

CS 3451: Computer Graphics

Blair MacIntyre

Fall 2019

Introductions

- Instructor
 - Blair MacIntyre
 - blair@cc.gatech.edu
- Augmented and Mixed Reality (“interactive 3D in the world”), Games, Graphics, HCI
 - Professor, Coc/IC; Adjunct Professor, LMC
 - Principal Research Scientist, Mozilla
 - Design of MR/AR experiences (3D in the world), web-based XR
 - Background in math, CS, 3D graphics
 - PhD research in distributed interactive 3D software design



Copyright 2019 Blair
MacIntyre ((CC BY-
NC-SA 4.0))

Copyright 2019 Blair MacIntyre ((CC BY-NC-SA 4.0))

TAs

- Elie Diaz (ediaz30@gatech.edu)
- Evan Goode (egoode6@gatech.edu)
- Ryan He (zhe66@gatech.edu)
- Daniel Kane (dkane6@gatech.edu)
- Austin Passmore (lpassmore3@gatech.edu)
- Isabelle Steffens (isteffens3@gatech.edu)
- Ethan Wilson (ewilson72@gatech.edu)

Elie Diaz

- ediaz30@gatech.edu



Copyright 20

Evan Goode

- egoode6@gatech.edu



Copyright 2019 Blair MacIntyre

Ryan He

- zhe66@gatech.edu



Copyright 2019 B

Daniel Kane

- dkane6@gatech.edu



Copyright 2019 Blair MacIntyre

Austin Passmore

- lpassmore3@gatech.edu



Copyright 2019 Blair MacIntyre ((CC BY-NC-ND))

Isabelle Steffens

- isteffens3@gatech.edu



Copyright 2019 Bl

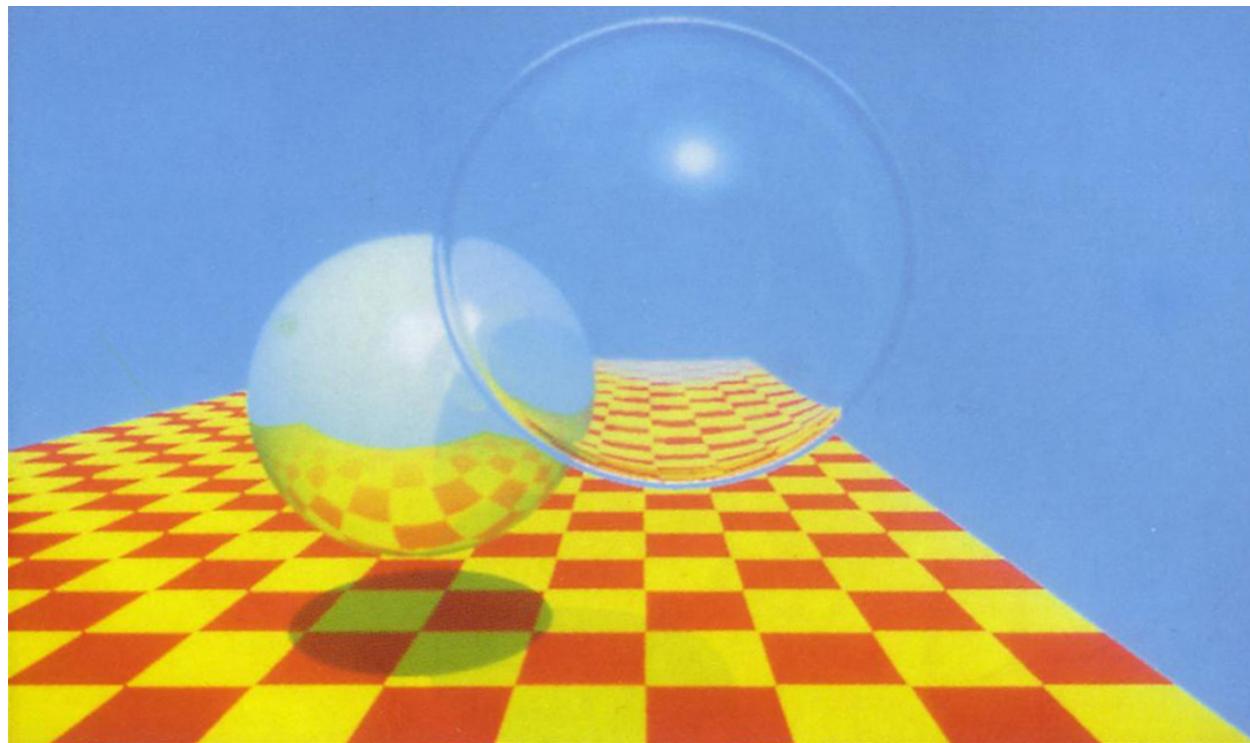
Ethan Wilson

- ewilson72@gatech.edu



Copyright 2019 Blair MacIntyre ((CC BY-

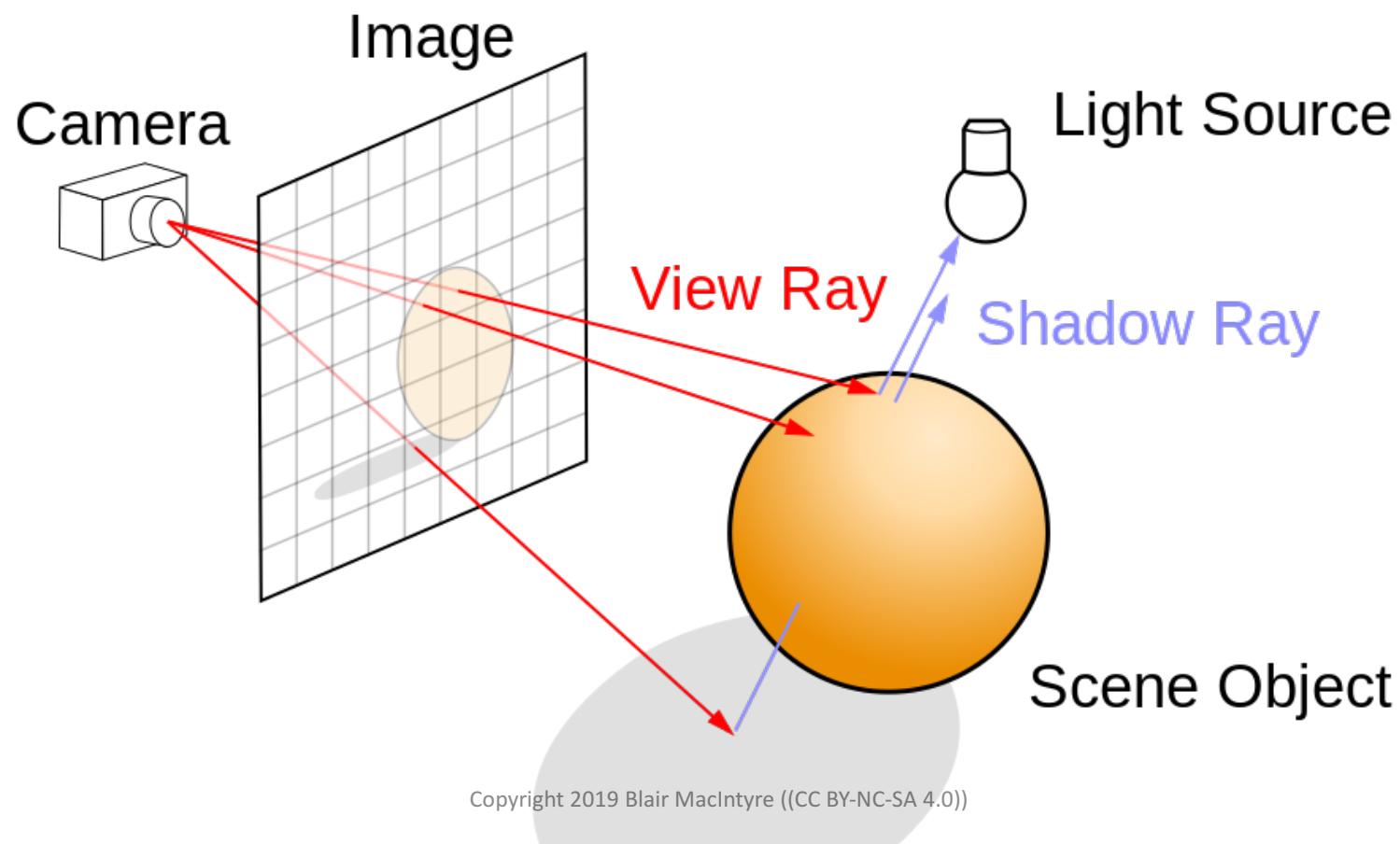
Ray Tracing: Global Illumination & Rendering



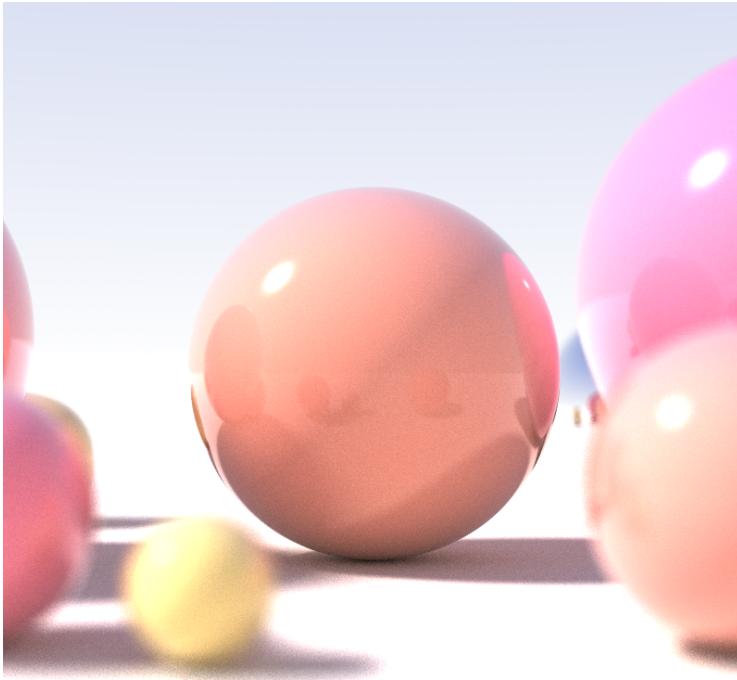
Turner Whitted. 1980. An improved illumination model for shaded display. *Commun. ACM* 23, 6 (June 1980), 343-349. DOI=10.1145/358876.358882 <http://doi.acm.org/10.1145/358876.358882>

Copyright 2019 Blair MacIntyre ((CC BY-NC-SA 4.0))

Intuitive, Simple to Understand



Basis of Modern Global Illumination

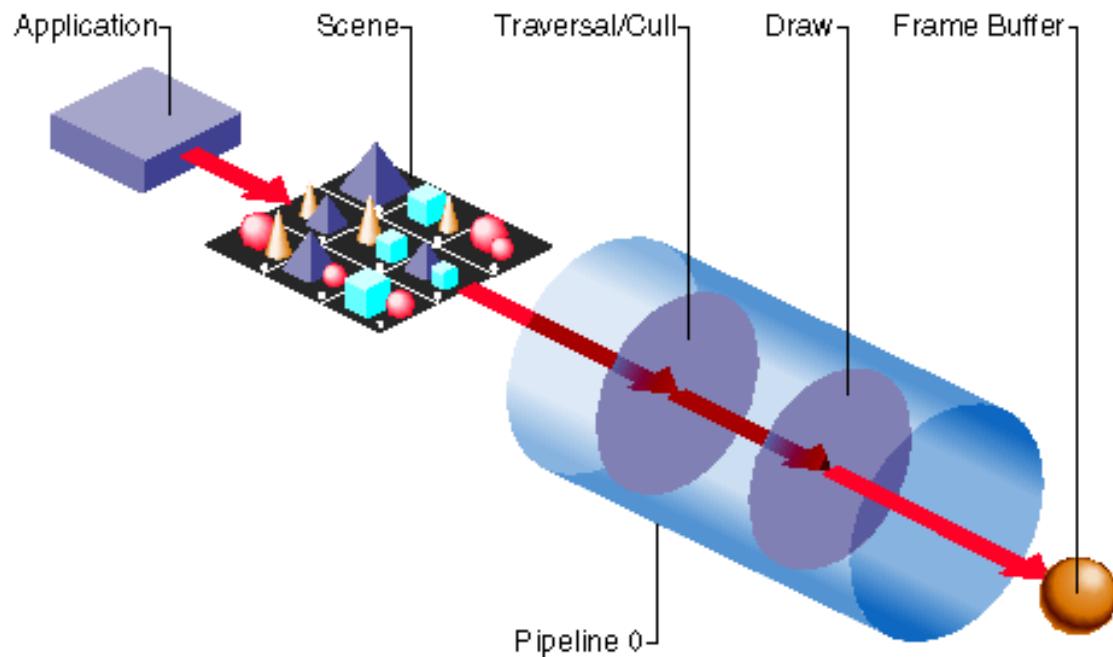


Images from [https://en.wikipedia.org/wiki/Ray_tracing_\(graphics\)](https://en.wikipedia.org/wiki/Ray_tracing_(graphics))
Copyright 2019 Blair MacIntyre ((CC BY-NC-SA 4.0))

Basis of Animated Film

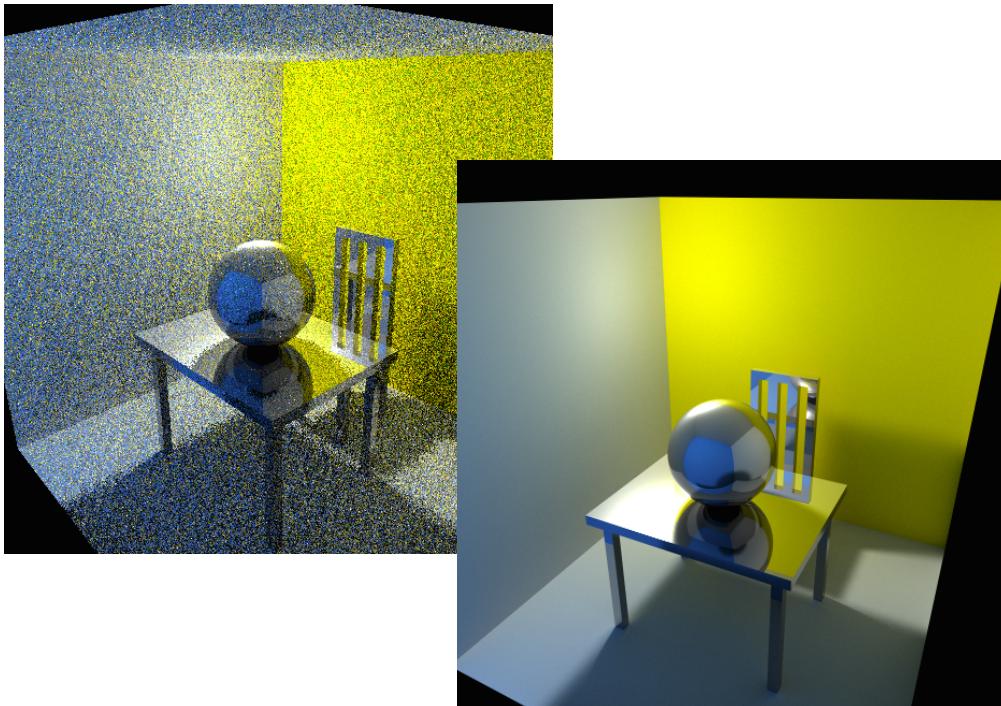
- Pixar's Renderman, recently Disney's Hyperion
 - <http://www.fxguide.com/featured/disneys-new-production-renderer-hyperion-yes-disney/>
 - <http://www.engadget.com/2014/10/18/disney-big-hero-6/>
 - <http://www.geek.com/chips/inside-the-199-million-core-hours-that-created-big-hero-6-1609488/>
- Big 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

Online (“real-time”) Rendering Inverts Things

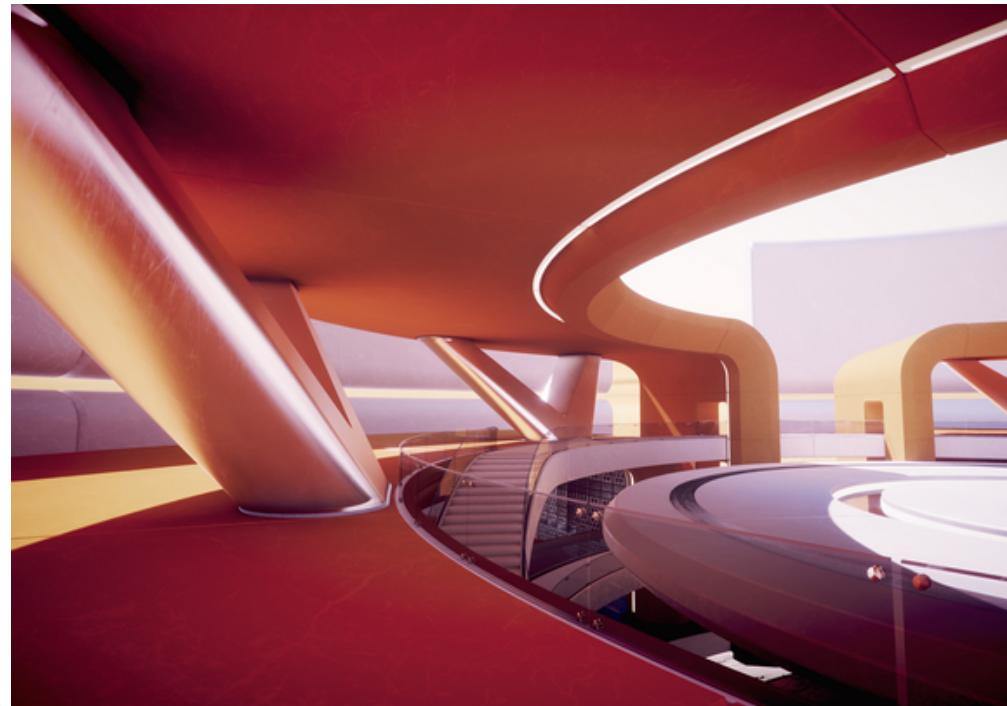


GPUs let us blur the lines

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



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



Copyright 2019 Blair MacIntyre (CC BY-NC-SA 4.0))



Copyright 2019 Blair MacIntyre ((CC BY-NC-SA 4.0))

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

Administrivia

Copyright 2019 Blair MacIntyre ((CC BY-NC-SA 4.0))

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

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
 - Tools, technologies and techniques for doing interactive 3D graphics on the web

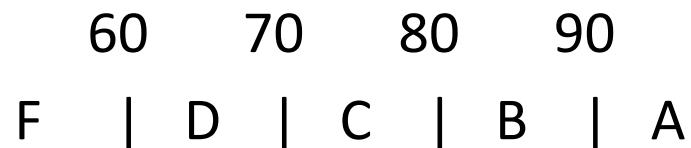
Class Resources on `github.com`

- <http://github.com/cs3451>
 - Source to everything (website, samples, assignment source code)
- Website will change over semester
 - I'll try to make the changes obvious! Will post announcements to the site
- Schedule is a starting point, may also change
 - But I'll try not to move assignment deadlines and midterm date

Grading

- Heavy emphasis on assignments
 - Assignments: 75%
 - Assignment 0: 5%, remaining 5 share 70%
 - Midterm exam: 10%
 - Final exam: 15%

- Final Grade:



Lectures

- Read the material
- Bring a laptop
- You are responsible for anything we talk about in class. If you miss class, talk to classmates!

Annoying Warning

- I've had too many complaints about students distracting others by goofing around on laptops. So: NO OPEN LAPTOPS in class, unless we ask you to take them out
 - If you like to take notes on your laptop, fine. But, I may ask you to show me what you are doing, at any moment. And I may ask you to send me your notes, just to see.
- No recording devices, without permission
 - If you want to use a Livescribe, audio recorder, etc., let me know
 - You may NOT record any video
 - You MUST NOT share ANY recording (pics, audio) with anyone outside the class

Questions?

Copyright 2019 Blair MacIntyre ((CC BY-NC-SA 4.0))

Next class

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

Setting up for Development

- Ensure you have a browser that supports WebGL and web debugging
 - Any up-to-date browser should work
- Install Git
 - <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 Node (including npm and other tools)
 - <http://nodejs.org>
- Install Typescript
 - <http://www.typescriptlang.org/#Download>
 - npm install -g typescript

- Install an editor with Typescript language service extension
 - I'm currently using Visual Studio Code
 - <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/f19-ex1>