Standard Graphics Pipeline

**Last week**

- Construct geometry
- Define transformations
- Assign material properties

**This week**

- Vertex Transformations
- Vertex Shader: Object's local space → viewport space
- Culling & Clipping: Determine front-facing triangles, determine which vertices are visible
- Rasterization: Fill the triangle with fragments
- Fragment Shading: Calculate correct color values
- Visibility Tests: Is the fragment visible?
- Blending: Blend together multiple fragments
Points and Vectors

- In computer graphics we distinguish:
  - **Point** – a location in space (location vector, *kohavektor*)
  - **Vector** – a direction in space (direction vector, *suunavektor*)
Points and Vectors

• Both are elements of a 2-, 3- or 4-dimensional vector space over the field $\mathbb{R}$

• More precisely, elements of a coordinate space.

• So both are vectors in terms of algebra.

• We distinguish them because some operations make sense for vectors, some for points.

• A space that contains both of them and defines an addition between a point and a vector is called an affine space.

• More precisely, an Euclidean space.
Points and Vectors

Vector space of points

Vector space of vectors
Points and Vectors

- Given a vector space over $\mathbb{R}^2$ with a basis and the origin, all the elements of the vector space can be represented as a...
Points and Vectors

\[ \mathbf{v} = \alpha_0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \]

So, the scalar coefficients for our \( \mathbf{v} \) would currently be?
Points and Vectors

- Because the elements of our vector space are \( n \)-tuples, we can call it a coordinate space.

![Coordinate space of points](image)

![Coordinate space of vectors](image)
Points and Vectors

• Besides just doing operations between points, or between vectors, we want to do operations between them.

• Or do we? Can you think of an operation we would want to do between a point and a vector?
Points and Vectors

• When we put those two spaces together, we get an affine space.
Points and Vectors

• Is this a point or a vector?

\[ x = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \]

• What about this?

\[ x = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \]
Points and Vectors

- **Row-major** and **column-major** formats.
- Which is which?
- How to get from one to another?

\[
x = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \quad \Rightarrow \quad x = \begin{pmatrix} 3 & 2 \end{pmatrix}
\]

Sometimes the reader is expected to guess which one is used based on the context.

Good explanation:
Points and Vectors

- **Homogeneous coordinates** – a notation where we add an additional coordinate to distinguish between points and vectors.

\[
p = (x \ y \ z) = \left( \frac{x}{z}, \frac{y}{z} \right)
\]

\[
p = (x \ y \ z \ w) = \left( \frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right)
\]
Points and Vectors

• In 2D homogeneous coordinates:

  • Point
    
    \[
    p = (x \ y \ z) \quad z \neq 0
    \]

  • Vector
    
    \[
    p = (x \ y \ z) \quad z = 0
    \]

• Vector is a point located in infinity
Points and Vectors

- What should \( z \) be if you want to define a point located at \((x, y)\)?
- How does addition work now?
- Addition between two vectors?
- Addition between two points?
- Subtraction of points?
You probably know about the **implicit** line equation: \( y = a \cdot x + b \)
It defines the relationship between the coordinates.

Can also be used to test if a given point is on the line or not. How?
Line

- How can we represent a line in our affine space?
- We do not know $a$ (the slope) or $b$ (y-intercept).
- What would we need to know to represent a line?
Line

- Two points that the line passes
- One point and a direction vector
Line

\[ \text{line} = (1 - \alpha) \cdot A + \alpha \cdot B \quad \text{line} = A + \alpha \cdot d \]

\[ d = B - A \]
Line Segment

- Knowing that: \( \text{line} = (1 - \alpha) \cdot A + \alpha \cdot B \)

How to represent a line segment?
Triangle

- How about a triangle?
Barycentric Coordinates

- The coefficients of a convex combination of the vertices are the **Barycentric coordinates** of all the points inside the triangle.

\[ \text{triangle} = \alpha_0 \cdot A + \alpha_1 \cdot B + \alpha_2 \cdot C \]

\[ \alpha_i \geq 0, \quad \alpha_0 + \alpha_1 + \alpha_2 = 1 \]
Barycentric Coordinates

- The coefficients of a **convex combination** of the vertices are the **Barycentric coordinates** of all the points inside the triangle.

\[
\text{triangle} = \alpha_0 \cdot A + \alpha_1 \cdot B + \alpha_2 \cdot C
\]

\[
\alpha_i \geq 0, \quad \alpha_0 + \alpha_1 + \alpha_2 = 1
\]

What are the coordinates of the vertices in the Barycentric system?

Find them for other easy points.
Dot Product

• Useful operation between vectors. Why?

• Definition
  • Geometric: \( u \cdot v = \|u\| \cdot \|v\| \cdot \cos(\angle uv) \)
  • Algebraic: \( u \cdot v = u_0 \cdot v_0 + u_1 \cdot v_1 + u_2 \cdot v_2 \)

• Also called: scalar product, inner product

• *Skalaarkorrutis*
Scalar Projection

- Dot product can be used to project one vector onto another.
- Scalar projection of \( \mathbf{u} \) onto \( \mathbf{v} \) is:
  \[
  c = \mathbf{u} \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|} = \mathbf{u} \cdot \hat{\mathbf{v}}
  \]
- It gives you the length, how much \( \hat{\mathbf{v}} \) you have to take in order to reach the orthogonal projection point of \( \mathbf{u} \).

*Eg* The Gram–Schmidt process
# Cross Product

- Returns a vector orthogonal to the operands.

- **Definition**
  \[ u \times v = n \cdot ||u|| \cdot ||u|| \cdot \sin(\angle uv) \]

  - Geometric
  - Algebraic

- Also called: vector product

- *Vektorkorrutis*

Direction of the result depends on the handedness of the coordinate system.
Scalar Triple Product

- Definition: \( u \cdot (v \times w) \)

- Useful in solving a system of equations of vectors, because:

\[
\begin{vmatrix}
  u_0 & u_1 & u_2 \\
  v_0 & v_1 & v_2 \\
  w_0 & w_1 & w_2
\end{vmatrix}
\]

- We can see this in Basic II, with triangle-ray intersection testing.

- *Segakorrutis.*
What was important for you today?

What more would you like to know?

Next time: Transformations
(scale, shear, rotate, translate)