

Painter Algorithm

View reference point is changing

polygons are fixed

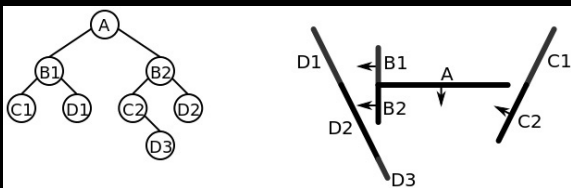
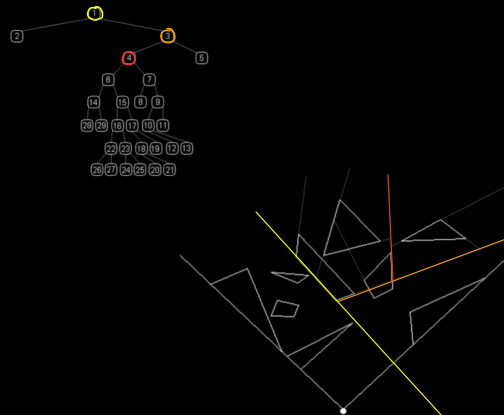
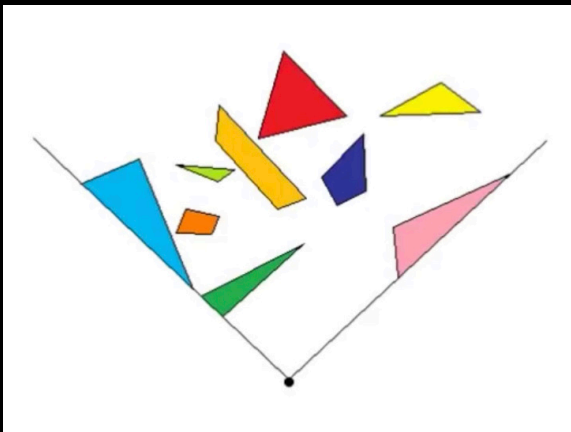
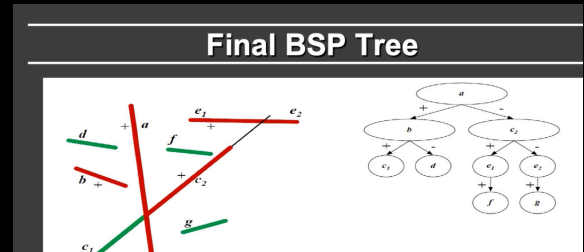
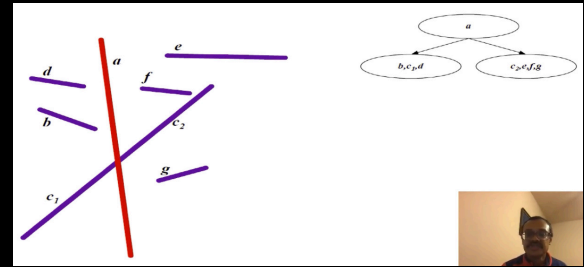
① Determine partitioning plane P_i

→ front < left edge >

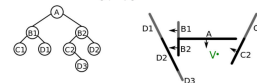
→ back < right edge >

② divide polygons recursively until the tree formed

? Δ Makes most polygons convex → make tree balanced
WHY HOW



1. If the current node is a leaf node, render the polygons at the current node.
2. Otherwise, if the viewing location V is in front of the current node:
 1. Render the child BSP tree containing polygons behind the current node
 2. Render the polygons at the current node
 3. Render the child BSP tree containing polygons in front of the current node
3. Otherwise, if the viewing location V is behind the current node:
 1. Render the child BSP tree containing polygons in front of the current node
 2. Render the polygons at the current node
 3. Render the child BSP tree containing polygons behind the current node
4. Otherwise, the viewing location V must be exactly on the plane associated with the current node. Then:
 1. Render the child BSP tree containing polygons in front of the current node
 2. Render the child BSP tree containing polygons behind the current node



Applying this algorithm recursively to the BSP tree generated above results in the following steps:

- The algorithm is first applied to the root node of the tree, node A . V is in front of node A , so we apply the algorithm first to the child BSP tree containing polygons behind A
 - This tree has root node $B1$. V is behind $B1$ so first we apply the algorithm to the child BSP tree containing polygons in front of $B1$:
 - This tree is just the leaf node $D1$, so the polygon $D1$ is rendered.
 - We then render the polygon $B1$.
 - We then apply the algorithm to the child BSP tree containing polygons behind $B1$:
 - This tree is just the leaf node $C1$, so the polygon $C1$ is rendered.
- We then draw the polygons of A
- We then apply the algorithm to the child BSP tree containing polygons in front of A
 - This tree has root node $B2$. V is behind $B2$ so first we apply the algorithm to the child BSP tree containing polygons in front of $B2$:
 - This tree is just the leaf node $D2$, so the polygon $D2$ is rendered.
 - We then render the polygon $B2$.
 - We then apply the algorithm to the child BSP tree containing polygons behind $B2$:
 - This tree has root node $C2$. V is in front of $C2$ so first we would apply the algorithm to the child BSP tree containing polygons behind $C2$. There is no such tree, however, so we continue.
 - We render the polygon $C2$.
 - We apply the algorithm to the child BSP tree containing polygons in front of $C2$
 - This tree is just the leaf node $D3$, so the polygon $D3$ is rendered.

The tree is traversed in linear time and renders the polygons in a far-to-near ordering ($D1, B1, C1, A, D2, B2, C2, D3$) suitable for the painter's algorithm.