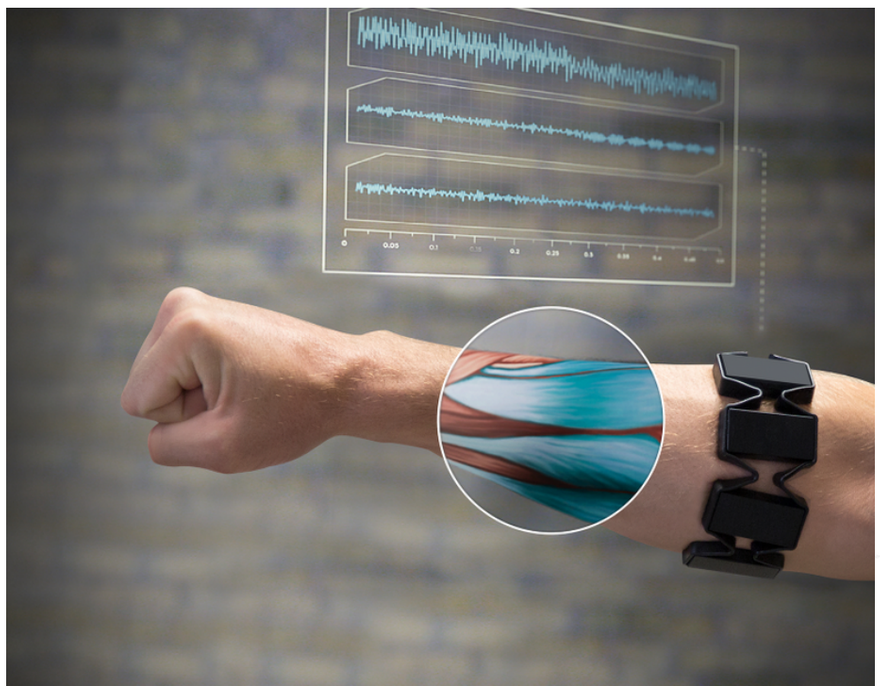
Input / Capturing Devices

- Mouse -> tablets (multi-finger interactions)
- Haptic devices, game control (Wii)
- Touchless sensors, human control
- Kinect
- Leap Motion
- Motion Capture (Qualisys)

Modern Graphics

Input / Capturing Devices

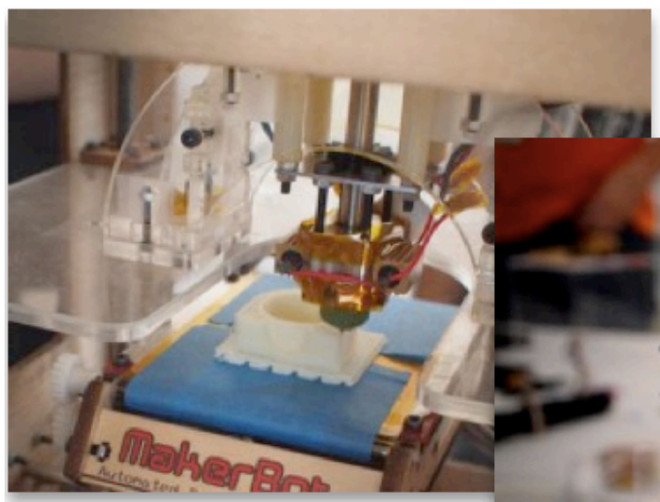
- Cameras (phones, webcams)
- EEG Technologies
  - Brain wear (EMOTIV Epoc +)
  - Myo armband
- Sensors (accelerometer, gyroscope, temperature…)
- Data Glove
- Eyetracker

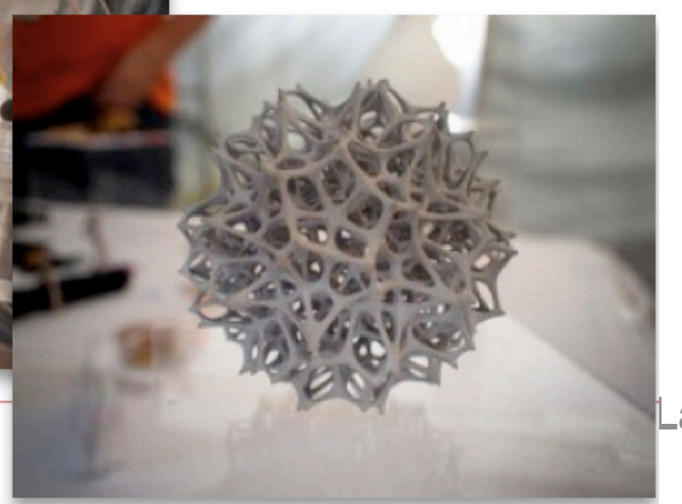


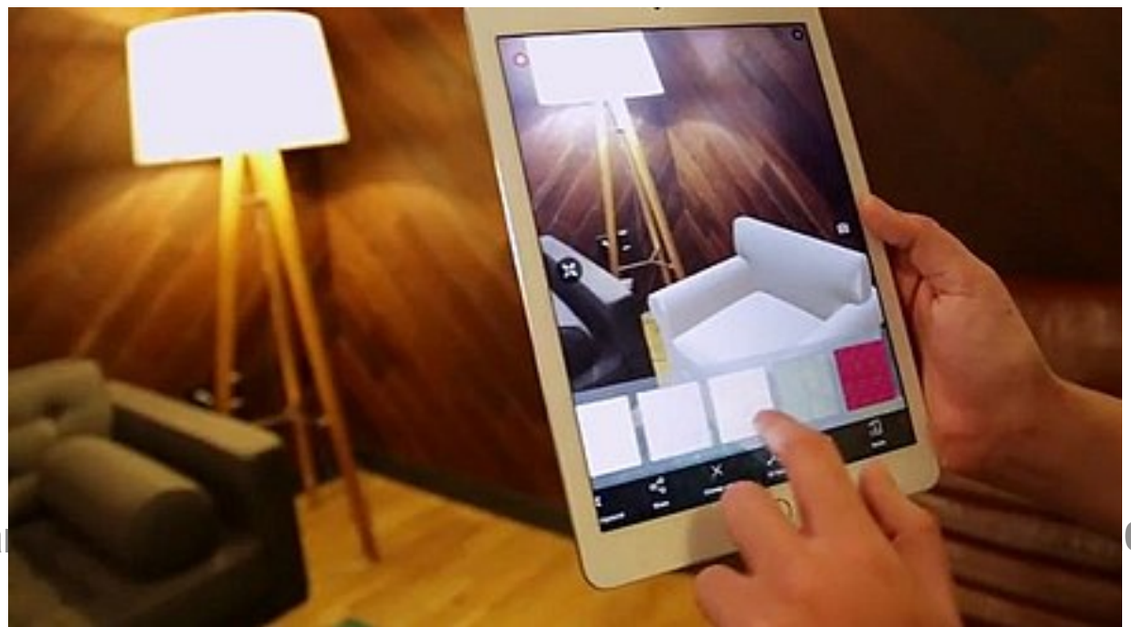
Modern Graphics

Output / Display Devices

- Polarized glasses (3D on TV)
- 3D printers
- Virtual Reality gear (HTC Vive)
- Augmented Reality gear (HoloLens)
- Quadcopters (drones)







Modern Graphics

Output / Display Devices

- New type of Screens
  - Smartphones, tablets
  - Large touchscreens, touch tables
  - Stereoscopic screens
  - Immersive rooms
  - Caves

Modern Graphics

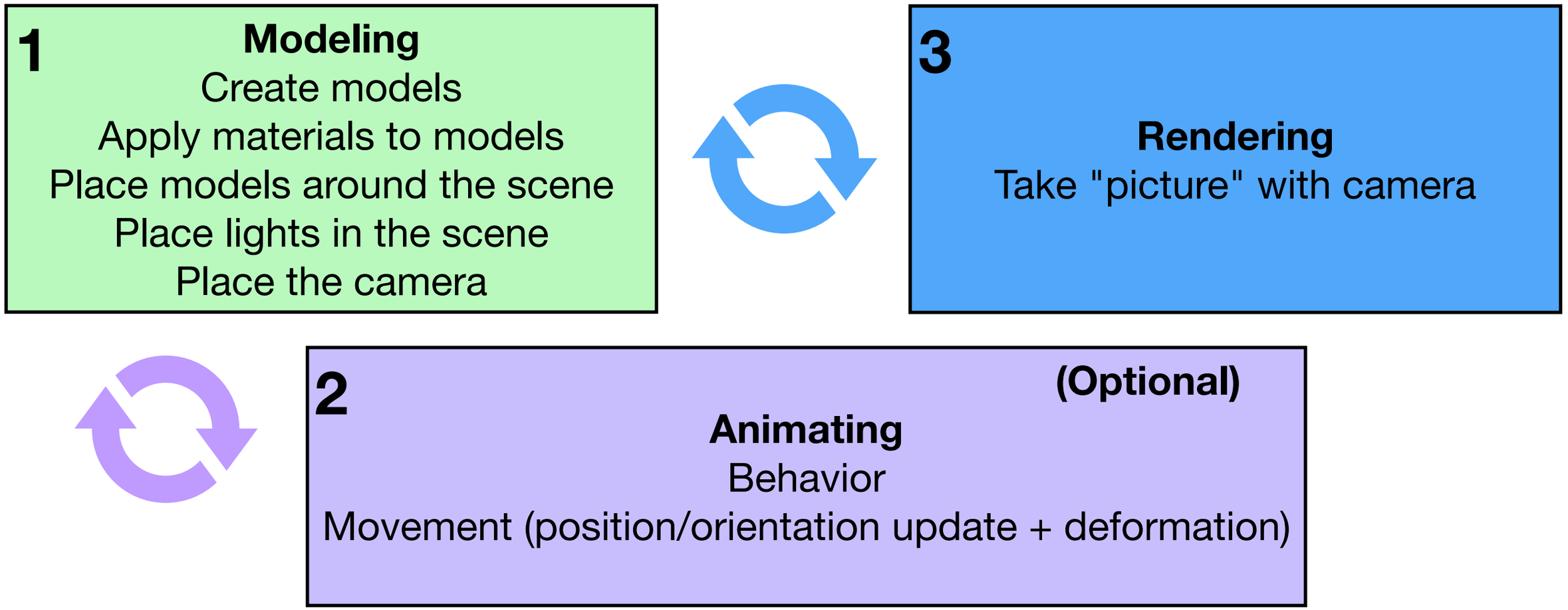
New Software

- Algorithms and Data Structures
  - Physics based modeling / simulation techniques
  - Big Data: optimized data structures
- Parallel Computing
  - Some instructions can be computed independently
    - Animation: nodes of mass-spring systems
    - Rendering: per pixel or per vertex operations (GLSL)
- Distributed Computing
  - Make use of several computers / cores for a computation
    - Render Farms (24 000 cores for pixar movie Monsters University 2013)
    - SETI program

Processing

Animation Pipeline

- 3 steps: modeling, animating, rendering



Commercial software: Autodesk Maya™ , Autodesk 3D Studio Max™, Blender™

Processing

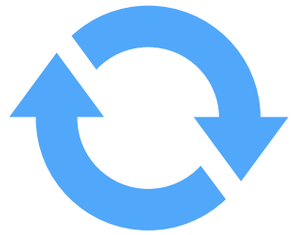
Animation Pipeline

- 3 steps: modeling, animating, rendering

1

Modeling

Create models
Apply materials to models
Place models around the scene
Place lights in the scene
Place the camera

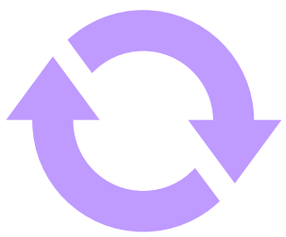


3

Rendering

Take "picture" with camera

Sending data to display devices



2 **(Optional)**

Animating

Behavior
Movement (position/orientation update + deformation)

Using data from capture devices

Processing

Animation and Rendering Loops

- `Initialize{}`
  - **Reads models designed offline or creates procedural models**
- `Animate{}`
  - **Called just before `Draw{}` for procedural techniques**
  - **Called at the frequency of the timestep for physically based techniques**
  - **Even faster when using haptics**
- `Draw{}`
  - **Called at least at 24 fps (frames per second)**

Note that `Animate{}` and `Draw{}` are loops that do not need to be synchronous

Code Example

Particles Simulation

```
class Particle
{
public :
  Particle();

  void init();
  void draw();
  void animate();

private :
  qglviewer::Vec speed_, pos_;
  int age_, ageMax_;
};


class Viewer : public QGLViewer
{
protected :
  virtual void draw();
  virtual void init();
  virtual void animate();
  virtual QString helpString() const;

private:
  int nbPart_;
  Particle* particle_;
};
```


Code Example

Particles Simulation

```
void Viewer::init()
{
  restoreStateFromFile();
  glDisable(GL_LIGHTING);
  nbPart_ = 2000;
  particle_ = new Particle[nbPart_];
  glPointSize(3.0);
  setGridIsDrawn();
  help();
  startAnimation();
}

void Viewer::draw()
{
  glBegin(GL_POINTS);
  for (int i=0; i<nbPart_; i++)
    particle_[i].draw();
  glEnd();
}

void Viewer::animate()
{
  for (int i=0; i<nbPart_; i++)
    particle_[i].animate();
}
```


Code Example

Particles Simulation - `init{}`

```
void Particle::init()
{
  pos_ = Vec(0.0, 0.0, 0.0);
  float angle = 2.0 * M_PI * rand() / RAND_MAX;
  float norm  = 0.04 * rand() / RAND_MAX;
  speed_ = Vec(norm*cos(angle), norm*sin(angle), rand() / static_cast<float>(RAND_MAX) );
  age_ = 0;
  ageMax_ = 50 + static_cast<int>(100.0 * rand() / RAND_MAX);
}
```


Code Example

Particles Simulation - `animate{}`

```
void Particle::animate()
{
   speed_.z -= 0.05f;
   pos_ += 0.1f * speed_;

   if (pos_.z < 0.0)
      {
         speed_.z = -0.8*speed_.z;
         pos_.z = 0.0;
      }

   if (++age_ == ageMax_)
      init();
}
```


Code Example

Particles Simulation - `draw{}`

```
void Particle::draw()
{
  glColor3f(age_/(float)ageMax_, age_/(float)ageMax_, 1.0);
  glVertex3fv(pos_);
}
```


Course Outline

1. Modeling and Geometric Deformation Techniques
2. Physically Based Deformation Techniques
3. Deformation of Complex Systems
4. Interactive Applications
5. Perceptually Based Graphics

Seminars

- SIGGRAPH & IEEEVR (see Moodle)
- Choose a paper
- Presentation during 8 minutes +- 10%
  - First presentation in 2 weeks
  - Main idea
  - Be able to explain to students in your class what the paper is about

Semester Project

- The project takes place during the entire semester. The project grade accounts for 50% of the final grade.
- The project can be mutualised with the project for UE Movement et Intelligence Artificielle (type C)

Semester Project

Content

3 types of projects

- A: graphics-animation project (UE 3001)
- B: capture / analysis / IA project (UE 3002)
- C: single project that includes aspects of project A and B, two independent grades, one for each UE

Semester Project

Groups

- 18 students:
  - groups of 3 to 5 students
- Make group according to common work interests ! (not personal affinity)

Semester Projects

Type A

- Graphics oriented project
  - VR
  - AR
  - Screen
- Do not use capture or very simple capture systems (e.g., the controller of the HTC Vive)

Semester Projects

A.1 Smoke VR

- Transform Eulerian smoke code into a .dll and load it into the Unity engine
- Use Laban descriptors computed on mocap to control simulation parameters
- Add real-time 3D interaction

Semester Projects

A.2 Poseidon

- Continue the M1 project…
- Add control
- Move it to VR

Semester Projects

A.3 Virtual Zoo VR

- Graphics
  - Add animated grass
  - Add animated water
  - ...
- Make in place 3D interaction metaphors
  - To navigate in the zoo
  - To activate the description of the animals
- Any other idea welcome…

Semester Projects

Type B

- Capture oriented project
- Capture data and analyse the data
  - Machine learning techniques
  - Classifiers
- Gesture recognition (IA)
- More details with S. Gibet

Semester Projects

Type C

- Create a data-based interactive application that integrates both aspects of projects type A and B
- Capture
- Graphics

Semester Project
C.1 Teddy VR

- Leap Motion
- HTC Vive
- Modeling in immersion

Semester Project

- Each group needs to define either
  - Two projects, one of type A and one of type B
  - One integrated project (type C)

Semester Project

INF3001

- Deliverables
  - A project proposal for the 16th of Oct
  - A report for the 4th of Jan (10%)
  - A demo video for the 4th of Jan (5%)
  - A 10 min presentation, week 4-8 Jan (10%)
- The quality of the project will be evaluated on the 25% remaining

Semester Project(s)

Project Proposal

- 4 pages maximum
- Describes the context, the idea in details, the repartition of the work among the team
- Due October 16 (but can be returned before)

Semester Project

- Goal: create a data-based interactive application or 2 distinct applications
- Capturing Devices
  - Motion Capture, Qualisys system
  - Microsoft Kinect (2+1)
  - Leap Motion (2)
  - Myo Armband (2)
  - EMOTIV Epoc + (1)
  - Data Gloves
  - Pupil Eyetracker (1)

Semester Project

Output Devices

- Computer screen / Projector
- HTC Vive (RV gear, immersive rendering)
- Hololens (RA gear)
- Quadcopter Crazyflie 2.0
- iPad