

Course Topics

- Modelling of objects and camera
- 2-D and 3-D transformations
- OpenGL
- Shading: lighting, reflection, etc.
- Clipping
- Object representation: splines

Prerequisites

You should be familiar with the following:

- Programming: basic C++
- Data structures: stacks, lists, trees, etc.
- Math: matrix operations, functions, trigonometry

What is Computer Graphics?

- Considers all aspects of producing images or videos using a computer.
- Generally starts with a (mathematical) model of the objects to display.
- Algorithms to convert the model into an image.

Applications

- Visualization
- Entertainment (games, movies)
- Design (CAD/CAM)
- GIS
- Simulations
- Art
- Virtual Reality

Graphics System

Major elements:

- Input devices
- Processing Units (CPU and GPU)
- Memory
- Framebuffer
- Output devices

Input Devices

Some examples:

- Traditional: keyboard, mouse, joystick
- Touch screen
- 3D input: Leap, Kinect, VR controllers, etc.

Output Devices

- Traditionally cathode-ray tube (CRT)
- Now LED, LCD, etc.
- 2D grid of pixels, each one can be independently controlled (raster)
- Projection systems: mostly raster devices
- Stereo: produce two distinct images (e.g. different polarization or different screens)

Processing Units

- In the past: central processing unit (CPU) performs all computations
- Now special-purpose graphics processing units (GPUs) carry out specific graphics functions
- Modern GPUs are highly parallelized and programmable.
- GPU has access to the framebuffer
- GPUs may have access to its own memory.

Framebuffer

- Framebuffer: memory representing the 2D grid of pixels
- Resolution: number of pixels in framebuffer
- Depth/precision: number of bits used for each pixel
- Common: 24-bit RGB, possibly an additional “alpha” value
- There can be multiple framebuffers. e.g. double buffering
- Rasterization or scan conversion: process of converting geometric objects to pixel colours and locations in framebuffer.
- One of the framebuffers correspond to the current display.

Graphics Programming (OpenGL)

We will discuss some of the following aspects:

- General processing, basic functions
- Pipelining
- Coordinate systems
- Geometric primitives
- Colours
- Viewing
- Control functions

Geometric Objects

Some examples:

- Points
- Vectors
- 2D objects: lines, polygons, circles, etc.
- 3D objects
- Bezier curves, B-spline

A coordinate system is needed to specify objects

Transformations

- Translation
- Rotation
- Scaling
- Combination

Viewing

- Camera model (e.g. pinhole camera)
- Orientation of camera
- Projection
- Hidden surface removal
- Clipping

Shading

- Lighting models
- Texture mapping

Rough Workflow

- Geometric objects are transformed (translation, rotation, scaling, etc.)
- Rasterization is done to draw pixels in the framebuffer
- Framebuffer is displayed on screen

Graphics Architecture

- Geometric objects are represented by vertices:
 - coordinate systems are needed
 - different coordinate systems: world vs. camera view vs. screen, 3D vs 2D
 - properties: colour, lighting, materials
- Transformations:
 - represented by matrix multiplication
 - a sequence of transforms can be represented by matrix product

Graphics Architecture

- Clipping:
 - apply a viewing rectangle to objects
 - decide if an object is visible completely or partially
 - efficient clipping algorithms are needed (for refreshing)
- Projection:
 - project 3D objects to 2D objects (need a projection plane)
 - matrix operations
- Rasterization:
 - Convert projected 2D objects into pixel info in framebuffer

Raster-based Graphic System

- Resolution: screen size (e.g. 1280×1024)
 - Each pixel usually needs 3 bytes (24-bit RGB)
 - What is the amount of memory needed?
 - Double-buffering: at least 2 framebuffers:
 - one holds the content being displayed (front buffer)
 - one holds the content being drawn (back buffer)
- The two buffers are swapped constantly (just “references”)

Line Segment Drawing Example

- How is a line segment drawn on screen?
- Line segments are specified by two endpoints
- Need to decide which pixels to turn on
- Jaggedness? Aliasing and anti-aliasing
- Modern graphics libraries take care of this automatically