



COMPUTER GRAPHICS
CS-314: Fall 2004
Instructor: Alla Sheffer



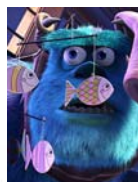



<http://www.ugrad.cs.ubc.ca/~cs314>



What is Computer Graphics ?

- Generation of static/dynamic (realistic) images on computer



Fake or Photo ?

<http://www.alias.com/eng/etc/fakeorfoto/quiz.html>

1



Fake or Photo ?

2



University of British Columbia

Fake or Photo ?

3



University of British Columbia

Fake or Photo ?



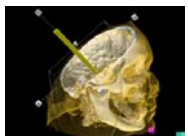
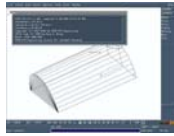

4



University of British Columbia

What is CG used for?




- GUI
 - Modeling systems
 - Applications
- Simulations & Visualization



University of British Columbia

What is CG used for?

- Movies
 - Animation
 - Special Effects



University of British Columbia

What is CG used for?





- Computer Games



University of British Columbia

What is CG used for?



- Images
 - Design
 - Advertising
 - Art



University of British Columbia

What is CG used for?


- Virtual reality




University of British Columbia

What This Course Is About

- Basic **algorithms** for
 - Modeling – generating models
 - Rendering – displaying models
 - (Animation – generating motion)




University of British Columbia




What This Course is **NOT** About

- NOT covered:
 - Artistic and design issues
 - Usage of commercial software packages




University of British Columbia




Other graphics courses

- CPSC 424: Geometric Modeling
- CPSC 426: Computer Animation

- CPSC 514: Image-based Modeling and Rendering
- CPSC 526: Computer Animation
- CPSC 533A: Digital Geometry
- CPSC 533B: Animation Physics
- CPSC 533C: Information Visualization




University of British Columbia



Basic Rendering – 2D



- Raster display – discrete grid of elements
- Terminology
 - **Pixel:** basic element on device
 - **Dynamic Range:** ratio between min (not zero!) & max light intensities of pixel
 - **Resolution:** number of rows & columns in device
 - Measured in
 - Absolute values (1K x 1K)
 - Density values (300 dots per inch)
 - **Screen Space:** Discrete 2D Cartesian coordinate system of the screen pixels



University of British Columbia


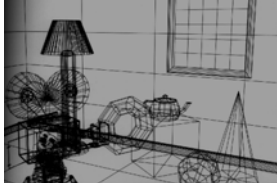
Basic Rendering – 2D

- Algorithms for:
 - Scan Conversion
 - Draw (lines)
 - Anti-Aliasing
 - Fill (polygons)
 - Clipping
 - Color





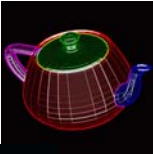

3D Graphics Components

- Geometric Modeling
 - polygons,
 - smooth surfaces,
 - etc...




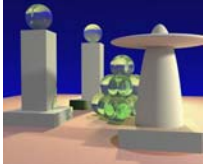

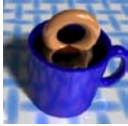
3D Graphics Components

- Material Properties
 - Opacity/Transparency
 - Texture
 - Reflectance



3D Graphics Components


- Light
 - Shading
 - Light absorbance/Reflection
 - Shadows



University of British Columbia

Rendering - 3D


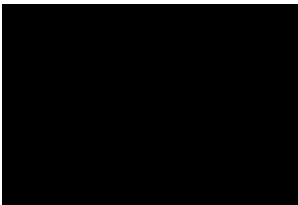
- Geometric transformations
 - Project 3D geometry to image plane
- Hidden surface removal
 - What is visible?
- Scan conversion (similar to 2D)
- Color
 - lighting, shading, texture mapping



University of British Columbia

Animation


- Generating motion
 - Frame/State interpolation
 - Physical simulation
 - Motion capture



University of British Columbia

Lecture Syllabus


- Introduction
- Math Review (week 2)
- Transformations (week 2/3)
- Rasterization (week 4)
- Hidden Surface Removal (week 5)
- Geometric Modeling
 - Solid & Free-form (week 6)
 - Meshes & Subdivision (week 7)
- Lighting Models (week 8)
- Color Theory (week 9)
- Methods for improving realism (week 10)
 - Shadows
 - Texture
- Image Improvement (week 11)
 - Aliasing and Anti-aliasing
 - Image manipulation
- Ray Tracing (week 12)



University of British Columbia

Course Organization


- Programming assignments:
 - C++, Windows or Linux
 - OpenGL graphics library / GLUT for user interface
- Collaboration:
 - Individual solutions unless stated otherwise
 - See plagiarism policy
<http://www.ugrad.cs.ubc.ca/~cs314/Vsep2004/plag.html>



University of British Columbia



Course Organization

- Up-to-date information:
 - <http://www.ugrad.cs.ubc.ca/~cs314>
 - WebCT (follow link from course home page)
 - Bulletin board
 - REQUIRES INTERCHANGE ACCOUNT!

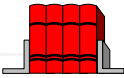




University of British Columbia


Teaching Staff

- Instructor: Alla Sheffer
 - Office hrs: CICSR 217, Tue 4-5pm
 - Contact info:
 - sheffa@cs.ubc.ca
 - do not use for
 - assignment related questions
 - anything else which might be relevant to other students
 - TAs: Dan Julius, TBD, Wei Li
 - Contact info:
 - TBD



Literature



- Fundamentals of Computer Graphics
 - Peter Shirley, A.K. Peters
- OpenGL Programming Guide
 - J. Neider, T. Davis and W. Mason, Addison-Wesley



Grading

- **Assignments (programming): 40%**
 - 3D Transformations (13%)
 - Rendering pipeline (13%)
 - OpenGL Extravaganza (14%)
- **Two Quizzes: (30%)**
 - 15% each
- **Final Exam: (30%)**