# Computer Graphics

Romrawin Chumpu 639506093 (Jinpu)

## Homework 4: Mid-Point Circle Drawing Algorithm

From class slide pseudocode,

**Midpoint Circle Algorithm**

1. ป้อนค่ารัศมี $r$ และจุดศูนย์กลางวงกลม $(x_c, y_c)$ แล้วกำหนดพิกัด **(coordinates)** สำหรับจุดแรกบนเส้นรอบวงของวงกลมที่มีจุดศูนย์กลางอยู่ที่จุด **origin** เป็น
$$(x_0, y_0) = (0, r)$$
2. คำนวณค่าเริ่มต้นของ **decision parameter** เป็น
$$p_0 = \frac{5}{4} - r$$
3. สำหรับแต่ละตำแหน่ง $x_k$ โดยเริ่มจาก $k = 0$ ให้ทำการทดสอบต่อไปนี้
   a. ถ้า $p_k < 0$ จุดถัดไปบนวงกลมที่มีจุดศูนย์กลางอยู่ที่ $(0, 0)$ คือจุด $(x_{k+1}, y_k)$ และ
   $$p_{k+1} = p_k + 2x_{k+1} + 1$$
   b. สำหรับกรณีอื่น จุดถัดไปบนวงกลม คือจุด $(x_k + 1, y_k - 1)$ และ
   $$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$
   โดยที่ $2x_{k+1} = 2x_k + 2$ และ $2y_{k+1} = 2y_k - 2$
4. กำหนดจุดสมมาตรในอีก **7 octants** ที่เหลือ
5. ย้ายแต่ละตำแหน่งของ **pixel** $(x, y)$ ที่คำนวณได้ ไปไว้บนเส้นของวงกลมที่มีจุดศูนย์กลางอยู่ที่ $(x_c, y_c)$ และ **plot** ค่าพิกัดใหม่เป็น
$$x = x + x_c, \quad y = y + y_c$$
6. ทำซ้ำตั้งแต่ ขั้นตอนที่ 3 ถึง 5 จนกระทั่ง $x \geq y$

## Python Code Implementation

Colab notebook: [Mid-Point Circle Algorithm]

```
def midPointCircle(x_c, y_c, radius):
  x = 0
  y = radius

  xs, ys = [], []
  # Print initial points
  print("Centre point: (%d, %d)"%(x+x_c, y+y_c))

  # Check if radius > 0
  update_x = [x+x_c, -x+x_c, x+x_c, -x+x_c, y+y_c, y+y_c, -y+y_c, -y+y_c]
  update_y = [y+y_c, y+y_c, -y+y_c, -y+y_c, x+x_c, -x+x_c, x+x_c, -x+x_c]
  if (radius > 0) :
    for i in range(8):
      print("(%d, %d)"%(update_x[i], update_y[i]), end=" ")
      xs.append(update_x[i])
      ys.append(update_y[i])
    print()
```

```
  # Initialising the value of P
  p_0 = 1 - radius
  print("(x, y)= (%d, %d)"%(x, y))
  print("-"*30)
  while x < y:
    x += 1
    if p_0 < 0:
      print("p_0 = %d"%(p_0), end=" | ")
      p_0 += (2*x)+1
      update_x = [x+x_c, -x+x_c, x+x_c, -x+x_c,
                  y+y_c, y+y_c, -y+y_c, -y+y_c]
      update_y = [y+y_c, y+y_c, -y+y_c, -y+y_c,
                  x+x_c, -x+x_c, x+x_c, -x+x_c]
      print("(x, y= (%d, %d)"%(x, y))
      for i in range(8):
        print("(%d, %d)"%(update_x[i], update_y[i]), end=" ")
        xs.append(update_x[i])
        ys.append(update_y[i])
      print()
      print("-"*30)
    else:
      y -= 1
      print("p_0 = %d"%(p_0), end=" | ")
      p_0 += (2*x)+1-(2*y)
      update_x = [x+x_c, -x+x_c, x+x_c, -x+x_c,
                  y+y_c, y+y_c, -y+y_c, -y+y_c]
      update_y = [y+y_c, y+y_c, -y+y_c, -y+y_c,
                  x+x_c, -x+x_c, x+x_c, -x+x_c]
      print("(x, y)= (%d, %d)"%(x, y))
      for i in range(8):
        print("(%d, %d)"%(update_x[i], update_y[i]), end=" ")
        xs.append(update_x[i])
        ys.append(update_y[i])
      print()
      print("-"*30)
else:
  print("Radius < 0, one dot is returned")
  xs.append(x_c)
  ys.append(y_c)

# Plot
fig = plt.figure(figsize=(5, 5))
plt.xlabel("x")
plt.ylabel("y")
plt.scatter(xs, ys)
plt.show()
```

**Output:**

```
midPointCircle(0, 0, 10)
```

```
Center point: (0, 10)
(0, 10) (0, 10) (0, -10) (0, -10) (10, 0) (10, 0) (-10, 0) (-10, 0)
(x, y)= (0, 10)
------------------------------
p_0 = -9 | (x, y= (1, 10)
(1, 10) (-1, 10) (1, -10) (-1, -10) (10, 1) (10, -1) (-10, 1) (-10, -1)
------------------------------
p_0 = -6 | (x, y= (2, 10)
(2, 10) (-2, 10) (2, -10) (-2, -10) (10, 2) (10, -2) (-10, 2) (-10, -2)
------------------------------
p_0 = -1 | (x, y= (3, 10)
(3, 10) (-3, 10) (3, -10) (-3, -10) (10, 3) (10, -3) (-10, 3) (-10, -3)
------------------------------
p_0 = 6 | (x, y)= (4, 9)
(4, 9) (-4, 9) (4, -9) (-4, -9) (9, 4) (9, -4) (-9, 4) (-9, -4)
------------------------------
p_0 = -3 | (x, y= (5, 9)
(5, 9) (-5, 9) (5, -9) (-5, -9) (9, 5) (9, -5) (-9, 5) (-9, -5)
------------------------------
p_0 = 8 | (x, y)= (6, 8)
(6, 8) (-6, 8) (6, -8) (-6, -8) (8, 6) (8, -6) (-8, 6) (-8, -6)
------------------------------
p_0 = 5 | (x, y)= (7, 7)
(7, 7) (-7, 7) (7, -7) (-7, -7) (7, 7) (7, -7) (-7, 7) (-7, -7)
------------------------------
```