Introduction to Computer Graphics

Prof. George Wolberg
Dept. of Computer Science
City College of New York

Course Description

- Intense introduction to computer graphics.
- Intended for advanced undergraduate and graduate students.
- Topics include:
  - OpenGL pipeline, API, GLSL shading language
  - Geometric transformations
  - 3D viewing
  - Geometric modeling, curves and surfaces
  - Shading, texture mapping, shadows

Syllabus

Week	Topic
1	Introduction, history, vector/raster graphics
2-4	OpenGL, GLSL, Qt
5-6	Geometry, 2D/3D transformations
7	Texture mapping
8	Projections, perspective
9	3D viewing
10	Midterm
11-12	Shading
13-14	Curves and surfaces

Required Text

V. Scott Gordon and John Clevenger, *Computer Graphics Programming in OpenGL with C++, 2nd* Edition, Mercury Learning and Information, 2021.

Supplementary Texts

- Supplementary Texts:
  - Edward Angel and Dave Shreiner, *Interactive Computer Graphics: A Top-Down Approach With Shader-Based OpenGL*, 6th Edition, Addison-Wesley, 2012.
  - Graham Sellers, Richard Wright, and Nicholas Haemel, *OpenGL SuperBible*, 6th Edition, Addison-Wesley, 2014.
  - Kouichi Matsuda and Rodger Lea, *WebGL Programming Guide*, Addison-Wesley, 2013.
  - Mike Bailey and Steve Cunningham, *Graphics Shaders*, 2nd Edition, CRC Press, 2012.
  - Dave Shreiner, Mason Woo, Jackie Nieder, and Tom Davis, *OpenGL Programming Guide,* 8th Edition, Addison-Wesley, 2013.
    - The definitive OpenGL programming reference

Grading

- The final grade is computed as follows:
  - Midterm exam: 25%
  - Final exam: 25%
  - Homework programming assignments: 50%
- Substantial programming assignments are due every three weeks.
- Proficiency in C/C++ is expected.
- Prereqs: CSc 221

Contact Information

- Prof. Wolberg
  - Email: wolberg@ccny.cuny.edu
- Teaching Assistant (TA): Siavash Zokai
  - Email: ccny.cs472@gmail.com
- See class web page for all class info such as office hours, homework and source code:

  www-cs.ccny.cuny.edu/~wolberg/cs472

Objectives

- Broad introduction to Computer Graphics
  - Software
  - Hardware
  - Applications
- Top-down approach
- Shader-Based
- Programs in C/C++ will be assigned to reinforce understanding of the material

Prerequisites

- Good programming skills in C (or C++)
- Data structures
  - including stacks, queues, trees, and recursion
- Geometry
- Simple linear algebra

OpenGL Resources

- Can run OpenGL on any system
  - Desktop OpenGL on Windows, Mac, Linux
  - OpenGL ES on mobile platforms: iOS, Android
- Get Qt from www.qt.io/download-open-source
  - Graphical user interface toolkit for all platforms
  - Adds sliders, pushbuttons, advanced widgets to GUI
- www.opengl.org
  - Standards documents and sample code
- www.opengl-tutorial.org
  - Informative tutorials on basic and intermediate topics
- www.khronos.org

What is Computer Graphics?

Prof. George Wolberg
Dept. of Computer Science
City College or New York

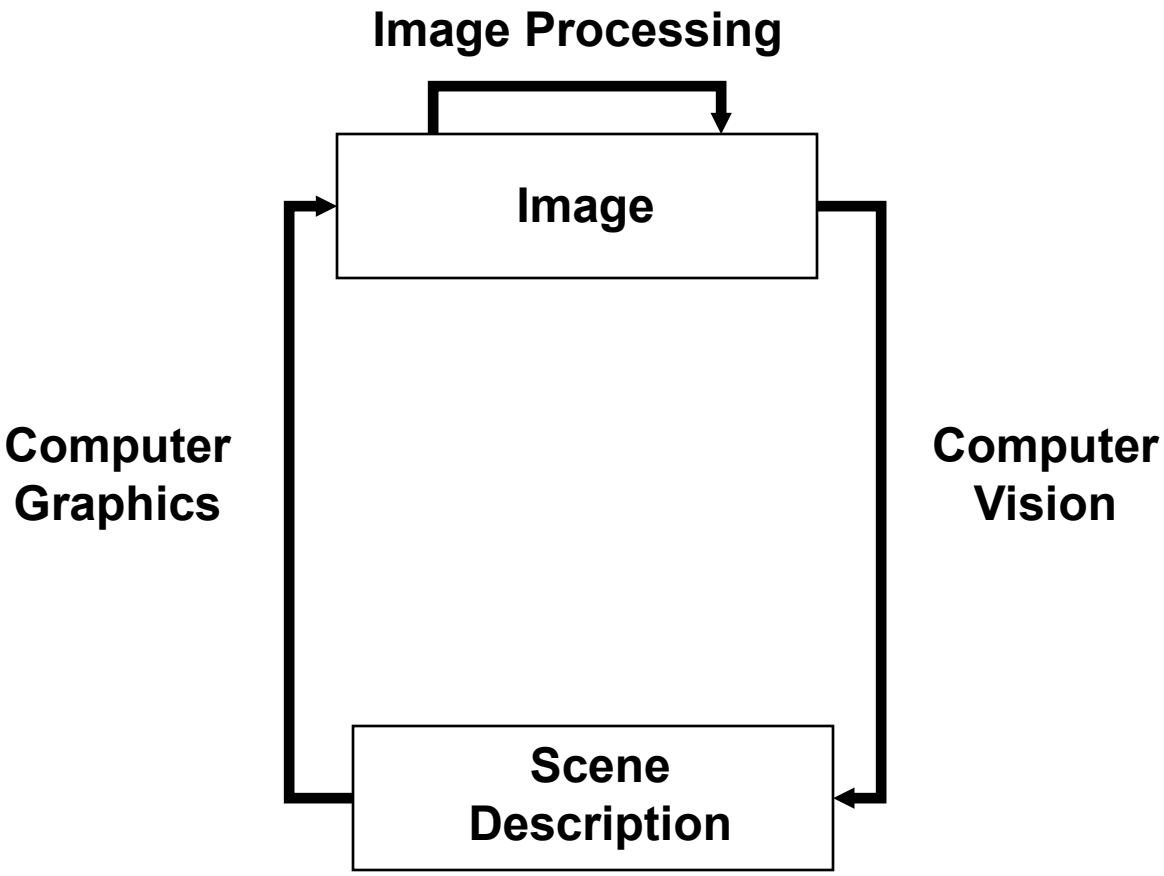
Objectives

- In this lecture, we explore what computer graphics is about and survey some application areas
- But we start with a historical introduction

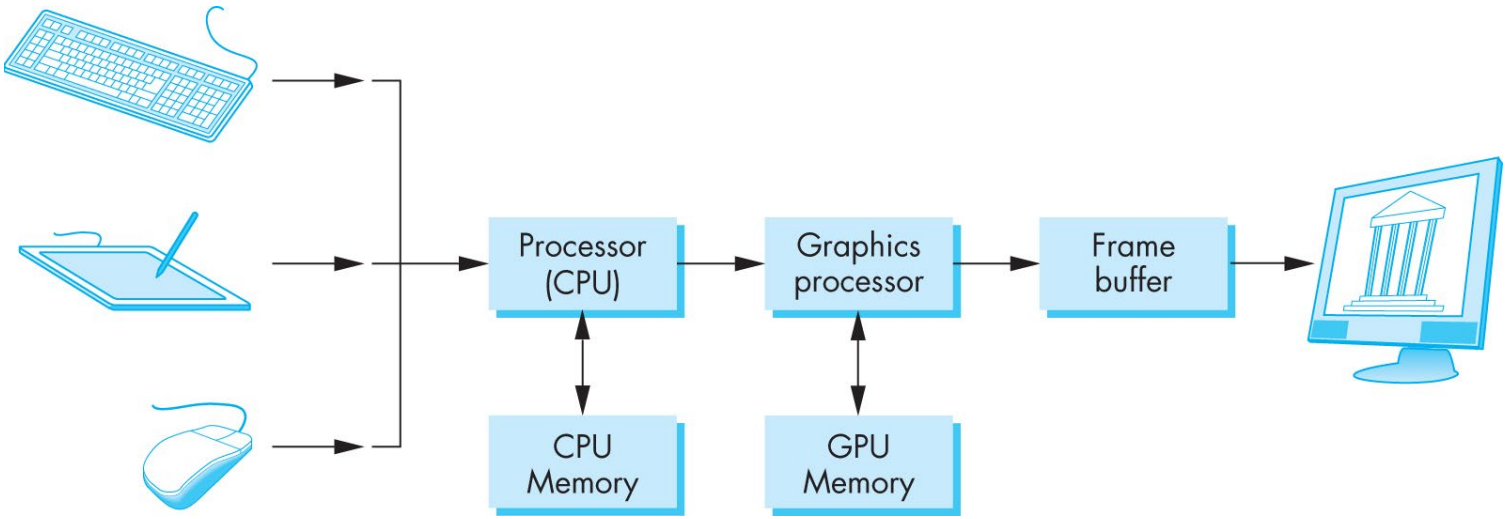
Computer Graphics

- *Computer graphics* deals with all aspects of creating images with a computer
  - Hardware
    - PC with discrete graphics (GPU) for modeling and rendering
  - Software
    - Autodesk Maya, 3D Studio Max, Houdini, and Lightwave for modeling and rendering; they are built on top of OpenGL
  - Applications
    - Detailed modeling for photo-realistic rendering and animation

Related Fields



Basic Graphics System



Input devices

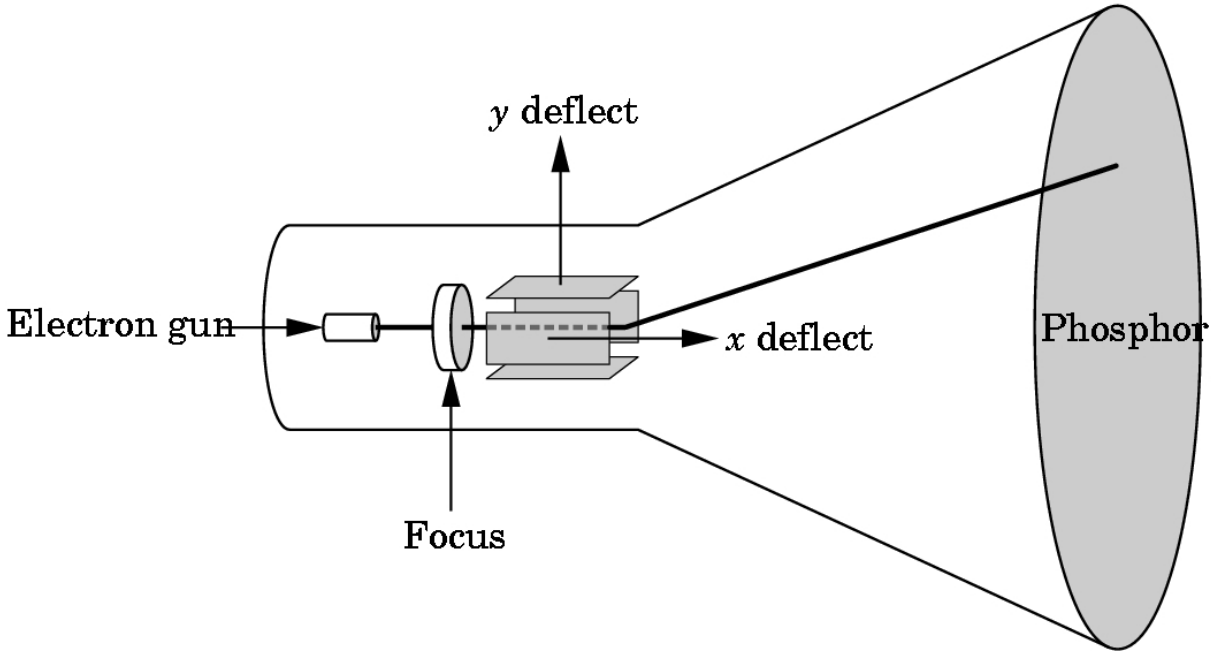
Output device

Image formed in frame buffer

Computer Graphics: 1950-1960

- Computer graphics goes back to the earliest days of computing
  - Strip charts
  - Pen plotters
  - Simple displays using A/D converters to go from computer to calligraphic CRT
- Cost of refresh for CRT too high
  - Computers slow, expensive, unreliable

CRT

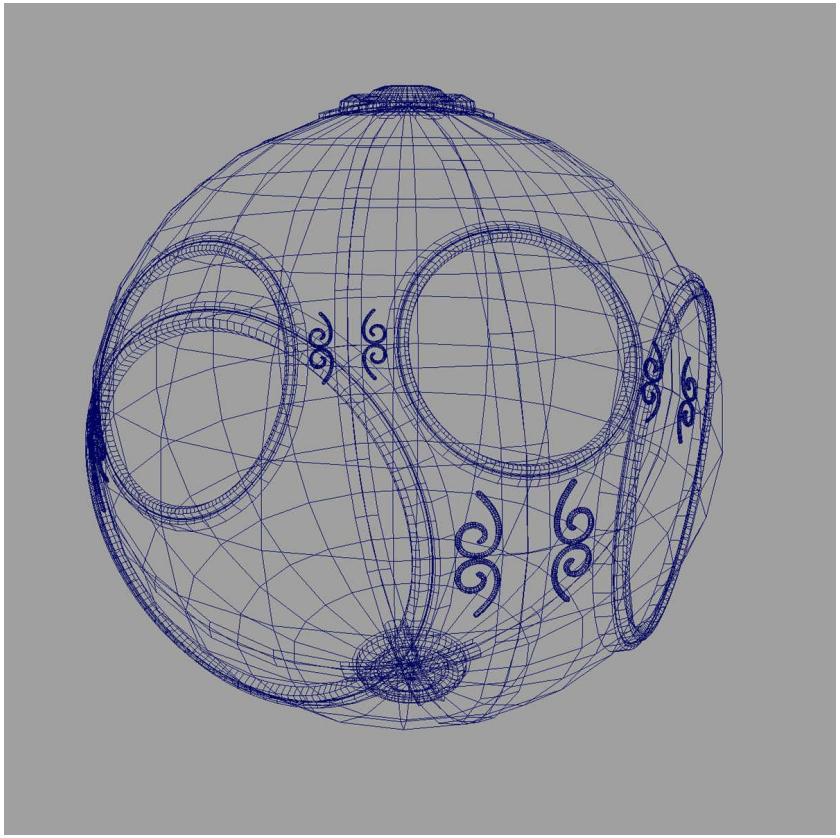


Can be used as a line-drawing device (vector graphics) or to display contents of frame buffer (raster graphics)

Computer Graphics: 1960-1970

- *Wireframe* graphics
  - Draw only lines
- Sketchpad
- Display Processors
- Storage tube

wireframe representation
of sun object

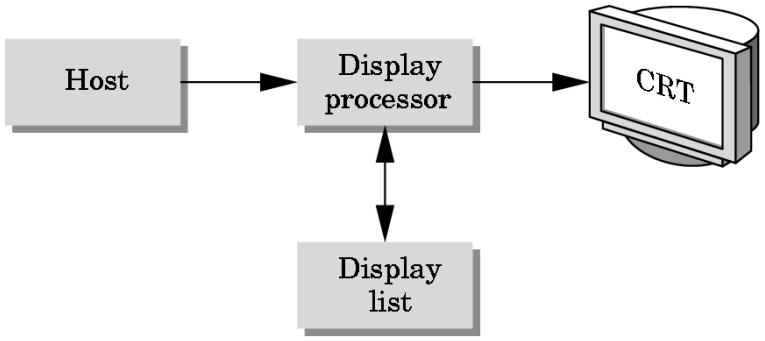


Project Sketchpad

- Ivan Sutherland's PhD thesis at MIT
  - Recognized the potential of man-machine interaction
  - Loop
    - Display something
    - User moves light pen
    - Computer generates new display
  - Sutherland also created many of the now common algorithms for computer graphics

Display Processor

- Rather than have host computer try to refresh display use a special purpose computer called a *display processor* (DPU)



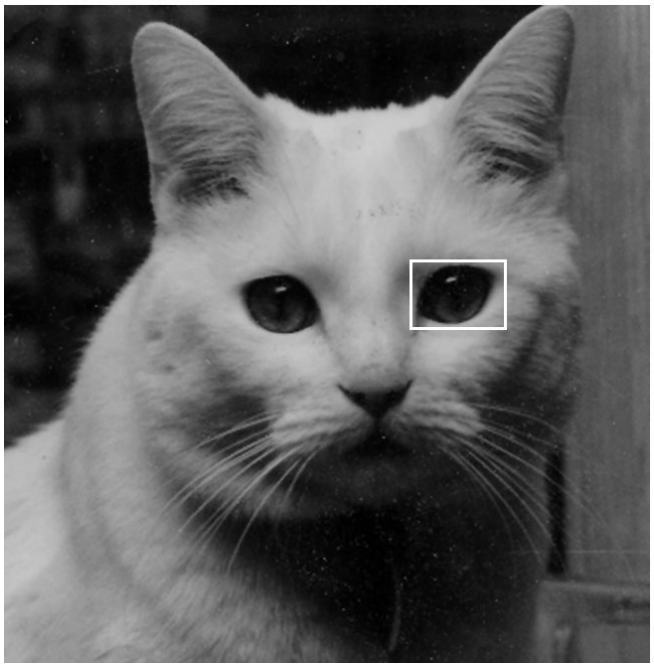
- Graphics stored in display list (display file) on display processor
- Host *compiles* display list and sends to DPU

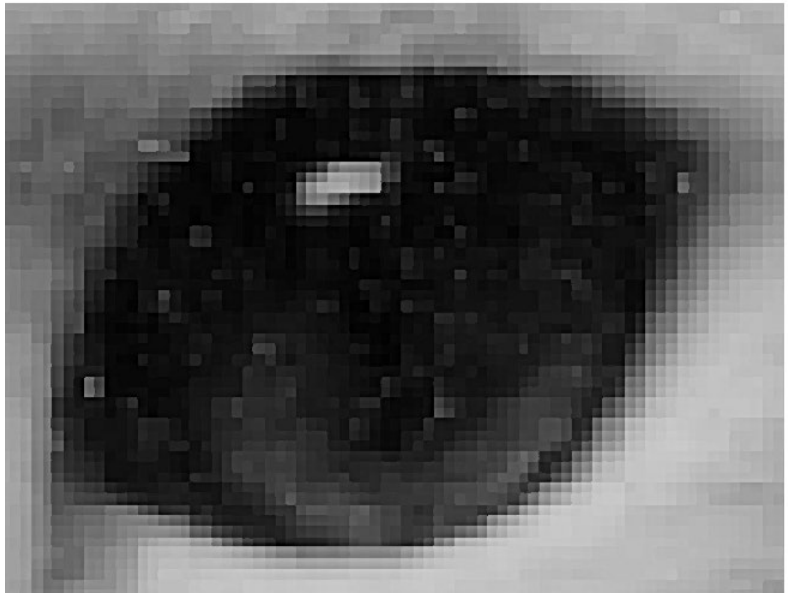
Computer Graphics: 1970-1980

- Raster Graphics
- Beginning of graphics standards
  - IFIPS
    - GKS: European effort
      - Becomes ISO 2D standard
    - Core: North American effort
      - 3D but fails to become ISO standard
    - Everyone loves a standard but there were too many!
- Workstations and PCs

Raster Graphics

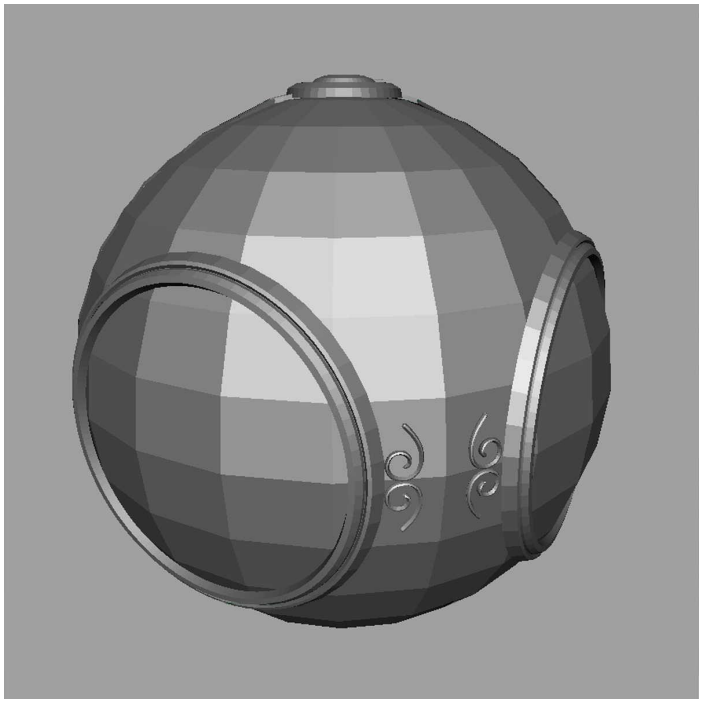
- Image produced as an array (the *raster*) of picture elements (*pixels*) in the *frame buffer*





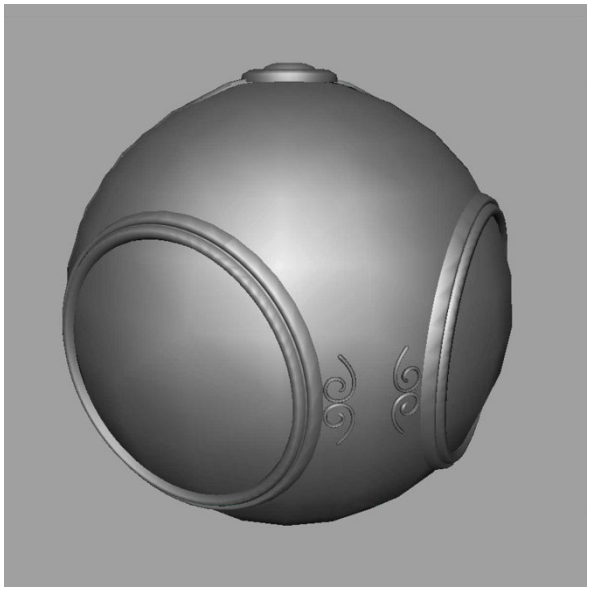
Raster Graphics

- Allow us to go from lines and wireframes to filled polygons



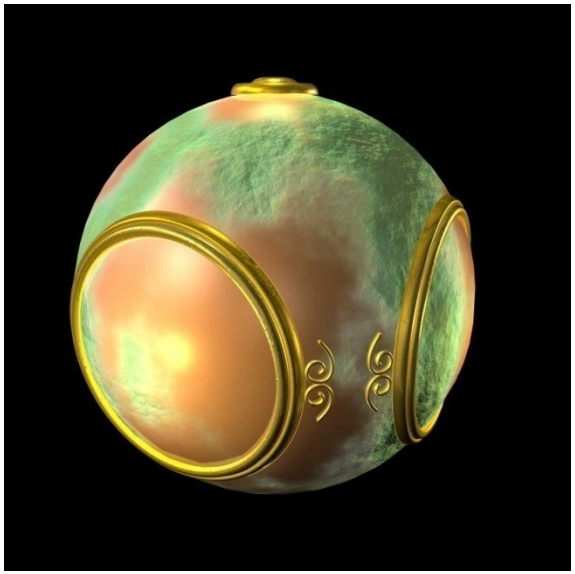
Computer Graphics: 1980-1990

Realism comes to computer graphics



smooth shading

environment mapping



bump mapping

Computer Graphics: 1980-1990

- Special purpose hardware
  - Silicon Graphics geometry engine
    - VLSI implementation of graphics pipeline
- Industry-based standards
  - PHIGS
  - RenderMan
- Networked graphics: X Window System
- Human-Computer Interface (HCI)

Computer Graphics: 1990-2000

- OpenGL API
- Completely computer-generated feature-length movies (Toy Story) are successful
- New hardware capabilities
  - Texture mapping
  - Blending
  - Accumulation, stencil buffers

Computer Graphics: 2000-2010

- Photorealism
- Graphics cards for PCs dominate market
  - Nvidia, ATI
- Game boxes and game players determine direction of market
- Computer graphics routine in movie industry: Maya, 3DS Max, Lightwave
- Programmable pipelines

Computer Graphics: 2010-2020

- Xbox, Playstation
  - Realistic rendering, animation
- Kinect sensor
  - Gesture recognition
- Touchscreen interfaces
  - Phones, tablets, Windows 10
- 3D scanning and printing
  - Editing tools for rapid prototyping 3D models
- Virtual reality
  - Oculus Rift, Samsung Gear VR, Google Cardboard

Getting Started

Prof. George Wolberg
Dept. of Computer Science
City College of New York

Languages and Libraries

- Modern graphics programming is done using a graphics library
  - Programmer invokes functions in a set of predefined libraries that provide support for lower-level graphical operations
  - Most common library for platform-independent graphics programming is OpenGL (Open Graphics Library)
  - OpenGL is based on the C language
  - We will use the library with C++

C++

- C++ is a general-purpose programming language that first appeared in the mid-1980s.
  - It is compiled to native machine code and its design makes it an excellent choice for systems that require high performance, such as 3D graphics computing.
  - Another advantage: the OpenGL call library is based on C.
- C++ development environments:
  - Microsoft Visual Studio (on Windows)
  - Xcode (on MacOS)

OpenGL / GLSL

- Revised and extended regularly
  - 1992: version 1.0 of OpenGL first appears
  - 2004: version 2.0 introduced the OpenGL Shading Language (GLSL), allowing “shader programs” to be installed and run directly in graphics pipeline stages
  - 2009: version 3.1 removed a large number of features that had been deprecated, to enforce the use of shader programming as opposed to earlier approaches (referred to as “immediate mode”)
  - 2010: version 4.0 added a tessellation stage to the programmable pipeline

Window Management

- OpenGL doesn't draw to a screen; it renders to a frame buffer
  - It is the job of the machine to draw the contents of the frame buffer to a window on the screen.
- Various libraries support drawing frame buffer to screen
  - GLUT was a historically popular option; it's now deprecated
  - freeglut is a modernized extension
  - GLFW (OpenGL Framework) is a popular and used by book
    - Has built-in support for Windows, Mac, and Linux
  - We will use Qt
    - It is an advanced C++-based 3D widget toolkit for creating GUIs using one code base that can be deployed on Windows, Mac, and Linux

Math Library

- 3D graphics makes heavy use of vector and matrix algebra
- Eigen and vmath are popular math libraries
- OpenGL Mathematics (GLM) is the most popular
  - It is a header-only C++ library
  - Uses the same naming conventions as those in GLSL
  - Contains utility classes for creating and using 3D graphics structures, such as perspective and look-at matrices.
  - We will use built-in functions in Qt

Texture Management

- Will use texture loading library for reading images for texture mapping
  - FreeImage, DevIL, OpenGL Image (GLI), and Glraw
- Book uses Simple OpenGL Image Loader (SOIL2).