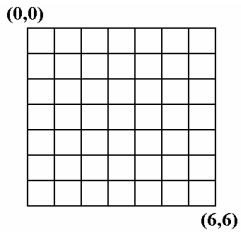
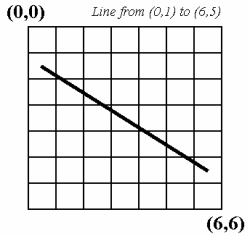
Drawing a line in 2D

- Consider a 7x7 pixel screen

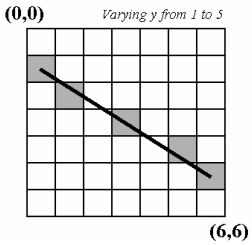


- Draw an oblique line across it

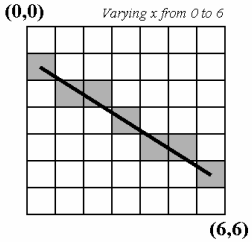


Selecting the pixels

- Indexing down through ***y***

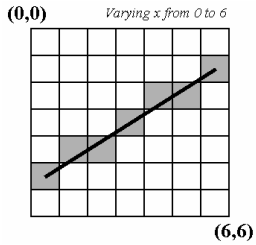


- Indexing across through ***x***



Selecting the pixels

- Indexing across through ***x*** with a negative slope



- Digital Differential Analyser (DDA) Algorithm

$$y_{k+1} = y_k + m$$

$$x_{k+1} = x_k + \frac{1}{m}$$

Bresenham's Algorithm

1. Obtain the two endpoints and place the left hand one in (x_s, y_s)
2. Plot this point
3. Calculate Δx, Δy, $2\Delta y$, $2\Delta x$ and p_0

$$p_0 = 2\Delta y - \Delta x$$

4. Determine p_k for each x_k along the line

$$p_k = p_{k-1} + 2\Delta y - 2\Delta x\left(y_k - y_{k-1}\right)$$

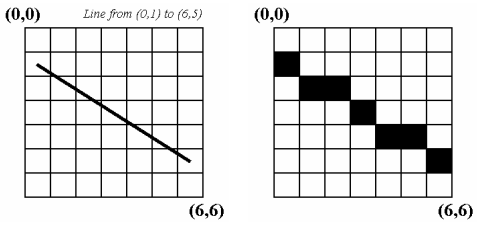
5. If $p_k<0$ then plot (x_{k+1}, y_k) otherwise plot (x_{k+1}, y_{k+1})

Bresenham's Circle Algorithm

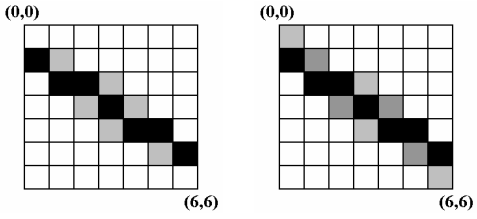
1. Consider a circle of radius r and centre (x_c, y_c)
2. We need only consider one octant - pick the octant just below the x axis
3. Set x_0 to 0 and y_0 to r
4. Plot the point $(x_0 + x_c, y_0 + y_c)$
5. Calculate $p_0 = 3 - 2r$
6. While $x_k < y_k$
   Set $x_{k+1} = x_k + 1$
   If $p_k < 0$ then set $y_{k+1} = y_k$
   and plot the point $(x_{k+1}+x_c, y_{k+1}+y_c)$
   and set $p_{k+1} = p_k + 4x_k + 6$
   Else set $y_{k+1} = y_k - 1$
   and plot the point $(x_{k+1}+x_c, y_{k+1}+y_c)$
   and set $p_{k+1} = p_k + 4x_k - 4y_k + 10$

Aliasing and Anti-aliasing

- A line suffering from aliasing



- Anti-aliased versions

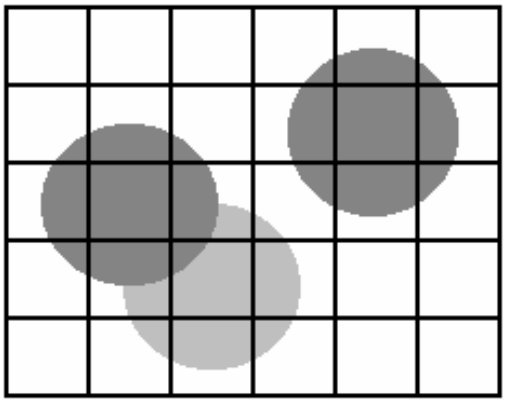


Anti-aliasing Methods - Filtering

- Method
  - Define an area of influence for each pixel
  - Determine the proportion of the area of influence lying within the rectangular line
  - Specify the filtering function to determine the illumination from this proportion
- Drawback
  - Filter function, area of influence, palette size are all fixed in pre-calculated tables

Filtering

- Circular areas of influence for a conical filter function



Anti-aliasing Methods - Supersampling

- Method
  - Break each pixel down into sub-pixels
  - Count number of sub-pixels whose top-left corner, say, lies within the rectangular line
  - Proportionally illuminate pixels based on number of sub-pixels counted
- Drawback
  - Line perceived to shift in direction of chosen reference points (e.g. top-left)

Supersampling

- Pixels divided into arrays of 3x3 sub-pixels