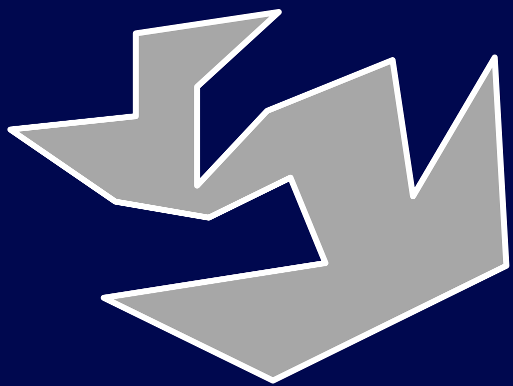
Clipping

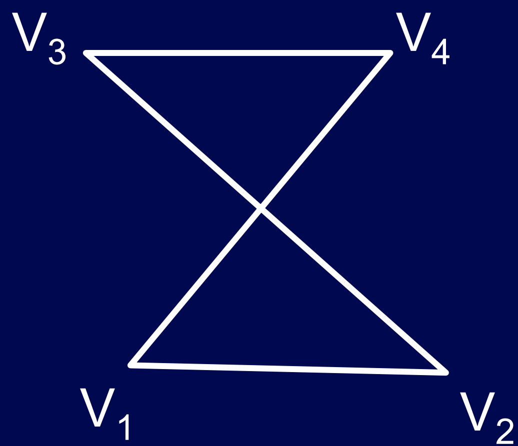
Polygon Clipping

Polygon : Area primitive

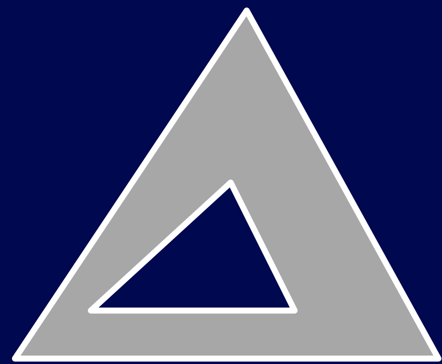
Simple Polygon: Planar set of ordered points
No line crossings
No holes



Simple Polygon



Line Crossing

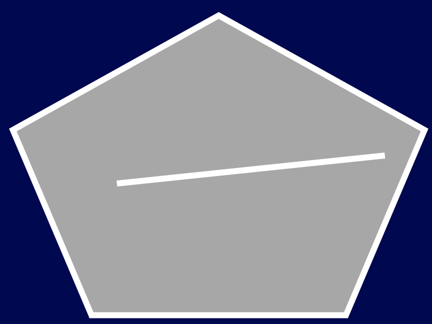


Hole

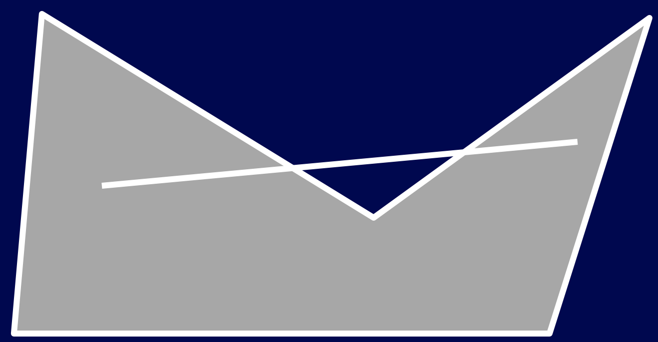
Clipping

Polygon Clipping

Polygon : Area primitive



Convex Polygon



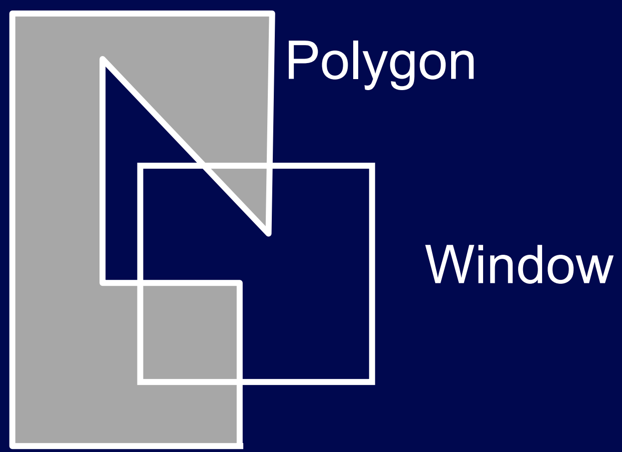
Non-Convex Polygon

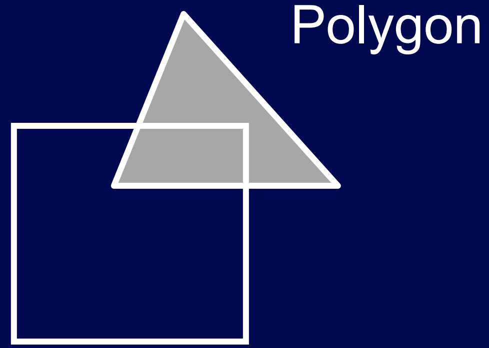
Clipping

Polygon Clipping

Sutherland-Hodgman

- Window must be convex
- Polygon to be clipped can be convex or non-convex



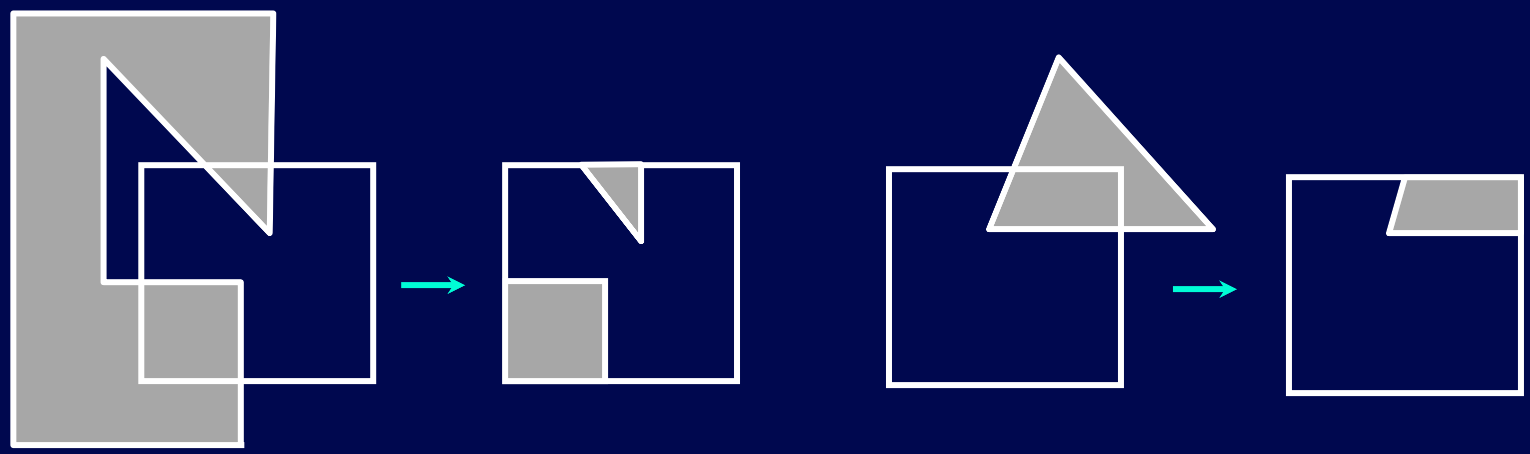


Clipping

Polygon Clipping

Sutherland-Hodgman

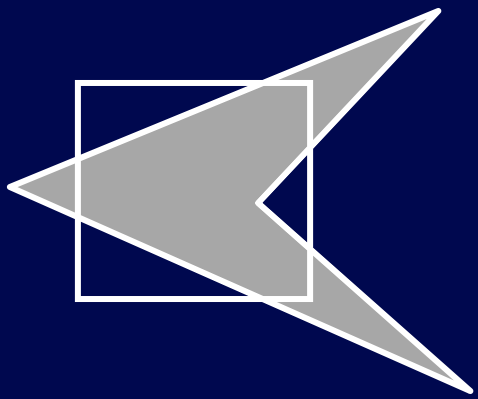
- Window must be convex
- Polygon to be clipped can be convex or non-convex



Clipping

Polygon Clipping

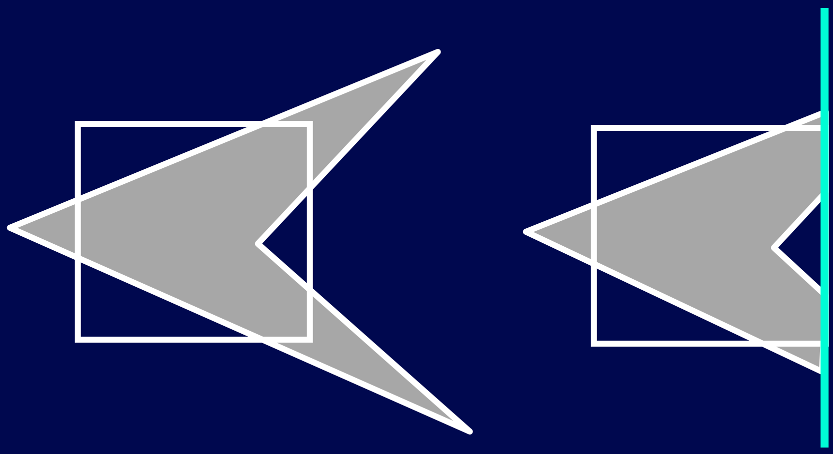
Sutherland-Hodgman



Clipping

Polygon Clipping

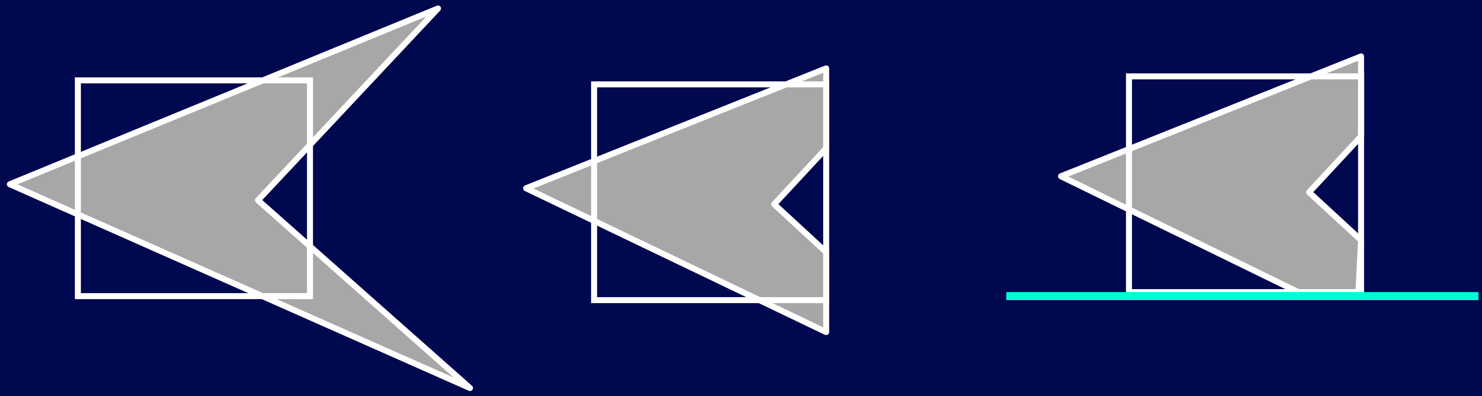
Sutherland-Hodgman



Clipping

Polygon Clipping

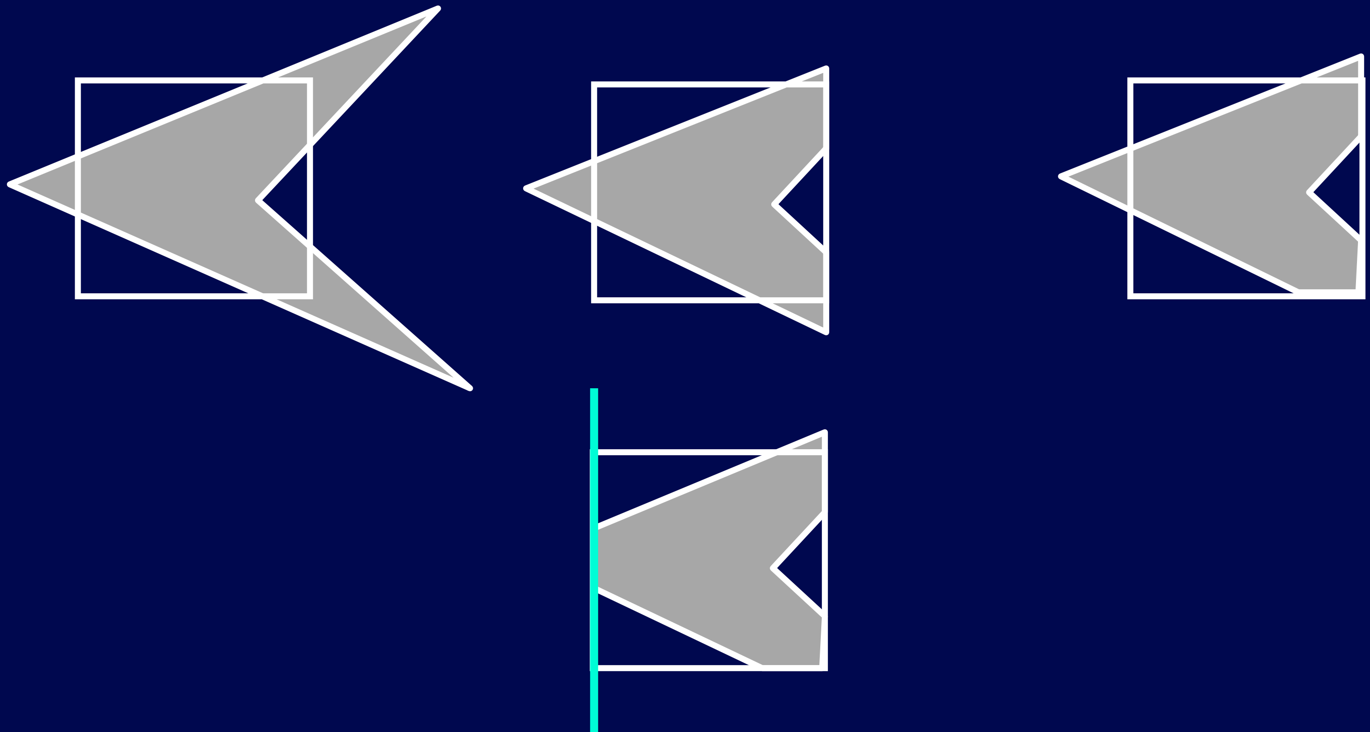
Sutherland-Hodgman



Clipping

Polygon Clipping

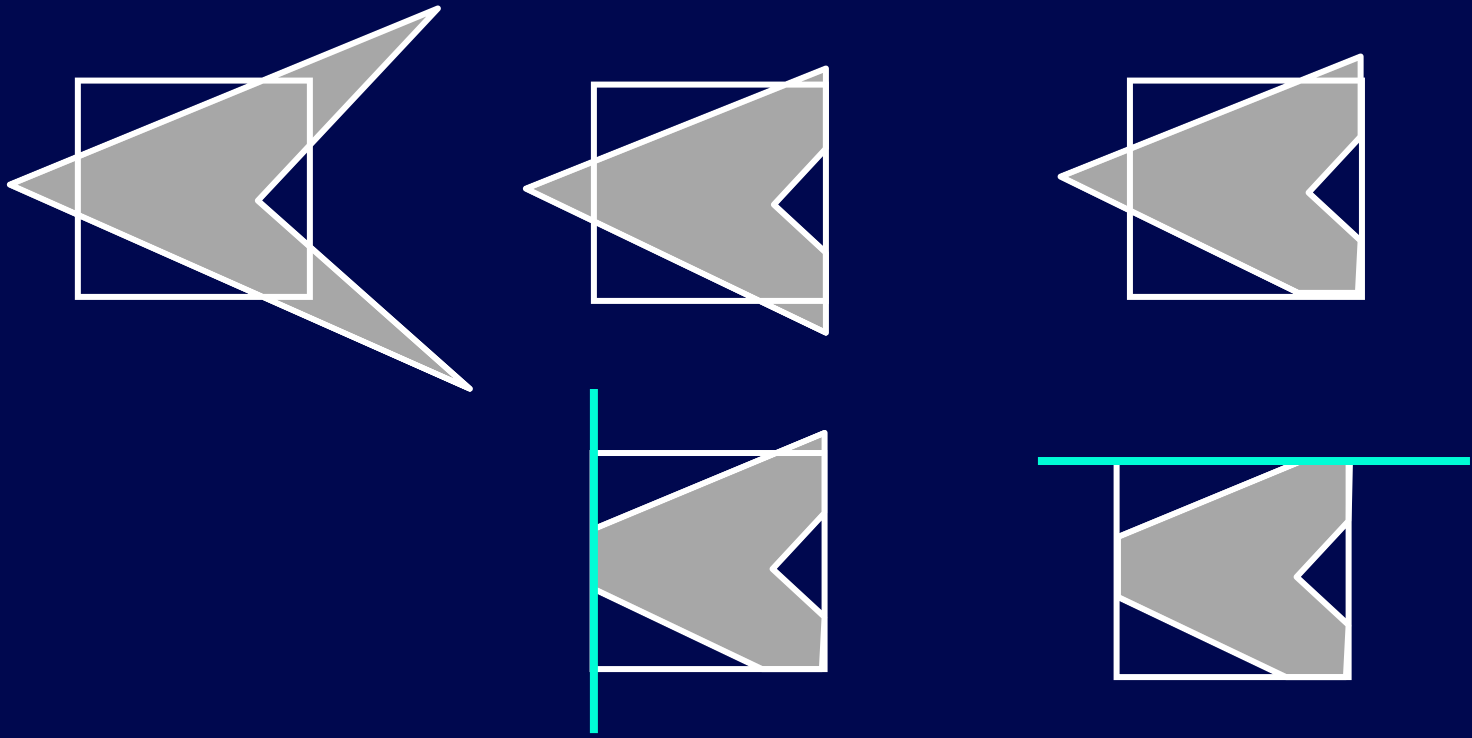
Sutherland-Hodgman



Clipping

Polygon Clipping

Sutherland-Hodgman



Clipping

Polygon Clipping

Sutherland-Hodgman

Approach

- Polygon to be clipped is given as $v_1, v_2, \ldots, v_n$
- Polygon edge is a pair $[v_i, v_{i+1}]$
- Process all polygon edges in succession against a window edge

  polygon $(v_1, v_2, \ldots, v_n) \rightarrow$ polygon $(w_1, w_2, \ldots, w_m)$
- Repeat on resulting polygon with next window edge

Clipping

Polygon Clipping

Sutherland-Hodgman

Approach

Four Cases

- ***s*** = v_i is the polygon edge starting vertex
- ***p*** = v_{i+1} is the polygon edge ending vertex
- ***i*** is a polygon-edge/window-edge intersection point
- w_j is the next polygon vertex to be output

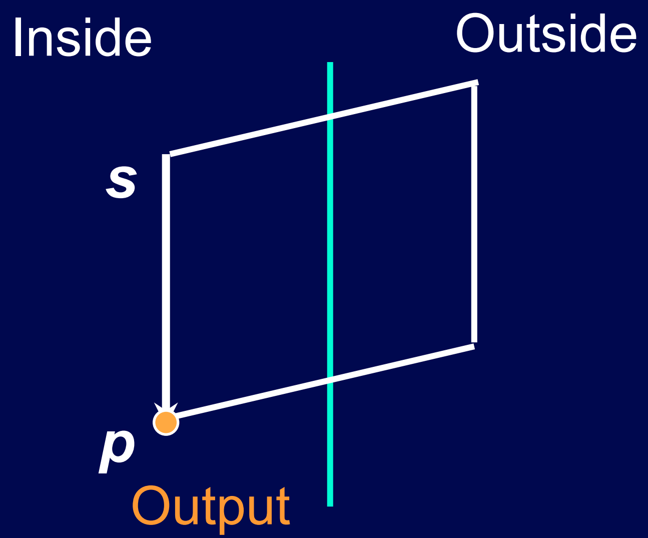
Clipping

Polygon Clipping

Sutherland-Hodgman

Approach

Case 1: Polygon edge is entirely inside the window edge



- ***p*** is next vertex of resulting polygon
- ***p*** $\rightarrow w_j$ and $j+1 \rightarrow j$

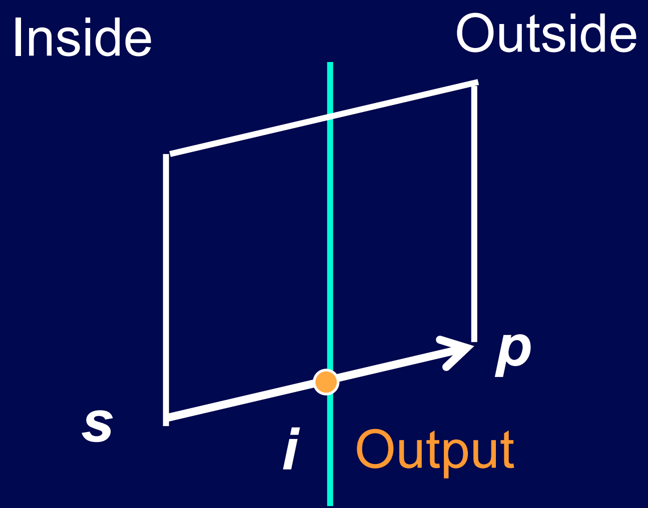
Clipping

Polygon Clipping

Sutherland-Hodgman

Approach

Case 2: Polygon edge crosses window edge going out



- Intersection point ***i*** is next vertex of resulting polygon
- ***i*** $\rightarrow$ w_j and $j+1 \rightarrow j$

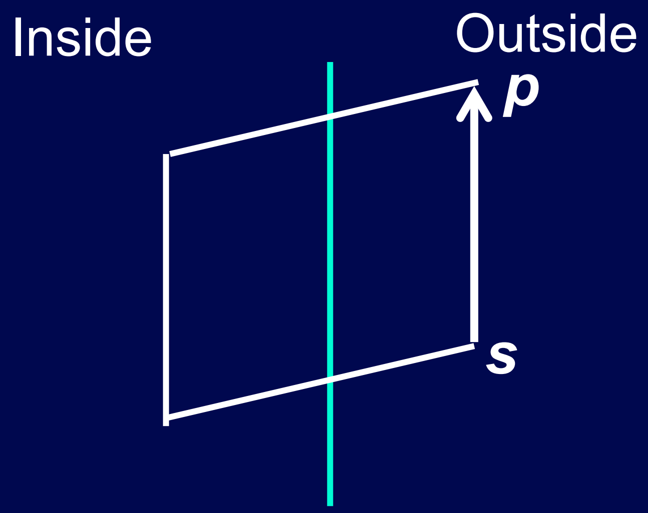
Clipping

Polygon Clipping

Sutherland-Hodgman

Approach

Case 3: Polygon edge is entirely outside the window edge



- No output

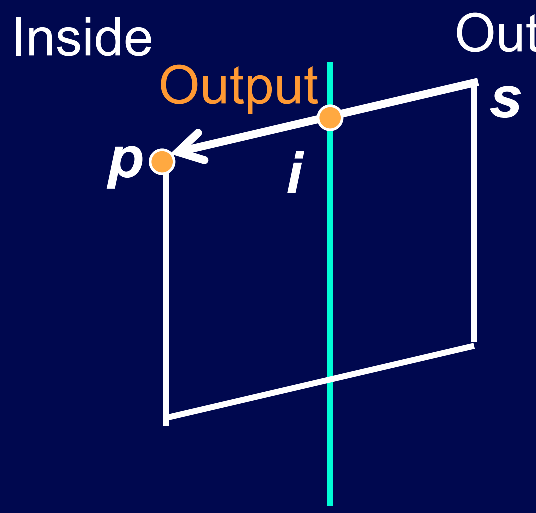
Clipping

Polygon Clipping

Sutherland-Hodgman

Approach

Case 4: Polygon edge crosses window edge going in



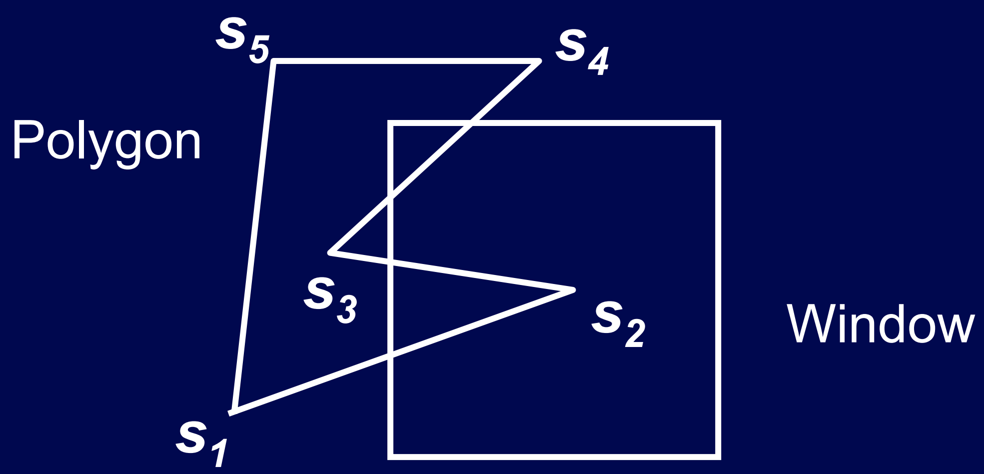
- Intersection point ***i*** and ***p*** are next two vertices of resulting polygon
- ***i*** $\rightarrow w_j$ and ***p*** $\rightarrow w_{j+1}$ and $j+2 \rightarrow j$

Clipping

Polygon Clipping

Sutherland-Hodgman

Example

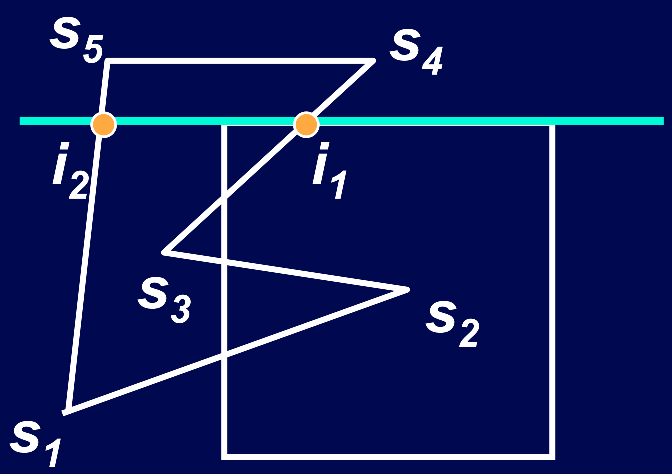


Clipping

Polygon Clipping

Sutherland-Hodgman

Example

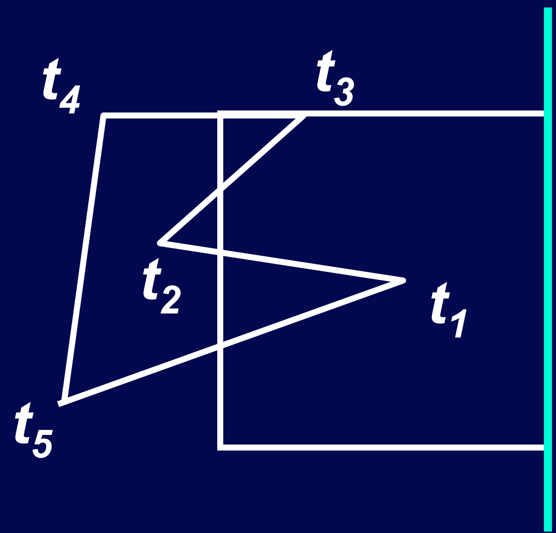


Clipping

Polygon Clipping

Sutherland-Hodgman

Example

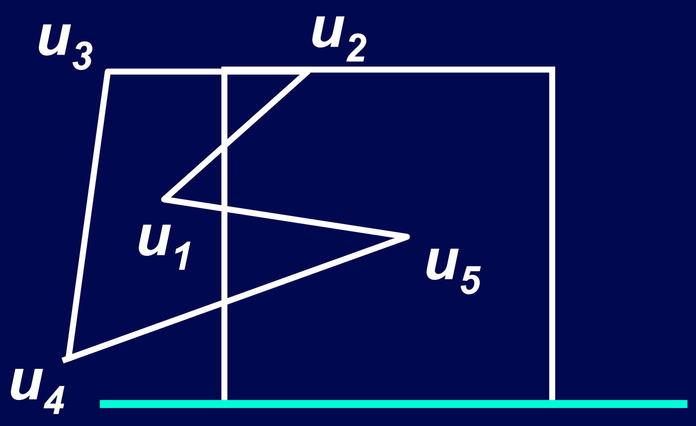


Clipping

Polygon Clipping

Sutherland-Hodgman

Example

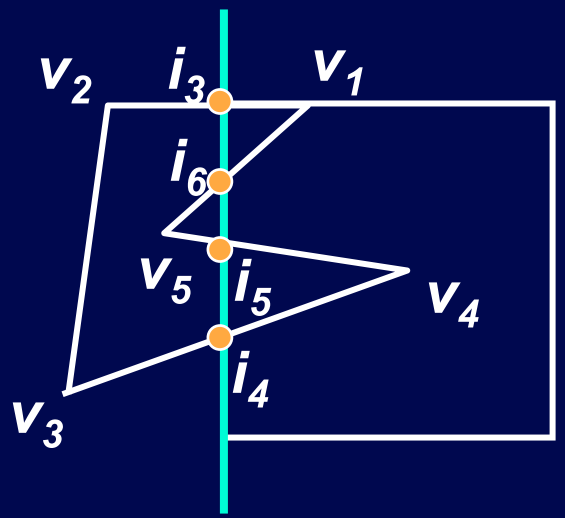


Clipping

Polygon Clipping

Sutherland-Hodgman

Example

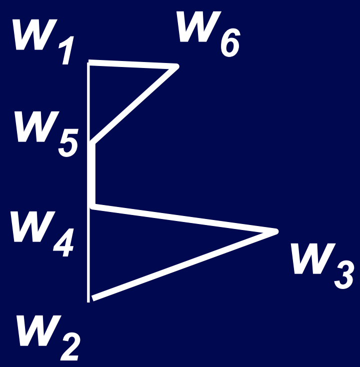


Clipping

Polygon Clipping

Sutherland-Hodgman

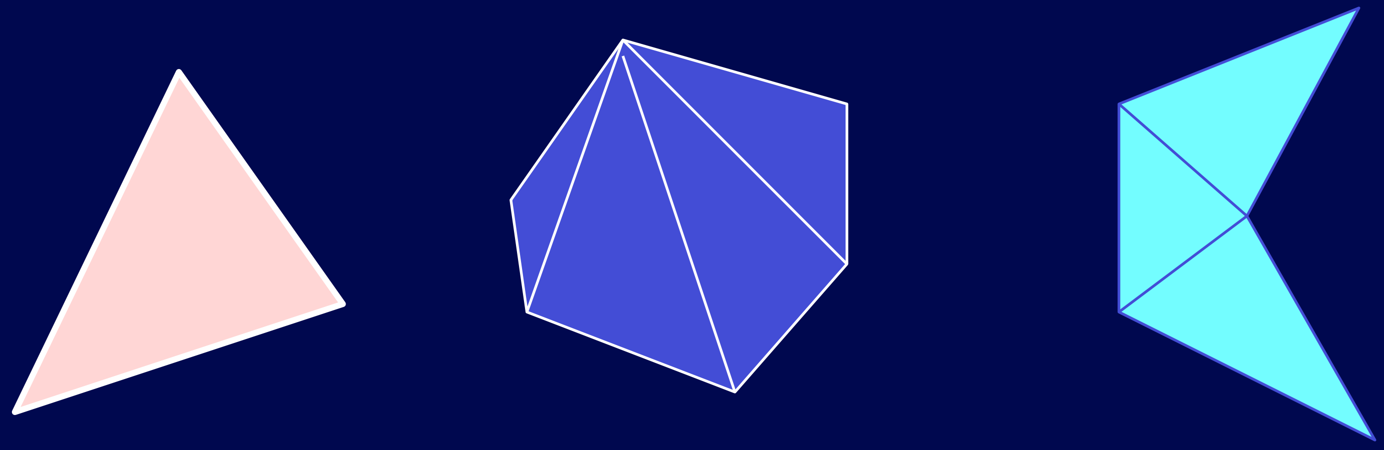
Example



Polygon Scan Conversion

Polygon Filling

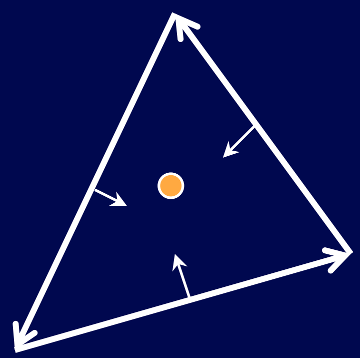
Consider first triangle



Polygon Scan Conversion

Polygon Filling

Consider first triangle

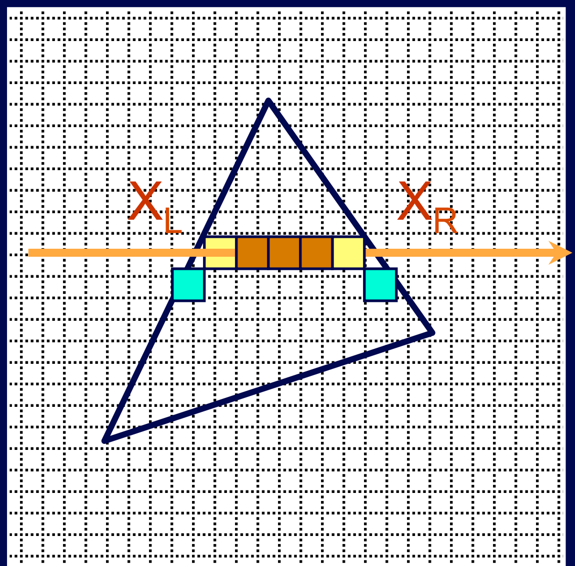


Color all pixels inside triangle
Inside (containment) test

Polygon Scan Conversion

Polygon Filling

Triangle

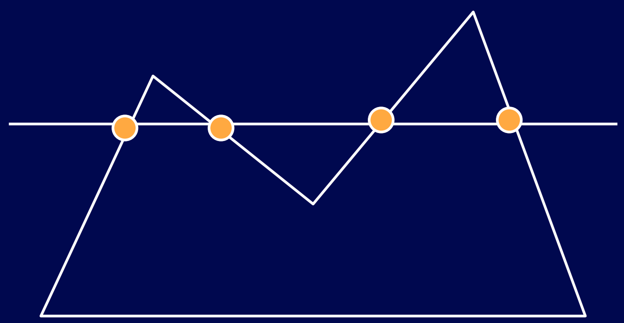


Use horizontal spans.
Process horizontal spans in scan-line order.
For the next spans use edge slopes

Polygon Scan Conversion

Polygon Filling

Polygon

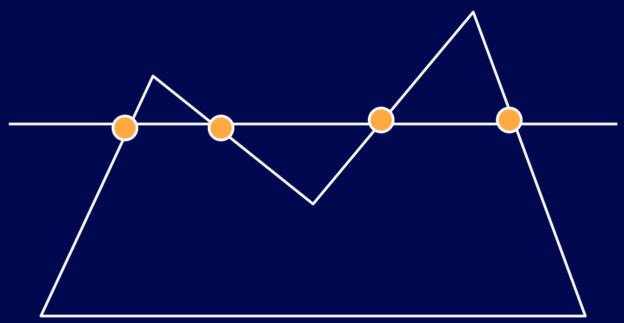


How do we decide what parts of the span should be filled ?

Polygon Scan Conversion

Polygon Filling

Polygon



How do we decide what parts of the span should be filled ?

Parity check

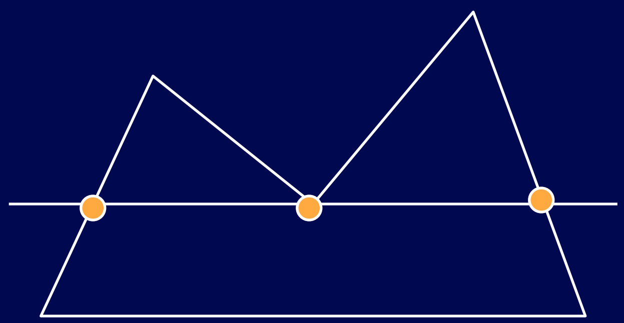
if odd fill

if even don't fill

Polygon Scan Conversion

Polygon Filling

Polygon

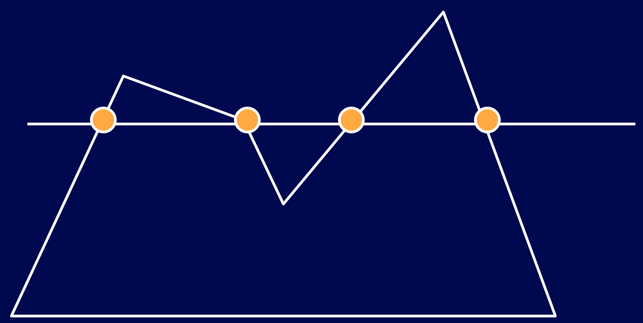


What happens here?

Polygon Scan Conversion

Polygon Filling

Polygon



What happens here?

Polygon Scan Conversion

Polygon Filling

Polygon

Recursive seed filling