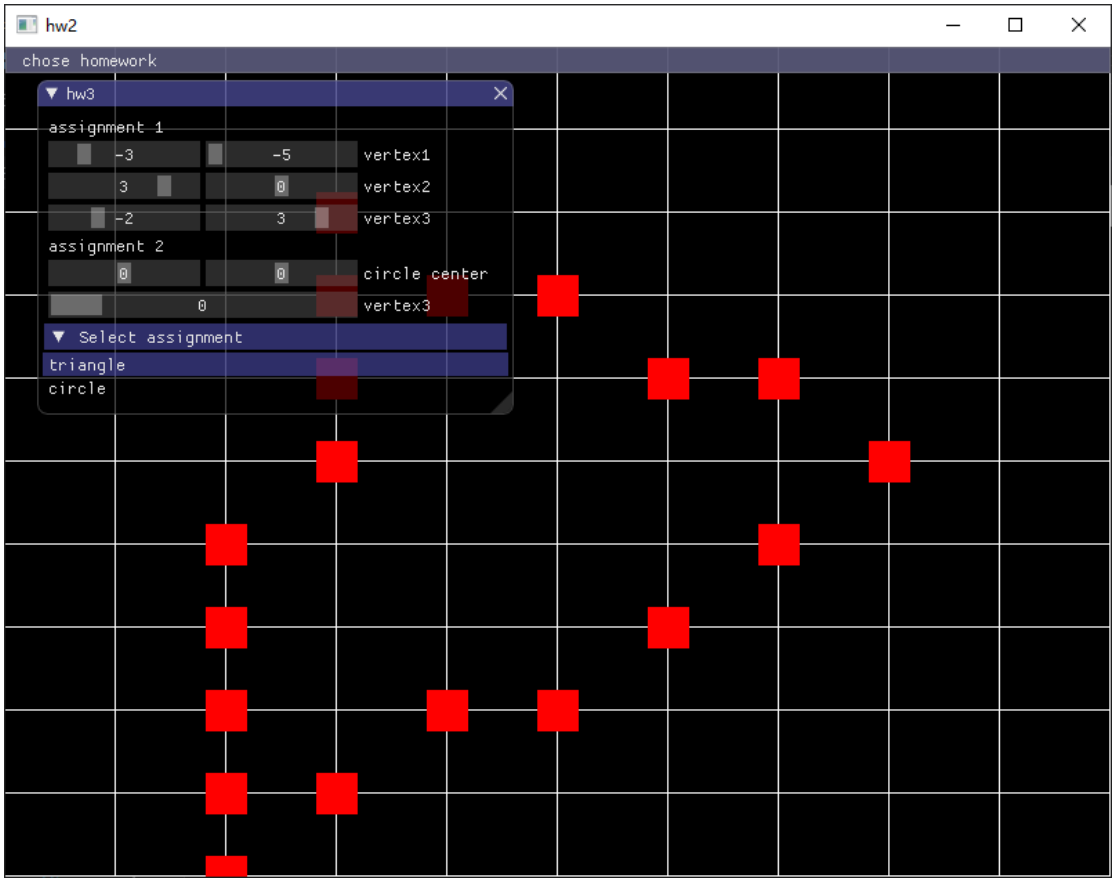
HW3

胡嘉鹏 16340076

1



输入顶点（-3，-5），（3，0），（-2，3），输出对应三角形。

实现思路：

将三角形的绘制分解到 3 条直线的绘制上，每条直线的绘制使用 *Bresenham's line algorithm* 。

`void HW3::draw_triangle()` ：绘制三角形。

`void HW3::plotLine(int x0, int y0, int x1, int y1)`：绘制直线

`void HW3::plotLineLow(int x0, int y0, int x1, int y1)`：斜率绝对值小于 1 的情况

`void HW3::plotLineHigh(int x0, int y0, int x1, int y1)`：斜率绝对值大于 1 的情况

`void HW3::plot(float x, float y)`：绘制坐标为（x，y）的点

算法解释：

Bresenham's line algorithm

（1）对于直线斜率为正且小于 1 的情况：

1. 计算detax, detay, $p = 2 * detax - detay$
2. y = y0, yi = 1
3. for x from x0 to x1
4. plot(x, y)
5. if p ≤ 0
6. y = y + yi
7. p = p − 2 ∗ detax
8. p = p + 2 ∗ detay

（2）对于直线斜率小于 0 但大于-1 的情况：

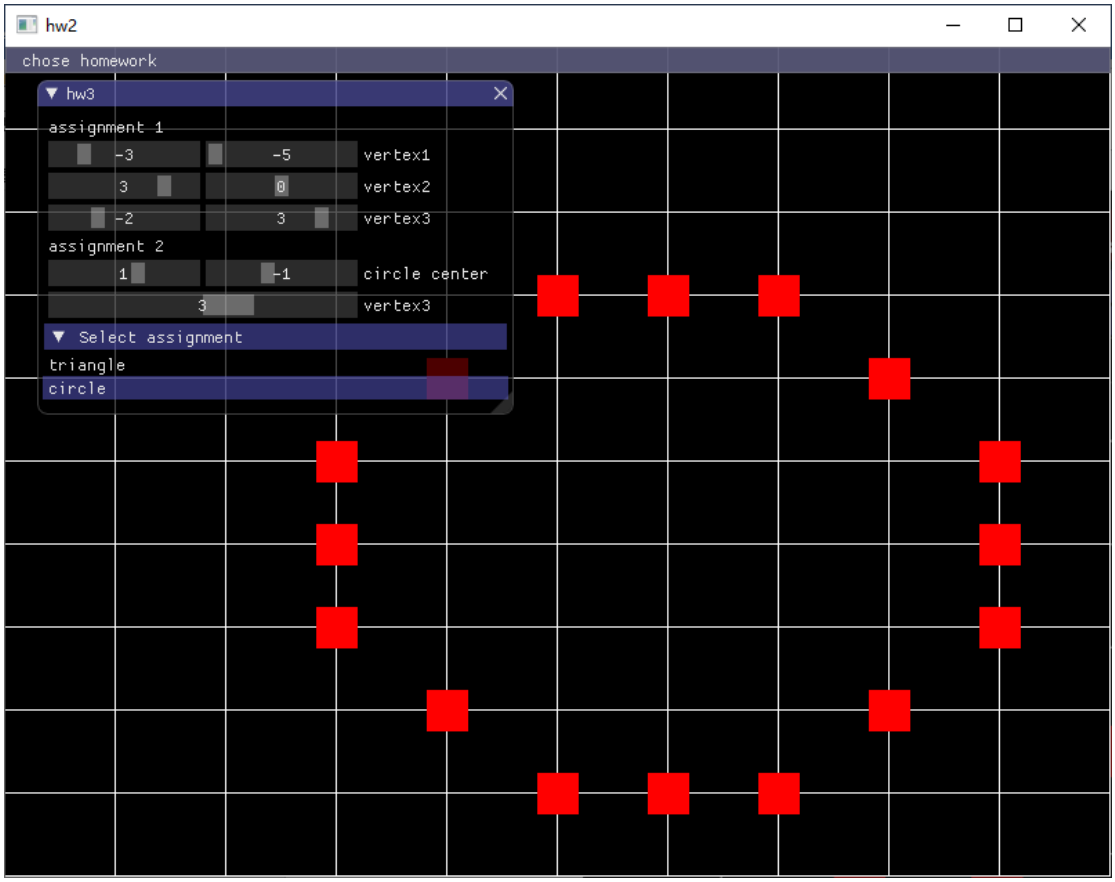
在（1）的基础上把 y 的递增改为递减，detay取正值

1. yi = −1
2. detay = −detay

（3）对于直线斜率大于 1 的情况：

将 x 和 y 调换，再执行（1）的步骤

2



输入圆心为（1，-1），半径为 3，输出对应位置大小的圆形。

实现思路

使用 *Bresenham's circle algorithm* 计算出 1/8 的圆形的点的坐标，再利用圆的对称性计算出剩下 7/8 的点的坐标，最后再根据圆心的坐标平移所有的点。

`void HW3::draw_circle()`：使用 *Bresenham's circle algorithm* 计算出 1/8 圆的坐标

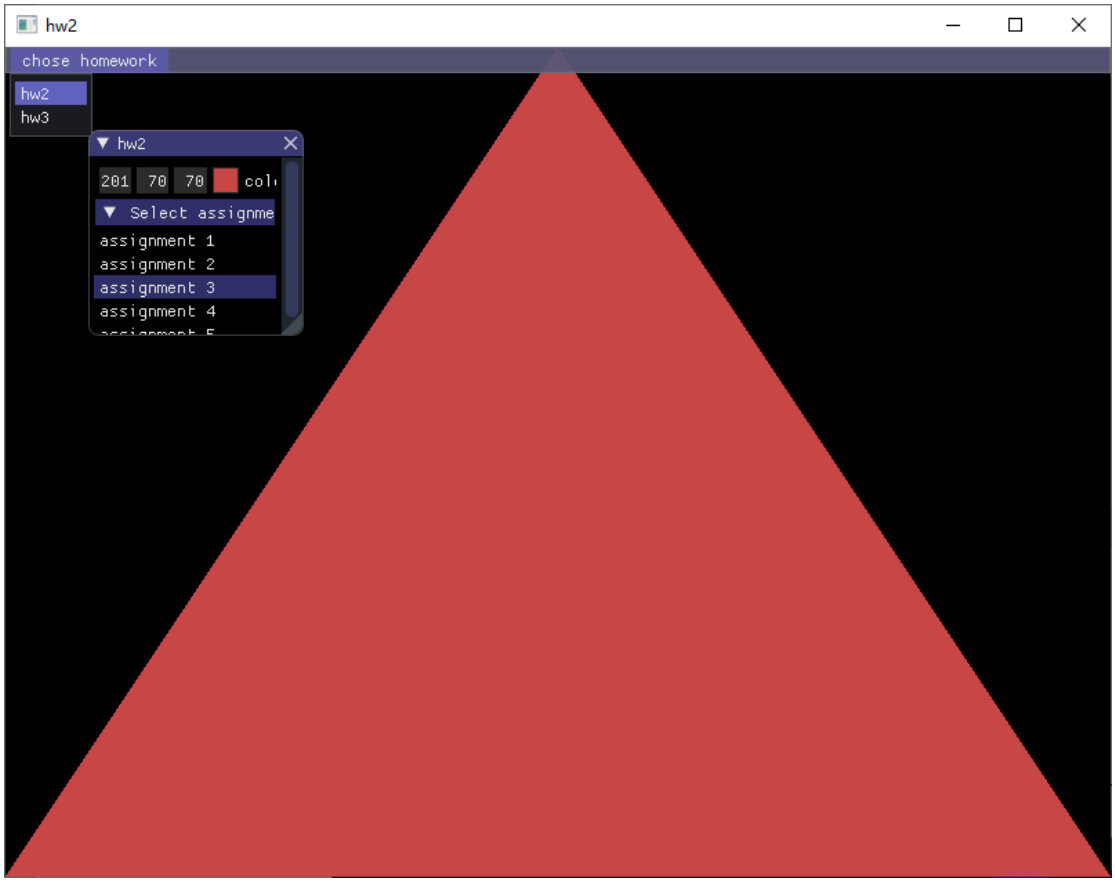
`void HW3::plot_symmetry(int x, int y)`：根据对称性求出剩下点的坐标，并根据圆心坐标平移，再分别绘制

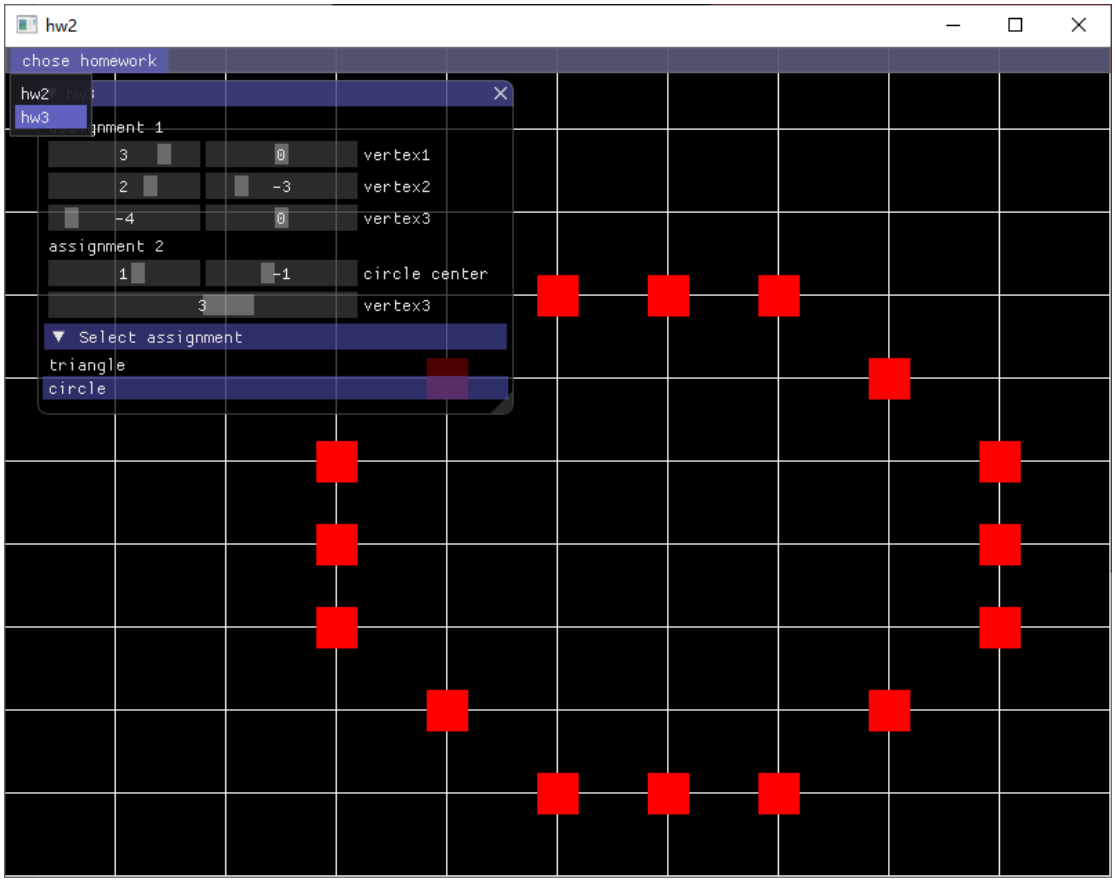
Bresenham's circle algorithm

1. $x = 0, y = r, p = 3 - 2 * r$
2. while $x \leq y$
3. plot_symmetry(x, y)
4. If $p < 0$
5. $p = p + 4 * x + 6$
6. $x = x + 1$
7. Else
8. $p = p + 4 * (x - y) + 10$
9. $x = x + 1$
10. $y = y - 1$

3

添加菜单栏选择 hw，hw3 内部可以通过 select assignment 选择是绘制圆形还是三角形。


hw2
chose homework
hw2
hw3
hw2
201 70 70 col
Select assignme
assignment 1
assignment 2
assignment 3
assignment 4


hw2
chose homework
hw2
hw3
3 0 vertex1
2 -3 vertex2
-4 0 vertex3
assignment 2
1 -1 circle center
3 vertex3
Select assignment
triangle
circle