9/24/2020

Moving to CG10 and above
Theory
OGL Transformations

Geometry
Linear space
Affine space
Affine homogenous coordinates – in these coordinates all of OGL transformations can be implemented
Using matrix multiplication 4x4 by 4x4 or 4x4 by 4x1
Coordinate system
3-D coordinates
Volume

OpenGL primitives
Geometry Primitives (Vertex, Line, Polygon)
Raster Primitives

Better understanding of the Geometry Primitives
The notion of Volumes and changing Volumes (transformation)

Transformations
Projection
- Parallel – glOrtho()
- Perspective – glFrustum()

Model View
- Affine Transformations
  - Translation,
  - Rotation,
  - Scaling,
  - ~~Shearing~~

Part of the motivation is to be able (understand) how these transformations translate into matrix multiplication

Linear Vector Space
- Scalars – real numbers (complex)
- Vectors
- No-points

Scala Scalar, Scalar vector and vector by vectors operation

Scalar Operations are real number operations (addition, subtraction, multiplication)

Linear Vector Space

Scalar Operations

1) Scalar Operations are real number operations (addition, subtraction, multiplication)

Vectors

Direction
Magnitude
No – location

Vector by Vector Operations

2) Addition / Subtraction
   Cross product

Vector by Scalar Operations

3) Multiplication

Slide ch10-7

aV a is a scalar and V is a vector9multiplication)
When $a = -1$ $a * -V$ ➔ -V Type equation here.

$U - V == U + -V$ vector subtraction

Slide 8 Linear Vector Space

Scale and Rotate via matrix multiplication, not possible to implement translation via mutrix multiplication.

v=u+2w-3r

Slide 10

Points denote location
Point Vector operations.

Point + Vector yields a point
Point – Point yields a vector

Affine Space

Vector Space
Points
Point vector
Point to point

Slide 11

Ambiguity

Vector (1,1)
Vector(1, 1, 1)

Point (1,1) (1, 1, 1)

(1, 1) point or vector?

(2, 2, ,2, 1) – point
(2, ,2, 2, 0) Vector might denote light source at infinity

In 3D origin is at (0, 0, 0, 1) == P

Consider Q at (1, 2, 3, 1)

Moved to 4D the last component can be used to **distinguish** points and vectors
Enables to use matrix multiplication for all transformations

Affine Pace Slide 11

Definition of a line 12

Consider P_0 add aV to P_0

Assume $0 \leq a \leq \infty$
The collection of vectors $P_0 + aV$ $0 \leq a \leq \infty$ → ray
Vectors through P_0 in the direction of V

Assume -∞ $\leq a \leq \infty$
The collection of vectors $P_0 + aV$ -∞ $\leq a \leq \infty$ → Vectors through P_0 in the direction of $\pm V$

Consider P_0 = (x_0, y_0, z_0, 1) , P_1 = (x_1, y_1, z_1, 1)

$P_1 - P_0$ provides a vector call it U

$P_0 + aU$ = $P_0 + a \times (P_1 - P_0)$

When $0 \leq a \leq 1$ we get the line from P_0 to P_1

When $a = 0$ we get P_0 When a is 1 we get P_1

What is the explicit definition of the line from P_0 to P_1

The line is given by $y = mx + b$ where m and b are determined by the point coordinates

Parametric definition of the line.
Slide 13 provides the parametric equation

Every point P on the line is given by $\begin{bmatrix} \mathrm{x(a)} \\ \mathrm{y(a)} \\ \mathrm{z(a)} \end{bmatrix}$ $([x(a), y(a)]^T)$ = $\begin{bmatrix} \mathrm{a}x_0 + (1-a)x_1 \\ \mathrm{a}y_0 + (1-a)y_1 \\ \mathrm{a}z_0 + (1-a)z_1 \end{bmatrix}$

How would we draw this given P_0 = (x_0, y_0, z_0, 1) , P_1 = (x_1, y_1, z_1, 1)

glBegin(points)
loop on a from 0 to 1

glVertex(aX0 + (1-a)X1, aY0 + (1-a)Y1)

glend()
glFlush()