Back-face Culling

First consider VCS

First idea: cull if \( N_z < 0 \)
Works, but sometimes misses polygons that should be culled

Better idea: cull if eye is below polygon plane

Back-face Culling (continued)

Summary of culling in VCS
- compute polygon normal, \( N \)
- cull if \(-N \cdot P < 0\)

Culling in NDCS

VCS

\[ \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} m_1 & m_2 & m_3 & T_x \\ m_4 & m_5 & m_6 & T_y \\ m_7 & m_8 & m_9 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \]

eye

cull if \( N_z > 0 \)

Back-face Culling (continued)

Computing Normals
- polygon: \( N = (P_2 - P_1) \times (P_3 - P_1) \)
- assume vertices ordered CCW when viewed from visible side of polygon
- normal for a vertex:
  - used for lighting
  - supplied by model (i.e., sphere), or computed from neighboring polygons

Back-face Culling (continued)

Transforming Normals
- first idea: set \( h=0 \)
- zero in order to avoid translations
- problem: only works for some transformation matrices (rotations, uniform scales, translations)
Transforming Normals (cont.)

Transform a plane

Plane = \(Ax + By + Cz + D\)

\[
\begin{bmatrix}
A & B & C & D
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= N^T P
\]

\(P\) \rightarrow \(P' = MP\)
\(N\) \rightarrow \(N' = QN\)

If we know \(M\),
what should \(Q\) be?

\[
\begin{bmatrix}
P \\
N
\end{bmatrix}
\rightarrow
\begin{bmatrix}
P' \\
N'
\end{bmatrix}
\]

\(N^T P = 0\)
\(N'^T P = 0\)
\((QN)^T (MP) = 0\)
\(N'^T Q^T M = 0\)

\[Q^T M = I\]

\[
Q = \left(M^{-1}\right)^T
\]