Computer Graphics

Romrawin Chumpu 639506093 (Jinpu)

Homework 2: Bresenham's Line Algorithm

From class slide pseudocode,

Bresenham's Line-Drawing Algorithm for $|m| < 1$

1. **Input the two line endpoints and store the left endpoint in (x_0, y_0).**
2. **Load (x_0, y_0) into the frame buffer; that is, plot the 1st point.**
3. **Calculate constants Δx, Δy, $2\Delta y$, and $2\Delta y - 2\Delta x$, and obtain the starting value for the decision parameter as**
$$p_0 = 2\Delta y - \Delta x$$
4. **At each x_k along the line, starting at $k = 0$, perform the following test:**
If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and
$$p_{k+1} = p_k + 2\Delta y$$
Otherwise, the next point to plot is $(x_k + 1, y_k + 1)$ and
$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$
5. **Repeat Step 4 Δx times.**

Assumptions

1. Line is drawn from left to right
2. Point x0 < x1 and y0 < y1
3. Slope of the line is between 0 and 1.

Python Code Implementation

Colab notebook: [Bresenham's Line Algorithm]

```
def BresenhamLine(x0,y0,x1, y1):
  # Different x, y
  dx = x1-x0
  dy = y1-y0

  m = dy/dx
  # Assumption 1:
  if m > 0:
    # Set intial point
    two_dy = 2 * dy
    p0 = two_dy - dx

    # set y0, x0
    y = y0
    x = x0
    k = 0
    pks = []
```

```
    xs, ys = [], []
    pks.append(p0); xs.append(x), ys.append(y)
    print("Initial | p_(k) = %d \t (%d, %d)"%(p0, x, y))
    print("-"*30)
    while True:
      # Conditional incrementation
      if p0 >= 0:
        p0 += two_dy - (2*dx)
        y += 1
        x += 1
      else:
        p0 += two_dy
        x += 1
      # Check whether the point reach the terminal (x1, y1)
      if x > x1 and y > y1:
        break
      # Print the result
      pks.append(p0); xs.append(x), ys.append(y)
      print("k %d | p_(%d) = %d \t (%d, %d)"%(k, k, pks[-2], x, y))
      k+=1
    print("-"*30)

    # Display the line
    sns.regplot(x = xs, y = ys, color="black")
    plt.xlabel("x")
    plt.xlabel("y")
    plt.title("Bresenham's Line")
    plt.show()
  else:
    print("Initial point does not follow assumption 1.")
```

Output:

Check if it is valid to assumption 1 or not

```
BresenhamLine(15, 30, 30, 20)
```

```
Initial point does not follow assumption 1.
```

Run BresenhamLine function

```
BresenhamLine(15, 10, 30, 20)
```

```
Initial | p_(k) = 5      (15, 10)
-------------------------------
k 0 | p_(0) = 5          (16, 11)
k 1 | p_(1) = -5         (17, 11)
k 2 | p_(2) = 15         (18, 12)
k 3 | p_(3) = 5          (19, 13)
k 4 | p_(4) = -5         (20, 13)
k 5 | p_(5) = 15         (21, 14)
k 6 | p_(6) = 5          (22, 15)
k 7 | p_(7) = -5         (23, 15)
k 8 | p_(8) = 15         (24, 16)
k 9 | p_(9) = 5          (25, 17)
k 10 | p_(10) = -5       (26, 17)
k 11 | p_(11) = 15       (27, 18)
k 12 | p_(12) = 5        (28, 19)
k 13 | p_(13) = -5       (29, 19)
k 14 | p_(14) = 15       (30, 20)
-------------------------------
```