gki3Line.java

```java
package gobalkrishnan_v_18_06_1995.graphics;

import gobalkrishnan_v_18_06_1995.color.gkiColor;

public class gki3Line {
@Override
    public String toString() {
        return  sp + ":" + ep ;
    }

public ArrayList<gkiColor>  color=new ArrayList<>();
public ArrayList<gki3Point> point=new ArrayList<>();

double sx,sy,sz,ex,ey,ez;
gkiColor sc,ec;
gki3Point sp,ep;

public void sx(double sx){this.sx=sx;sp.x=sx;}
public void sy(double sy){this.sy=sy;sp.y=sy;}
public void sz(double sz){this.sz=sz;sp.z=sz;}

public void ex(double ex){this.ex=ex;ep.x=ex;}
public void ey(double ey){this.ey=ey;ep.y=ey;}
public void ez(double ez){this.ez=ez;ep.z=ez;}

public void sc(gkiColor sc){this.sc=sc;}
public void ec(gkiColor ec){this.ec=ec;}
public void sp(gki3Point sp){
    this.sp=sp;
    sx=sp.x;
    sy=sp.y;
    sz=sp.z;
```

```
}
public void ep(gki3Point ep){
    this.ep=ep;
   ex=ep.x;
   ey=ep.y;
   ez=ep.z;
}
public void set(gki3Point sp,gkiColor sc,gki3Point ep,gkiColor ec){

    sp(sp);
    ep(ep);
    sc(sc);
    ec(ec);
}
gkiColor  col;
public void process(){
    color.removeAll(color);
    point.removeAll(point);

    double xdiff=ex-sx;
    double ydiff=ey-sy;
    double zdiff=ez-sz;

    if(xdiff == 0 && ydiff == 0 && zdiff == 0){
        point.add(new gki3Point(sp));
        color.add(sc);
    }

    double x_=fabs(xdiff);
    double y_=fabs(ydiff);
    double z_=fabs(zdiff);
```

```
if(x_>y_ && x_>z_){

    double xmin,xmax;
    if(sx<ex){
        xmin=sx;
        xmax=ex;
    }else{
        xmin=ex;
        xmax=sx;
    }

    double slopeY=ydiff/(double)xdiff;
    double slopeZ=zdiff/(double)xdiff;


    for(double x=xmin;x<=xmax;x++){
        double y=(double)(sy+((x-sx)*slopeY));
        double z=(double)(sz+((x-sx)*slopeZ));
        try{
            double ratio = (x-sx)/(double)xdiff;
            int alpha=(int)(ec.alpha*ratio + sc.alpha*(1-ratio));
            int red=(int)(ec.red*ratio+sc.red*(1-ratio));
            int green=(int)(ec.green*ratio+sc.green*(1-ratio));
            int blue=(int)(ec.blue*ratio+sc.blue*(1-ratio));
            if(alpha<0){alpha=0;}
            if(alpha>255){alpha=255;}
            if(red<0){red=0;}
            if(red>255){red=255;}
            if(green<0){green=0;}
            if(green>255){green=255;}
            if(blue<0){blue=0;}
        if(blue>255){blue=255;}
         col=new gkiColor(alpha, red, green, blue);
        color.add(col);
        }catch(NullPointerException e){

        }
        gki3Point p=new gki3Point(x,y,z);
        p.color(col);

        point.add(p);
    }
  }else if(y_>z_){
    double ymin,ymax;
    if(sy<ey){
      ymin=sy;
      ymax=ey;
    }else{
      ymin=ey;
      ymax=sy;
    }
    double slopeX=xdiff/(double)ydiff;
    double slopeZ=zdiff/(double)ydiff;

    for(double y=ymin;y<=ymax;y++){
      double x=sx+((y-sy)*slopeX);
      double z=sz+((y-sy)*slopeZ);

      try{
        double ratio=(y-sy)/(double)ydiff;
        int alpha= (int) (ec.alpha*ratio+sc.alpha*(1-ratio));
        int red=(int)(ec.red*ratio+sc.red*(1-ratio));
        int green=(int)(ec.green*ratio+sc.green*(1-ratio));
        int blue=(int)(ec.blue*ratio+sc.blue*(1-ratio));
        if(alpha<0){alpha=0;}
        if(alpha>255){alpha=255;}
        if(red<0){red=0;}
        if(red>255){red=255;}
        if(green<0){green=0;}
        if(green>255){green=255;}
        if(blue<0){blue=0;}
        if(blue>255){blue=255;}
        col=new gkiColor(alpha, red, green, blue);
        color.add(col);
        }catch(NullPointerException e){

        }
      gki3Point p=new gki3Point(x,y,z);
      p.color(col);

      point.add(p);


    }




  }else{
    double zmin,zmax;
    if(sz<ez){
      zmin=sz;
      zmax=ez;
    }else{
      zmin=ez;
      zmax=sz;
```

```
}
double slopeX=xdiff/(double)zdiff;
double slopeY=ydiff/(double)zdiff;

for(double z=zmin;z<=zmax;z++){
    double x=sx+((z-sz)*slopeX);
    double y=sy+((z-sz)*slopeY);

    try{
        double ratio=(y-sy)/(double)ydiff;
        int alpha= (int) (ec.alpha*ratio+sc.alpha*(1-ratio));
        int red=(int)(ec.red*ratio+sc.red*(1-ratio));
        int green=(int)(ec.green*ratio+sc.green*(1-ratio));
        int blue=(int)(ec.blue*ratio+sc.blue*(1-ratio));
        if(alpha<0){alpha=0;}
        if(alpha>255){alpha=255;}
        if(red<0){red=0;}
        if(red>255){red=255;}
        if(green<0){green=0;}
        if(green>255){green=255;}
        if(blue<0){blue=0;}
        if(blue>255){blue=255;}
        col=new gkiColor(alpha, red, green, blue);
        color.add(col);
        }catch(NullPointerException e){

        }

        gki3Point p=new gki3Point(x,y,z);
        p.color(col);

        point.add(p);
        }

    }


}
public void reverse(){
    gki3Point[] li=new gki3Point[point.size()];
    int a=0;
    for(int i=point.size()-1;i>=0;i--){
        li[a++]=point.get(i);
    }
    point.removeAll(point);
    for(int i=0;i<li.length;i++){
        point.add(li[i]);
    }

}
private double fabs(double g){
    if(g<0){
        g*=-1;
    }
    return g;
}


}
```