Welcome to

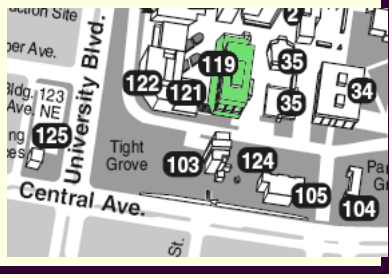
CS 413
Ray Tracing and Vector Graphics

Instructor: **Joel Castellanos**

e-mail: joel@unm.edu

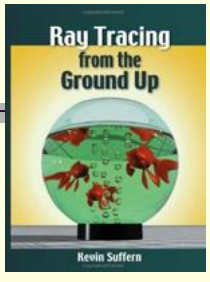
Web: http://cs.unm.edu/~joel/

Farris Engineering Center: 2110



2/5/2019

Course Resources



Textbook: Ray Tracing from the Ground Up by Kevin Suffern

Blackboard Learn: https://learn.unm.edu/

- Assignement Drop-box
- Discussions
- Grades

Class website: http://cs.unm.edu/~joel/cs413/

- Syllabus
- Projects
- Lecture Notes
- Readings Assignments
- Source Code

Structure of the Course

- Studio: Each of you will, by the end of the semester, build a single, well featured ray tracer / procedural texture renderer.
- Stages this project total to 70% of course grade.
- No exams
- Class time (30% of course grade) :
  - Lecture
  - Discussion of Reading
  - Quizzes
  - Show & Tell
  - Code reviews

3

Language and Platform

- The textbook examples are all in C++:
  - STL (Standard Template Library – this is multiplatform).
  - wxWidgets application framework (Windows and Linux).
  - Class examples replace wxWidgets with OpenFrameworks.
- Decide on an IDE. If you are using Windows, I recommend MS Visual Studio as it is an industry standard and many employers will expect you to be comfortable with it. For Linux, there are a number of good options including:, Eclipse C++, Netbeans for C/C++ Development, Code::Blocks, Sublime Text editor and others.
- Your project must be in an on-going repo in GitHub. This can be public or private. Building a good GitHub repo to show prospective employers is essential to a young computer scientist in the current job market.

4

Assignment 1

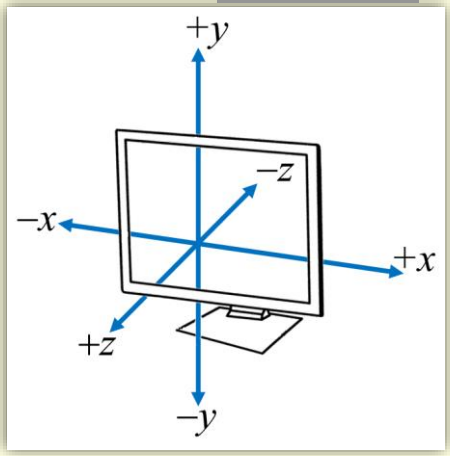
- For Monday, Jan 29:
  - Read Chapters 1, 2 (skim and reference later) and 3.
  - Be prepared to discuss solutions to questions 3.1, 3.2, 3.3 & 3.4, 3.5 & 3.6.
- "Bare-Bones Ray Tracer +".
  - If not using wxWidgets, then it is fully enough to get the single red sphere displayed in a simple materialess, lightingless, orthographic projection straight down the z-axis.
  - If using wxWigets, then get the supplied code working with an additional 3 menu options:
    a) Exercise 3.1 (single, simple orthographic sphere)
    b) Exercise 3.2 (part a, with z_w = 100.0)
    c) Exercise 3.3 (sphere with different center, radius & color).
    d) Some feature you think is cool.

5

5

3D Graphics Coordinate System

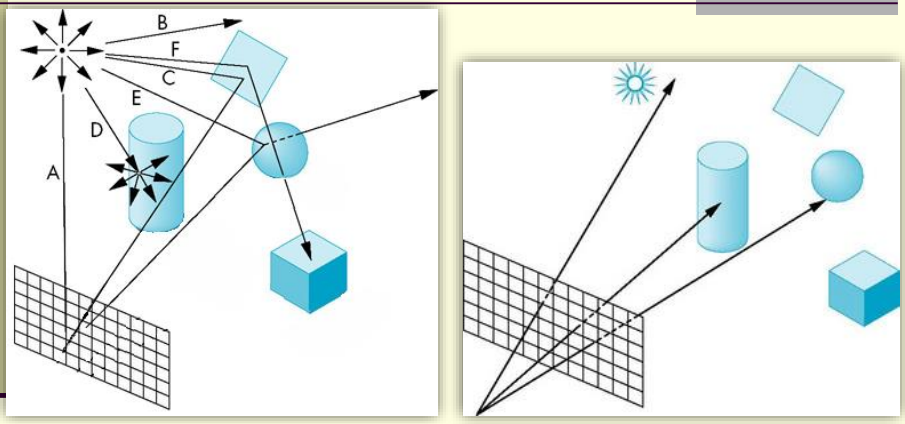
Throughout this course, the usual way of specifying the location of a point in 3D space is as Cartesian coordinates using an ordered triplet, (x, y, z), of floating point numbers (`float` or `double`).



6

6

Ray Tracing verses *Ray Casting*



Ray Tracing

Ray Casting

Simple *Ray Casting* Ray Tracer

1) Define some objects.
2) Specify a material for each object.
3) Define a window whose surface is covered with pixels.
4) For each pixel:
   a) From the center of each pixel, cast a ray towards the objects.
   b) Compute the nearest (if any) hit point of the ray with the objects.
   c) If the ray hits an object:
      Use object's material and the lights to compute pixel color.
   d) else:
      Set the pixel color to black.

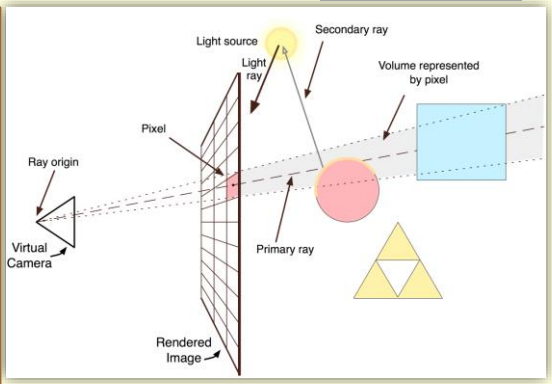
Four Common Types of Rays

Primary Rays start at the centers of pixels for parallel viewing and at the camera for perspective viewing.

Secondary Rays are reflected and transmitted rays that start on object surfaces.

Shadow Rays are used for shading and start at object surfaces.

Light Rays start at the lights and are used to simulate some aspects of global illumination.



Chapter 3 covers only Primary Rays and only those starting at the centers of pixels.

9

OpenFrameworks (getting started: 1 of 7)

- OpenFrameworks 0.10.1
- Visual Studio 2017 (v 15.9.2)
- OpenFrameworks Website: https://openframeworks.cc/
- There are instructions for installing the OpenFrameworks plugin into Visual Studio. The instructions were written for Visual Studio 2015 and updated for VS 2017. However, following the instructions does not work.
- The next few slides show steps that do work for creating a simple circle drawing app with slider control.

10

10

OpenFrameworks (getting started: 2 of 7)

- For Microsoft Windows 10 and Visual Studio 2017.
- Download of_v0.10.1_vs2017_release.zip from https://openframeworks.cc/download/
- Unzip the folder and place the folder in your working directory.
- Change the name of the folder from "of_v0.10.1_vs2017_release" to something more sensible such as "OpenFrameworks_v0.10.1".
- Copy and rename the example folder: OpenFrameworks_v0.10.1\apps\myApps\emptyExample to your project name (i.e. RayTracer_00).
- Your project folder must be in the same folder as emptyExample (two levels below the OpenFrameworks folder).

11

11

OpenFrameworks (getting started: 3 of 7)

- Open Visual Studio 2017.
- From the "Tools" menu, select "Extensions and Updates". Then select "Online" and search for "openframeworks".
- Click "download". When it finishes, restart Visual Studio.

12

12

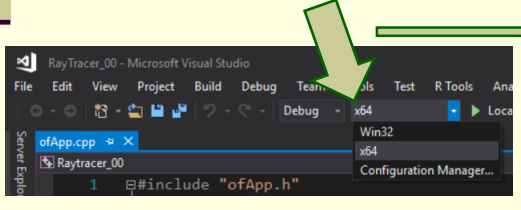
OpenFrameworks (getting started: 4 of 7)

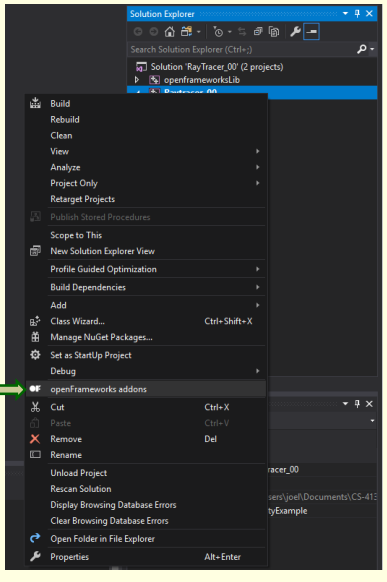
- Reopen Visual Studio
- From the “File” menu, select: “Open/Project Solution…” and use the file dialog to open your project folder, then, from within the project folder, open “emptyExample.sln” (note: if your system is set to “Hide Extensions of Known Types”, then, open Windows “File Explorer Options” and uncheck that option under the “View” tab).
- There should be two projects in the Visual Studio Solution Explorer Panel: emptyExample and openframeworksLib.
- Rename “Solution emptyExample” and the “emptyExample” project (i.e. rename to ‘RayTracer_00’).
- From the “Debug” menu, select “Start Without Debugging”. This should compile, build and open a blank window with a close box.

13

OpenFrameworks (getting started: 5 of 7)

- Now, it is time to add the slider and draw something.
- Right click on your project and select “openframeworks addons”
- From the “Addons” dialog, check “ofxGui”, then click “Ok”.
- Set the build target to x64!





14

OpenFrameworks (getting started: 6 of 7)

- Now, it is time to add the slider and draw something.
- Right click on your project and select “openframeworks addons”
- From the “Addons” dialog, check “ofxGui”, then click “Ok”.
- From the Solution Explorer, open Raytracer_00/src/ofApp.cpp
- After the first include at the top of ofApp.cpp, add

```
#include "ofxGui.h"
ofxFloatSlider radius;
ofxPanel gui;
```

- Change the **`setup()`** method to:

```
void ofApp::setup() {
    gui.setup();
    // add a slider with <label>, <initial value>, <min>, <max>
    gui.add(radius.setup("radius", 140, 10, 300));
}
```

15

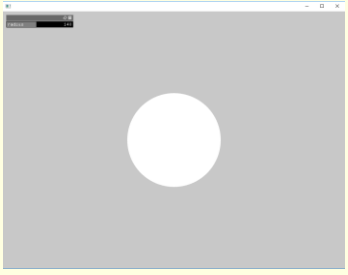
OpenFrameworks (getting started: 7 of 7)

- Change the **`draw()`** method to:

```
void ofApp::draw() {

  //Default circle resolution 10, which draws a regular Icosagon.
  ofSetCircleResolution(100);

  int centerX = ofGetWidth() / 2;
  int century = ofGetHeight() / 2;
  ofDrawCircle(centerX, centerY, radius);
  gui.draw();
}
```



16