Rasterization Basics

Image Formation

- Light Source
- Camera
- Image
- World

Reflected Ray
- Incident Ray
- Transmitted Ray

In this model, can you reason about:
- Shadows?
- Area light sources?
- Diffraction?
- Energy transfer?
An image is an array of raster elements called pixels. Every pixel has at least a colour value.

Let's take a closer look.

How is the sphere drawn using the pixels?

To draw a geometrical figure...
...we assign the correct pixels with the correct colour. This process is called rasterization.

Continue the pixel colouring to get regions filled with colour.

The framebuffer is a memory buffer storing the colour value for each pixel displayed.

So the image model mimics the memory model from hardware.

An alternate image model is a vector image model. Bresenham's Line Drawing Algorithm

Function line(x0, y0, x1, y1)
  let delta_x = x1 - x0
  let delta_y = y1 - y0
  let error = 0
  let delta_error = delta_y / delta_x

  // Assume delta_x != 0 (line is not vertical),
  // note that this division needs to be done in a way
  // that preserves the fractional part
  let y = y0
  for x = x0 to x1
    plot(x, y)
    error = error + delta_err
    if error >= 0.5
      y = y + 1
      error = error - 1.0

Bresenham's Line Drawing Algorithm

O(0,0) X

(x0,y0) (x1,y1)
How to colour the correct pixels?

- Extension for all line directions.
- Optimize.
- Demo
- Curves - Read!

Bresenham's Line Drawing Algorithm

```c
function line(int x0, int y0, int x1, int y1)
    int deltax = x1 - x0
    int deltay = y1 - y0
    float error = 0
    float deltaerr = deltay / deltax
    // Assume deltax != 0 (line is not vertical),
    // note that this division needs to be done in a way
    // that preserves the fractional part
    int y = y0
    for x = x0 to x1
        plot(x,y)
        error = error + deltaerr
        if error >= 0.5
            y = y + 1
            error = error - 1.0

    return 0;
```

How to fill pixels?

- Scanfill Algorithm

The Edge List

- Edges in the edge list become active when the y-coordinate of the current scan line matches their $Y_{min}$ value.
- First intersection point between an active edge and a scan line is always the endpoint corresponding to $Y_{max}$. 

Scanfill Algorithm
Inside / Outside

Special Cases

- For monotonically increasing/decreasing edges across a shared vertex count one intersection.
- Else count two.
- Ignore horizontal edges.

Scanfill Algorithm