# CS 3451: Computer Graphics

Blair MacIntyre

Fall 2019

As scientists and engineers, we should follow the best evidence and best practices available

The Board of Regents has said no school in the USG may require vaccines or masks in the classroom

I am asking a favor from you:

**Please wear a mask in this class**

# Introductions

Instructor

- Blair MacIntyre (he/him)
- blair@cc.gatech.edu

Augmented and Mixed Reality (“interactive 3D in the world”), Games, Graphics, HCI

- Professor, Coc/IC; Adjunt Professor, LMC
- Recently a Principal Research Scientist, Mozilla (2016-2020)
- Design of MR/AR experiences (3D in the world), web-based XR
- Background in math, CS, 3D graphics
- Interesting facts: I play the bagpipes (not enough), and Overwatch (too much)

https://cs3451-f21.github.io/



## Time and Place

TTh, 12:30 – 1:45, Paper Tricentennial 109

**INSTRUCTORS AND TAS**

**RESOURCES**

Canvas
Teams
Github

**SITE LINKS**

All Posts
Announcements
Assignments

# Introduction to Computer graphics

This course is designed to be a first course in computer graphics, and as such, no previous experience in graphics is assumed. If you have had any prior course in computer graphics, this course may be too basic for you. The only knowledge that you need coming into this course is basic linear algebra and strong programming skills.

Course topics include: output hardware, rasterization, 2D and 3D transformations, projection, hidden surfaces, color vision, surface reflectance and illumination, texture mapping, shadows, anti-aliasing, hardware graphics pipeline, GPU programming, ray tracing, polyhedral models, polynomial curves and surfaces, subdivision surfaces. A few student-chosen topics will be covered near the end of the course.

For this class, we will use modern web technology to create interactive 3D graphics for the web. Students will program in Typescript, a modern, object-oriented superset of Javascript, and will use WebGL, CSS3 3D Transforms, and three.js for their assignments.

# TAs

# Evan Goode
(he/him)

- Preferred email: egoode6@gatech.edu
- Program: MS in Computer Science specializing in Computing Systems=
- Interesting fact: I'm starting a perfume company

# Kevin Huynh
(he/him)

- Preferred email: khuynh44@gatech.edu
- Program: 4th year computer science major with a concentration in media and information internetworks
- Interesting fact: I enjoy breakdancing on my free time

# Sarang Joshi
(he/him)

- Preferred email: sjoshi92@gatech.edu
- Program: PhD student in Computer Science (4th year)
- Interesting fact: I love playing (and watching) badminton.

# Diana Kim
(she/they)

- Preferred email: dkim806@gatech.edu
- Program: 3rd year CS (Media/Intelligence threads)
- Interesting fact: I own a small art shop, where I sell stickers and shirts I design!!!

# Katia Lucas
(she/her)

- Preferred email: klucas32@gatech.edu
- Program: BS/MS - 1st year of MSCS with a specialization in Computer Graphics
- Interesting fact: I have 3 nationalities: American, French, and Russian!

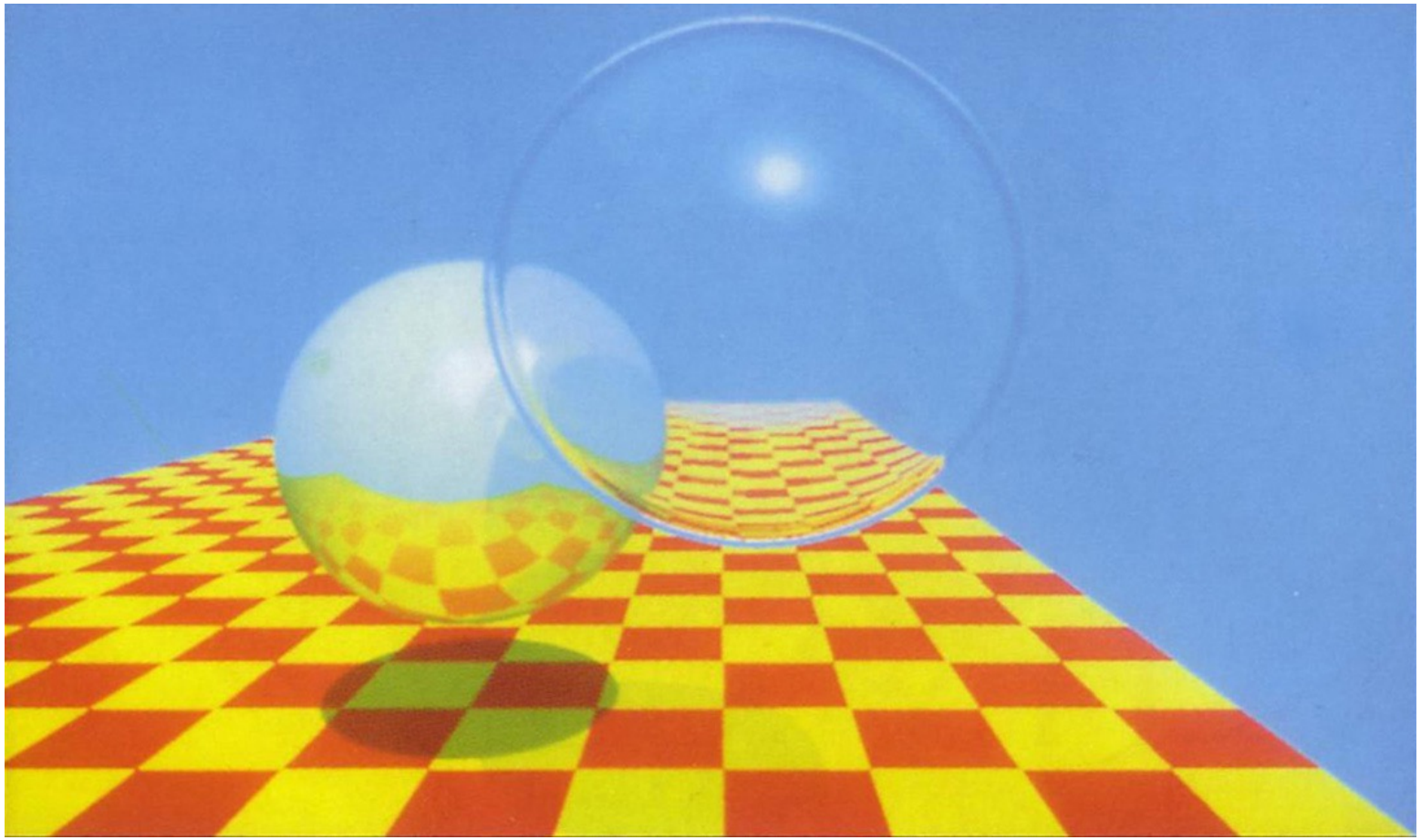
# Josh McCord
(he/him)

- Preferred email: jmccord34@gatech.edu
- Program: (Senior) Computer Science major with threads in Info-Internetworks and Media.
- Interesting fact: I am currently obsessed with retro gaming consoles and own every Nintendo console with each of their respective Zelda games.

# Ryan Miles
(he/him)

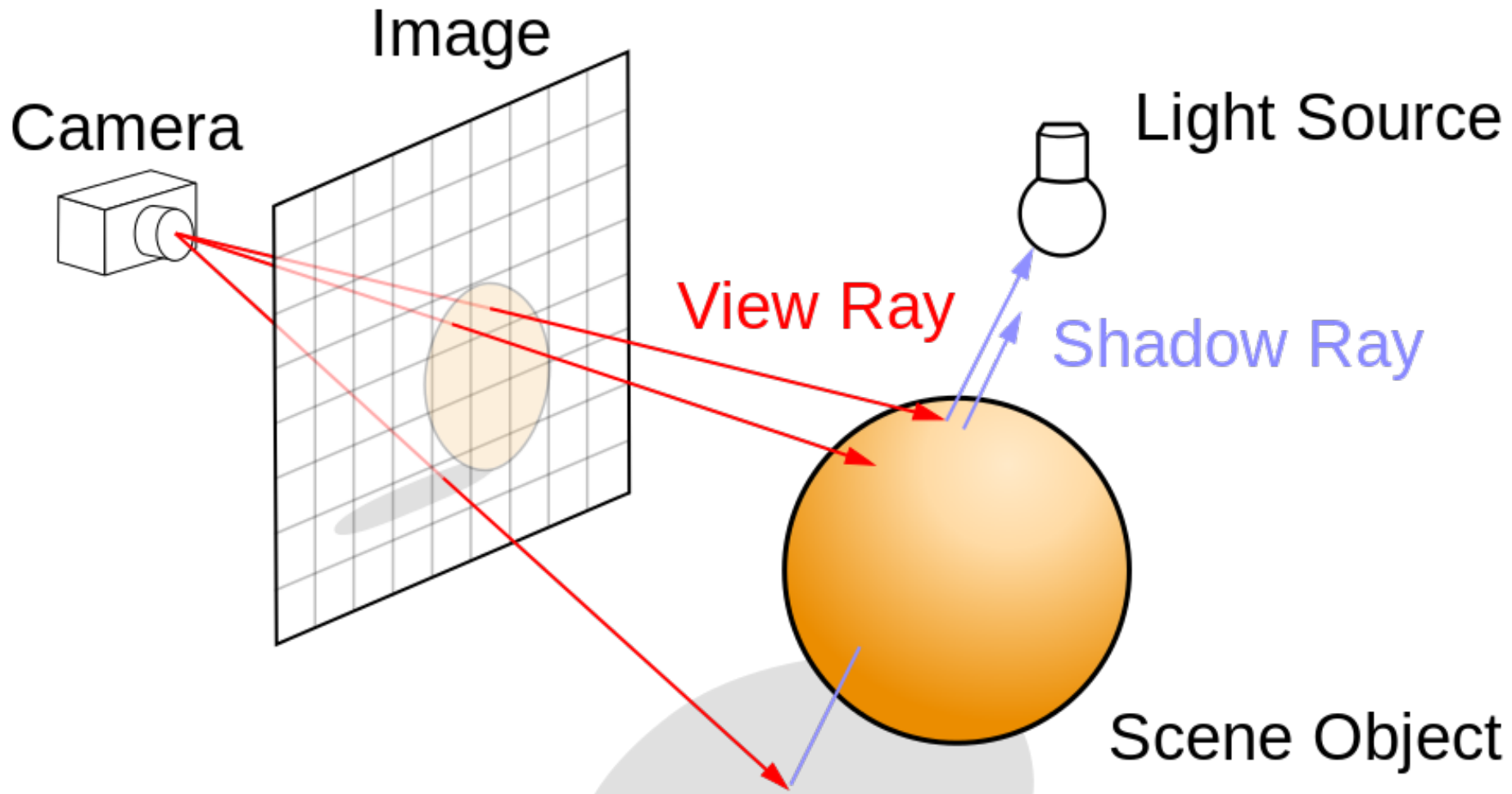
- Preferred email: rmiles30@gatech.edu
- Program: 1st year MSCS student in BS/MS program
- Interesting fact: I type using the Colemak keyboard layout

# Ray Tracing: Global Illumination & Rendering

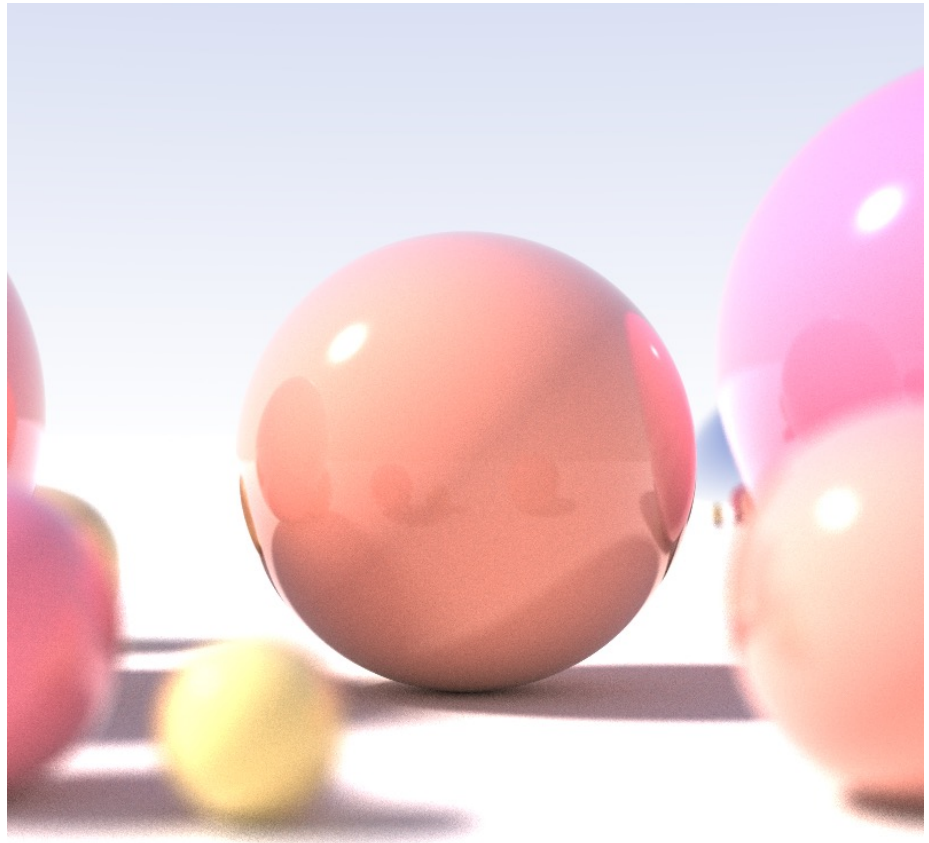


Turner Whitted. 1980. An improved illumination model for shaded display. *Commun. ACM* 23, 6 (June 1980), 343-349. DOI=10.1145/358876.358882 http://doi.acm.org/10.1145/358876.358882

# Intuitive, Simple to Understand



# Basis of Modern Global Illumination



Images from https://en.wikipedia.org/wiki/Ray_tracing_(graphics)

# Basis of Animated Film

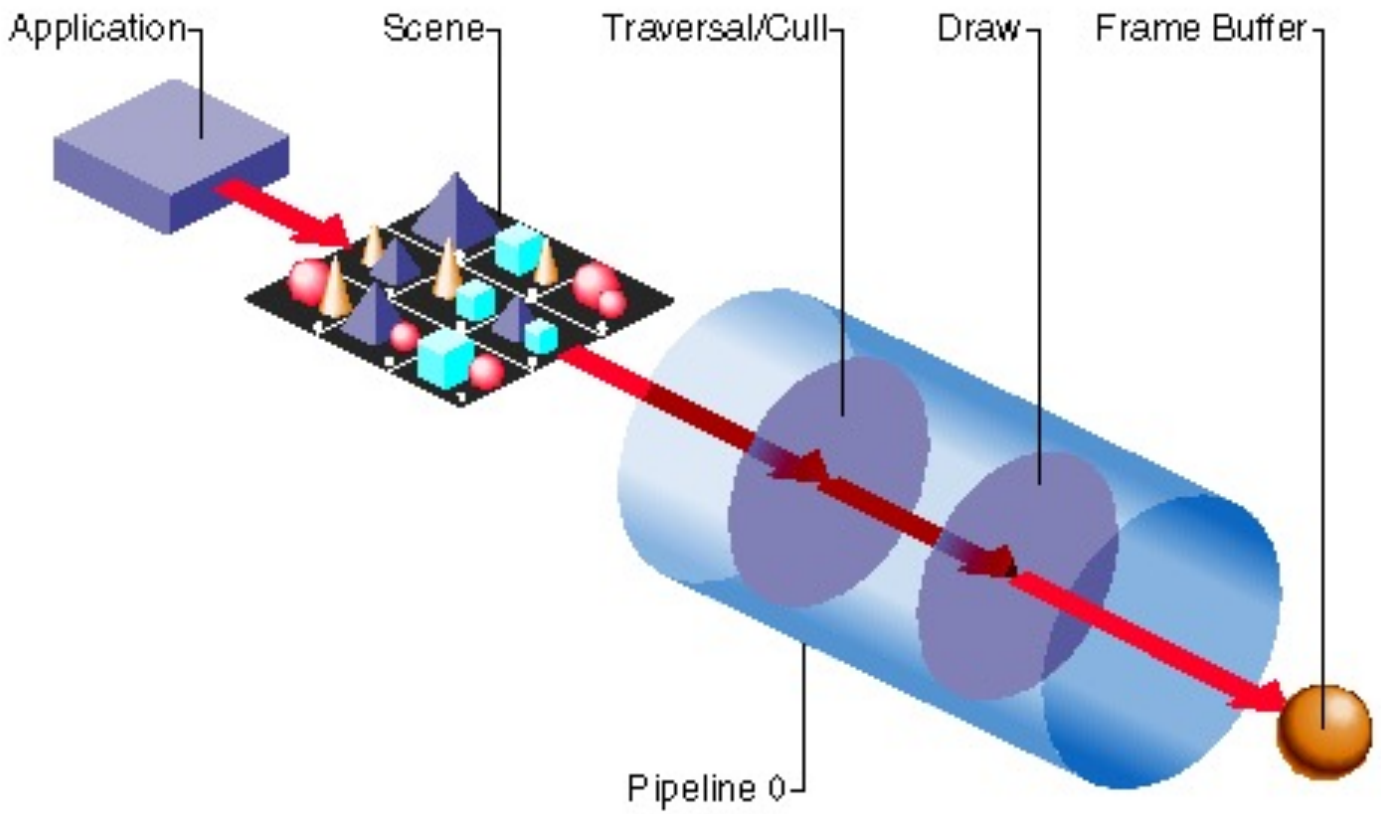
- Pixar's Renderman, recently Disney's Hyperion
  - http://www.fxguide.com/featured/disneys-new-production-renderer-hyperion-yes-disney/
  - http://www.engadget.com/2014/10/18/disney-big-hero-6/
  - http://www.geek.com/chips/inside-the-199-million-core-hours-that-created-big-hero-6-1609488/
- Big Hero 6
  - Videos (Sizzle Trailer https://www.youtube.com/watch?t=39&v=ckAGKzTKsvw, Graphics Tech https://www.youtube.com/watch?v=w4WD15EgLWc)
  - Hyperion renders 10 – 20 bounces of indirect lighting
  - Create new render farm: 55,000-cores across four geographic locations
  - San Fransokyo contains around 83,000 buildings, 260,000 trees, 215,000 streetlights and 100,000 vehicles (plus thousands of crowd extras)
  - 20 million microbots onscreen in a given shot, on average
  - 199 million core-hours to render
    - Peak production ran at 1.1 million core-hours per day
    - Tangled was 11.5 core-hours, could render on the BH6 farm in under 10 days

# Question:  How do we get close to reality?

- Look around, wherever you are.  You'll see places (e.g., under a desk) that do not obviously have any direct illumination fall on them.  How will this work?

- (Think about this for a minute or two)

# Online (“real-time”) Rendering Inverts Things




http://techpubs.sgi.com/library/dynaweb_docs/0630/SGI_Developer/books/Perf_PG/sgi_html/ch02.html

# Question: How do we render objects?

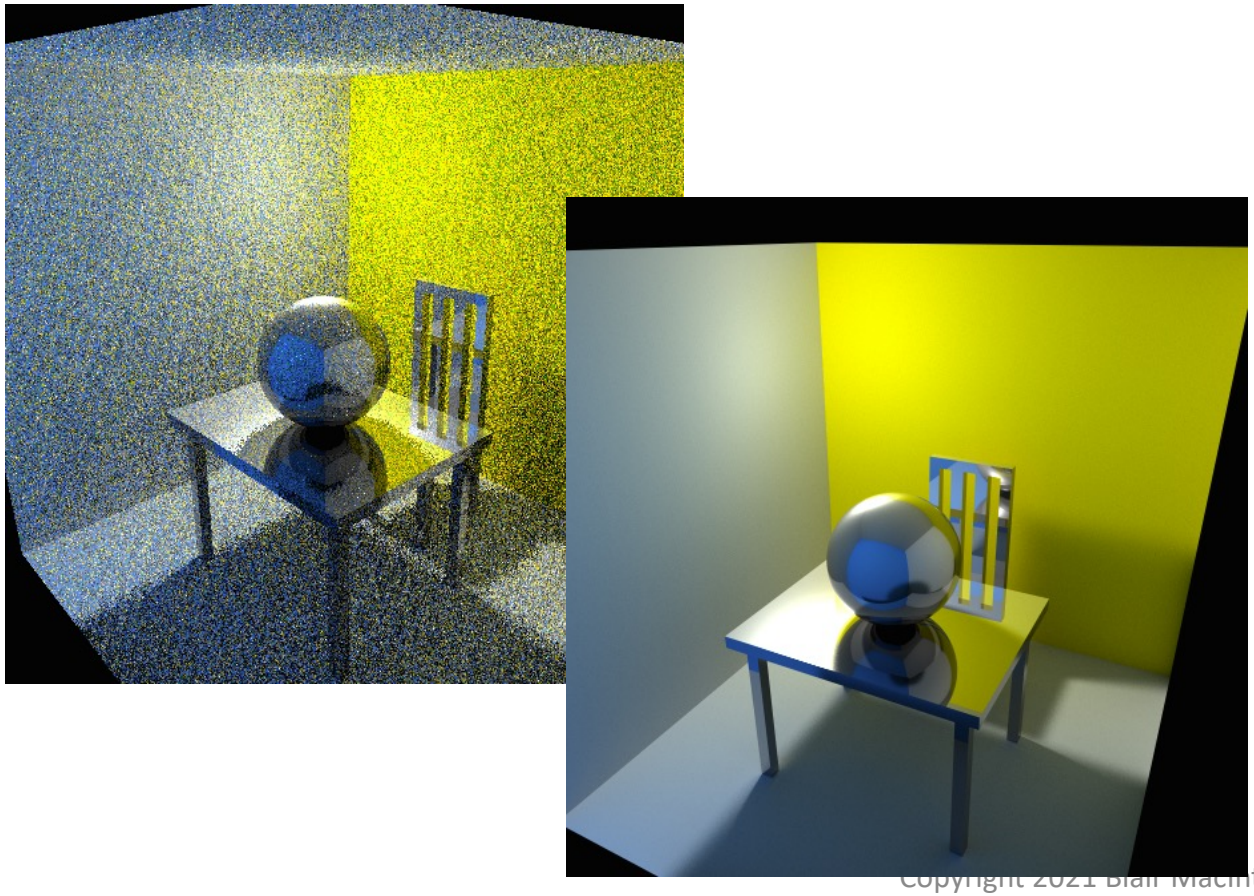
- With ray tracing, we see things based on if rays intersect with object, lights, etc.  What does it mean to “draw a cube” or even “draw a line” on the screen?

- Consider a line on a display, going from pixel 0,0 to pixel 2,5.  How does this get “drawn”?

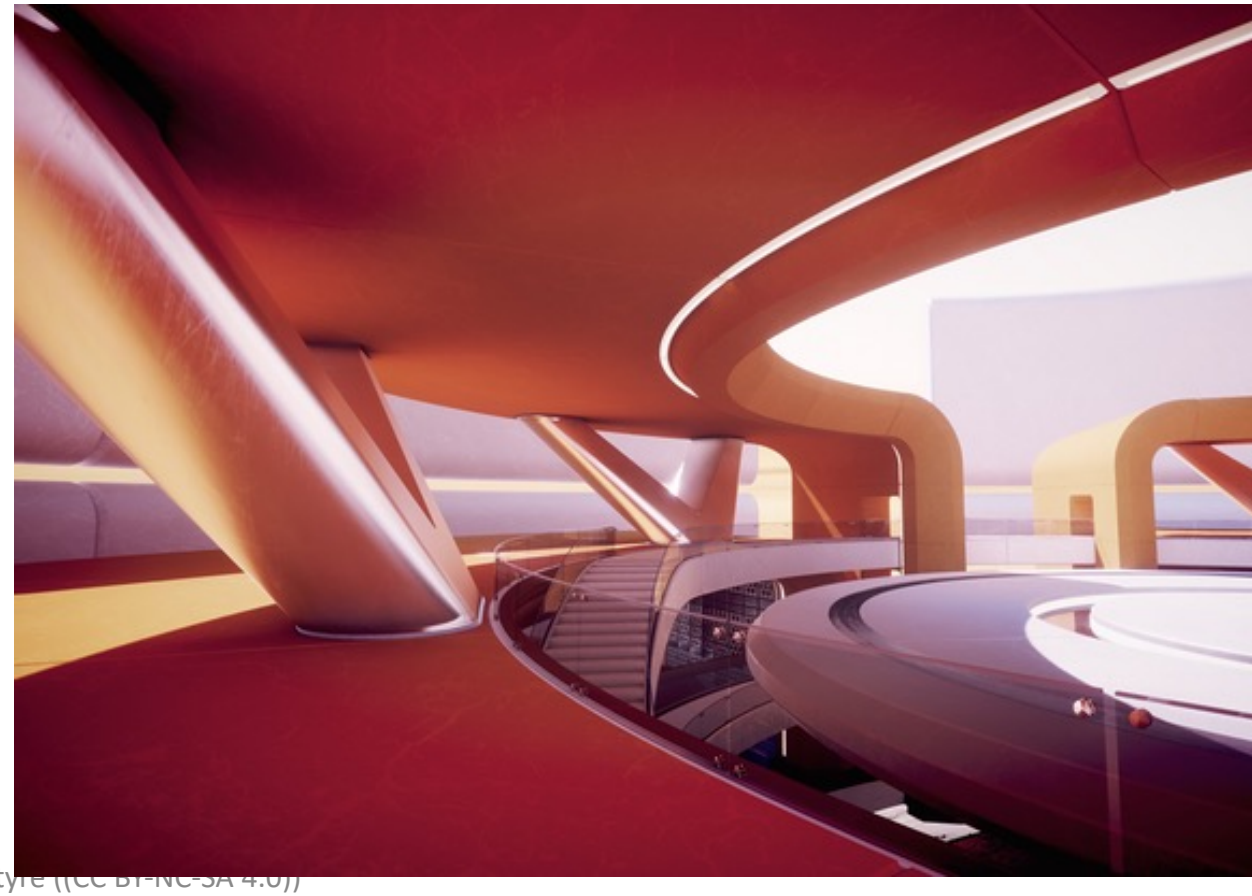
- Think about this for a minute or two.  Perhaps draw some pictures.

# GPUs let us blur the lines

http://madebyevan.com/webgl-path-tracing/



http://docs.unity3d.com/Manual/GIIntro.html



# Overview of the Class

- First 10 weeks: Global Illumination (offline) vs Interactive Graphics (online)
  - Interactive 3D graphics
    - 3D mathematical foundations
    - Scene graphs, the graphics pipeline
  - Shading and lighting
  - Hardware and GPU programming
  - Intro to Global Illumination with Ray tracing
- Return to other topics later
  - Curves, surfaces, meshes, data structures and models
  - More rendering (Radiometry, Light and Color)
  - Real time interactive graphics, AR/VR

# Learning Objectives

- Understand
  - Mathematical and algorithmic foundations of computer graphics
  - Limits of approximating real visual phenomena digitally
  - Differences between online and offline graphics techniques
  - Modern graphics hardware architectures, GPU programming
  - How to create interactive graphical applications

- Additionally
  - Modern tools, technologies, and techniques for doing interactive 3D graphics on the web

# *Administrivia*

# Topics

- From the catalog
  - Geometric constructions; transformations; perception; reflection models; photorealistic; non-photorealistic, and image-based rendering; rendering software and API's; triangle-mesh processing; graphic acceleration; user-interaction, design and animation.
- Essentially
  - Modern 3D graphics, both online and offline
  - Creating interactive graphical applications

# Prerequisites

- MATH 2605 or MATH 2401 or MATH 2411 or MATH 24X1, and
  - Calc III:  Linear algebra, calculus concepts, numerical methods
  - Why? Graphics uses math, especially these topics
- CS 2110 or CS 2261, and
  - Hardware/software systems, how machines work
  - Why? Understanding graphics hardware, GPU programming
- CS 1332, and
  - Data structures and algorithms
  - Why? Trees, lists, queues, more complex spatial data structures
- CS 2340
  - OOP, software design, debugging and testing
  - Why? Graphics programs and libraries are complex, we'll use objects and OOP

# Class Resources on github.com

- http://github.com/cs3451-f21
  - Source to everything (website, samples, assignment source code)

- Website will change over semester
  - I'll try to make the changes obvious!  Will post announcements on Canvas

- Schedule is a starting point, may also change
  - Will get the assignment due dates up in the next few days
  - But I'll try not to move assignment deadlines and test dates after this week

# Grading

| Component | Grade |
|---|---|
| Warmup project | 5% |
| Programming assignments: | 65% |
| Quizzes | 10% |
| Midterm exam: | 10% |
| Final exam: | 10% |

Final Grade:

60 70 80 90

F | D | C | B | A

# Lectures

- Read the material
- Bring a laptop to the lecture

- *Intend* to stream the lectures on Teams, and record them
  - You are responsible for anything we talk about in class.
  - If you miss class, and the stream doesn't work for some reason, talk to classmates!
    - I don't expect this to be common, and I'm trying to have backups.
    - Worst case, assuming it's rare, I'll just re-run the session online in the evening to create a recording

# Annoying Warning

- Over the years, I've had too many complaints about students distracting others by goofing around on laptops. So, please don't open your laptop in class except for class related activities
  - If you like to take notes on your laptop, great!
  - Just please don't spend time watching videos, skimming social media, etc. It's very distracting to people behind you

- No recording devices beyond the recorded lectures, without permission (FERPA)
  - You may not record any video aside from what I record on Teams
  - You MUST NOT share ANY recording (pics, audio, recorded streams) with anyone outside the class. Doing so is considered academic misconduct by GT and will be taken seriously

# Questions?

# Next class

- Interactive walkthrough of some web programming, Typescript and Javascript concepts

# Setting up for Development

- Git and Github Desktop
  - https://mac.github.com or https://windows.github.com
  - Log into github.com, create an account if you don't have one, doesn't need to be personally identifying (we'll get you to submit it)
- Install Nvm, use it to install Node
  - Mac/linux (https://github.com/nvm-sh/nvm), Windows (https://github.com/coreybutler/nvm-windows)
  - node (https://nodejs.org) and npm (https://npmjs.org)
- (optional) Install Typescript
  - http://www.typescriptlang.org/#Download
  - *npm install –g typescript*

# GITHUB.COM, not GITHUB.GATECH.EDU

- Install an editor with Typescript language service extension
  - I'm currently using Visual Studio Code (works on many platforms)
    - https://code.visualstudio.com
  - Also: Visual Studio, WebStorm, Atom, Sublime Text, Eclipse all have some level of support
- Check out the Typescript samples
  - https://github.com/Microsoft/TypeScriptSamples
- Clone ex1 from class repository
  - https://github.com/cs3451-f21/ex1