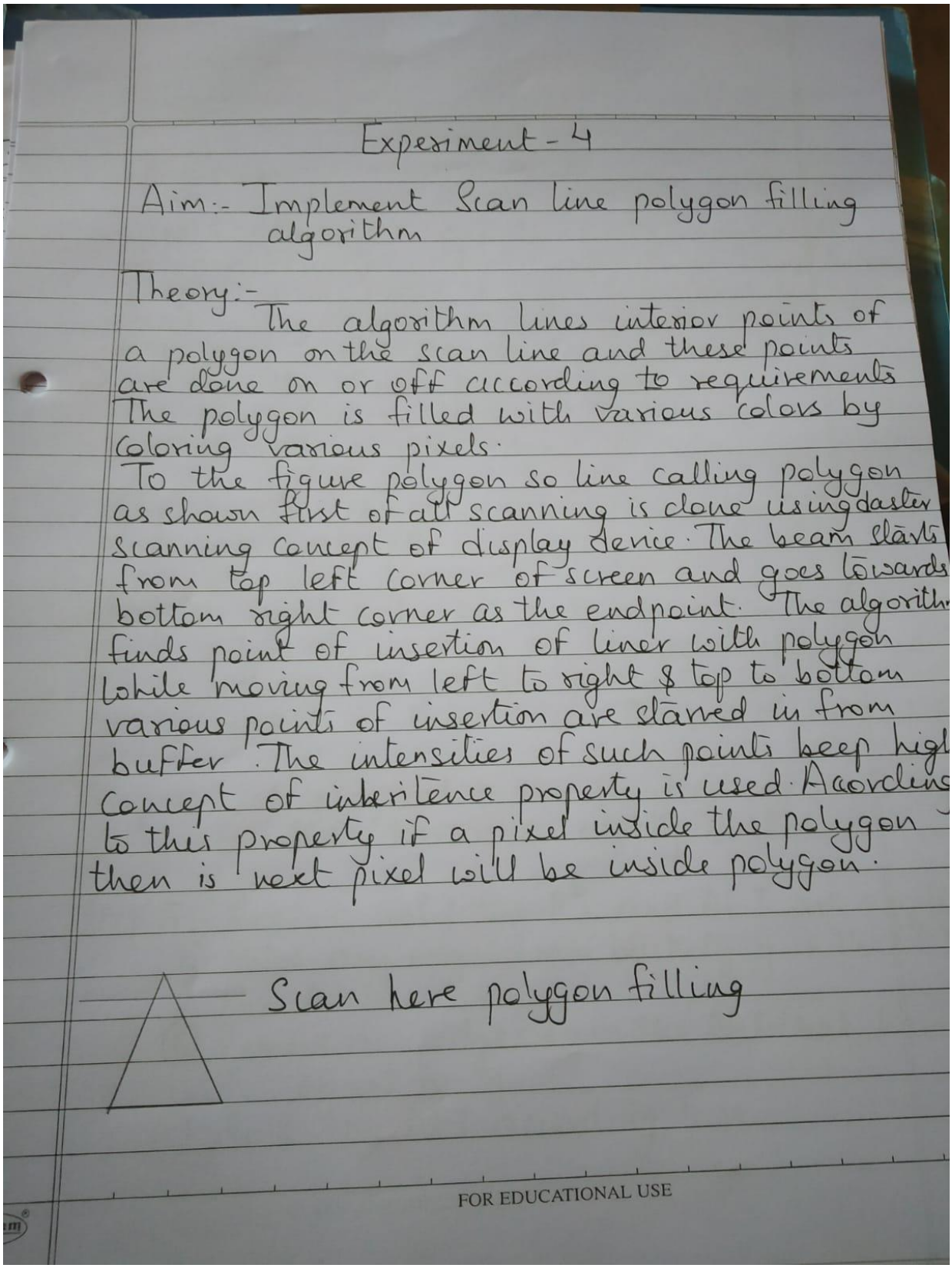
COMPUTER ENGINEERING

CG ODD SEM 2021-22/EXPERIMENT 4

NAME:- GAURAV AMARNANI (D7A, 67)



Experiment - 4

Aim:- Implement Scan line polygon filling algorithm

Theory:-

The algorithm lines interior points of a polygon on the scan line and these points are done on or off according to requirements The polygon is filled with various colors by coloring various pixels.

To the figure polygon so line calling polygon as shown first of all scanning is done using raster scanning concept of display device. The beam starts from top left corner of screen and goes towards bottom right corner as the endpoint. The algorith finds point of insertion of liner with polygon while moving from left to right & top to bottom various points of insertion are stored in from buffer. The intensities of such points keep high Concept of inheritence property is used. Accordin to this property if a pixel inside the polygon then is next pixel will be inside polygon.

Scan here polygon filling

FOR EDUCATIONAL USE

Algorithm:-

Step 1:- For each non-horizontal edge of the Polygon boundary identify the upper and lower endpoint (x_1, y_1) and (x_4, y_4) such that $y_4 > y_1$ and construct a record for each that contains

- y_4, they-coordinate at upper endpoint
- $x = x_1$, the current x intersection
- $w = 1/m = (x_4 - x_1)/(y_4 - y_1)$, the raprocal of the slope of the edge.

Step 2:- Set the AET (the active edge table) to be empty

Step 3:- Apply a bucket sort algorithm to sort the edges using they, as the primary key, and x_1 as secondary and w as the tertiary N.B. Each bucket contains a list The set of called the ET

Step 4:- Set y equal to the smallest index in the ET that has a non empty bucket

Step 5:- Repeat until the ET and AET are empty

(a) Move any edges from bucket y in the ET to the AET

(b) Remove any edges from the AET that have a y_4 equal to y.

(c) Sort the AET according to x

FOR EDUCATIONAL USE

(d) fill in the requisite pixels between the even and odd adjacent pairs of intersections in the AET: round up. [x] the x-coordinate of left intersections, round down, [x-1] that of the right intersections.
(e) increment y by one
(f) Update $x \leftarrow x + 1/m$ for every non-vertical edge in the AET

Side effects of scan conversion:-

1) Staircase as jogged:- It deals with unequal apperance of brightness as lines, as inclined the appears less bright as compared to horizontal or vertical

Conclusion:-

Thus we have implemented scanline Polygon filling algorithm and seen how it work

FOR EDUCATIONAL USE

Program-

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void main() {
int n,i,j,k,gd,gm,dy,dx;
int x,y,temp;
int a[20][2],xi[20];
float slope[20];
clrscr();
printf("\n\n\tEnter the no. of edges of polygon : ");
scanf("%d",&n);
printf("\n\n\tEnter the cordinates of polygon :\n\n\n ");

for(i=0;i<n;i++) {
printf("\tX%d Y%d : ",i,i);
scanf("%d %d",&a[i][0],&a[i][1]);
}

a[n][0]=a[0][0];
a[n][1]=a[0][1];

detectgraph(&gd,&gm);
initgraph(&gd,&gm,"C:\\TurboC3\\BGI");

for(i=0;i<n;i++) {
line(a[i][0],a[i][1],a[i+1][0],a[i+1][1]);
}

getch();

for(i=0;i<n;i++) {
dy=a[i+1][1]-a[i][1];
dx=a[i+1][0]-a[i][0];
if(dy==0) slope[i]=1.0;
if(dx==0) slope[i]=0.0;
if((dy!=0)&&(dx!=0)) {
slope[i]=(float) dx/dy;
}
}
for(y=0;y< 480;y++) {
k=0;
for(i=0;i<n;i++) {
if( ((a[i][1]<=y)&&(a[i+1][1]>y))|| ((a[i][1]>y)&&(a[i+1][1]<=y))) {
xi[k]=(int)(a[i][0]+slope[i]*(y-a[i][1]));
k++;
}
}
```

```
for(j=0;j<k-1;j++) {
for(i=0;i<k-1;i++) {
if(xi[i]>xi[i+1]) {
temp=xi[i]; xi[i]=xi[i+1]; xi[i+1]=temp;
}
}
setcolor(3);
for(i=0;i<k;i+=2) {
line(xi[i],y,xi[i+1]+1,y);
getch();
}
}
}
```

Output: