3D Graphics Programming

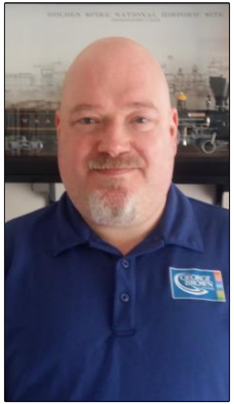
T163 - Game Programming

Fall 2020

Instructors

Alex Richard (Lectures)

- arichard6@georgebrown.ca
- Other contact info on Blackboard



Hooman Salamat (Labs)

- Hooman.Salamat@georgebrown.ca
- Other contact info on Blackboard

Evaluation System

Assessment Tool:	Description:	Outcome(s) assessed:	EES assessed:	Date / Week:	% of Final Grade:
Assignment 1	Practical coding OpenGL exercise	1, 4-8	1-11	3	**10**
Assignment 2	Practical coding OpenGL exercise	1, 4-8	1-11	5	**10**
Assignment 3	Practical coding OpenGL exercise	1, 4-8	1-11	7	**10**
Assignment 4	Practical coding OpenGL exercise	1, 4-8	1-11	11	**10**
Assignment 5	Practical coding OpenGL exercise	1, 4-8	1-11	13	**10**
Midterm Exam	Test on code and theory	2, 3, 6	2, 4-7, 11	7	**30**
Project	Practical coding OpenGL project	1, 4-8	1-11	15	**20**
				TOTAL:	100%

Course Outcomes

1. Create various 3D programs using OpenGL
2. Explain the basic concepts of 3D programming
3. Explain the fixed and programmable graphical pipelines
4. Manipulate and animate 3D objects to produce games
5. Apply textures and lighting to 3D objects to produce realistic and/or stylized effects
6. Apply special effects to enhance the visual quality of 3D scenes
7. Use primitive 3D objects as well as complex models to produce games
8. Format all deliverables to comply with Canadian laws and policies

Books 📖

Interactive Computer Graphics - A top-down approach with shader-based OpenGL - 6th edition

By: Edward Angel & Dave Shreiner

ISBN: 978-0-13-254523-5

OpenGL 4.0 Shading Language Cookbook

By: David Wolff

ISBN: 978-1-849514-76-7

Blackboard

- You will submit all classwork, i.e. labs, assignments, etc. via the related link on Blackboard
  - Could be in Assignments folder or Week folder for labs

- This course is delivered online
  - Follow the Blackboard Collaborate link in the Content page
  - Lectures and labs will use the Course Room
  - Don't forget to click the small button with three lines to access video recordings from BBCU main page
    - You can also widen the search to find older videos

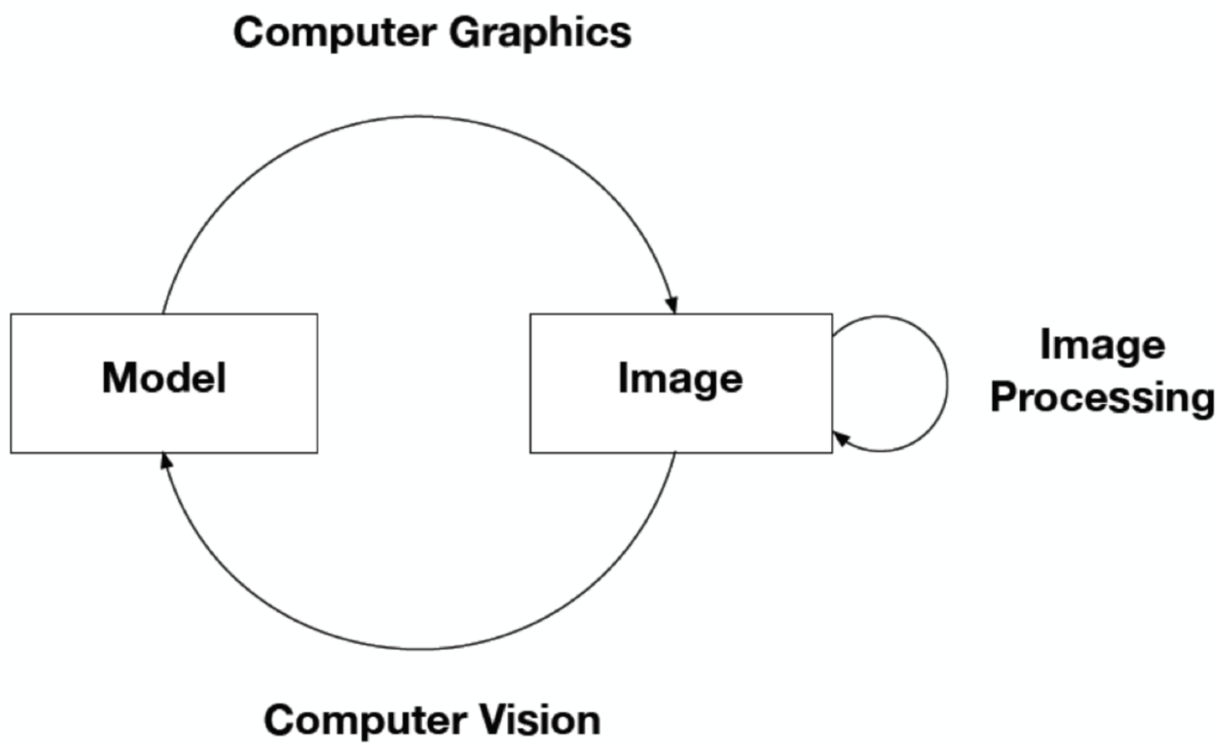
Week 1

Intro to OpenGL & Useful Tools

Framework Options (GLFW/GLUT/SDL)

Math Review

What is Computer Graphics?



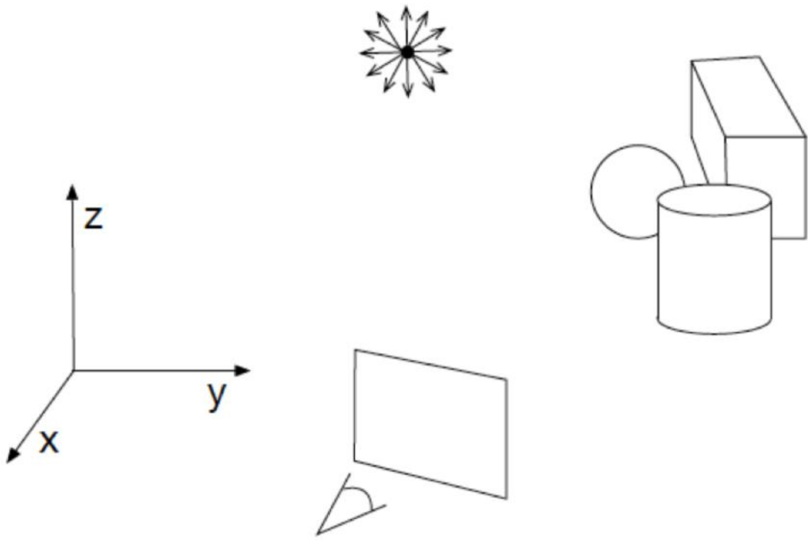
Applications?

- Display of information
- Design
- Games
- Simulation
- Animation
- User Interfaces

Process

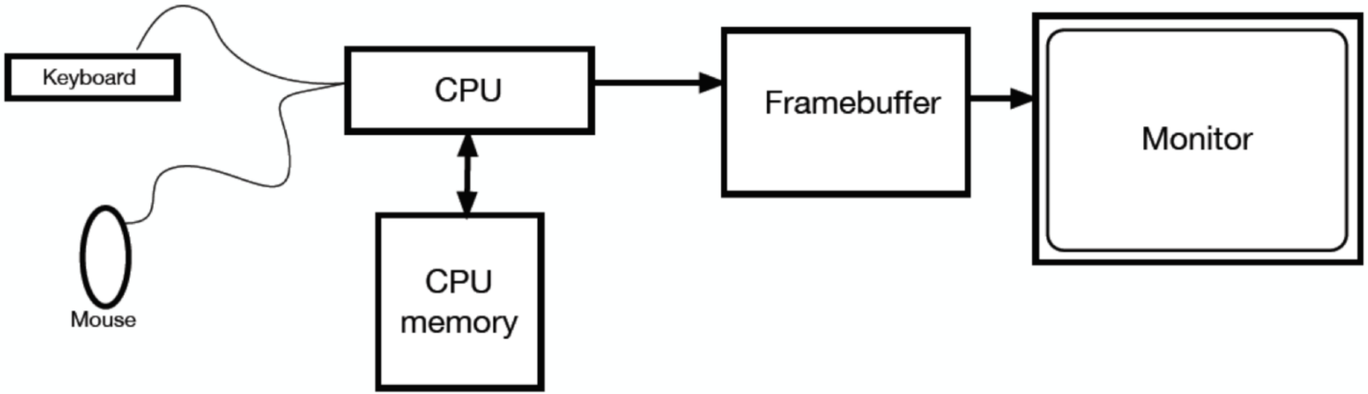
<input>: Given 3D model, material properties, eye, camera, lights

<output>: Generate 2D image



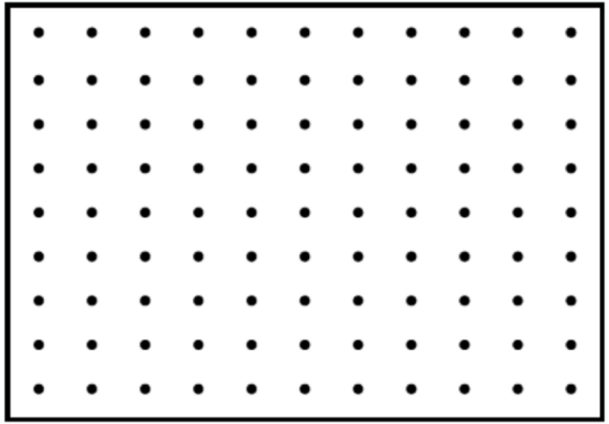
Early Hardware

❖ CPU does all the work



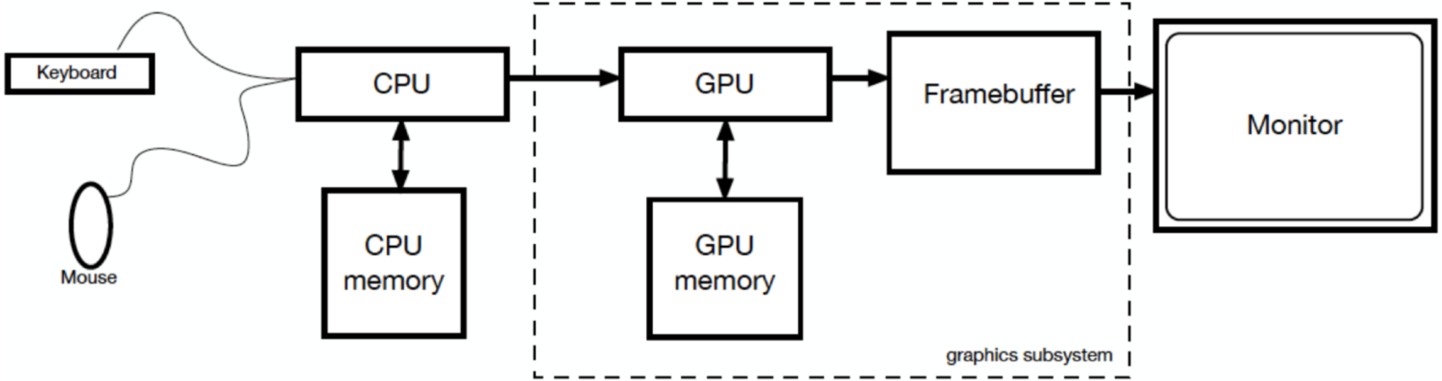
What is a Frame Buffer?

- 2D array of colorful pixels
- Intensity and color (R, G, B, A)
- Number of pixels (resolution)
- Bits per pixel: 1, 8, 24, 36

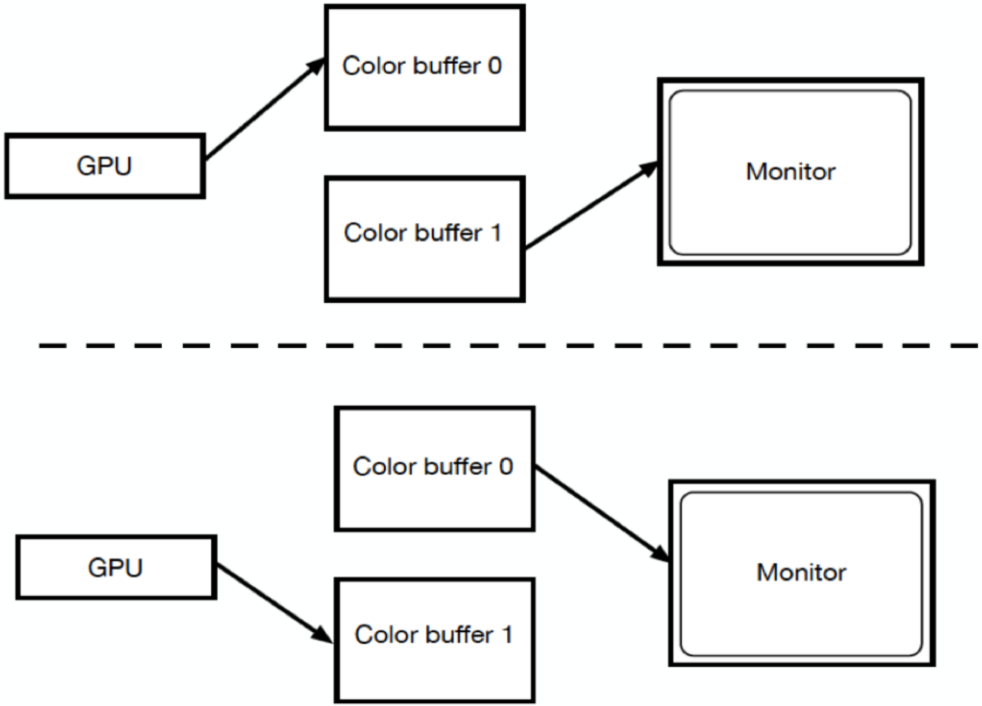


Modern Hardware

❖ Enter the GPU!



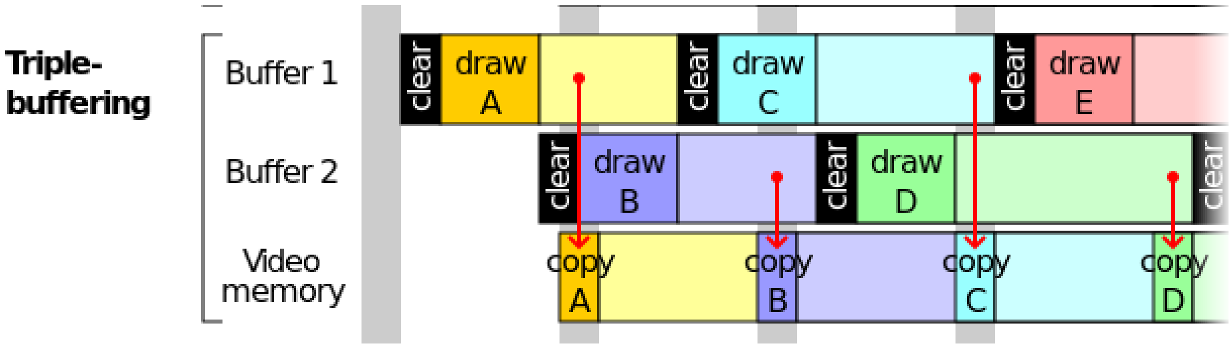
Double Buffering



Triple Buffering

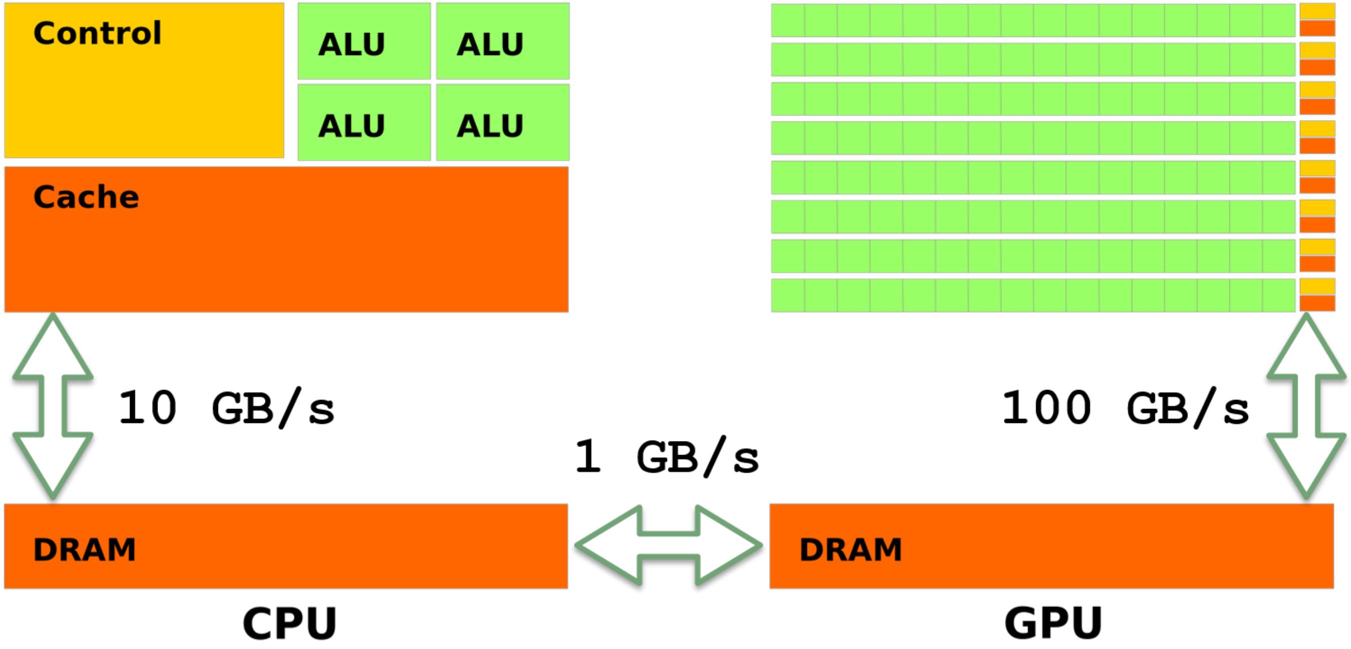
Note: This was a joke from the former instructor who doesn't work here anymore…

Triple Buffering



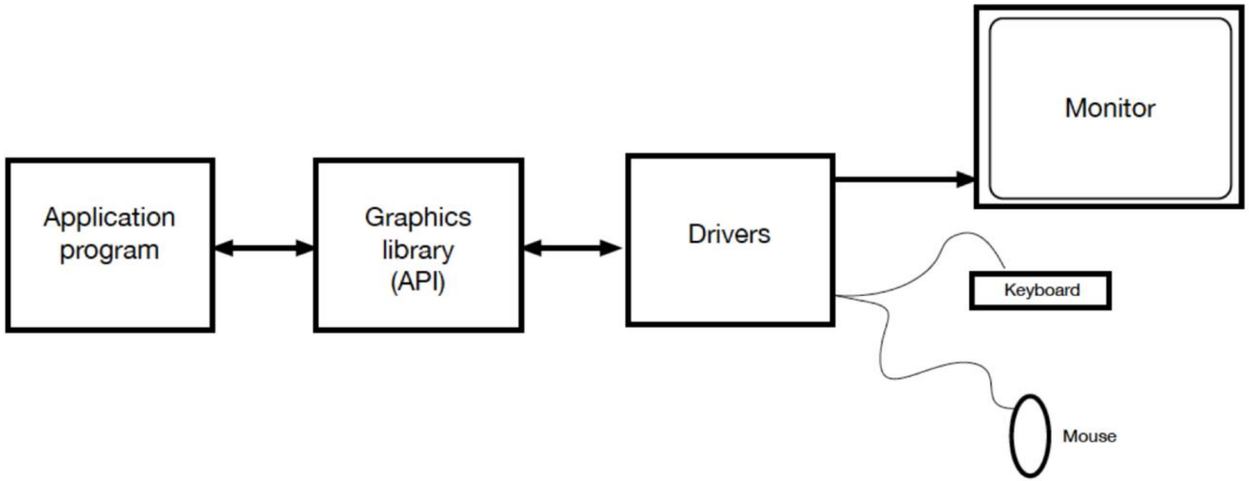
- ❖ Higher framerate potential
- ❖ Requires extra GPU memory

CPU vs GPU


Control
ALU
ALU
ALU
ALU
Cache
10 GB/s
100 GB/s
1 GB/s
DRAM
DRAM
CPU
GPU

GPU

- ❖ Graphical Processing Unit
- ❖ API:



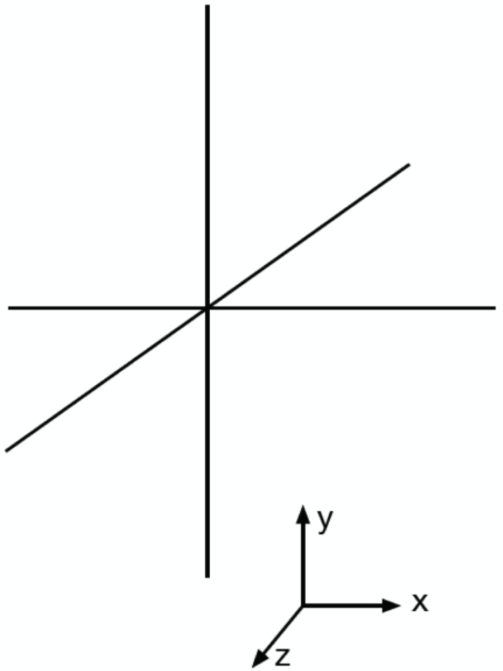
GPU

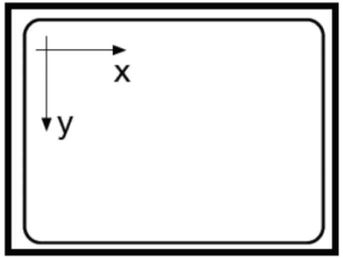
- API Major Tasks
  - Specify objects to be viewed
  - Specify properties of these objects
  - Specify how these objects to be viewed

Coordinate Systems

- Model/Object coordinate system
  - Where you define the object
- World coordinate system
  - Where objects are placed relative to each other (2D or 3D)
- Screen coordinate system
  - Device specific coordinates

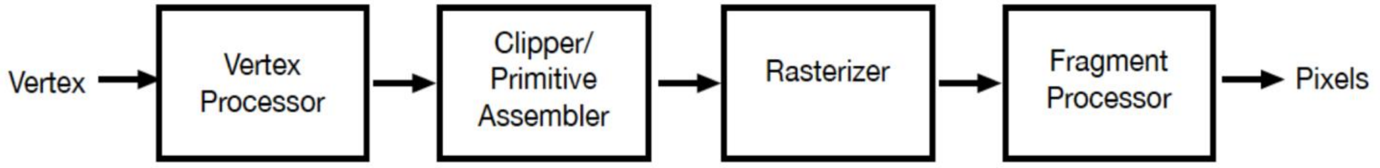
Coordinate Systems





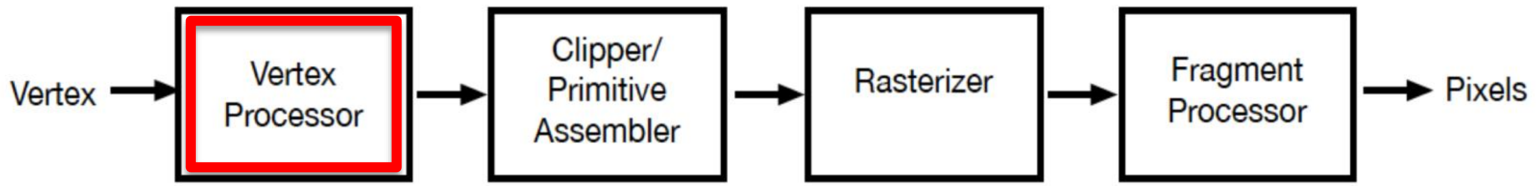
Graphics Pipeline

- World to screen…



- Transform and project graphics primitives onto projection screen
- Determine what's inside (clipping)
- Determine what's visible
- Break down into pixels
- Shade approximately

Graphics Pipeline



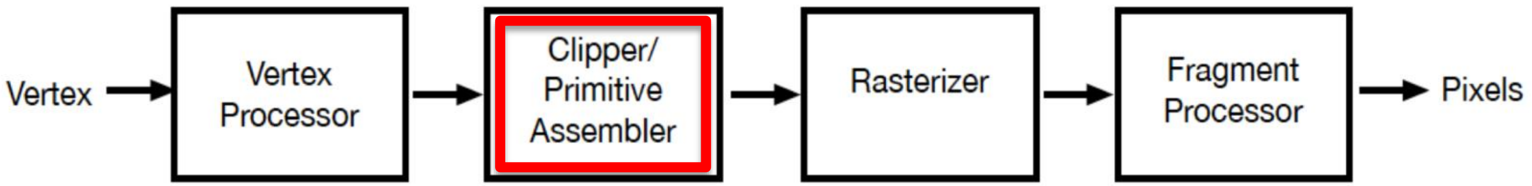
- Vertex Specification
  - VAOs (Vertex Array Objects)
  - VBOs (Vertex Buffer Objects)

- Vertex Shader (programmable)
- Tessellation (programmable)
- Geometry Shader (programmable)

Shaders

- Vertex Shaders
  - Programs that describe the traits (position, colors, and so on) of a vertex
  - The vertex is a point in 2D/3D space, such as the corner or intersection of a 2D/3D shape
- Fragment Shaders
  - Programs that deal with the per-fragment processing such as lighting
  - The fragment is a WebGL term that you can think of as a kind of pixel and contains color, depth value, texture coordinates, and more
- All shaders run on GPU!

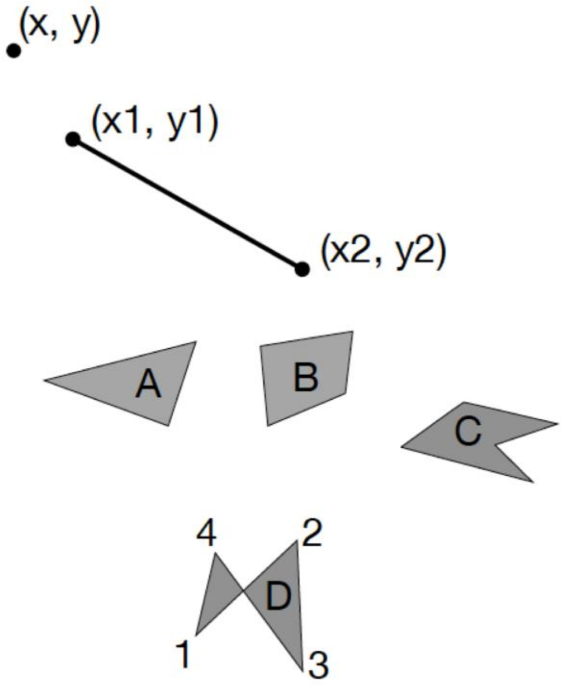
Graphics Pipeline



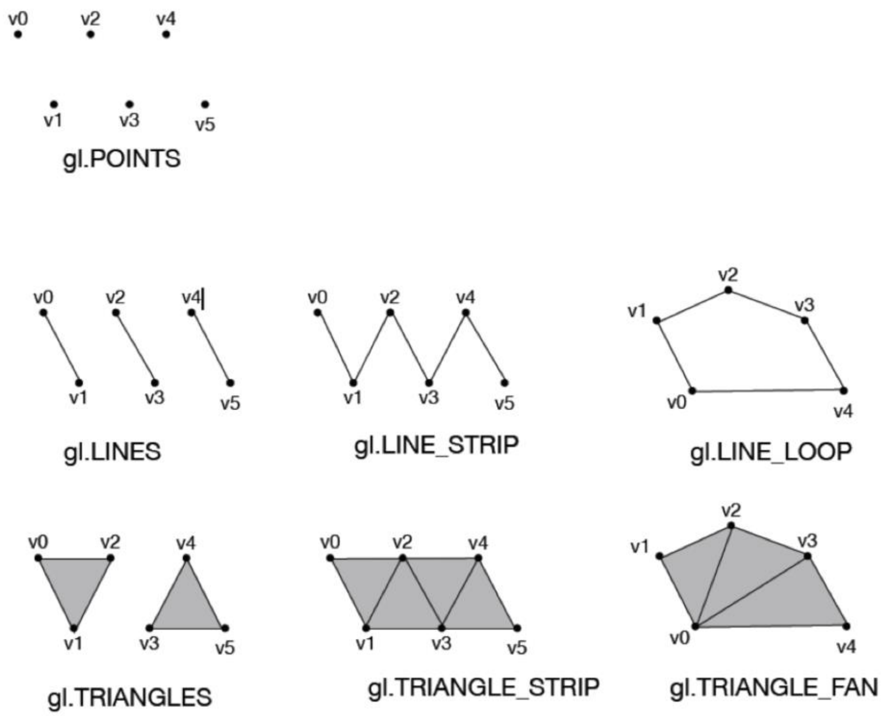
- Vertex Post-Processing
- Primitive Assembly
  - Vertices converted in to a series of primitives
  - Remove primitives that can't be seen (culling)

Primitives

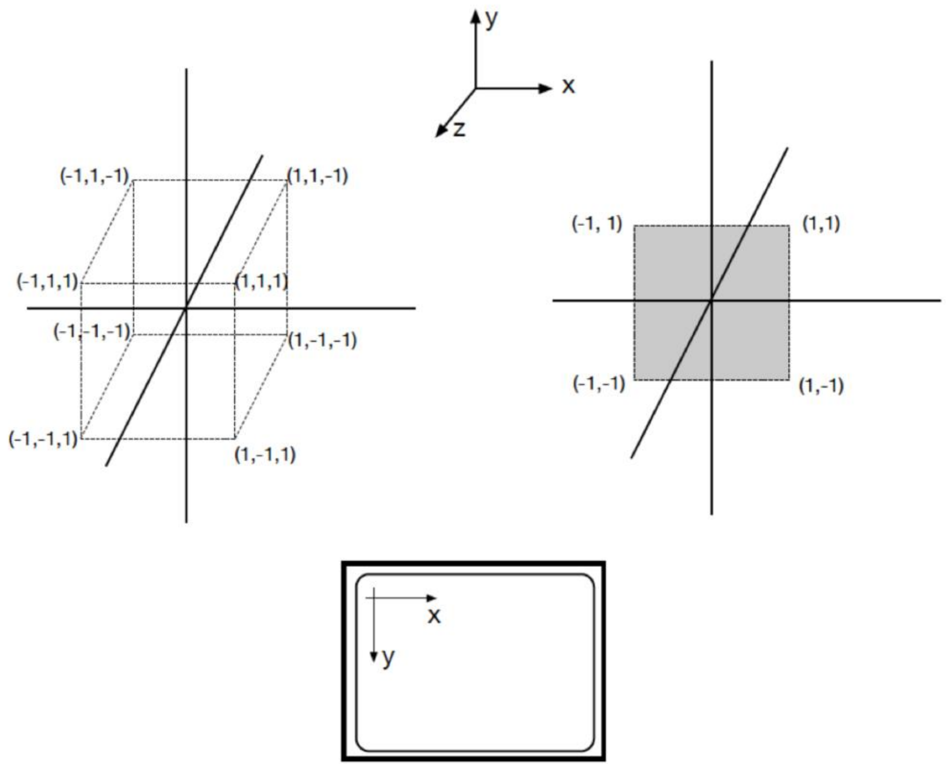
- Points
- Lines
- Triangles
- Quads
- Polygons
- Curves
- Surfaces



OpenGL Primitives

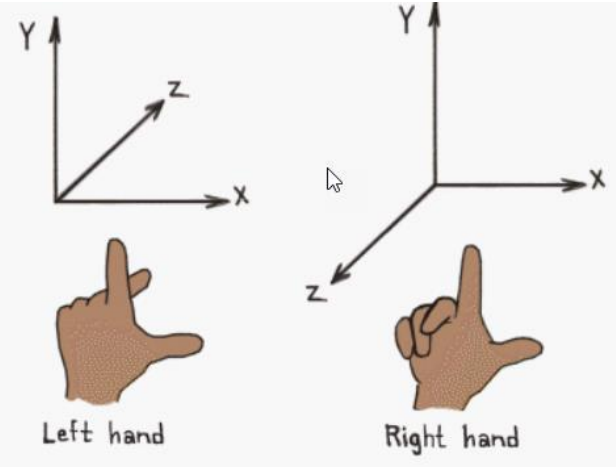


OpenGL Defaults



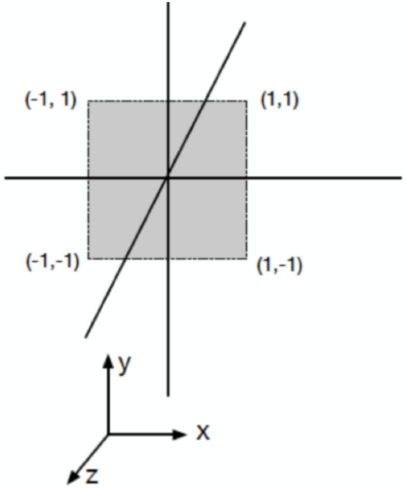
OpenGL Defaults

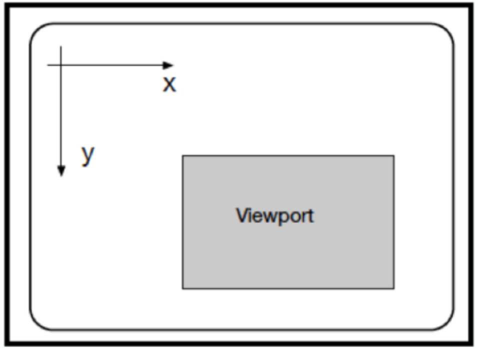
- So are OpenGL coordinates left-handed or right-handed?
- What does that even mean?



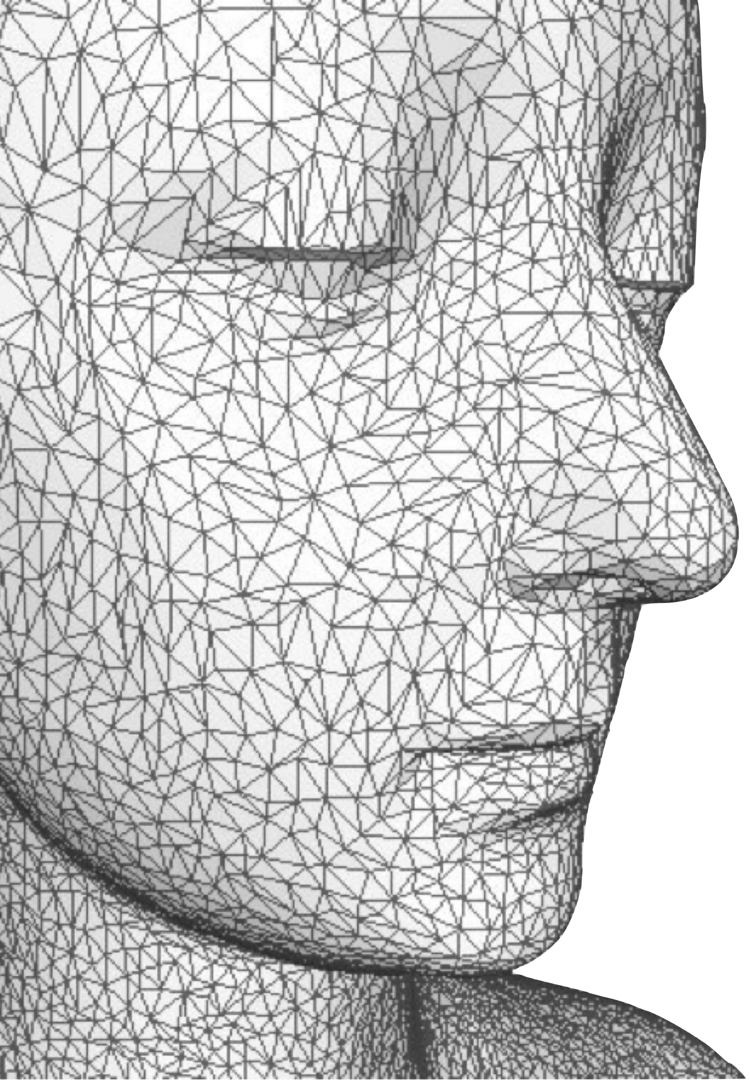
Viewport

- Specifies where on the screen the window will appear






```
gl.viewport(x, y, width, height)
```

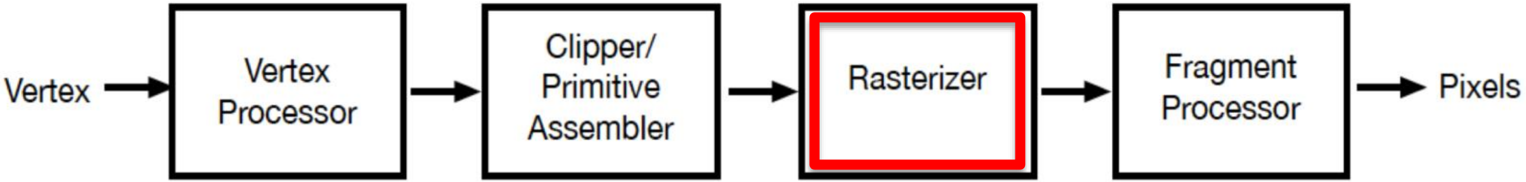


Polygonal Meshes

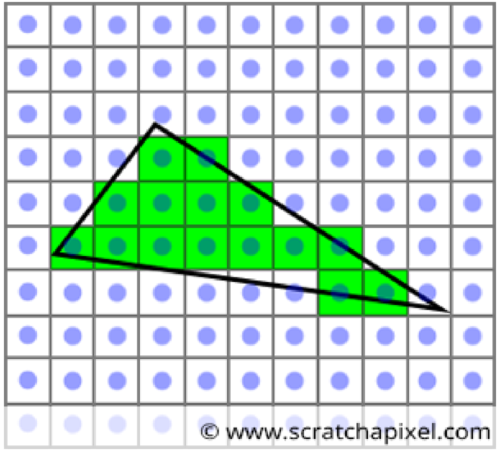
- - List of vertices
- - List of polygons
  - Each polygon has list of vertices

  - V: 1, 2, 3, 4, 5
  - P: A(1,2,5), B(2,3,5), C(3,4,5)

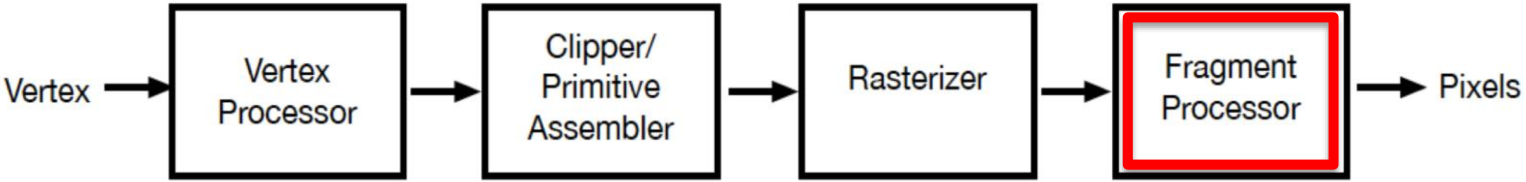
Graphics Pipeline



- Rasterization
  - Converts primitives to fragments (Pixel data)



Graphics Pipeline



- Fragment Shader (programmable)
- Per-Sample Operations
  - Depth test, color blending
  - Swap the buffers and done!

Graphics Pipeline

- The normal graphics pipeline has limitations, however
  - So We will look at some advanced rendering techniques much later

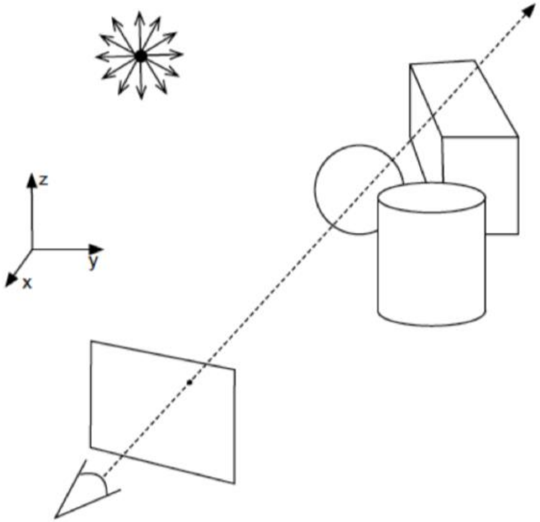
- What are some of the most impressive graphics and/or graphical details you've seen in games?

Ray Tracer

Radical obscure reference!

Ray Tracing (Screen to World)

- Shoot ray from eye through screen into world
- Intersect objects with ray
- Find closest intersection
- Do shading/lighting calculation



Week 1

Lecture Example

OpenGL Getting Started

- Open Alex's lecture example
  - Uses GLUT as a "framework"
  - Renders line strips
  - First part of Assignment 1

Week 1

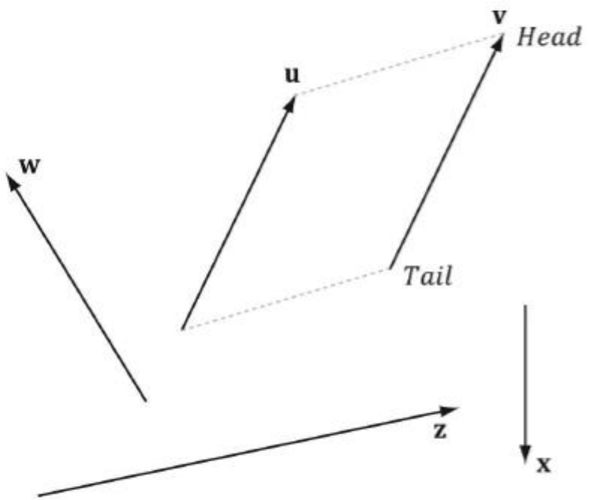
Math Review

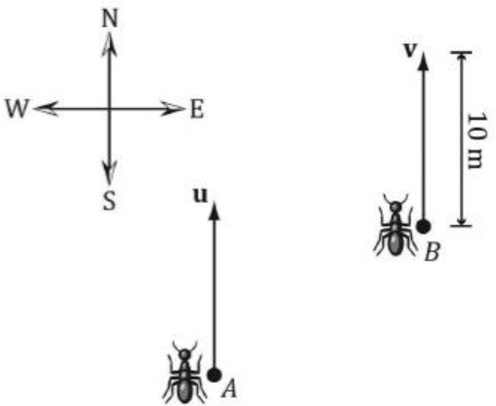
Vector Review

- Vectors play a crucial role in computer graphics
  - Also collision detection and physical simulation
  - Common components in modern video games
- Vector review
  - A vector possesses both magnitude and direction
  - Quantities that possess both magnitude and direction are called vector-valued quantities
    - E.g.: forces, displacements (change) and velocities

Vector Review

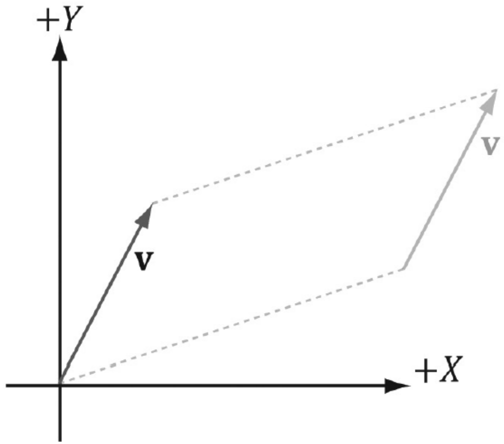
- Graphical vectors

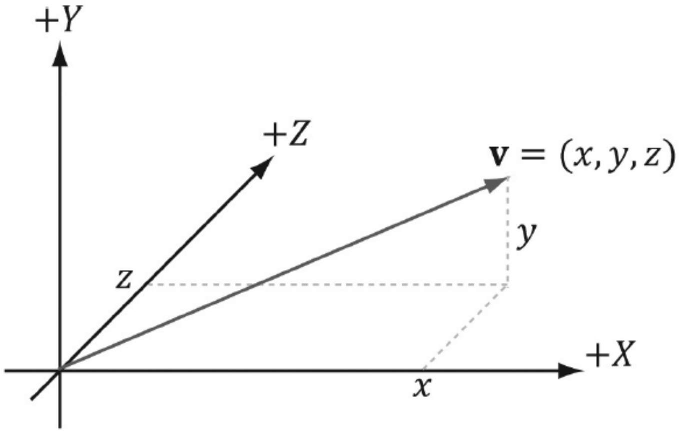




Vector Review

- 2D vs 3D





Vector Algebra

- For definitions of $u = (u_x, u_y, u_z)$ and $v = (v_x, v_y, v_z)$
  - Addition is $u + v = (u_x + v_x, u_y + v_y, u_z + v_z)$
  - Multiplication with a scalar k with u is $ku = (ku_x, ku_y, ku_z)$
  - Subtraction of $u - v = u + (-v)$ or $(u_x - v_x, u_y - v_y, u_z - v_z)$
  - Magnitude of a vector $||u||$ is: $\sqrt{x^2 + y^2 + z^2}$
  - A normalized vector is: $\hat{\mathbf{u}} = \frac{\mathbf{u}}{||\mathbf{u}||} = \left(\frac{x}{||\mathbf{u}||}, \frac{y}{||\mathbf{u}||}, \frac{z}{||\mathbf{u}||} \right)$

Vector Algebra

- Dot product
  - A form of vector multiplication that results in a scalar value
  - Sometimes referred to as the scalar product
  - Defined as: $\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y + u_z v_z$
  - Given the law of cosines, we can get: $\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}|| \, ||\mathbf{v}|| \cos\theta$
  - See example on next slide

Vector Algebra

- Example:

  - Find the angle between u and v where
    u = (1, 2, 3) and v = (−4, 0, −1)

  - Start with dot product: $\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y + u_z v_z$
  - Let's look at the entire process…

Vector Algebra

- Find the angle between u and v where u = (1, 2, 3) and v = (−4, 0, −1)

1) $\mathbf{u}\cdot\mathbf{v} = (1,\ 2,\ 3)\cdot(-4,\ 0,\ -1) = -4 - 3 = -7$

Vector Algebra

- Find the angle between u and v where u = (1, 2, 3) and v = (−4, 0, −1)

1) $\mathbf{u}\cdot\mathbf{v} = (1,\ 2,\ 3)\cdot(-4,\ 0,\ -1) = -4-3 = -7$

2) $||\mathbf{u}|| = \sqrt{1^2+2^2+3^2} = \sqrt{14}$

Vector Algebra

- Find the angle between u and v where u = (1, 2, 3) and v = (−4, 0, −1)

1) $\mathbf{u}\cdot\mathbf{v} = (1, 2, 3)\cdot(-4, 0, -1) = -4 - 3 = -7$

2) $||\mathbf{u}|| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$

3) $||\mathbf{v}|| = \sqrt{(-4)^2 + 0^2 + (-1)^2} = \sqrt{17}$

Vector Algebra

- ❖ Find the angle between u and v where u = (1, 2, 3) and v = (−4, 0, −1)

1) $\mathbf{u}\cdot\mathbf{v} = (1,\ 2,\ 3)\cdot(-4,\ 0,\ -1) = -4-3 = -7$

2) $||\mathbf{u}|| = \sqrt{1^2+2^2+3^2} = \sqrt{14}$

3) $||\mathbf{v}|| = \sqrt{(-4)^2+0^2+(-1)^2} = \sqrt{17}$

4) $\cos\theta = \dfrac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} = \dfrac{-7}{\sqrt{14}\sqrt{17}}$

Vector Algebra

❖ Find the angle between u and v where u = (1, 2, 3) and v = (−4, 0, −1)

1) $\mathbf{u} \cdot \mathbf{v} = (1,\ 2,\ 3) \cdot (-4,\ 0,\ -1) = -4 - 3 = -7$

2) $||\mathbf{u}|| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$

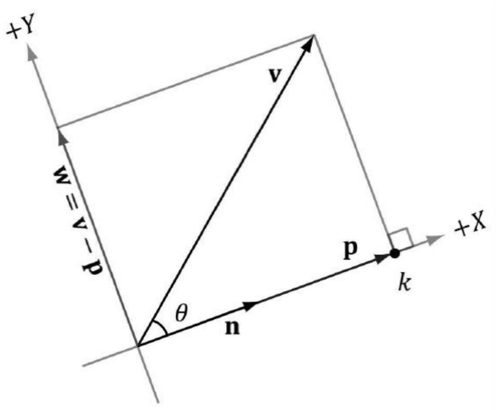
3) $||\mathbf{v}|| = \sqrt{(-4)^2 + 0^2 + (-1)^2} = \sqrt{17}$

4) $\cos\theta = \dfrac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}||\,||\mathbf{v}||} = \dfrac{-7}{\sqrt{14}\sqrt{17}}$

5) $\theta = \cos^{-1} \dfrac{-7}{\sqrt{14}\sqrt{17}} \approx 117°$

Vector Algebra

- Vector projection
  - Given the following diagram:



  - Using v and vector n, find a formula for p in terms of v and n using the dot product

Vector Algebra

- Vector projection, cont'd
  - We have a scalar k such that: $\mathbf{p} = k\mathbf{n} = (||\mathbf{v}||\cos\theta)\mathbf{n}$
  - Let's assume that n is a unit vector, so we have the following result:

$$\mathbf{p} = (||\mathbf{v}||\cos\theta)\mathbf{n} = (||\mathbf{v}|| \cdot 1\cos\theta)\mathbf{n} = (||\mathbf{v}||\,||\mathbf{n}||\cos\theta)\mathbf{n} = (\mathbf{v}\cdot\mathbf{n})\mathbf{n}$$

  - We call p the orthogonal projection of v on n, and can be expressed as:

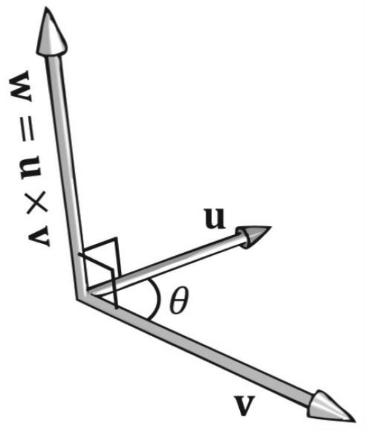
$$\mathbf{p} = \mathrm{proj}_{\mathbf{n}}(\mathbf{v})$$

  - And finally, we get: $\mathbf{p} = \mathrm{proj}_{\mathbf{n}}(\mathbf{v}) = \left(\mathbf{v}\cdot\frac{\mathbf{n}}{||\mathbf{n}||}\right)\frac{\mathbf{n}}{||\mathbf{n}||} = \frac{(\mathbf{v}\cdot\mathbf{n})}{||\mathbf{n}||^2}\mathbf{n}$

Vector Algebra

- Cross product:
  - Given u = (2, 1,3) and v = (2, 0, 0), compute w = u × v

$$\mathbf{w} = \mathbf{u} \times \mathbf{v} = (u_y v_z - u_z v_y, u_z v_x - u_x v_z, u_x v_y - u_y v_x)$$



$$\mathbf{w} = \mathbf{u} \times \mathbf{v}$$
$$= (2, 1, 3) \times (2, 0, 0)$$
$$=(1\cdot 0 - 3\cdot 0, 3\cdot 2 - 2\cdot 0, 2\cdot 0 - 1\cdot 2)$$
$$= (0, 6, -2)$$

Matrices Review

- An m × n matrix is a rectangular array of real numbers with m rows and n columns
  - The product of the number of rows and columns gives its dimensions
  - The numbers in a matrix are called elements or entries

  - Examples:

$$\mathbf{A} = \begin{bmatrix} 3.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 2 & -5 & \sqrt{2} & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{bmatrix} \quad \mathbf{u} = [u_1, u_2, u_3] \quad \mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ \sqrt{3} \\ \pi \end{bmatrix}$$

  - A is a 4 × 4 matrix, B is a 3 × 2 matrix; u is a 1 × 3 matrix and v is a 4 × 1 matrix

Matrices Review

- Matrix operations:
  - Two matrices must have the same number of rows and columns in order to be compared
  - We add two matrices by adding their corresponding elements so it only makes sense to add matrices that the same number of rows and columns
  - We multiply a scalar and a matrix by multiplying the scalar with every element in the matrix
  - We define subtraction in terms of matrix addition and scalar multiplication, e.g. $A - B = A + (-1\,B) = A + (-B)$

Matrix Operations Examples

- Matrices have some nice algebraic properties:

$$A + B = B + A$$

(Commutative Law)

$$r(A + B) = rA + rB$$

$$(r + s)A = rA + sA$$

(Distributive Law)

Matrix Operations Examples

❖ Given: $\mathbf{A}=\begin{bmatrix}1 & 5\\ -2 & 3\end{bmatrix}, \mathbf{B}=\begin{bmatrix}6 & 2\\ 5 & -8\end{bmatrix}, \mathbf{C}=\begin{bmatrix}1 & 5\\ -2 & 3\end{bmatrix}, \mathbf{D}=\begin{bmatrix}2 & 1 & -3\\ -6 & 3 & 0\end{bmatrix}$

$$\mathbf{A}+\mathbf{B}=\begin{bmatrix}1 & 5\\ -2 & 3\end{bmatrix}+\begin{bmatrix}6 & 2\\ 5 & -8\end{bmatrix}=\begin{bmatrix}1+6 & 5+2\\ -2+5 & 3+(-8)\end{bmatrix}=\begin{bmatrix}7 & 7\\ 3 & -5\end{bmatrix}$$

$$\mathbf{3D}=3\begin{bmatrix}2 & 1 & -3\\ -6 & 3 & 0\end{bmatrix}=\begin{bmatrix}3(2) & 3(1) & 3(-3)\\ 3(-6) & 3(3) & 3(0)\end{bmatrix}=\begin{bmatrix}6 & 3 & -9\\ -18 & 9 & 0\end{bmatrix}$$

$$\mathbf{A}-\mathbf{B}=\begin{bmatrix}1 & 5\\ -2 & 3\end{bmatrix}-\begin{bmatrix}6 & 2\\ 5 & -8\end{bmatrix}=\begin{bmatrix}1-6 & 5-2\\ -2-5 & 3-(-8)\end{bmatrix}=\begin{bmatrix}-5 & 3\\ -7 & 11\end{bmatrix}$$

Matrix Operations Examples

- Vector-Matrix Multiplication: $\mathbf{uA} = [x, y, z]\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = [x, y, z]\begin{bmatrix} \uparrow & \uparrow & \uparrow \\ \mathbf{A}_{*,1} & \mathbf{A}_{*,2} & \mathbf{A}_{*,3} \\ \downarrow & \downarrow & \downarrow \end{bmatrix}$

$$\begin{aligned}
\mathbf{uA} &= \begin{bmatrix} \mathbf{u}\cdot\mathbf{A}_{*,1} & \mathbf{u}\cdot\mathbf{A}_{*,2} & \mathbf{u}\cdot\mathbf{A}_{*,3} \end{bmatrix} \\
&= [xA_{11} + yA_{21} + zA_{31}, \quad xA_{12} + yA_{22} + zA_{32}, \quad xA_{13} + yA_{23} + zA_{33}] \\
&= [xA_{11}, xA_{12}, xA_{13}] + [yA_{21}, yA_{22}, yA_{23}] + [zA_{31}, zA_{32}, zA_{33}] \\
&= x[A_{11}, A_{12}, A_{13}] + y[A_{21}, A_{22}, A_{23}] + z[A_{31}, A_{32}, A_{33}] \\
&= x\mathbf{A}_{1,*} + y\mathbf{A}_{2,*} + z\mathbf{A}_{3,*}
\end{aligned}$$

$$\mathbf{uA} = x\mathbf{A}_{1,*} + y\mathbf{A}_{2,*} + z\mathbf{A}_{3,*}$$

Matrix Operations Examples

- Vector-Matrix Multiplication:

$$\mathbf{uA} = [x, y, z]\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = [x, y, z]\begin{bmatrix} \uparrow & \uparrow & \uparrow \\ \mathbf{A}_{*,1} & \mathbf{A}_{*,2} & \mathbf{A}_{*,3} \\ \downarrow & \downarrow & \downarrow \end{bmatrix}$$

$$\begin{aligned} \mathbf{uA} &= \begin{bmatrix} \mathbf{u}\cdot\mathbf{A}_{*,1} & \mathbf{u}\cdot\mathbf{A}_{*,2} & \mathbf{u}\cdot\mathbf{A}_{*,3} \end{bmatrix} \\ &= [xA_{11} + yA_{21} + zA_{31}, \quad xA_{12} + yA_{22} + zA_{32}, \quad xA_{13} + yA_{23} + zA_{33}] \\ &= [xA_{11}, xA_{12}, xA_{13}] + [yA_{21}, yA_{22}, yA_{23}] + [zA_{31}, zA_{32}, zA_{33}] \\ &= x[A_{11}, A_{12}, A_{13}] + y[A_{21}, A_{22}, A_{23}] + z[A_{31}, A_{32}, A_{33}] \\ &= x\mathbf{A}_{1,*} + y\mathbf{A}_{2,*} + z\mathbf{A}_{3,*} \end{aligned}$$

$$\mathbf{uA} = x\mathbf{A}_{1,*} + y\mathbf{A}_{2,*} + z\mathbf{A}_{3,*}$$

Matrix Operations Examples

- Given: $\mathbf{A} = \begin{bmatrix} -1 & 5 & -4 \\ 3 & 2 & 1 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & -2 & 1 \\ -1 & 2 & 3 \end{bmatrix}$

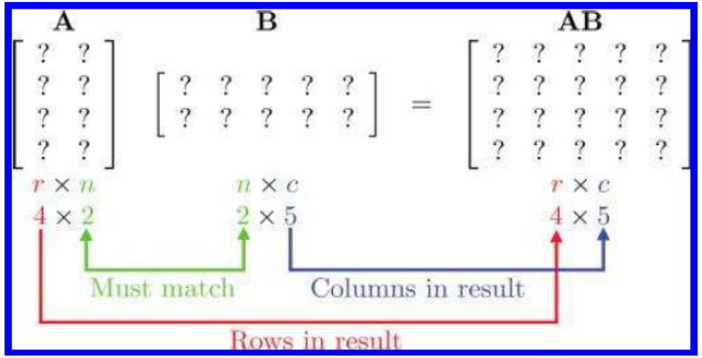
$$\mathbf{AB} = \begin{bmatrix} -1 & 5 & -4 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 0 & -2 & 1 \\ -1 & 2 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} (-1,5,-4)\cdot(2,0,-1) & (-1,5,-4)\cdot(1,-2,2) & (-1,5,-4)\cdot(0,1,3) \\ (3,2,1)\cdot(2,0,-1) & (3,2,1)\cdot(1,-2,2) & (3,2,1)\cdot(0,1,3) \end{bmatrix}$$

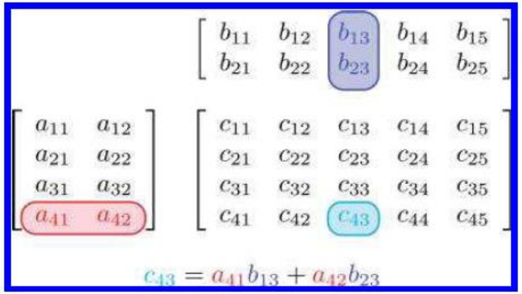
$$= \begin{bmatrix} 2 & -19 & -7 \\ 5 & 1 & 5 \end{bmatrix}$$

Matrix Operations Examples

- Clarification:



- Helps to visualize:



Matrix Operations Examples

- Matrix multiplication has some nice algebraic properties

$$A(B+C) = AB + AC$$
$$(A+B)C = AC + BC$$
$$(AB)C = A(BC)$$

Matrix Operations Examples

- Matrix Transpose
  - Interchange the rows and columns of the matrix!
  - Given these matrices:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 8 \\ 3 & 6 & -4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

  - Their matrix transpose is found by swapping rows and columns thus:

$$\mathbf{A}^T = \begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 8 & -4 \end{bmatrix}, \quad \mathbf{B}^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}, \quad \mathbf{C}^T = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$$

Matrix Operations Examples

- Transpose has some useful properties

$(A+B)^T = A^T + B^T$

$(cA)^T = cA^T$

$(AT)^T = A$

$(AB)^T = B^TA^T$

$(A-1)^T = (A^T)-1$

Matrix Operations Examples

- Matrix Identity
  - The identity matrix is a square matrix that has zeros for all elements except for ones along the main diagonal

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Example: Let $\mathbf{u} = [-1, 2]$ and let $\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Verify that $\mathbf{uI} = \mathbf{u}$

$$\mathbf{uI} = \begin{bmatrix} -1, & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} (-1,2)\cdot(1,0), & (-1,2)\cdot(0,1) \end{bmatrix} = \begin{bmatrix} -1, & 2 \end{bmatrix}$$

Matrix Operations Examples

- Matrix Minors
  - Given an n × n matrix A, the minor matrix is the (n – 1) × (n – 1) matrix found by deleting the ith row and jth column of A
  - Example: Find the minor matrices $\bar{\mathbf{A}}_{11}, \bar{\mathbf{A}}_{22}$, and $\bar{\mathbf{A}}_{13}$ of the following matrix:

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

For $\bar{\mathbf{A}}_{11}$ we eliminate the first row and first column to obtain:

$$\bar{\mathbf{A}}_{11} = \begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix}$$

For $\bar{\mathbf{A}}_{22}$ we eliminate the second row and second column to obtain:

$$\bar{\mathbf{A}}_{22} = \begin{bmatrix} A_{11} & A_{13} \\ A_{31} & A_{33} \end{bmatrix}$$

For $\bar{\mathbf{A}}_{13}$ we eliminate the first row and third column to obtain:

$$\bar{\mathbf{A}}_{13} = \begin{bmatrix} A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix}$$

Matrix Operations Examples

- Matrix Determinant
  - The determinant is a special function which inputs a square matrix and outputs a real number
  - The determinant of a square matrix A is commonly denoted by det A
  - Determinants are used to solve systems of linear equations
  - Determinants give us an explicit formula for finding the inverse of a matrix
  - Matrix A is invertible if and only if det A $\neq 0$

  - The determinant of a matrix is defined recursively:
    - The determinant of a 4×4 matrix is defined in terms of the determinant of a 3×3 matrix
    - The determinant of a 3×3 matrix is defined in terms of the determinant of a 2×2 matrix
    - The determinant of a 2×2 matrix is defined in terms of the determinant of a 1×1 matrix
    - The determinant of a 1×1 matrix $A = [A_{11}]$ is trivially defined to be $\det [A_{11}] = A_{11}$

Matrix Operations Examples

- Matrix Determinant, cont'd
  - For 2x2 matrices, we have the formula:

$$\det\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = A_{11}\det[A_{22}] - A_{12}\det[A_{21}] = A_{11}A_{22} - A_{12}A_{21}$$

  - For 3x3 matrices, we have the formula:

$$\det\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$
$$= A_{11}\det\begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix} - A_{12}\det\begin{bmatrix} A_{21} & A_{23} \\ A_{31} & A_{33} \end{bmatrix} + A_{13}\det\begin{bmatrix} A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix}$$

Matrix Operations Examples

- Matrix Determinant, cont'd
  - For 4x4 matrices, we have the formula:

$$\det\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} = A_{11}\det\begin{bmatrix} A_{22} & A_{23} & A_{24} \\ A_{32} & A_{33} & A_{34} \\ A_{42} & A_{43} & A_{44} \end{bmatrix} - A_{12}\det\begin{bmatrix} A_{21} & A_{23} & A_{24} \\ A_{31} & A_{33} & A_{34} \\ A_{41} & A_{43} & A_{44} \end{bmatrix}$$

$$+A_{13}\det\begin{bmatrix} A_{21} & A_{22} & A_{24} \\ A_{31} & A_{32} & A_{34} \\ A_{41} & A_{42} & A_{44} \end{bmatrix} - A_{14}\det\begin{bmatrix} A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \\ A_{41} & A_{42} & A_{43} \end{bmatrix}$$

Matrix Operations Examples

- Matrix Determinant, cont'd
  - Find the determinant of: $\mathbf{A} = \begin{bmatrix} 2 & -5 & 3 \\ 1 & 3 & 4 \\ -2 & 3 & 7 \end{bmatrix}$

$$\det \mathbf{A} = A_{11} \det \begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix} - A_{12} \det \begin{bmatrix} A_{21} & A_{23} \\ A_{31} & A_{33} \end{bmatrix} + A_{13} \det \begin{bmatrix} A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix}$$

$$\begin{aligned} \det \mathbf{A} &= 2\det \begin{bmatrix} 3 & 4 \\ 3 & 7 \end{bmatrix} - (-5)\det \begin{bmatrix} 1 & 4 \\ -2 & 7 \end{bmatrix} + 3\det \begin{bmatrix} 1 & 3 \\ -2 & 3 \end{bmatrix} \\ &= 2(3\cdot 7 - 4\cdot 3) + 5(1\cdot 7 - 4\cdot(-2)) + 3(1\cdot 3 - 3\cdot(-2)) \\ &= 2(9) + 5(15) + 3(9) \\ &= 18 + 75 + 27 \\ &= 120 \end{aligned}$$

Matrix Operations Examples

- Matrix Adjoint
  - With a matrix adjoint, we can find an explicit formula for computing matrix inverses
  - Given an n x n matrix A, the product $C_{ij} = (-1)^{i+j} \det \overline{\mathbf{A}}_{ij}$ is called the cofactor of A_{ij}
  - After computing C_{ij} and placing it in a matrix CA as follows:

$$\mathbf{C}_{\mathrm{A}} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{bmatrix}$$

  - The adjoint of A is a transpose of C_A which we can denote by

$$\mathbf{A}^{*} = \mathbf{C}_{\mathbf{A}}^{T}$$

Matrix Operations Examples

- Matrix Inverses
  - Only square matrices have inverses
    - Not every square matrix has one though
  - The inverse of an $n \times n$ matrix M is an $n \times n$ matrix denoted by M^{-1}
  - The inverse is unique when it exists
  - Multiplying a matrix with its inverse results in the identity matrix: $MM^{-1} = M^{-1}M = I$
  - The formula for finding inverses can be given as follows:

$$\mathbf{A}^{-1} = \frac{\mathbf{A}^{*}}{\det \mathbf{A}}$$

Matrix Operations Examples

- Matrix Inverses
  - Example:
  - Take: $$\mathbf{M} = \begin{bmatrix} -4 & -3 & 3 \\ 0 & 2 & -2 \\ 1 & 4 & -1 \end{bmatrix}.$$
  - Step 1: Compute the co-factors, given the pattern:

$$\begin{bmatrix} + & - & + & - & \cdots \\ - & + & - & + & \cdots \\ + & - & + & - & \cdots \\ - & + & - & + & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Matrix Operations Examples

- Step 1a: Get the minors:
- Bomberman time!

$$\mathbf{M} = \begin{bmatrix} 2 & -2 \\ 4 & -1 \end{bmatrix}. \qquad \mathbf{M} = \begin{bmatrix} 0 & -2 \\ 1 & -1 \end{bmatrix}.$$

- Etc…

Matrix Operations Examples

- Step 1a: Get the minors:

$$C^{\{11\}} = +\begin{vmatrix} 2 & -2 \\ 4 & -1 \end{vmatrix} \qquad C^{\{12\}} = -\begin{vmatrix} 0 & -2 \\ 1 & -1 \end{vmatrix} \qquad C^{\{13\}} = +\begin{vmatrix} 0 & 2 \\ 1 & 4 \end{vmatrix}$$

$$C^{\{21\}} = -\begin{vmatrix} -3 & 3 \\ 4 & -1 \end{vmatrix} \qquad C^{\{22\}} = +\begin{vmatrix} -4 & 3 \\ 1 & -1 \end{vmatrix} \qquad C^{\{23\}} = -\begin{vmatrix} -4 & -3 \\ 1 & 4 \end{vmatrix}$$

$$C^{\{31\}} = +\begin{vmatrix} -3 & 3 \\ 2 & -2 \end{vmatrix} \qquad C^{\{32\}} = -\begin{vmatrix} -4 & 3 \\ 0 & -2 \end{vmatrix} \qquad C^{\{33\}} = +\begin{vmatrix} -4 & -3 \\ 0 & 2 \end{vmatrix}$$

Matrix Operations Examples

- Step 1b: Get the determinants, given: $|\mathbf{M}| = \begin{vmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{vmatrix} = m_{11}m_{22} - m_{12}m_{21}$

$$C^{\{11\}} = +\begin{vmatrix} 2 & -2 \\ 4 & -1 \end{vmatrix} = 6, \quad C^{\{12\}} = -\begin{vmatrix} 0 & -2 \\ 1 & -1 \end{vmatrix} = -2, \quad C^{\{13\}} = +\begin{vmatrix} 0 & 2 \\ 1 & 4 \end{vmatrix} = -2,$$

$$C^{\{21\}} = -\begin{vmatrix} -3 & 3 \\ 4 & -1 \end{vmatrix} = 9, \quad C^{\{22\}} = +\begin{vmatrix} -4 & 3 \\ 1 & -1 \end{vmatrix} = 1, \quad C^{\{23\}} = -\begin{vmatrix} -4 & -3 \\ 1 & 4 \end{vmatrix} = 13,$$

$$C^{\{31\}} = +\begin{vmatrix} -3 & 3 \\ 2 & -2 \end{vmatrix} = 0, \quad C^{\{32\}} = -\begin{vmatrix} -4 & 3 \\ 0 & -2 \end{vmatrix} = -8, \quad C^{\{33\}} = +\begin{vmatrix} -4 & -3 \\ 0 & 2 \end{vmatrix} = -8.$$

$$\begin{vmatrix} -4 & -3 & 3 \\ 0 & 2 & -2 \\ 1 & 4 & -1 \end{vmatrix} = \begin{array}{l} (-4)\big((\ \ 2)(-1) - (-2)(\ \ 4)\big) \\ +(-3)\big((-2)(\ \ 1) - (\ \ 0)(-1)\big) \\ +(\ \ 3)\big((\ \ 0)(\ \ 4) - (\ \ 2)(\ \ 1)\big) \end{array}$$

$$= \begin{array}{l} (-4)\big((-2) - (-8)\big) \\ +(-3)\big((-2) - (\ \ 0)\big) \\ +(\ \ 3)\big((\ \ 0) - (\ \ 2)\big) \end{array} = \begin{array}{l} (-4)(\ \ 6) \\ +(-3)(-2) \\ +(\ \ 3)(-2) \end{array} = \begin{array}{l} (-24) \\ +(\ \ \ 6) \\ +(\ -6) \end{array}$$

$$\boxed{= -24.}$$

Matrix Operations Examples

- Step 2: Get the adjoint of M:
- Which is the transpose matrix of co-factors, thus:

$$\text{adj } \mathbf{M} = \begin{bmatrix} C^{\{11\}} & C^{\{12\}} & C^{\{13\}} \\ C^{\{21\}} & C^{\{22\}} & C^{\{23\}} \\ C^{\{31\}} & C^{\{32\}} & C^{\{33\}} \end{bmatrix}^{\mathrm{T}}$$

$$= \begin{bmatrix} 6 & -2 & -2 \\ 9 & 1 & 13 \\ 0 & -8 & -8 \end{bmatrix}^{\mathrm{T}} = \boxed{\begin{bmatrix} 6 & 9 & 0 \\ -2 & 1 & -8 \\ -2 & 13 & -8 \end{bmatrix}}.$$

Matrix Operations Examples

- Step 3: Divide the classical adjoint (from Step 2) by the determinant:

$$\mathbf{M} = \begin{bmatrix} -4 & -3 & 3 \\ 0 & 2 & -2 \\ 1 & 4 & -1 \end{bmatrix};$$

$$\mathbf{M}^{-1} = \frac{\text{adj } \mathbf{M}}{|\mathbf{M}|} = \frac{1}{-24} \begin{bmatrix} 6 & 9 & 0 \\ -2 & 1 & -8 \\ -2 & 13 & -8 \end{bmatrix} = \begin{bmatrix} -1/4 & -3/8 & 0 \\ 1/12 & -1/24 & 1/3 \\ 1/12 & -13/24 & 1/3 \end{bmatrix}.$$

Week 1

Lab Activities

Week 1 Lab

- For the lab, see Hooman's material (with video)
- OpenGL examples covered:
  - GLUT and GLFW basics
  - Fixed and programmable pipeline examples
  - Rendering different shapes
  - Using vertex and fragment shaders

Week 1

End