

Designing Graphical System

- Now we're going to bring it all together.
- How do we build graphical systems for the real world?
- How to see through the hype when you work for a company and you have to deal with marketing people.
- How not to look like an idiot in front of your boss when he/she asks for your advice.

Main issues

- Interface: How will user interact?
- Software: How to implement? What algorithms to use? What API to use?
- Hardware: Input, output, processing
- Data modelling: How to model data?

Constraints

- Performance
 - Real-time/offline
- Quality of output
 - High quality required, or just simple graphics?
- Cost
 - What is budget?
 - Handheld all the way up to Render farm. Which is appropriate?

Designing graphical systems

- Art as well as science.
- Needs good knowledge of whole field.
- In my opinion the approach to take is to ask:
"What are the requirements/specifications?"
"What do I want my user's experience of the system to be?"
followed by
"What is appropriate?"

User experience

- Often a diagram really helps.
- Olympic site ... I actually drew a kiosk.
- Forced me to think of means of interaction.
- Touches on other areas: human-computer interaction, hardware, software, modelling. So that's why we do it first.

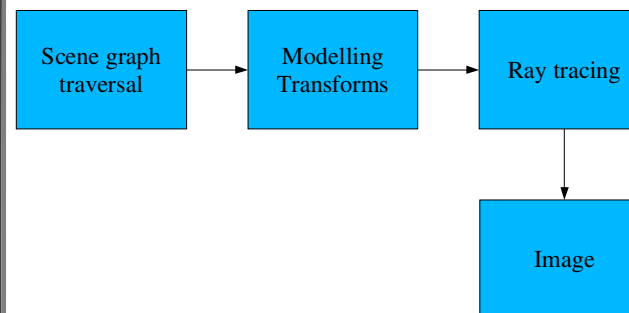
User experience and I/O

- Once we've defined the user experience, it's easy to work out possible I/O hardware options.
- Look at best possible and cheapest possible.
- Context is important.
 - Touch screen is appropriate for an information kiosk, VR gloves are not.
 - Touch screens are not appropriate for 3D modelling, VR gloves are.

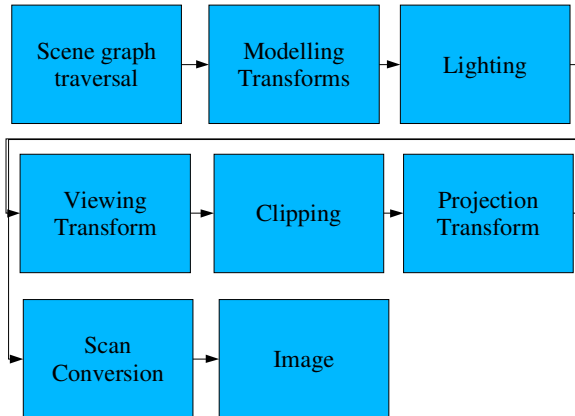
Implementation

- For graphical aspects, I often draw the rendering pipeline. Where is data coming from, where is it going to?
- Once I have decided on approach, I can then select the algorithms.
- Once the algorithms are chosen, then consider, hardware, software and APIs to use.

Generic ray tracing pipeline



Example: Polygon-Gouraud-Z buffer rendering pipeline



Issues in algorithm selection

- Basic style: Polygon, Ray tracing
- Above is basically a quality vs time tradeoff.
- Which illumination equation to use? specular? reflection/transparency/shadows?
- How to handle visible surface determination? Painter's vs BSP vs Z-buffer vs 2D.
- What type of polygon rendering? Flat vs Gouraud vs Phong

More issues

- What types of projection? Parallel vs Perspective, 3D, moving, etc.
- Texture mapping support? texture maps, bump maps, procedural vs image, 2D vs 3D, environment maps, mipmaps
- Antialiasing? Postfilter vs Prefilter

APIs

- Once implementation is chosen, can choose appropriate API, e.g. OpenGL, VRML, Direct3D, POV-Ray, GIF/JPEG.

Graphics hardware

- Choose appropriate graphics hardware for task.
- What are performance requirements?
- Software is more flexible + cheaper vs hardware is better performance and rigid
- Options: plain framebuffer, simple 3D card, high-end 3D card, 3D systems, custom raytracers etc.

Modelling data

- One weakness of students
- Depends on application
- Basic problem: How do I store data about scene I have to render?
- E.g. scene graph for basic structure.
- Read data in from file, over network, dynamically generated, what?

Make sure your system integrates!

- Bits have to fit together and be sensible.
- Example: Don't need Gouraud shading/z buffer when doing raytracing
- Using a high end graphics card won't speed up your CPU raytracing speed.
- You can't use the OpenGL API directly for doing ray tracing.

More pitfalls

- Don't suggest practically impossible things (e.g. real time ray tracing for a mobile phone).
- Or say we should use BSP trees or painter's algorithm when you have a graphics card with hardware Z buffer support.
- Can't build a CAVE for \$10,000.

Exam

- Four questions
- Three hours
- Question 1: Definitions/comparisons (20 marks)
- Question 2: Calculations (25 marks)
- Question 3: Design (25 marks)
- Question 4: Assignment (30 marks)

Question 1

- Definitions of terms:
 - environment mapping
- Comparison
 - Distinguish between Phong illumination model and Phong shading
 - solid vs surface textures
- Easy marks, just know your stuff

Question 2

- Calculations, like:
 - Normals + lighting
 - BSP tree arrangement
 - modelling and viewing transforms
 - Bezier curve evaluation
- Check your calculations! Normalise normals, BRING A CALCULATOR, do matrix multiplies in correct order.

Question 3

- Design: No one correct answer
- Given a very general specification
- Marked based on the "vision", the integration, and coverage of all important issues (i.e. hardware, software, APIs, interface, modelling).
- Modelling and interface is where people lose most marks. Lots of people forget I/O!
- Make a sketch!!

Question 4

- Assignment question
- Asks you to extend an assignment solution in a particular way.
- Goes from easiest to hardest.
- Last part or two are the "discriminators", hence they are hard.

Much much more to Graphics

- Physics, fluid simulation
- Advanced OpenGL: GLSL, particle systems, shadows, geometry shaders
- Deferred shading
- Quaternions
- Ray tracing with photon maps ...
- COMP9018 Advanced Graphics
- Special Projects ...