Scene graphs - arrange objects in a hierarchy of transformations; each node has transformation matrix used for rendering:

\[ x' = x \cos \theta - y \sin \theta \]
\[ y' = x \sin \theta + y \cos \theta \]

can perform transformations using matrices:

\[
\begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
x' \\
y'
\end{bmatrix}
\]

rotation

\[
\begin{bmatrix}
1 & 0 & \Delta x \\
0 & 1 & \Delta y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

translation

\[
\begin{bmatrix}
1 & 0 & \Delta x \\
0 & 1 & \Delta y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

rotation followed by translation

\[
= x \cos \theta - y \sin \theta + \Delta x \\
y \sin \theta + x \cos \theta + \Delta y
\]
Quad and Oct Trees

divide area into four sections,
stop when area has fewer
than two objects

test for collision by traversing tree
from root until reaching a leaf,
test for collision with other object
in leaf (if it exists)
if object spans multiple subdivisions
must traverse each
Binary space partition

divide area in half, alternating vertically and horizontally, stop when area has less than two objects in it

do not have to divide evenly, can create a more flat tree by choosing partition location
BSP for rendering

each partition is a plane
coincident with a polygon

can traverse the BSP based
on viewport location to find
order to draw polygons

GFBCEDA