Scanline Polygon filling algorithm

1. #include <stdio.h>
2. #include <conio.h>
3. #include <graphics.h>
4. void main() 5. {
5. int n,i,j,k,gd,gm,dy,dx;
6. int x,y,temp;

8. int a[20][2],xi[20];

1. float slope[20];
2. clrscr();
3. printf("\n\n\tEnter the no. of edges of polygon : ");

12. scanf("%d",&n);

13. printf("\n\n\tEnter the cordinates of polygon :\n\n\n ");

14. for(i=0;i<n;i++)

15. {

16. printf("\tX%d Y%d : ",i,i);

|  |  |  |
| --- | --- | --- |
| 17. | scanf("%d %d",&a[i][0],&a[i][1]); |  |
| 18. | } |
| 19. | a[n][0]=a[0][0]; |
| 20. | a[n][1]=a[0][1]; |
| 21. | detectgraph(&gd,&gm); |
| 22. | initgraph(&gd,&gm,"c:\\TURBOC3\\bgi"); |
| 23. | /\*- draw polygon -\*/ |
| 24. | for(i=0;i<n;i++) |
| 25. | { |
| 26. | line(a[i][0],a[i][1],a[i+1][0],a[i+1][1]); |
| 27. | } |
| 28. | getch(); |
| 29. | for(i=0;i<n;i++) |
| 30. | { |
| 31. | dy=a[i+1][1]-a[i][1]; |
| 32. | dx=a[i+1][0]-a[i][0]; |
| 33. | if(dy==0) slope[i]=1.0; |
| 34. | if(dx==0) slope[i]=0.0; |
| 35. | if((dy!=0)&&(dx!=0)) /\*- calculate inverse | slope -\*/ |
| 36. | { |  |
| 37. | slope[i]=(float) dx/dy; |  |
| 38. | } |  |
| 39. | } |  |

|  |  |  |
| --- | --- | --- |
| 40. | for(y=0;y< 480;y++) |  |
| 41. | { |
| 42. | k=0; |
| 43. | for(i=0;i<n;i++) |
| 44. | { |
| 45. | if( ((a[i][1]<=y)&&(a[i+1][1]>y))|| |
| 46. | ((a[i][1]>y)&&(a[i+1][1]<=y))) |
| 47. | { |
| 48. | xi[k]=(int)(a[i][0]+slope[i]\*(y-a[i][1])); |
| 49. | k++; |
| 50. | } |
| 51. | } |
| 52. | for(j=0;j<k-1;j++) /\*- Arrange x-intersections | in order -\*/ |
| 53. | for(i=0;i<k-1;i++) |  |
| 54. | { |  |
| 55. | if(xi[i]>xi[i+1]) |  |
| 56. | { |  |
| 57. | temp=xi[i]; |  |
| 58. | xi[i]=xi[i+1]; |  |
| 59. | xi[i+1]=temp; |  |
| 60. | } |  |
| 61. | } |  |
| 62. | setcolor(35); |  |
| 63. | for(i=0;i<k;i+=2) |  |
| 64. | { |  |
| 65. | line(xi[i],y,xi[i+1]+1,y); |  |
| 66. | getch(); |  |
| 67. | } |  |
| 68. | } |  |
| 69. | } |  |