P2 CG 2012.1

	Computação Gráfica 1 Prof. Rodrigo de Toledo	Data: 18/06/2012 P2

1) (3 pontos) Bresenham
a) Dado dois pontos (X1,Y1) e (X2,Y2), calcule quantos pixels serão acesos pelo algoritmo de Bresenham. Pode usar pseudo-código apenas com estruturas se-senão. Dica: atenção aos 8 casos de Bresenham!
b) Para desenhar uma linha branca sobre um fundo preto, foi usado o algoritmo de Bresenham considerando que cada pixel tinha 16 subpixels (4x4) para diminuir o efeito de aliasing. Quantos tons de cinzas diferentes um pixel pode vir a ter? Por que?

2) (4 pontos) Ray-tracing
A função `bool = intersection(vec3 ray_origin, vec3 ray_direction, float radius, vec3 center, float &t1, float &t2)` retorna `verdadeiro/falso` se houve interseção entre uma esfera e um raio, assim como guarda os parâmetros dos pontos de interseção em `t1` e `t2`.
a) Quais linhas de código devem ser acrescentas na função `intersection` para que ela também calcule a normal do ponto de interseção mais próximo (guardando-o normalizado no parâmetro `&normal`), se houver?
b) Implemente a função `bool = intersectionIntersection (vec3 ray_origin, vec3 ray_direction, float radius1, vec3 center1, float radius2, vec3 center2, float &t1, float &t2, &normal)`; que retorna se houve interseção entre o raio e a interseção de duas esferas (CSG). A função também deve calcular o parâmetro do ponto de entrada e saída do raio, assim como a normal do ponto mais próximo).

3) (3 pontos) GPU programming
vertex:

```
varying vec3 v_V;
varying vec3 v_N;

void main () {
    gl_Position = ftransform();
    v_V = (gl_ModelViewMatrix * gl_Vertex).xyz;
    v_N = gl_NormalMatrix * gl_Normal;
}
```

fragment:

```
varying vec3 v_V;
varying vec3 v_N;

void main () {
  vec3 color = vec3(1.0, 0.0, 0.0);
  vec3 N = normalize(v_N);
  vec3 V = normalize(v_V);
  vec3 L = normalize(vec3(gl_LightSource[0].position));
  vec3 R = L - 2*dot(L,N)*N;

  vec3 a = color * 0.1;
  vec3 b = color * (1.0 - a) * max(dot(L, N), 0.0);
  vec3 c = vec3(1.0, 1.0, 1.0) * pow(max(dot(R, V), 0.0), 8.0);

  gl_FragColor = vec4(a+b+c, 1.0);
}
```

a) Quais das componentes de iluminação (ambiente, difusa e especular) são aplicadas nesse código? Em que proporção?
O Código acima tenta aplicar o modelo de iluminação de Phong.
b) Explique o porquê.
c) Quais modificações deveriam ser feitas para aplicar o modelo de Gouraud ao invés de Phong?