In Computer Graphics, Inside Outside is performed to test whether a given point lies inside of a closed polygon or not. Mainly, there are two methods to determine a point is interior/exterior to polygon:

1. Even-Odd / Odd-Even Rule or Odd Parity Rule
2. Winding Number Method



**Even-Odd Rule / Odd Parity Rule**

It is also known as crossing number and ray casting algorithm. The algorithm follows a basic observation that if a ray coming from infinity crosses through border of polygon, then it goes from outside to inside and outside to inside alternately. For every two crossings, point lies outside of polygon.

**Algorithm:**

1. Construct a line segment from point to be examined to point outside of a polygon.
2. Count the number of intersections of line segment with polygon boundaries.
3. If Odd number of intersection, then Point lies inside of Polygon.
4. Else, Point lies outside of polygon.

# even-oddFilled Area Primitives:

Region filling is the process of filling image or region. Filling can be of boundary or interior region as shown in fig. Boundary Fill algorithms are used to fill the boundary and flood-fill algorithm are used to fill the interior.



## Boundary Filled Algorithm:

This algorithm uses the recursive method. First of all, a starting pixel called as the seed is considered. The algorithm checks boundary pixel or adjacent pixels are colored or not. If the adjacent pixel is already filled or colored then leave it, otherwise fill it. The filling is done using four connected or eight connected approaches.



Four connected approaches is more suitable than the eight connected approaches.

**1. Four connected approaches:** In this approach, left, right, above, below pixels are tested.

**2. Eight connected approaches:** In this approach, left, right, above, below and four diagonals are selected.

Boundary can be checked by seeing pixels from left and right first. Then pixels are checked by seeing pixels from top to bottom. The algorithm takes time and memory because some recursive calls are needed.

# Flood Fill Algorithm:

In this method, a point or seed which is inside region is selected. This point is called a seed point. Then four connected approaches or eight connected approaches is used to fill with specified color.

The flood fill algorithm has many characters similar to boundary fill. But this method is more suitable for filling multiple colors boundary. When boundary is of many colors and interior is to be filled with one color we use this algorithm.



In fill algorithm, we start from a specified interior point (x, y) and reassign all pixel values are currently set to a given interior color with the desired color. Using either a 4-connected or 8-connected approaches, we then step through pixel positions until all interior points have been repainted.

## Disadvantage:

1. Very slow algorithm
2. May be fail for large polygons

Initial pixel required more knowledge about surrounding pixels.

**Cell array**

A cell array is a data structure that allows for the storage of data in an array-like format. It is similar to a regular array, but with the added benefit of being able to store data of different types in each individual cell. This makes it ideal for storing data for use in computer graphics applications.

One example of where a cell array would be useful is when storing images for use in an image processing application. Each cell could contain an image, and the application could then operate on all the images in the cell array simultaneously. This would be much more efficient than having to process each image individually.

Another example where a cell array can be used is when storing 3D models for use in a computer game or other 3D applications. Each cell can contain a different model, and the application can then render all the models in the cell array at once. This can save a lot of time, as opposed to having to load and render each model individually.

In computer graphics, a cell array is an array of cells, each of which stores data associated with a particular graphical object. For example, a cell might store the data for a line or polygon, or it might store the data for an image.

### Cell Array **Advantages:**

A cell array is a data structure that is used to store data in an ordered manner. It is similar to a matrix, but it can store data of different types and sizes. A cell array is used in computer graphics to store information about the colors of pixels in an image.

The advantage of using a cell array is that it can store data of different types and sizes, which makes it more flexible than a matrix. The disadvantage is that it takes up more memory than a matrix.

### **Disadvantages:**

There are several disadvantages to using cell arrays in computer graphics. One is that they can be difficult to work with and understand. Another is that they can take up a lot of memory, which can be a problem when working with large images or files. Finally, they can be slow to render, which can impact the overall performance of your system.

# What is a Character Generator (CG)?

In the world of video production, a character generator (CG) is a software application that produces static or animated text for use in 2D and 3D videos. A CG can be used to create anything from simple Lower Thirds text to full-blown 3D animations. A character generator, or CG, is a tool used to create digital characters. These characters can be used in video games, movies, and other digital media. CGs are created by artists who design the characters and then use software to bring them to life. There are many different types of character generators, but the most common one is the 3D character generator. This type of CG allows artists to create realistic-looking characters that can be used in movies and video games. 3D character generators are usually very expensive and require a lot of experience to use.

2D character generators are also common, but they are not as realistic as 3D CGs. 2D CGs are often used for cartoons and other types of artwork. They are usually less expensive than 3D character generators and easier to use. No matter what type of character generator you use, the process of creating a digital character generally follows the same steps: first, the artist designs the character; then, they build the model using software; finally, they animate the character using motion capture or keyframing techniques.

Another great benefit of using a character generator is that you can try out different looks for your characters without having to commit to one right away. This can be helpful if you’re unsure of what kind of look you want for your story, or if you want to experiment with different designs before finalizing anything. You can also easily change up a character’s appearance if you decide you want to go in a different direction later on.

Overall, using a character generator can be a huge time-saver and allow you to create more polished and professional-looking characters than if you were to attempt to create them yourself from scratch. If you’re looking to save time and create high-quality characters for your stories, then using a character generator is definitely something worth considering.

**Types of Character Generators:**

There are several different types of character generators, each with its own unique capabilities. Here are a few of the most common:

1. 2D character generators create two-dimensional characters that can be used in a variety of applications, such as video games or animated films.
2. 3D character generators create three-dimensional characters that can be used in a variety of applications, such as video games or animated films.
3. Motion capture character generators use motion capture technology to record the movement of real people and then generate realistic character animations from that data.
4. Facial recognition character generators use facial recognition algorithms to generate characters that look like specific people or celebrities.

## Advantages of Character Generator:

There are many benefits to using a character generator when creating characters for your stories. Perhaps the most obvious benefit is that it can save you a lot of time. If you’re not experienced in drawing or creating digital art, it can be very time-consuming to create believable and detailed characters. With a character generator, you can simply input your desired characteristics and have a professional-looking character in minutes.

Another great benefit of using a character generator is that you can try out different looks for your characters without having to commit to one right away. This can be helpful if you’re unsure of what kind of look you want for your story, or if you want to experiment with different designs before finalizing anything. You can also easily change up a character’s appearance if you decide you want to go in a different direction later on.

Overall, using a character generator can be a huge time-saver and allow you to create more polished and professional-looking characters than if you were to attempt to create them yourself from scratch. If you’re looking to save time and create high-quality characters for your stories, then using a character generator is definitely something worth considering.

**Types of Character Generators:**

There are several different types of character generators, each with its own unique capabilities. Here are a few of the most common:

1. 2D character generators create two-dimensional characters that can be used in a variety of applications, such as video games or animated films.
2. 3D character generators create three-dimensional characters that can be used in a variety of applications, such as video games or animated films.
3. Motion capture character generators use motion capture technology to record the movement of real people and then generate realistic character animations from that data.
4. Facial recognition character generators use facial recognition algorithms to generate characters that look like specific people or celebrities.

# Anti-Aliasing - Computer Graphics

***Antialiasing is a computer graphics method that removes the aliasing effect***. The aliasing effect occurs when rasterised images have jagged edges, sometimes called "jaggies" (an image rendered using pixels). Technically, jagged edges are a problem that arises when scan conversion is done with low-frequency sampling, also known as **under-sampling**, this under-sampling causes distortion of the image. Moreover, when real-world objects made of continuous, smooth curves are rasterised using pixels, aliasing occurs.

Under-sampling is an important factor in anti-aliasing. The information in the image is lost when the sample size is too small. When sampling is done at a frequency lower than the Nyquist sampling frequency, under-sampling takes place. **We must have a sampling frequency that is at least two times higher than the highest frequency appearing in the image in order to prevent this loss.**

## Anti-Aliasing Methods:

A high-resolution display, post-filtering (super-sampling), pre-filtering (area sampling), and pixel phasing are the techniques used to remove aliasing. The explanations of these are given below:

1. **Using High-Resolution Display** - Displaying objects at a greater resolution is one technique to decrease aliasing impact and boost the sampling rate. When using high resolution, the jaggies are reduced to a size that renders them invisible to the human eye. As a result, sharp edges get blurred and appear smooth.
***Real-Life Applications:***
**For example**, OLED displays and retina displays in Apple products both have high pixel densities, which results in jaggies that are so microscopic that they are blurry and invisible to the human eye.
2. **Post-Filtering or Super-Sampling** - With this technique, we reduce the adequate pixel size while improving the sampling resolution by treating the screen as though it were formed of a much finer grid. The screen resolution, however, does not change. Now, the average pixel intensity is determined from the average of the intensities of the subpixels after each subpixel's intensity has been calculated. In order to display the image at a lesser resolution or screen resolution, we do sampling at a higher resolution, a process known as supersampling. Due to the fact that this process is carried out after creating the rasterised image, this technique is also known as post filtration.
***Real-Life Applications:***
The finest image quality in gaming is produced with SSAA (Super-sample Antialiasing) or FSAA (full-scene Antialiasing). It is frequently referred to as the "**pure AA,"** which is extremely slow and expensive to compute. When no better AA techniques were available, this technique was frequently utilised in the beginning. Other SSAA modes are available, including 2X, 4X, 8X, and others that indicate sampling that is done x times (greater than) the present resolution.
MSAA (multisampling Antialiasing), a quicker and more accurate version of super-sampling AA, is a better AA type.
Its computational cost is lower. Companies that produce graphics cards, such as CSAA by NVIDIA and CFAA by AMD, are working to improve and advance super-sampling techniques.
3. **Pre-Filtering or Area-Sampling** - The areas of each pixel's overlap with the objects displayed are taken into account while calculating pixel intensities in area sampling. In this case, the computation of pixel colour is centred on the overlap of scene objects with a pixel region.
**Example**: Let's say a line crosses two pixels. A pixel that covers a larger amount of a line (90%) displays 90% intensity, whereas a pixel that covers a smaller piece (10%) displays 10-15% intensity. If a pixel region overlaps with multiple colour areas, the final pixel colour is calculated as the average of those colours. Pre-filtering is another name for this technique because it is used before rasterising the image. Some basic graphics algorithms are used to complete it.
4. **Pixel Phasing** - It is a method to eliminate aliasing. In this case, pixel coordinates are altered to virtually exact positions close to object geometry. For dispersing intensities and aiding with pixel phasing, some systems let you change the size of individual pixels.