Clipping 2D

Disciplina: Computação Gráfica

Professora: Deller James Ferreira

1) **Compile, execute e analise o código a seguir:**

```
// Um programa OpenGL que implementa o Clipping

#include <GL/glut.h>

#include <stdlib.h>

#include <math.h>

// estrutura que descreve um ponto (x,y)

typedef struct XY {

      GLfloat x;

      GLfloat y;

} PontoXY;

PontoXY P1, P2, P3,P4;



 GLint round (const GLfloat a) {return GLint (a + 0.5);}

/*Define ocodigo de 4 bits*/

const GLint winLeftBitCode = 0x1;

const GLint winRightBitCode = 0x2;
const GLint winBottomBitCode = 0x4;
const GLint winTopBitCode = 0x8;

 GLint inside (GLint code) {return GLint (!code);}
 GLint reject (GLint code1, GLint code2) {return GLint (code1 & code2);}
 GLint accept (GLint code1, GLint code2) {return GLint (!(code1 | code2));}
GLubyte encode (PontoXY pt, PontoXY winMin, PontoXY winMax)
{
GLubyte code = 0x00;
if(pt.x < winMin.x)
     code = code | winLeftBitCode;
if(pt.x  > winMax.x)
    code = code | winRightBitCode;
if(pt.y < winMin.y)
    code = code | winBottomBitCode;
if(pt.y  > winMax.y)
    code = code | winTopBitCode;
return (code);
}
void  swapPts (PontoXY  * p1, PontoXY * p2)
{
PontoXY tmp;
tmp = *p1;
```

```
*p1=*p2;
*p2=tmp;
}
void  swapCodes (GLubyte   * c1, GLubyte   * c2)
{
GLubyte    tmp;
tmp = *c1;
*c1=*c2;
*c2= tmp;
}
void lineClipCohSuth (PontoXY winMin, PontoXY winMax, PontoXY p1, PontoXY p2)
{
GLubyte code1, code2;
GLint done = false;
GLfloat m;

while (!done)
     {

      code1 = encode (p1, winMin, winMax);
      code2 = encode (p2, winMin, winMax);
    if (accept (code1, code2))
          done = true;
```

```
else

  {

       if (reject (code1, code2))
        done = true;
       else
         {/* estabelece o ponto fora da janela como p1*/
         if(inside (code1))
              {
              swapPts (&p1,&p2);
              swapCodes(&code1,&code2);
              }
         if (p2.x !=p1.x)
            m = (p2.y - p1.y)/ (p2.x - p1.x);
         if ( code1 & winLeftBitCode) {
            p1.y += (winMin.x -p1.x) * m;
            p1.x = winMin.x;
            }
         else
            if ( code1 & winRightBitCode) {
               p1.y += (winMax.x -p1.x) * m;
```

```
                 p1.x = winMax.x;
                 }
              else
                 if ( code1 & winBottomBitCode) {
                    p1.y += (winMin.y -p1.x) * m;
                    p1.x = winMin.y;
                   }
                 else
                   if ( code1 & winTopBitCode) {
                       p1.y += (winMax.y -p1.x) * m;
                       p1.x = winMax.y;}
          }
    }



}

    glClear (GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 0.0,0.0);
    glBegin (GL_LINES);
    glVertex2i (round(p1.x), round(p1.y));
    glVertex2i (round(p2.x), round(p2.y));
```

```
      glEnd ();

     glFlush();

}

void  Desenha(void)
{
P1.x=0.0;
P1.y=0.0;
P2.x=300.0;
P2.y=300.0;
P3.x=0.0;
P3.y=0.0;
P4.x=270.0;
P4.y=270.0;

lineClipCohSuth (P1,  P2,  P3, P4);
//Executa os comandos OpenGL

}
// Inicializa parâmetros de rendering
void Inicializa (void)
```

```
{      // Define a cor de fundo da janela de visualização como preta
       glClearColor (1.0,1.0,1.0,0.0);
       glMatrixMode (GL_PROJECTION);
       gluOrtho2D (0.0, 300.0, 0.0, 300.0);
}
// Programa Principal
int main(int argc, char **argv)
{         glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
         glutCreateWindow("Clipping");
          glutDisplayFunc(Desenha);
          Inicializa();
          glutMainLoop();
}
```

2)Altere a janela de clipping e o tamanho do segmento para testar o programa.