Computer Graphics & User Interfaces
Lecture 1

Dr. Marc Eduard Frîncu
West University of Timisoara

Feb 21st 2012
Introduction

What is Computer Graphics (CG)?

- Technically: production, manipulation & display of images
- Practically: art, design, movies, games, training, advertising, communication, ...

What will we do?
- Understand how CG systems:
  - Are constructed
  - Use mathematics, physics, algorithms and data structures

What will we NOT do?
- Use CG systems to produce these images
What is Computer Graphics (CG)?

- **method** of dealing with all aspects of creating images and animations (i.e., sequences of images) using a computer:
  - technically: production, manipulation & display of images
  - practically: art, design, movies, games, training, advertising, communication, ...
What is Computer Graphics (CG)?

- **method** of dealing with all aspects of creating images and animations (i.e., sequences of images) using a computer:
  - technically: production, manipulation & display of images
  - practically: art, design, movies, games, training, advertising, communication, ... 
- a **technology** oriented towards the descriptive and accurate communication of information
Introduction

What is Computer Graphics (CG)?

- **method** of dealing with all aspects of creating images and animations (i.e., sequences of images) using a computer:
  - **technically**: production, manipulation & display of images
  - **practically**: art, design, movies, games, training, advertising, communication, ...

- a **technology** oriented towards the descriptive and accurate communication of information

**What will we do?** understand how CG systems:

- are constructed
- use mathematics, physics, algorithms and data structures
Introduction

What is Computer Graphics (CG)?

- **method** of dealing with all aspects of creating images and animations (i.e., sequences of images) using a computer:
  - **technically**: production, manipulation & display of images
  - **practically**: art, design, movies, games, training, advertising, communication, ...  
- a **technology** oriented towards the descriptive and accurate communication of information

What will we do? understand how CG systems:
- are constructed
- use mathematics, physics, algorithms and data structures

What will we **NOT** do? use CG systems to produce these images
CG is not just rendering (image reproduction) but incorporates

- graphic design – arts
CG is not just **rendering** (image reproduction) but incorporates

- **graphic design** – *arts*
- **ray tracing and rendering** – *geometry, physics*
Introduction

CG is not just rendering (image reproduction) but incorporates

- graphic design – arts
- ray tracing and rendering – geometry, physics
- mathematical models of places and entities within them – mathematics
CG is not just **rendering** (image reproduction) but incorporates

- **graphic design** – *arts*
- **ray tracing** and **rendering** – *geometry, physics*
- **mathematical models of places and entities** within them – *mathematics*
- **graphics hardware, libraries, and window systems APIs** – *computer science*
Introduction

CG is not just rendering (image reproduction) but incorporates

- graphic design – arts
- ray tracing and rendering – geometry, physics
- mathematical models of places and entities within them – mathematics
- graphics hardware, libraries, and window systems APIs – computer science
- user interfaces and graphical input – computer science & engineering
Introduction

CG is not just rendering (image reproduction) but incorporates

- graphic design – arts
- ray tracing and rendering – geometry, physics
- mathematical models of places and entities within them – mathematics
- graphics hardware, libraries, and window systems APIs – computer science
- user interfaces and graphical input – computer science & engineering

CG includes
Introduction

CG is not just rendering (image reproduction) but incorporates

- graphic design – arts
- ray tracing and rendering – geometry, physics
- mathematical models of places and entities within them – mathematics
- graphics hardware, libraries, and window systems APIs – computer science
- user interfaces and graphical input – computer science & engineering

CG includes

1. hardware, e.g., graphics cards for modeling & rendering
Introduction

CG is not just rendering (image reproduction) but incorporates:

- **graphic design** – *arts*
- **ray tracing** and **rendering** – *geometry, physics*
- **mathematical models of places and entities** within them – *mathematics*
- **graphics hardware, libraries, and window systems APIs** – *computer science*
- **user interfaces and graphical input** – *computer science & engineering*

CG includes:

1. **hardware**, e.g., graphics cards for modeling & rendering
2. **software**, e.g., 3D Studio Max, Maya – for modeling & rendering
Introduction

CG is not just **rendering** (image reproduction) but incorporates

- **graphic design** – *arts*
- **ray tracing** and **rendering** – *geometry, physics*
- **mathematical models of places and entities** within them – *mathematics*
- **graphics hardware, libraries, and window systems APIs** – *computer science*
- **user interfaces** and **graphical input** – *computer science & engineering*

CG includes

1. **hardware**, e.g., graphics cards for modeling & rendering
2. **software**, e.g., 3D Studio Max, Maya – for modeling & rendering
3. **applications**, e.g., display of a solar system inside a planetarium dome
Introduction

Theoretical instruments

- **computer science**: programming languages, software engineering, ...
Introduction

Theoretical instruments

- **computer science**: programming languages, software engineering, . . .
- **mathematics**: analytical, projective & descriptive geometry, algebra, calculus, coding algorithms, . . .
Introduction

Theoretical instruments

- **computer science**: programming languages, software engineering, . . .
- **mathematics**: analytical, projective & descriptive geometry, algebra, calculus, coding algorithms, . . .
- **physics**: optics, colorimetry, mechanics & thermodynamics (for simulations), . . .
Introduction

Theoretical instruments

- **computer science**: programming languages, software engineering, . . .
- **mathematics**: analytical, projective & descriptive geometry, algebra, calculus, coding algorithms, . . .
- **physics**: optics, colorimetry, mechanics & thermodynamics (for simulations), . . .
- **electronics**: signal generation & analysis, device technologies, . . .
Introduction

Theoretical instruments

- **computer science**: programming languages, software engineering, . . .
- **mathematics**: analytical, projective & descriptive geometry, algebra, calculus, coding algorithms, . . .
- **physics**: optics, colorimetry, mechanics & thermodynamics (for simulations), . . .
- **electronics**: signal generation & analysis, device technologies, . . .
- **human psychology**: human perception of 3D images, virtual reality, . . .
Introduction

Theoretical instruments

- **computer science**: programming languages, software engineering, . . .
- **mathematics**: analytical, projective & descriptive geometry, algebra, calculus, coding algorithms, . . .
- **physics**: optics, colorimetry, mechanics & thermodynamics (for simulations), . . .
- **electronics**: signal generation & analysis, device technologies, . . .
- **human psychology**: human perception of 3D images, virtual reality, . . .
- . . .
Introduction

CG applications

- science:
- engineering:
- education, learning & research:
- medicine:
- economy & management:
- commerce & publicity:
- recreation & entertainment:
- art:
- ...
Introduction

CG applications

- **science**: simulations, 2D/3D visualization
- **engineering**: CAD
- **education, learning & research**: presentations, demos
- **medicine**: 3D/4D visualization
- **economy & management**: simulations, 2D/3D visualization
- **commerce & publicity**: commercial ads
- **recreation & entertainment**: animations, movies
- **art**: 2D/3D visualization
- ...
Basic graphics system

– set of **equipment** ⊕ **programs** specialized in visual representation of the information

1. systems for image synthesis
2. systems for image processing & analysis

**Figure:** Angel: Interactive Computer Graphics 4E ©Addison-Wesley 2005
When & where did it all began?

Non-interactive graphics since the early stone age (30,000 B.C.)

Figure: Aurachs on a cave painting in Lascaux, France.
http://en.wikipedia.org
Introduction

History of CG

Syllabus

Coursework

When & where did it all began?

Non-interactive graphics since the early stone age (30,000 B.C.)

Figure: Aurachs on a cave painting in Lascaux, France.
http://en.wikipedia.org

1963: Ivan Sutherland was one of the people to pioneer the field (interactive graphics for a system called “Sketchpad” – Ph.D. thesis MIT)

Figure: Ivan Sutherland demonstrating Sketchpad. http://en.wikipedia.org
## History of CG

### 1950–1960

- CG goes back to the earliest days of computing: strip charts, **light pens**, pen plotters, calligraphic CRT, **vector displays**, CRT refresh rate costs too high (slow, expensive, unreliable computers)
1950–1960

- CG goes back to the earliest days of computing: strip charts, light pens, pen plotters, calligraphic CRT, vector displays,
- CRT refresh rate costs too high (slow, expensive, unreliable computers)

1960–1970

- **wireframe graphics** (draw only lines NOT surfaces)
- **Sketchpad**: demonstrated the potential of human-machine interaction
- **display processors**: computer $\oplus$ display file $\oplus$ refresh display processor (DPU)
- **storage tube**: (Tektronix) standard interface to computers, inexpensive, no constant refresh (e.g., Plot3D in Fortran)
- CAD & CAM ('64)
- photorealism at Univ. of Utah
- flight simulator (commercial) ('69)
- 3D vector pipeline, matrix multiplier, clipping ('69)

Figure: Wireframe representation of an object. Angel: Interactive Computer Graphics 4E ©Addison-Wesley 2005
History of CG

1970–1980

- Xerox PARC: bitmapped raster graphics, “windows”
- CG at NYIT (“Catmull”)
- ACM SIGGRAPH (Foley, Van Dam, Feiner) ('77)
- first standards: GKS – 2D ISO, attempt to 3D ISO (Core '77)
History of CG

1970–1980

- Xerox PARC: bitmapped raster graphics, “windows”
- CG at NYIT (“Catmull”)
- ACM SIGGRAPH (Foley, Van Dam, Feiner) (’77)
- first standards: GKS – 2D ISO, attempt to 3D ISO (Core ’77)

1980–1990

- realism arrives: smooth shading, environment mapping, bump mapping
- ray tracing (’82), fractals (’83)
- Clarke: geometry engine (’82)
- IRIS: Integrated Raster Imaging System
- Special purpose hardware: Silicon Graphics Geometry Engine, VRAM
- PHIGS, RenderMan, X Window System, Human-Computer Interface (HCI)
- TRON, Star Trek II – genesis effect
- Computer Generated Interfaces (CGI) still cannot replace models in movies & series (e.g., Star Trek TNG)

Figure: Angel: Interactive Computer Graphics 4E ©Addison-Wesley 2005
History of CG

1990–2000

- OpenGL ('93)
- QuickDraw 3D ('95)
- Direct3D ('95)
- completely computer-generated movies (e.g., Toy Story '95)
- new hardware capabilities: texture mapping, blending, accumulation & stencil buffers
- CG starts to widely replace models in movies & series (e.g., Babylon 5, Star Trek Voy)

Figure: Reflection showing the use of texture mapping & stencil buffer
http://oreilly.org
# History of CG

## 1990–2000

- **OpenGL** ('93)
- **QuickDraw 3D** ('95)
- **Direct3D** ('95)
- completely computer-generated movies (e.g., Toy Story '95)
- **new hardware capabilities**: texture mapping, blending, accumulation & stencil buffers
- CG starts to widely replace models in movies & series (e.g., Babylon 5, Star Trek Voy)

![Figure: Reflection showing the use of texture mapping & stencil buffer](http://oreilly.org)

## 2000–present

- **photorealism**
- **graphics card** dominion (e.g., Nvidia, ATI, 3DLabs, . . .)
- **game boxes** (e.g., XBox, Play Station)
- **CG routine** in movies & series (e.g., Maya, Lightwave)
- **programmable pipelines**

![Figure: Example of photo-realist CGI.](http://http://originaltrilogy.com)
Syllabus

1. Graphics devices
2. Image processing
3. Modeling & projections
4. Clipping & visibility
5. Lighting, shadowing, coloring & texturing
6. Animations
7. Simulation reality
Graphics Devices

- graphics systems: raster-based displays
- rendering pipeline
- graphics devices for visualization & user interactivity

**Figure:** Rendering pipeline
Image processing

- signal processing
- anti aliasing
- image processing operations

1st assignment: implement several image operations

Figure: Anti aliasing Xbox 360 vs. PS3
http://www.eurogamer.net/articles/diglfoundry-saboteur-aa-blog-entry
Modeling & projections

- recap of elementary notions of algebra and geometry
- reference systems
- modeling: basic scene transformations (rotation, translation, scaling)
- drawing basic shapes (primitives, curves, surfaces, 3D objects)
- solid & procedural models
- projection types

2nd assignment: implement a simple library for the basic scene transformations
3rd assignment: implement a simple library for drawing basic shapes
4th assignment: implement a simple library for using various types of projections. Apply them on several shapes

Figure: Various reference systems and spaces http://eraser85.wordpress.com/
Clipping & visibility

- clipping algorithms
- visibility test algorithms

5th assignment: implement a simple library for using various types of clipping algorithms. Apply them on several shapes

6th assignment: implement a simple library for using various types visibility algorithms. Apply them on several shapes

Figure: Sutherland-Hodgman clipping algorithm steps
Lighting, coloring & texturing

- light types
- ray tracing
- colors
- textures and texture mapping

**7th assignment:** implement a simple library for creating and applying one or more light types. Apply it on several 3D objects

**8th assignment:** implement a simple library for applying a texture on a 2D surface. Apply it on a 2D object

*Figure:* Various lighting types applied on a 3D object

http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter05.html
Animations

- 2D/3D animation
- key frames
- articulated figures
- kinematics & dynamics

Figure: Animation keyframes http://unkn0wnorigin.deviantart.com/art/Animation-keyframes-189968037
Simulating reality

- scene reconstruction
- physics

9th assignment implement a 3D scene by using Photosynth (http://photosynth.net/)

**Figure:** Scene reconstruction using Photosynth
Materials

for the course:

- books:
  - (RO) D. Petcu, L. Cucu, *Grafica pe calculator*, Tipografia Universitatii de Vest, 1999
  - ...

- URLs:
  - ...

for the lab:

- ...

Coursework

- written exam (30%)
  - 6 questions 1p each. No materials allowed

- course assignments (20%)
  - at least 5 assignments from the ones given during lecture time. Deadline: 2 weeks from their assignation

- lab assignments (50%)
  - 2 grades based on lab exercises. Students are picked randomly

Cheating is not tolerated! Whoever is caught automatically fails the exam
For passing the exam: you need to have at least 5/10p at the lab assignments & 5/10p at the final grade