## CPSC 414
Computer Graphics

**Instructor:**
Michiel van de Panne

---

### What This Course Is About

**Topics covered**

- basics of interactive rendering
  - *modeling, geometric transformations*
  - *projections, hidden surface removal*
  - *lighting, texture mapping*
  - *input and output hardware, human perception*
- as time allows
  - *shadows, current hardware*
  - *ray-tracing, global illumination, animation*
  - ...
What This Course is NOT About

Topics NOT covered:

- Artistic and design issues
- Usage of commercial software packages
- Other graphics courses
  - CPSC 424: Geometric Modeling
  - CPSC 448: Video Game Programming
  - CPSC 526: Animation
  - CPSC 514: Image-based Modeling and Rendering
  - CPSC 533: Topics in Information Visualization

Course Organization

Programming assignments:

- C++, Windows or Linux
- OpenGL graphics library / GLUT for user interface

Collaboration:

- Individual solutions unless stated otherwise
Course Organization

Up-to-date information:
- [http://www.ugrad.cs.ubc.ca/~cs414](http://www.ugrad.cs.ubc.ca/~cs414)
- WebCT (follow link from course home page)
  - Bulletin board
  - REQUIRES INTERCHANGE ACCOUNT!

3D Graphics: Applications

- simulation
  - flight, driving, surgical, ...
- arts and entertainment
  - film and video
  - games
  - simulator rides
- design
  - product, building, mechanical
- scientific visualization
  - weather, MRI, fluid flow, ...
- information visualization
  - network structure, data mining, ...
3D Graphics: History

• 2000 BC: Orthographic projection
• 1600s:
  – coordinate systems (Descartes)
  – optics (Huygens)
  – calculus, physics, optics (Newton)
• 1897: oscilloscope (Braun)
• 1950-70: vector display computers
• 1966: first raster display
• 1993: 500k tri/s, texmap @ 60Hz for $150,000
• 1995: feature length CG films
• 2002: 100M tri/s for $400

Images...
Images...

Images...
Images...

Images...
Graphics: State of the Art

• Displays:
  – IBM: 2560x2048 pixels, 3km wiring, 200ppi
  – electronic paper
  – 3D printers

• Input:
  – Z-cam, Triclops
  – motion capture

• Modelling: TEDDY

• Rendering: Video Textures

• Animation: Ski Stunt Simulator

• Animation: Fracture Simulation

Projective Rendering Pipeline

- project vertices
- scan convert
- eye
- image plane
Projective Rendering Pipeline

- **OCS** - object coordinate system
- **WCS** - world coordinate system
- **VCS** - viewing coordinate system
- **CCS** - clipping coordinate system
- **NDCS** - normalized device coordinate system
- **DCS** - device coordinate system

```
void glVertex3f(float x, float y, float z)
void glTranslatef(float x, float y, float z)
void glRotatef(float angle, float x, float y, float z)
void glFrustum(...)
void glutInitWindowSize(int x, int y)
```

Coming Up…

**Thursday, January 9:**

- math review:
  - points, vectors, coordinate frames
  - basis vectors, basis functions
  - dot product, cross product
- introduction to geometric transformations