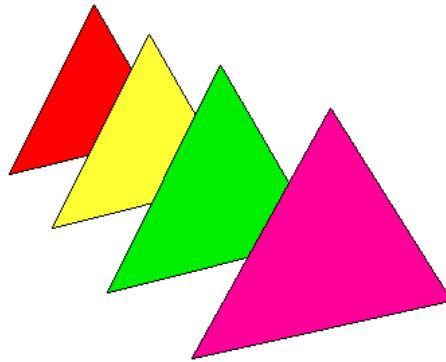




Painter's algorithm

Depth-sorting method



Render from back to front
Both image and object space method



Tests for sorting polygons

0. Z interval overlap
1. Bounding rectangle overlap
- 2-3. Tests using surface normal
 2. Surface completely behind
 3. Surface completely in front
4. Polygon-level check of projected polygon



Tests for sorting polygons

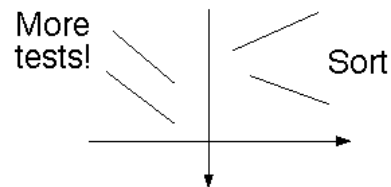
0. Z interval overlap

1. Bounding rectangle overlap

2-3. Tests using surface normal

- 2. Surface completely behind
- 3. Surface completely in front

4. Polygon-level check of projected polygon



Tests for sorting polygons

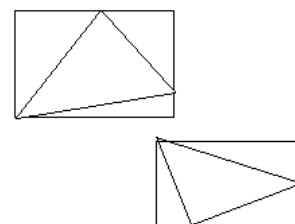
0. Z interval overlap

1. Bounding rectangle overlap

2-3. Tests using surface normal

- 2. Surface completely behind
- 3. Surface completely in front

4. Polygon-level check of projected polygon





Tests for sorting polygons

0. Z interval overlap

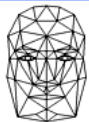
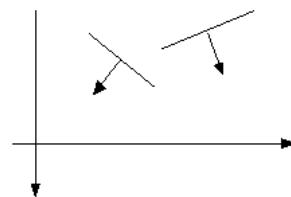
1. Bounding rectangle overlap

2-3. Tests using surface normal

2. Surface completely behind

3. Surface completely in front

4. Polygon-level check of projected polygon



Tests for sorting polygons

0. Z interval overlap

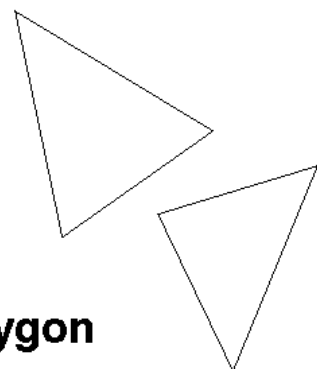
1. Bounding rectangle overlap

2-3. Tests using surface normal

2. Surface completely behind

3. Surface completely in front

4. Polygon-level check of projected polygon

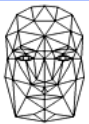
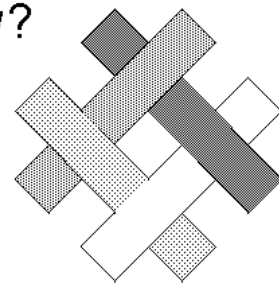




Painter's algorithm

Some scenes can not be sorted at all!

Solution: Figure out a way to split polygons to resolve the sort. But how?



Painter's algorithm

- Slow – may paint many pixels more than once
- Easy to implement in its simplest forms
- Slow and complicated in its full form - needs many checks to avoid errors



Painter's Algorithm for transparency

- Work on object level, not polygon level
- Only transparent objects need to be sorted
- Approximate sorting is often sufficient



BSP trees

BSP = Binary Space Partitioning

Variant on Painter's algorithm

Pre-generated sorting

Suitable for static scenes