Polygon clipping

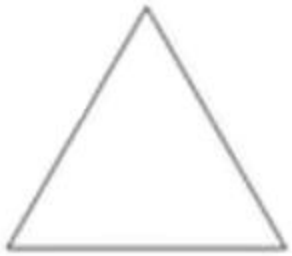
Nikhil K Pawanikar
Department of Information Technology
University of Mumbai
nikhilpawanikar@gmail.com

polygon

- It is a plane figure that is bounded by a finite chain of straight line segments closing in a loop to form a closed chain or *circuit*.
- These segments are called its *edges* or *sides*, and the points where two edges meet are the polygon's *vertices*

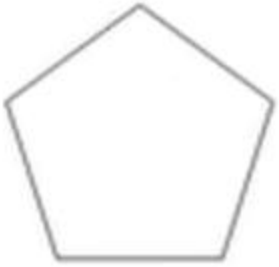
GEOMETRIC SHAPES

POLYGONS



Triangle – 3 sides

Square – 4 sides



Pentagon – 5 sides

Hexagon – 6 sides

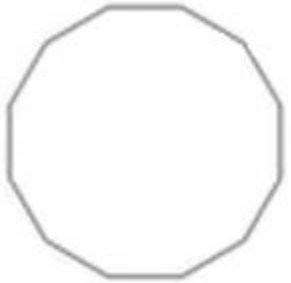
Heptagon – 7 sides

Octagon – 8 sides

Nonagon – 9 sides

Decagon – 10 sides

Hendecagon – 11



Dodecagon – 12

WWW.FUN-STUFF-TO-DO.COM

Types- Concave or Convex

- A **convex** polygon has no angles pointing inwards. More precisely, no internal angle can be more than 180°.
- If any internal angle is greater than 180° then the polygon is **concave**.

  (*Think: concave has a "cave" in it*)

Convex Concave

Polygon Clipping (Sutherland Hodgman Algorithm)

- Sutherland Hodgman Polygon Clipping algorithm
  1. The polygon is stored by its vertices and edges, say v1, v2, v3,……vn and e1, e2, e3,…. En.
  2. Polygon is clipped by a window we need 4 clippers. Left clipper , Right Clipper, Bottom Clipper, Top Clipper
  3. After clipping we get a different set of vertices say v1’ , v2’ , v3’,…… vn’
  4. Redraw the polygon by joining the vertices v1’ , v2’ , v3’ ,……vn’ appropriately.

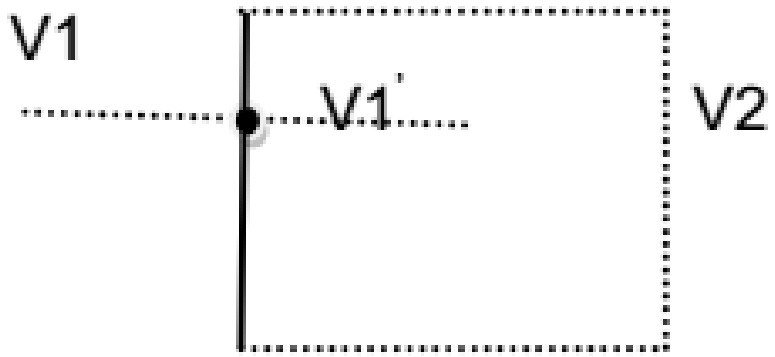
Algorithm

1. Read v1, v2, v3 ,……vn coordinates of polygon.
2. Readcliping window. (Xmin, Ymin)(Xmax, Ymax)
3. For every edge Compare the vertices of each edge of the polygon with the plane taken as the clipping plane.
4. Save the resulting intersections and vertices in the new list according to the possible relationships between the edge and the clipping boundary.
5. Draw the resulting polygon.

The output of the algorithm is a list of polygon vertices all of which are on the visible side of the clipping plane.

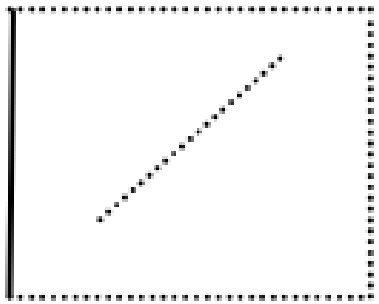
- Every edge is individually compare with the clipping plane.
- This is achieved by considering two vertices of each edge which lies around the clipping boundary or plane.
- This results in 4 possible relationships between the edge and the clipping plane.

- **1st possibility:**
  If the 1st vertex of an edge lies outside the window boundary and the 2nd vertex lies inside the window boundary.
  Here, point of intersection of the edge with the window boundary and the second vertex are added to the output vertex list
  (V1, v2)→(V1', v2)



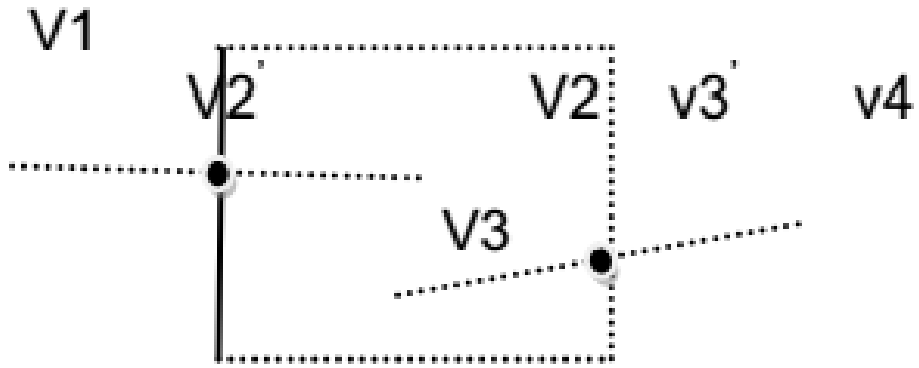
2nd possibility:

If both the vertices of an edge are inside of the window boundary only the second vertex is added to the vertex list



3rd possibility:

If the 1st vertex is inside the window and 2nd vertex is outside only the intersection point is add to the output vertex list.



4th possibility:

If both vertices are outside the window nothing is added to the vertex list.

Once all vertices are processed for one clipped boundary then the output list of vertices is clipped against the next window boundary going through above 4 possibilities. We have to consider the following points.

1) The visibility of the point. We apply inside-outside test.
2) Finding intersection of the edge with the clipping plane.

Text CLipping

- Character or text generation may be done by lines or pixels.
- They may have different style and size and are called fonts or typeface.

types

1. Dot Matrix (bit mapped)
2. Vector (line)
3. Outline

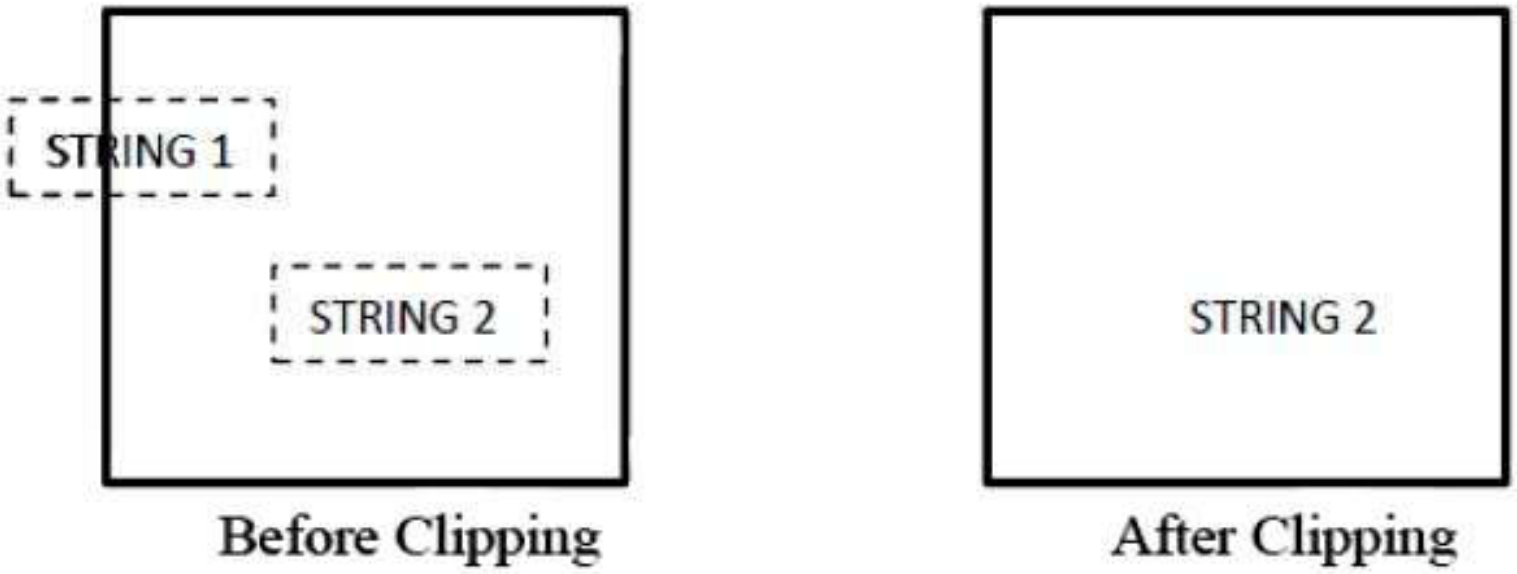
- **Bitmap** fonts consist of a matrix of dots or pixels representing the image of each glyph in each face and size.
- **Outline** fonts (also called **vector** fonts) use Bézier curves, drawing instructions and mathematical formulae to describe each glyph, which make the character outlines scalable to any size.

- **Stroke** fonts use a series of specified lines and additional information to define the *profile*, or size and shape of the line in a specific face, which together describe the appearance of the glyph.

Text Clipping

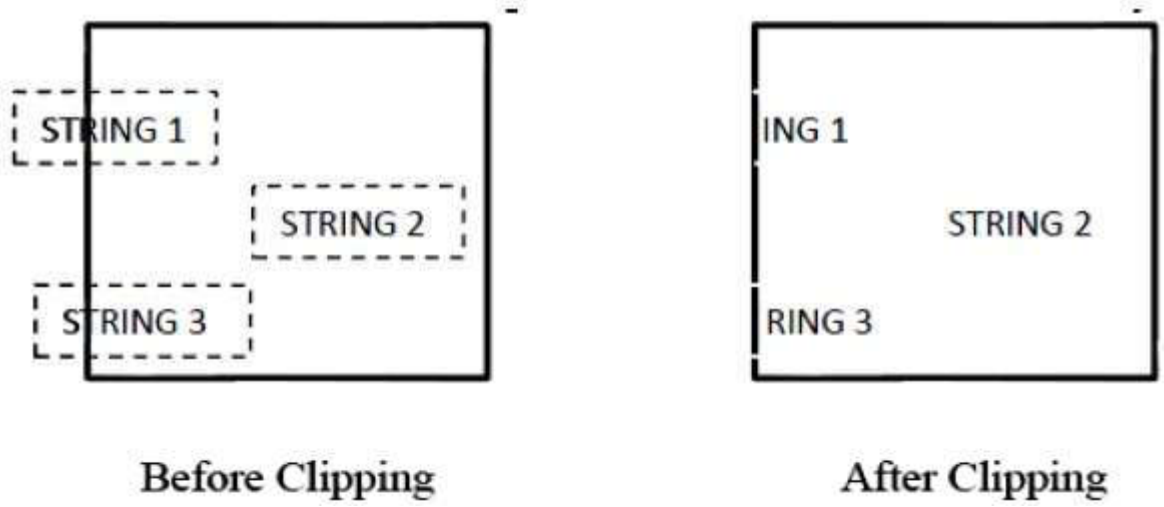
- Various techniques are used to provide text clipping in a computer graphics. It depends on the methods used to generate characters and the requirements of a particular application.
- There are three methods for text clipping which are listed below
  - All or none string clipping
  - All or none character clipping
  - Text clipping

All or none string clipping



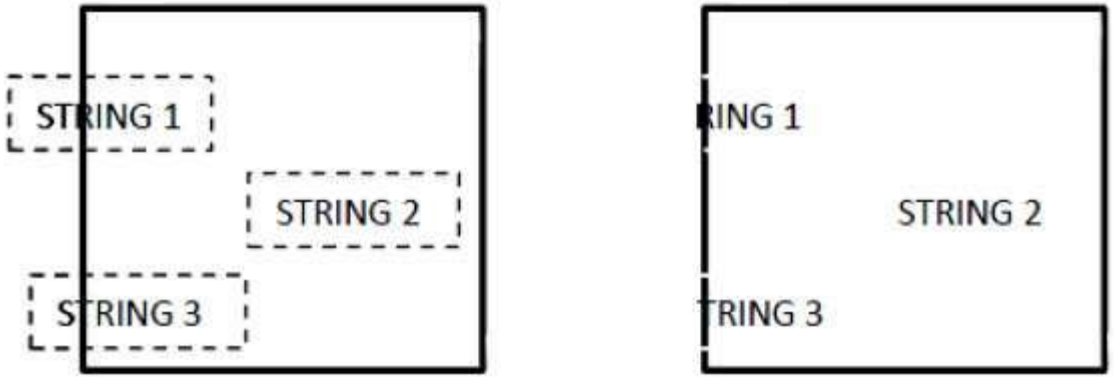
- In all or none string clipping method, either we keep the entire string or we reject entire string based on the clipping window. As shown in the above figure, STRING2 is entirely inside the clipping window so we keep it and STRING1 being only partially inside the window, we reject.

All or none character clipping



- This clipping method is based on characters rather than entire string. In this method if the string is entirely inside the clipping window, then we keep it. If it is partially outside the window, then
- You reject only the portion of the string being outside
- If the character is on the boundary of the clipping window, then we discard that entire character and keep the rest string.

Text clipping



Before Clipping

After Clipping

- This clipping method is based on characters rather than the entire string. In this method if the string is entirely inside the clipping window, then we keep it. If it is partially outside the window, then
- You reject only the portion of string being outside.
- If the character is on the boundary of the clipping window, then we discard only that portion of character that is outside of the clipping window.