COMP 3490 – Computer Graphics 1

Course Description

Calendar entry

An introductory course in computer graphics including topics such as raster graphics, two and three dimensional transforms, and simple rendering. Prerequisite: [(COMP 2150 or ECE 3740) or ((COMP 2140 or the former COMP 2061) and 3 credit hours of MATH courses at the 2000 level)] and [one of MATH 1220, MATH 1300 (B), MATH 1301 (B), MATH 1310 (B), MATH 1210 (B), or MATH 1211 (B)] and [one of MATH 1230, MATH 1500 (B), MATH 1501 (B), MATH 1510 (B), the former MATH 1520 (B), or MATH 1524 (B)].

General Course Description

This course introduces some of the foundational mathematics, algorithms, and implementations for 2D and 3D computer graphics. This broad approach to the concepts of computer graphics will make it possible for students who complete the course to understand the behaviour of 2D and 3D graphics engines and APIs from data to pixels.

Detailed Prerequisites

Before entering this course, a student should be able to:

- Evaluate and plot simple linear, quadratic, and cubic functions.
- Apply trigonometric functions to compute relationships between angles and distances.
- Compute matrix and vector operations such as sums, determinants, and products.
- Recognize how linear transformations can be represented as matrices.
- Write programs in an object-oriented programming language.
- Implement algorithms that process ADTs such as lists, stacks, and queues.
- Write complete applications that combine multiple algorithms operating on data structures containing several different types of data; this requires "programming maturity" corresponding to a 3rd-year level.

Course Goals

By the end of this course students will:

- Implement the rasterization of simple 2D primitives.
- Apply affine transformations to 2D and 3D geometry.
- Write interactive and real-time graphical applications.
- Animate objects in a scene using a variety of strategies.

- Describe and apply techniques like texture mapping to improve the output quality of graphical scenes.
- Describe and apply the coordinate systems used to render 2D and 3D scenes.
- Describe and apply the geometry pipeline to project 3D graphics to a 2D display.

Learning Outcomes

Graphics Fundamentals

Students should be able to:

1. Identify the characteristics of basic hardware-related components of graphical output, such as pixels, frame buffers, and display-producing technology such as LCD panels.
2. Recognize the difference between bitmap and vector graphics.
3. Identify how geometry is used to represent vector objects in 2D and 3D.
4. Identify the coordinate systems that are used when transforming from geometry to the display.
5. Describe the steps of a viewing pipeline.

Primitives and Rasterization

Students should be able to:

1. Describe how geometric primitives are used in 2D to represent objects in modern graphics systems.
2. Identify basic properties of 2D primitives, such as the classification of simple vs complex polygons, and triangle winding.
3. Describe and implement rasterization algorithms, such as scan-line rendering, Bresenham's algorithm for lines, and/or parity for rendering complex polygons.
4. Recognize how and why the point-in-triangle test is the foundational calculation for rasterizing 2D geometry.

Transformations in 2D and 3D

Students should be able to:

1. Identify the set of affine transformations in 2D and 3D and their properties.
2. Apply sequences of elementary transformations to produce composite transformations in their matrix representation.
3. Define and explain the use of homogeneous coordinate systems.
4. Describe how affine transformations change the coordinate system (basis) of a geometry.
5. Identify the challenges with 3D Euler angle rotation and solutions such as axis-angle rotation and quaternions.

6. Apply 2D and 3D transformations as part of a viewing pipeline.

Animation

Students should be able to:

1. Identify the low-level principles of animation, such as frame rate and double buffering.
2. Use linear interpolation to animate different parameters of a geometric model.
3. Describe and implement simple animation systems, such as keyframe-based, kinematic, procedural, and particle systems.
4. Define and implement hierarchical models to simplify modeling, and for hierarchical animation using simple (predefined) forward kinematics.

Viewing and Projection

Students should be able to:

1. Define a synthetic camera and apply the corresponding viewing transformations to geometry.
2. Apply orthographic projection as part of a viewing pipeline.
3. Recognize the steps of the derivation of a perspective projection matrix.
4. Apply perspective projection as part of a viewing pipeline.
5. Construct and apply a view volume for 3D clipping.
6. Implement an interactive 3D scene consisting of multiple animated geometries, viewed through a synthetic camera, and displayed using perspective projection (this is the “capstone” assignment of the course).

Colour and Visibility

Students should be able to:

1. Describe how colour models affect the output produced by graphical systems and its relationship to our visual perception of colour.
2. Describe and apply the process of texture mapping, and the use of barycentric coordinates to interpolate perspective-correct texture coordinates.
3. Recognize the limitations of a graphical system based on samples, and the use of anti-aliasing techniques to reduce visual artefacts.
4. Describe the components of a Phong illumination model applied using Gouraud and Phong shading.
5. Recognize and use simple clipping and culling techniques, such as clipping to a viewport or view volume, and backface culling.
6. Describe and apply depth buffering to determine visibility in 3D scenes.