

PROVA 2 - COMPUTAÇÃO GRÁFICA

MATEUS CARVALHO GONÇALVES 201810245 JOA

① EM UM CÓDIGO MODULARIZADO, TEMOS AS SEQUENTES FUNÇÕES:

```
VOID LINHA (INT X, INT Y) {  
    GLBEGIN (GL_LINE_STRIP);  
    FOR (FLOAT I=X; I < X+6, I+=0.01) {  
        GLVERTEX2F (I, Y);  
    }  
    GLEND ();  
}  
  
VOID PONTO (INT X, INT Y) {  
    GLBEGIN (GL_POINTS);  
    GLVERTEX2F (X, Y);  
    GLEND ();  
}
```

DESSA FORMA, NA FUNÇÃO DISPLAY BASTA INSERIR O CÓDIGO SEGUINTE (LEMBRANDO QUE ESTAMOS DESTACANDO APENAS AS PARTES IMPORTANTES, UM CÓDIGO OPENGL POSSUI UMA ESTRUTURA BEM MAIOR):

```
GLCOLOR3F (1.0, 0.0, 0.0); // VERMELHO  
GLPointSize (5.0);  
GLLineWidth (5.0);  
INT ESPACO = 10;  
FOR (INT I=0; I < 12; I++) {  
    LINHA (50 + ESPACO * I, 100);  
    PONTO (58 + ESPACO * I, 100);  
}
```

② O TRECHO A SEGUIR POSSUI O RECORTE DO CÓDIGO REFERENTE À PARTE DA FUNÇÃO DISPLAY QUE É SIGNIFICANTE PARA A QUESTÃO.

```
glColor3f(0.0, 0.0, 1.0)
```

```
for(int i = 50; i >= -50; i -= 15) {
```

```
    glBegin(GL_QUADS);
```

```
        for(int j = -50; j <= 50; j += 25) {
```

```
            glVertex2f(j, i);
```

```
            glVertex2f(j, i - 10);
```

```
            glVertex2f(j + 20, i - 10);
```

```
            glVertex2f(j + 20, i);
```

```
        }
```

```
    glEnd();
```

```
}
```

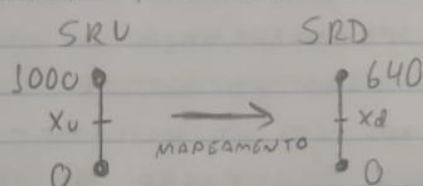
③ A IMPLEMENTAÇÃO DEPENDERIA SE A FIGURA É VAZADA OU SÓLIDA. NO PRIMEIRO CASO, É POSSÍVEL UTILIZAR A DIRETIVA GL-LINE-LOOP, DEFININDO OS VÉRTICES. JÁ NO CASO SÓLIDO, PODE-SE USAR A DIRETIVA GL-POLYGON, TAMBÉM COM OS VÉRTICES DEFINIDOS

④ É NECESSÁRIO UM MARGAMENTO SRU/SRT, SENDO OS LIMITES:

$$SRU \rightarrow \min = (0, 0), \max = (1000, 1500)$$

$$SRD \rightarrow \min = (0, 0), \max = (640, 480)$$

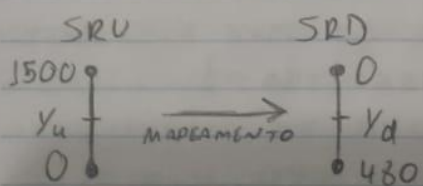
PARA X, TEMOS:



$$\frac{x_d - 0}{x_u - 0} = \frac{640 - 0}{1000 - 0}$$

$$x_d = (x_u \cdot 640) / 1000$$

PARA Y, TEMOS (ATENTANDO QUE O EIXO CRESCE EM SENTIDOS OPOSTOS NOS DOIS SISTEMAS):



$$\frac{y_d - 480}{y_u - 0} = \frac{0 - 480}{1500 - 0}$$

$$y_d = \frac{y_u \cdot (-480)}{1500} + 480$$

⑤ A IMAGEM VETORIAL É DE NATUREZA GEOMÉTRICA E É DEFINIDA POR ELEMENTOS E PARÂMETROS, ESSE TIPO ACEITA BEM OPERAÇÕES DE ESCALA, POR EXEMPLO. POR OUTRO LADO, IMAGENS MATRICIAIS SÃO REPRESENTAÇÕES DISCRETAS, EM QUE CADA ELEMENTO INDEPENDENTE É ARMAZENADO EM UMA POSEÇÃO DA MATRIZ.

FAZER A CONVERSÃO VETORIAL \rightarrow MATRICIAL É UMA OPERAÇÃO MAIS SIMPLES, POIS PODE-SE COLOCAR OS PONTOS DEFINIDOS DA REPRESENTAÇÃO VETORIAL NA MATRIZ. SÃO CONTRÁRIO É MAIS COMPLICADO E SOFRE PERDAS MAIS FACILMENTE, VISTO QUE É PRECISO EXTRAIR DA MATRIZ INFORMAÇÕES GEOMÉTRICAS.

⑥ TRANSFORMAÇÃO GEOMÉTRICA É UMA FUNÇÃO BISETORA QUE GERA, A PARTIR DE UMA FIGURA ORIGINAL, OUTRA GEOMETRICAMENTE IGUAL OU SEMELHANTE.

A TRANSLAÇÃO MOVE TODOS OS PONTOS DE UM ELEMENTO NA MESMA DISTÂNCIA EM UMA DETERMINADA DIREÇÃO. A ROTAÇÃO É UMA TRANSFORMAÇÃO DE UM SISTEMA DE COORDENADAS, QUE GIRA UM OBJETO EM TORNO DE UM OU MAIS EIXOS EM UM DETERMINADO ÂNGULO. POR FIM, A ESCALA É A RAZÃO DAS MEDIDAS DO ELEMENTO FINAL PELAS MEDIDAS DO REAL.

⑦ ROTAÇÃO DE 180° EM TORNO DO EIXO X É TRANSFORMAÇÃO DE ESCALA COM PARÂMETRO -1 .

⑧ COM OPENGL É POSSÍVEL ROTACIONAR COM A FUNÇÃO `glRotatef(ÂNGULO, BOOL-X, BOOL-Y, BOOL-Z)`, LEMBRANDO QUE O REFERENCIAL É O PONTO EM QUE A APLICAÇÃO SE ENCONTRA.

⑨ O PROCESSO EXEMPLIFICADO É O RECORTE, E DENTRE SEUS VÁRIOS TIPOS O POINT CLIPPING SERÁ EXEMPLIFICADO A SEGUIR. ELE É RÁPIDO E SIMPLES, É **tiltbro** NECESSÁRIO APENAS RESOLVER AS INEQUAÇÕES

$X_{min} < X < X_{max}$ e $Y_{min} < Y < Y_{max}$, DELIMITANDO AS COORDENADAS DO RECORTE. É IMPORTANTE PARA OTIMIZAÇÃO, SELECIONANDO APENAS AS PRIMITIVAS QUE SERÃO RASTERIZADAS PARA PROCESSAMENTO.

10) O MODELO DE BOUKNIGHT CALCULA O VALOR DA INTENSIDADE LUMINOSA NO PONTO QUE O RAIOS DE LUZ ATINGE O VETOR NORMAL E APLICA EM TODO O POLÍGONO. ENQUANTO O MODELO DE PHONG CONSIDERA AS VARIAÇÕES DO VETOR NORMAL AO LONGO DO ESPAÇO, DESSA FORMA É POSSÍVEL SIMULAR MELHOR OS EFEITOS DA LUZ EM CADA PONTO, ENTANTO É UM PROCESSO ONEROSO EM TERMOS DE PROCESSAMENTO.

11) O PROCESSO DE PHONG LEVA EM CONSIDERAÇÃO OS FATORES DE COR E POSIÇÃO DA LUZ, POSIÇÃO DO OBSERVADOR, DO PONTO E DA ORIENTAÇÃO DA FACE DO PONTO. PARA CADA PONTO, A INTENSIDADE LUMINOSA É CALCULADA A PARTIR DO VETOR NORMAL. PONTOS DE BRILHO SÃO DADOS COM MAIS NITIDEZ E POSSUEM ALTO CUSTO DE PROCESSAMENTO. POR OUTRO LADO, MÉTODOS COMO O DE BOUKNIGHT E GOURAUD FOCAM EM MENOR CUSTO DE PROCESSAMENTO, O PRIMEIRO FOE DISCUTIDO NA QUESTÃO ANTERIOR E O SEGUNDO BUSCOU MELHORA-LO COM INTERPOLAÇÕES.

12) A VISÃO PERSPECTIVA SERVE PARA CRIAR CENAS COM DIFERENTES PONTOS DE VISTA, OS QUAIS

FORAM "VISTOS" POR UM OBSERVADOR MÓVEL.

③ A IDEIA DESSE ALGORITMO É QUE A PARTIR DE UM OBSERVADOR, POSICIONADO EM FRENTE UMA TELA TRANSPARENTE E PLANA, PARTAM MÚLTIPLOS RAIOS VISUAIS DE SEUS OLHOS. ESSES, POR SUA VEZ, ATRAVESSAM O PLANO E PINTAM-NO COM A COR DO OBJETO Atingido NO FIM.

ALGORITMO BÁSICO:

- CALCULE UMA LINHA RETA UNINDO O OLHO DO OBSERVADOR A ESTE PONTO

- DESCUBRA AS INTERSEÇÕES DESTA RETA COM OS OBJETOS 3D QUE ESTÃO ATRÁS DA TELA

- PINTO O PONTO COM A COR DO OBJETO MAIS PRÓXIMO

④ É O PROCESSO DE CONVERSÃO DA REPRESENTAÇÃO VETORIAL PARA MATRICIAL. O ALGORITMO DE DDA É DEFINIDO POR DOIS PONTOS GENÉRICOS QUE FORMAM UMA RETA, E A PARTIR DA EQUAÇÃO DA RETA E DEFININDO 'm', É POSSÍVEL ACHAR 'n'.

PONTOS (x_1, y_1) \rightarrow EQUAÇÃO $y = mx + n \rightarrow m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$
 (x_2, y_2)

PARA UMA RETA COM OS PONTOS (6,9) E (11,12), TEMOS

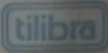
$m = 0,6$

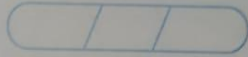
$n = 5,4$

E O ALGORITMO:

PARA $i = x_1$ ATÉ x_2 FAÇA:

$y_{Pixel} = 0,6 * i + 5,4;$

 $PLOTANPixel(i, ROUND(y_{Pixel}));$



15) UM MÉTODO SIMPLES PARA PREENCHIMENTO DE POLÍGONOS É VARRER CADA PIXEL DA IMAGEM E VERIFICAR SE ELE ESTÁ DENTRO DOS LIMITES DO POLÍGONO. PARA MELHOR EFICIÊNCIA DA VARREDURA UTILIZA-SE A TÉCNICA DO BOUNDING BOX, QUE É O RECORTE DO MENOR RETÂNGULO QUE CONTÉM O POLÍGONO.