Report

LeshanWang Leeds ID: 201388927SWJTU ID:2019110023

1. Assignment 1 and Assignment 2

In function **DrawLine** takes four floating point values as the coordinates of two points, and two RGB values in RGBVal struct format for the color of the points. In order to draw a line which covers every pixel that contains the mathematical ideal of the line, the parameterised version of a line is used and the strp sizes are calculated.

The parameterised version of a line can be expressed as P = (1-t)*A + t*B, where A and B are the two points define the line, P is the point which is on the line and t is the parameter. When 0<=t<=1, every calculated P is considered to be on segment AB, including A and B.

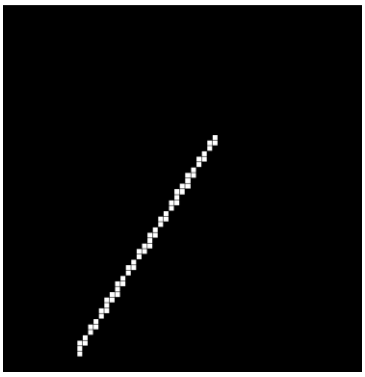
However, t is a changing variable, therefore the change rate of t, which is the step, can be neither too large or too small. If too large, some points P will be missed. If too small, some points will be overly lighted, resulting in lower efficiency.

After several attempts, the recipocal of the double of line length is implemented as step size. Its coding format is s = 1 / 4 * (floor(sqrt(pow(abs(int(x0-x1)),2) + pow(abs(int(y0-y1)),2)))). The reason for using this function is that, by observation and testing on various occasions, the number of pixels that should be lighted is larger than the physical distance of two vertices, but smaller than the quad of it, because the neighbors of a couple of lighted pixels will not exceed four. Although this is not the best way to calculate the step size, it is relatively efficient and is certain to light all the pixels that meet requirement.

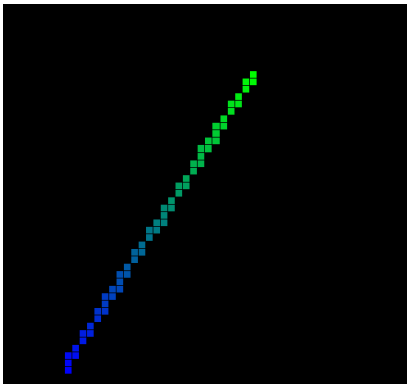
For gradient color interpolation, two RGB values of stating and ending point are entered as parameters, then their difference is calculated and divided into equal small floats as deltas, every time a point is colored, red, green, and blue color values are added with color deltas accordingly. By doing this, the line will have a gradual change of color.

Finally, when drawing pixels, the function **DefinePixelValues** is called within loops.

Result for assignment 1:



Result for assignment 2:



2. Assignment 3

The function called **Barycentric** is created which takes three points and an RGBVal struct to paint a triangle. It uses barycentric coordinates to interpolate pixels. For triangle ABC, a point P belonging to it can be expressed in format of barycentric as:

$$Px = w1*Ax+w2*Bx+w3*Cx$$

$$Py = w1*Ax+w2*Bx+ w3*Cx$$

$$w1+w2+w3=1$$

As $\alpha,\ \beta,\ \gamma$ are all in the range of 0 to 1, P is certain to be a part of triangle ABC. The functions above can be processed through mathematical reasoning as:

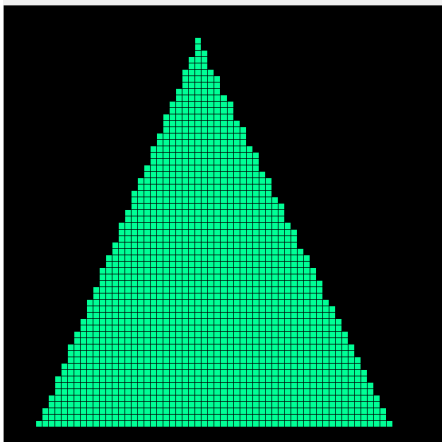
```
w1 = ((y0-y1)*px+(x1-x0)*py+x0*y1-x1*y0)/((y0-y1)*x2+(x1-x0)*y2+x0*y1-x1*y0);
```

w2 = ((y0-y2)*px+(x2-x0)*py+x0*y2-x2*y0)/((y0-y2)*x1+(x2-x0)*y1+x0*y2-x2*y0);

w3 = 1-w1-w2;

Again, DefinePixelValues is called to paint pixels.

Result of assignment 3:



3. **Assignment 4**

Several functions are created to realize the meet the requirements of this assignment.

Function **hpt** is created to implement the half plane test for points given. It takes four points, the first three are the vertices of the triangle, and the last one is the point which needs to be judged whether it's inside the triangle. For example, there is a triangle ABC and a point P. If P and A are the same side of B and C, P and B are on the same side of A and C, P and C are on the same side of A and B, then P is certainly inside triangle ABC.

Thus, expressions of line AB, AC and BC must be calculated. Then insert the x and y value of P and one vertex, if the product of their results is positive, it means they are on the same side.

There is an exception when Two of the vertices have the same y coordinate, which slope cannot be calculated. In this scenario, point half plane test can be conducted by checking whether the product of P's and one vertex's distance to the line is positive, which indicates they are on the same side.

Function **IsInside** is designed to run function **hpt** three times to fulfill all possible scenarios, thus, to determine whether point P is inside triangle ABC. If every test indicates P with one vertex is on the same side of the line formed by the other two vertices, the function will return true as result; otherwise return false.

Fucntion **bc** serves to integrate the two functions above, which two together conduct the half plane test for point P; **bc** also uses the barycentric coordinates to judge whether P is inside the triangle ABC, as a parallel method to half plane test.

Results from half plane test and barycentric coordinate formula are compared, if they are both true or false at the same time, that means the judging of position of P is correct. Afterwards, the results of w1, w2, w3, result from half plane test, result from barycentric coordinate formula, and the comparison result, are all written into a csv file.

A snapshot of the file generated:

A	B	C	D	E	F	G	H	I	J	K
-0.03	0.36	0.67	FALSE	FALSE	TRUE					
-0.02	0.37	0.65	FALSE	FALSE	TRUE					
-0.01	0.38	0.63	FALSE	FALSE	TRUE					
0	0.39	0.61	TRUE	TRUE	TRUE					
0.01	0.4	0.59	TRUE	TRUE	TRUE					
0.02	0.41	0.57	TRUE	TRUE	TRUE					
0.03	0.42	0.56	TRUE	TRUE	TRUE					
0.04	0.43	0.54	TRUE	TRUE	TRUE					
0.05	0.44	0.52	TRUE	TRUE	TRUE					
0.05	0.45	0.5	TRUE	TRUE	TRUE					
0.06	0.46	0.48	TRUE	TRUE	TRUE					
0.07	0.47	0.46	TRUE	TRUE	TRUE					
0.08	0.48	0.44	TRUE	TRUE	TRUE					
0.09	0.49	0.42	TRUE	TRUE	TRUE					
0.1	0.5	0.4	TRUE	TRUE	TRUE					
0.11	0.51	0.39	TRUE	TRUE	TRUE					
0.12	0.52	0.37	TRUE	TRUE	TRUE					
0.13	0.53	0.35	TRUE	TRUE	TRUE					
0.14	0.54	0.33	TRUE	TRUE	TRUE					
0.15	0.55	0.31	TRUE	TRUE	TRUE					
0.15	0.55	0.29	TRUE	TRUE	TRUE					
0.16	0.56	0.27	TRUE	TRUE	TRUE					
0.17	0.57	0.25	TRUE	TRUE	TRUE					
0.18	0.58	0.23	TRUE	TRUE	TRUE					
0.19	0.59	0.21	TRUE	TRUE	TRUE					
0.2	0.6	0.2	TRUE	TRUE	TRUE					
0.21	0.61	0.18	TRUE	TRUE	TRUE					
0.22	0.62	0.16	TRUE	TRUE	TRUE					
0.23	0.63	0.14	TRUE	TRUE	TRUE					
0.24	0.64	0.12	TRUE	TRUE	TRUE					
0.25	0.65	0.1	TRUE	TRUE	TRUE					
0.25	0.66	0.08	TRUE	TRUE	TRUE					
0.26	0.67	0.06	TRUE	TRUE	TRUE					
0.27	0.68	0.04	TRUE	TRUE	TRUE					
0.28	0.69	0.03	TRUE	TRUE	TRUE					
0.29	0.7	0.01	TRUE	TRUE	TRUE					

After calculation, there are 4891 true values among 4900 values, which is 99.82% precision, it is a decent result. The errors might be caused by the approximation of coordinates which are extremely close to zero but is negative, if they are viewed as zero, then pixels that outside the triangle are actually being defined as inside.

4. Assignment 5

The function c_ppm is created to generate ppm file of a figure. It calls function IsInside to interpolate pixels that belongs to a specific triangle. If traverses all the pixels, if its outside triangle, then painted black; if inside, then painted with designated RGB value.

Result of assignment 5:

The upper image is ppm file, the lower is triangle drawn by assignment 3, with the same coordinates. They are identical.