CS 543: Computer Graphics
Midterm review

Emmanuel Agu

Exam Overview

- Wednesday, Oct. 17, in-class
- Will cover up to today's lecture (Illumination and Shading)
- Can bring:
  - One page cheat-sheet, hand-written (not typed)
  - Calculator
- Will test:
  - Theoretical concepts
  - Mathematics
  - Algorithms
  - Programming
  - OpenGL knowledge (program structure and some commands)

What am I Really Testing?

- Understanding of
  - concepts (NOT only programming)
  - programming (pseudocode/syntax)
- Test that:
  - you can plug in numbers by hand to check your programs
  - you did the projects
  - you understand what you did in projects

General Advise

- **Read your projects** and refresh memory of what you did
- **Read the slides**: worst case – if you understand slides, you're more than 50% prepared
- Focus on **Mathematical results, concepts, algorithms**
- Plug numbers: calculate by hand
- Should be able to **predict subtle changes** to algorithm.. What ifs?..
- **Past exams**: Some on website, I taught fall 2005. This exam will resemble mostly fall 2005
- Every lecture has references. Look at refs to focus reading
- Do all readings I asked you to do on your own

Grading Policy

- I do **ALL** grading myself
- Give you all the points, take away only what I have to
- In time constraints, laying out outline of solution gets you healthy chunk of points
- Try to write something for each question
- Many questions will be easy, exponentially harder to score higher in exam

Introduction

- Motivation for CG
- Uses of CG (simulation, image processing, movies, viz, etc)
- Elements of CG (polylines, raster images, filled regions, etc)
- Device dependent graphics libraries (OpenGL, DirectX, etc)

OpenGL/GLUT

- High-level:
  - What is OpenGL?
  - What is GLUT?
  - Functionality, how do they work together?
- Design features: low-level API, event-driven, portability, etc
- Sequential Vs. Event-driven programming
- OpenGL/GLUT program structure (create window, init, callback registration, etc)
- GLUT callback functions (registration and response to events)

OpenGL Drawing

- glBegin( ), glEnd( ), glVertex( )
- OpenGL :
  - Drawing primitives: GL_POINTS, GL_LINES, etc (should be conversant with the behaviors of major primitives)
  - Command format
  - Data types
  - Interaction: keyboard, mouse (GLUT_LEFT_BUTTON, etc)
  - OpenGL state

2D Graphics: Coordinate Systems

- Screen coordinate system/Viewport
- World coordinate system/World window
- Window to Viewport mapping:
  - Motivation: why is it necessary?
  - OpenGL way: gluOrtho2D(**`left, right, bottom, top`**)
    glViewport **`(left, bottom, right-left, top-bottom)`**
  - Our way: calculate mapping
  - Applications: tiling, zooming, flipping, maintaining aspect ratio
- Cohen-sutherland clipping
  - algorithm operation
  - Why and how to do trivial accept/reject, chop
  - Given vertices, clip!!

Fractals

- What are fractals?
  - Self similarity
  - Applications (clouds, grass, terrain etc)
- Koch curves/snowflakes
  - How to build K1, K2, etc... S1, S2, etc.
  - Pseudocode: how to draw
- Mandelbrot set
  - Complex numbers: s, c, orbits, complex number math
  - Dwell function
  - Assigning colors
  - Mapping mandelbrot to screen

Points, Scalars Vectors

- Vector Operations:
  - Addition, subtraction, scaling
  - Magnitude
  - Normalization
  - Dot product
  - Cross product
  - Finding angle between two vectors
- Standard unit vector
- Normal of a plane

Transforms

- Homogeneous coordinates Vs. Ordinary coordinates
- 2D/3D affine transforms: rotation, scaling, translation, shearing
- Should be able to take problem description and build transforms and apply to vertices
- 2D: rotation (scaling, etc) about arbitrary center:
  - T(Px,Py) R(θ) T(-Px,-Py) * P
- Composing transforms
- OpenGL transform commands (glRotate, glTranslate, etc)
- 3D rotation:
  - x-roll, y-roll, z-roll, about arbitrary vector (Euler theorem) if given azimuth, latitude of vector or (x, y, z) of normalized vector
- Matrix multiplication!!

Modeling

- GLUT models (teapot, sphere, cube, etc)
- Overview of openGL
  - Modelview matrix (M and V part)
  - Projection matrix
  - Clipping
  - Viewport
- Should know high-level what each stage does
- OpenGL matrices: what are they? How to select, initialize, compose
- Hierachical modeling using OpenGL (glPopMatrix, glPushMatrix)
- SDL

Illumination and Shading

- Illumination models
  - Light types (point, extended, etc)
  - Global vs local illumination
  - Ambient, diffuse, specular
  - Phong light model
  - OpenGL lighting, material commands