Computer graphics 2: Graduate seminar in computational aesthetics Angus Forbes

evl.uic.edu/creativecoding/cs526

Computer Graphics 2

instructor

angus graeme forbes

course syllabus

http://evl.uic.edu/creativecoding/cs526

lab page http://evl.uic.edu/creativecoding

Computer Graphics 2

Goals

- to become familiar with 3D graphics programming, especially with GPU shaders.
- to survey interesting topics in computer graphics and new media arts
- to create meaningful projects in "computational aesthetics", at the intersections of culture and technology

Pedagogical Philosophy

Project-centered classroom

 learning occurs much more effectively when you have a compelling idea that drives you to learn

Aware of research contexts

- you will have better ideas when you are familiar with current research contexts

Projects

Project based

- you will be involved in (at least) two larger projects which have both a *technical* component and a *conceptual* component.

 projects should be of sufficient quality and novelty to be accepted to either a top-tier graphics conference or a new media arts symposium or exhibition.

Projects

Project based

 target SIGGRAPH conference (submissions due in January) for technical and/or creative projects.

 target ISEA symposium (submissions due in November) for creative demos and project write-ups.

- exhibition during the last week of class or finals, TBA, but possibly at gallery 400.

Writing

Focus on "writing and thinking"

- research journals will help guide your exploration of new ideas, of interesting projects and papers, and of conferences and exhibitions dedicated to computer graphics topics.
- informal (but rigorous) writing assignment every week, lead to end-ofthe-term write-up of final project.

Today's class

- **0. Logistic details, introductions, space**
- 1. Topics and themes in media arts history
- 2. Discussion of papers from RJ
- 3. Small group discussion ideas
- 4. Brief overview of rendering pipeline
- 5. Lab session three.js

Media arts

What is "media arts"?

Term is a relatively recent invention, as really all art involves some kind of medium. Generally used to refer to "new media" or new technology: computers, sensors, displays, biotech, drones, robotics, etc.

Media arts

Artistic computer game modification; ASCII art; Bio Art; Computer art; Digital art; Digital poetry; Evolutionary art; Fax art; Generative art; Glitch art; Hacktivism; Hypertext; Information art; Interactive art; Internet art; Kinetic art; Light art; Motion graphics; Net art; Performance art; Radio art; Robotic art; Software art; Sound art; Systems art; Telematic art; Video art; Video games; Virtual art

https://en.wikipedia.org/wiki/New_media_art

Media arts

- Broadly speaking, situated in a number of different areas:
- Modern explorations/challenges to representation
- Examination/replication of systems and dynamics, in emergent properties of complexity
- Creation of new concepts and metaexploration of concept generation

Computer graphics

Two streams of (overlapping) graphics research:

- Photorealism, "The Rendering Equation"
- Visual information for practical applications

Computer graphics

- Entertainment

Film, animation, video games, motion graphics

- Design

Architecture, industrial design, media arts installations, advertising

- Research

Visualization/visual analytics, virtual reality, physical simulation

Rendering equation

James Kajiya, 1986

$$I(x,x') = g(x,x') \left[\epsilon(x,x') + \int_{S} \rho(x,x',x'') I(x',x'') dx'' \right]$$

where:

 $I(x, x') \quad \text{is the related to the intensity of light} \\ passing from point x' to point x \\ g(x, x') \quad \text{is a "geometry" term} \\ \epsilon(x, x') \quad \text{is related to the intensity of emitted light} \\ from x' to x \\ \rho(x, x'x'') \quad \text{is related to the intensity of light scattered} \\ from x'' tox by a patch of surface at x' \\ \end{cases}$

The rendering equation describes the total amount of light emitted from a point *x* along a particular viewing direction, given a function for incoming light and a bidirectional reflectance distribution function (BRDF).



How to represent a realistic world as quickly and accurately as possible.

Pablo Picasso: "She is rather small, isn't she? And flat?" (from Gregory Bateson).

In a sense, the more realistic, the more elaborate the artifice... eg, holodeck

What's in a photo? What is a photo made up of? What elements does it contain? What makes it interesting? What you could you add to make it more interesting (if anything)? What could you take out while keeping the main meaning intact?

What's makes something interesting?

"It's got to do with the contention between content and form. Invariably that's what's responsible for its energies, its tensions, its being interesting or not." – Garry Winograd











Argument: focus on realism in graphics has largely ignored fundamental aspects of seeing, which are emphasized in the works of expert artists.

Can you quantify Winograd's "energies" and "tensions"? Can they be applied to interactive computer graphics?

Elements of photorealism

- Rhythm
- Form
- Pattern
- Contrast
- Texture
- Content
- Intention
- Context

What's makes something interesting?

"There is a transformation, you see, when you just put four edges around it. That changes it. A new world is created." – Garry Winograd

What's makes something interesting?

What is the "frame" in contemporary computer graphics?

- interactive
- modifiable
- infinite

A subset of research on photorealism is called "non-photorealistic rendering". Somewhat ironically, this term tends to mean photorealistic renderings of artworks (usually paintings), which themselves are stylized representations of the real world.

Nonetheless, a very interesting topic...

"The reason I'm painting this way is that I want to be a machine, and I feel that whatever I do and do machine-like is what I want to do"

-Andy Warhol

"When Picasso died I read in a magazine that he had made four thousand masterpieces in his lifetime and I thought, 'Gee, I could do that in a day."

-Andy Warhol

- How do you represent an individual style?

- How can you apply an *interpretive intelligence* to a scene?

Swarm Vision, George Legrady



Light Space Modulator, Moholy-Nagy

http:// www.sfmoma.org/ explore/ multimedia/ videos/1



Rendering Pipeline



The Object or Local coordinate system is defined in terms of the Geometry itself. The origin is usually the center or the lower-left of the object.

The Model or World coordinate system defines the x, y, and z axes which serve as a basis for the 3D space. Where is the origin? Which way is up?

The Eye, Camera, or View coordinate system defines another set of x, y, and z axes which server as a different basis for the 3D space. The camera is always positioned at the origin of this coordinate system.

The *Clip* coordinate system describes the bounded view of the visible by the camera in terms of both the "lens" of the camera, its "depth of focus", and the aspect ratio of the screen bounds.

The Normalized Device coordinates is the same view normalized from -1 to +1 along each axis.

The Window or Screen coordinates are these x and y coordinates positioned within the screen bounds. The z is used for depth-testing and is bound between 0 and 1

CPU / GPU

CPU side Define geometry; handle interaction

GPU side

<u>Vertex shader</u>: transform geometry from 3D space to 2D space

<u>Fragment/Pixel shader</u>: determine pixel color of every pixel inside the geometry

Homework 2 – RJ ongoing...

- Follow the suggestions in the Research Journal handout and find and describe THREE papers of your choice.
- Come up with (or refine) FIVE interesting ideas in computer graphics or new media arts
- Research ONE academic or creative venue

Research

- ACM DL
- ISEA
- IEEE Xplore
- Google Scholar
- Lab pages

Homework 2 – Emulation Project

- Choose one artist from the Inventing Abstraction website

a) recreate the artwork with as much detail as possible (using Three.js) (due 9/15)

b) create a program that *applies* the style of the artist to any input image (or video) (due 9/22)

Lab session, Three.js

Create a scene containing a series of geometric objects. Animate the objects (your choice).

Add point lights to the scene. Animate the movement of the lights (your choice).