3D Graphics

The rendering pipeline
Reminder : Scene

Light Source

3D Objects

Camera
The rendering pipeline

Helps us go from a 3D scene to a 2D image

Minimal rendering pipeline
Re-express vertices in the camera coordinates system

Project vertices in the frustum
Find which pixel is inside which triangle

Emit fragments (candidates pixel)
Compute the color to give each fragment.
Choose which Fragment get to become a pixel using a Depth test
Vertex Shader
Re-express vertices in the camera coordinates system

Project vertices in the frustum
Camera Space

“Up” vector

“Right” vector

“Look at” vector

{Xs, Ys, Zs}

{Xc, Yc, Zc}
Reminder Camera

Re-express positions in the camera space
Reminder Camera

Our current coordinates:

\[ \{Xs, Ys, Zs\} = Xs \cdot X + Ys \cdot Y + Zs \cdot Z \]

\[ X = \{1, 0, 0\} \]

\[ Y = \{0, 1, 0\} \]

\[ Z = \{0, 0, 1\} \]
Reminder Camera

The camera space

\{X_s,Y_s,Z_s\} = X_s' \times R + Y_s' \times U + Z_s' \times L
Reminder Camera

The Mathematical problem

\[ \{X_s, Y_s, Z_s\} = X_s' \ast R + Y_s' \ast U + Z_s' \ast L \]

\[ \{X_s, Y_s, Z_s\} = X_s \ast X + Y_s \ast Y + Z_s \ast Z \]

\[ X_s \ast X + Y_s \ast Y + Z_s \ast Z = X_s' \ast R + Y_s' \ast U + Z_s' \ast L \]

_The problem Find X_s’, Y_s’, and Z_s’_
Linear Transformations

https://youtu.be/kYB8lZa5AuE?si=vo2LXj1Vz_gwdasT
Linear Transformations

https://youtu.be/rHLEWRxRGiM?si=aKhXMsqAgGqWmSQv
Question:

\[ Xs \times X + Ys \times Y + Zs \times Z = Xs' \times R + Ys' \times U + Zs' \times L \]

**What is the matrix that solve this problem**

(1 minute alone)

(2 minutes with your neighbors)

(5 minutes with the whole group)
Question:

\[ Xs \times X + Ys \times Y + Zs \times Z = Xs' \times R + Ys' \times U + Zs' \times L \]

"Right" vector

\[
\begin{bmatrix}
Xs \\
Ys \\
Zs
\end{bmatrix}
= \begin{bmatrix}
R_x & U_x & L_x \\
R_y & U_y & L_y \\
R_z & U_z & L_z
\end{bmatrix}
\begin{bmatrix}
Xs' \\
Ys' \\
Zs'
\end{bmatrix}

"Up" vector

"Look at" vector
Question:

\[ Xs \times X + Ys \times Y + Zs \times Z = Xs' \times R + Ys' \times U + Zs' \times L \]

“Right” vector

\[
\begin{bmatrix}
Xs' \\
Ys' \\
Zs'
\end{bmatrix}
= \begin{bmatrix}
Rx & Ux & Lx \\
Ry & Uy & Ly \\
Rz & Uz & Lz
\end{bmatrix}^{-1}
\begin{bmatrix}
Xs \\
Ys \\
Zs
\end{bmatrix}
\]

“Look at” vector

“Up” vector
Question:

\[ Xs \times X + Ys \times Y + Zs \times Z = Xs' \times R + Ys' \times U + Zs' \times L \]

“Right” vector

\[
\begin{bmatrix}
Xs' \\
Ys' \\
Zs'
\end{bmatrix}
= \begin{bmatrix}
Rx & Ry & Rz \\
Ux & Uy & Uz \\
Lx & Ly & Lz
\end{bmatrix}
\begin{bmatrix}
Xs \\
Ys \\
Zs
\end{bmatrix}
\]

“Up” vector

“Look at” vector
Problem with Linear transformations

The camera space

{Xs, Ys, Zs} = Xs' * R + Ys' * U + Zs' * L

What about the camera coordinates?
Homogeneous Coordinates

Allow us to “move” the origin of the frame

Using 4 coordinates instead of 3: Homogeneous coordinates
Linear Transformation

\[ Xs \cdot X + Ys \cdot Y + Zs \cdot Z = Xs' \cdot R + Ys' \cdot U + Zs' \cdot L \]

\[
\begin{bmatrix}
Xs' \\
Ys' \\
Zs' \\
Ws
\end{bmatrix} =
\begin{bmatrix}
Rx & Ry & Rz & 0 \\
Ux & Uy & Uz & 0 \\
Lx & Ly & Lz & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
Xs \\
Ys \\
Zs \\
Ws
\end{bmatrix}
\]
Affine Transformation

General case

\[
\begin{bmatrix}
  x + \omega Tx \\
  y + \omega Ty \\
  z + \omega Tz \\
  w
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 0 & Tx \\
  0 & 1 & 0 & Ty \\
  0 & 0 & 1 & Tz \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  w
\end{bmatrix}
\]
Affine Transformation

For a vertex: \( w = 1 \)

\[
\begin{bmatrix}
  x + Tx \\
  y + Ty \\
  z + Tz \\
  1
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 0 & Tx \\
  0 & 1 & 0 & Ty \\
  0 & 0 & 1 & Tz \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]

Add the vector \( T \) to the vertex
Translate the camera to the origin

\[
\begin{bmatrix}
1 & 0 & 0 & -Xc \\
0 & 1 & 0 & -Yc \\
0 & 0 & 1 & -Zc \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

What about the camera coordinates?

\{Xc, Yc, Zc\}
Camera Space

\[
\begin{bmatrix}
X_{s'} \\
Y_{s'} \\
Z_{s'} \\
W_{s}
\end{bmatrix}
= 
\begin{bmatrix}
R_x & R_y & R_z & 0 \\
U_x & U_y & U_z & 0 \\
L_x & L_y & L_z & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & -X_c \\
0 & 1 & 0 & -Y_c \\
0 & 0 & 1 & -Z_c \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X_s \\
Y_s \\
Z_s \\
W_s
\end{bmatrix}
\]
Matrix multiplication

https://www.youtube.com/watch?v=XkY2DOUCWMU
Camera Space

\[
\begin{bmatrix}
R_x & R_y & R_z & 0 \\
U_x & U_y & U_z & 0 \\
L_x & L_y & L_z & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & -X_c \\
0 & 1 & 0 & -Y_c \\
0 & 0 & 1 & -Z_c \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Reminder: Camera

Frustum : the visible part of the scene :

- Near plane
- Far plane
- Aspect ratio
- Field of View
Projection Perspective

Objective: we want to express the visible space in the following space

\[ x \text{ in } [-1,1] \]
\[ y \text{ in } [-1,1] \]
\[ z \text{ in } [0,1] \]
Perspective Projection matrix

- Near plane = n
- Far plane = f
- Aspect ratio = a
- Field of View = fov

\[
s = \frac{1}{\tan\left(\frac{\text{fov}}{2}\right)}
\]

\[
\begin{bmatrix}
\frac{s}{a} & 0 & 0 & 0 \\
0 & s & 0 & 0 \\
0 & 0 & \frac{f}{f-n} & -\frac{f*n}{f-n} \\
0 & 0 & 1 & 0
\end{bmatrix}
\]
Problem

After the projection the w coordinate of the vertex is modified.

To solve this issue we normalize all vertex coordinates by dividing them by w
Summary Vertex shader

For each vertex $v$, the vertex shader computes a vertex $v'$ such that:

$$v_{tmp} = \begin{bmatrix} \text{proj} \end{bmatrix} \begin{bmatrix} \text{view} \end{bmatrix} v$$

$$v' = \frac{v_{tmp}}{v_{tmp} \cdot w}$$