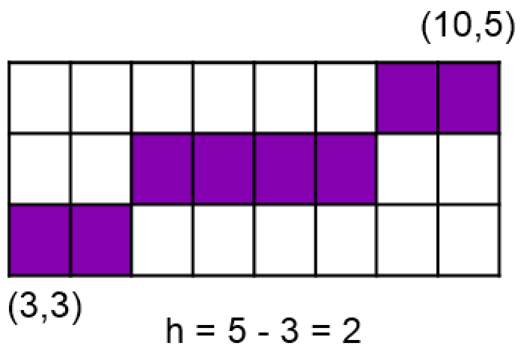
Week 9 Tutorial Solutions

Question 1

a) Use Bresenham's algorithm to draw a line from P(3,3) to Q(10,5). Repeat for the line (3,3) to (9,6)



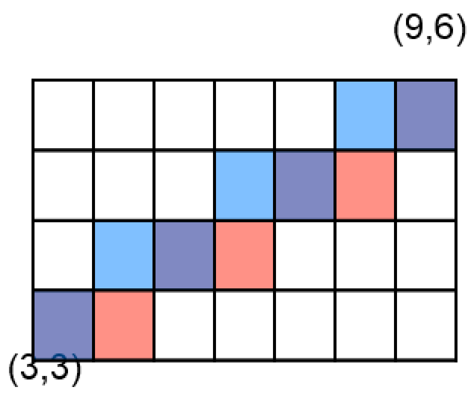
h = 5 - 3 = 2
w = 10 - 3 = 7

Initally:
F = 2h -w
= -3

Steps:
2h = 4
2h - 2w = -10

x	y	F
3	3	-3
4	3	1
5	4	-9
6	4	-5
7	4	-1
8	4	3
9	5	-7
10	5	-3

Ignore the pink pixels in this answer, it is part of the answer for question 1b.



h = 6 - 3 = 3
w = 9 - 3 = 6

Initially:
F = 2h -w
= 0

Steps:
2h = 6
2h - 2w = -6

x	y	F
3	3	0
4	4	-6
5	4	0
6	5	-6
7	5	0
8	6	-6
9	6	0

b) Consider the symmetrical problem of drawing for Q to P. What changes do you need to make to the algorithm to guarantee you get the same set of pixels?

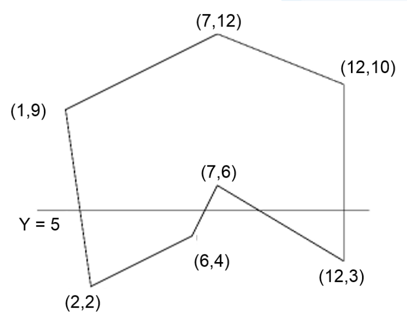
The first line was symmetrical, so there would be no difference drawing the line the other way from (10,5) to (3,3). However this is not the case with the second line. Asymmetries occur when you have a point of the line with F=0 and you must choose whether to increment at this point.

The pink pixels show the line from (3,3) to (9,6) which does step when F=0. The overlapping blue line goes from (9,6) to (3,3) and does step while F=0. Clearly they are not the same line.

To make sure lines are drawn the same way the code for lines the other way must be adjusted so that when tracing the line in the other direction the F=0 case is combined with the if condition that does not cause a decrement in y.

Question 2

Consider the scan line algorithm described in class. For the polygon drawn here, what does the edge list look like? What does the Active Edge List (AEL) look like just before filling the pixels on scan line 5? What pixels will be filled?



The Edge List

The inc is the inverse gradient, which is used to increment the x value later in the algorithm.

y in	x	inc	y out
2	2	2	4
2	2	-1/7	9
3	12	-5/3	6
3	12	0	10
4	6	½	6
9	1	2	12
10	12	-5/2	12

Active Edge List

By the time we get to a Y of 5, the first active edge has been incremented 3 times (as it started at y=2) and so the X is now 2 - 1/7 -1/7 -1/7. Similar calculations were used for the rest of the list.

x	inc	y out
1 4/7	-1/7	9
6 ½	1/2	6
8 2/3	-5/3	6
12	0	10

(2,5) (3,5)(4,5)(5,5)(6,5) ,(9,5),(10,5),(11,5)

The general rules are:

- **On the left edge, round up to the nearest integer, with round(n) = n if n is an integer.**
- **On the right edge, round down to the nearest integer, but with round(n) = n-1 if n is an integer.**

Question 3:

Suppose you are interested in implementing a particle system to represent a fire. How would your particles evolve over time? What variables would you want to change? How would their values change over time?

Making a realistic fire involves a lot of subtle effects, but in the very least you would want:

- **Particles move upwards (constant speed looks reasonable)**
- **Particles decrease in size**
- **Particles change colour from yellow/white to dull red.**

An appropriate animated sprite for fire particles could handle size and colour transitions for you and provide greater realism.

Question 4:

In the assignment specification it says for the particle system "For the full marks this would need to include alpha blended billboarded particles, creation and destruction ,some kind of evolution over time (position, size, colour, as is appropriate for your kind of particles)."

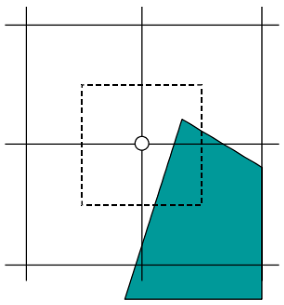
Discuss what alpha blended billboarded particles are.

Particles that are partially transparent (alpha blended) and are 2d texured quads that are transformed so they are always facing the camera (billboarding). They need to face the camera so that as the user moves the camera around they can not see that the particles are simply 2d textured quads.

What are some issues that may arise when implementing blending in opengl? What are some ways these issues can be overcome?

The interaction between using the depth buffer for hidden surface removal and drawing translucent polygons in opengl can cause undesired results. For example if a transparent polygon is drawn first, a polygon that is behind it will never be drawn if the depth buffer is on. One solution is to draw all opaque polygons first and then draw the transparent polygons last. If you are drawing many transparent polygons (as in the case of a particle system) to do the it accurately you would need to sort your polygons and draw in back to front order. You may instead wish to turn off depth buffer writing using gl.glDepthMask(GL3.GL_FALSE); while you are drawing your transparent polygons. (Remember to set it back to true again for rendering the next frame). This may give results that look ok.

Question 5



Suppose we are determining the intensity of the pixel above (The pixel is shown by the dotted line box) with a value between 0 and 1. What value will the pixel have if sampling is performed by:

- Sampling in the centre of the pixel **0**
- Sampling at the corners and taking the average **1/4**
- Double sampling and taking the average **2/9**
- Double sampling and using the following mask to perform a weighted average

```
1/16 1/16 1/16
1/16 1/2  1/16
1/16 1/16 1/16
```

1/16 + 1/16 = 1/8

Assignment Use any left over time to discuss the initial milestone of the assignment with your tutor