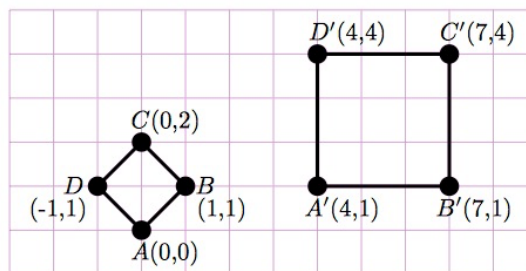


Prova 1
MAC-420/5744 – Introdução à Computação Gráfica
1º. Semestre de 2014
Prof. Marcel P. Jackowski

1. **(0,5 pts)** Qual a relação entre as disciplinas de visão computacional e computação gráfica?
2. **(0,5 pts)** Por que as coordenadas homogêneas são utilizadas em transformações geométricas ?
3. **(0,5 pts)** Qual a diferença entre um ponto e um vetor? Como eles são representados na forma homogênea?
4. **(0,5 pts)** Existe alguma vantagem em representar transformações como matrizes ? Explique.
5. Existem dois modelos básicos de iluminação: o modelo local e o modelo global. Responda:
 - a. **(0,5 pts)** Qual a diferença entre eles ?
 - b. **(0,5 pts)** Qual deles é utilizado pela API do OpenGL ?
 - c. **(0,5 pts)** Cite uma técnica de renderização capaz de reproduzir a refração da luz.
6. **(1,0 pts)** Explique a diferença entre as fórmulas *explícita (reduzida)*, *implícita* e *paramétrica* da reta e dê exemplos. Qual delas é útil para produção de coordenadas de pontos em um segmento de reta ?
7. **(1,0 pts)** Defina as inversas das seguintes transformações rígidas em 2D: translação, rotação e escala.
8. **(1,0 pts)** Dê a matriz ou o produto de matrizes, que transformará o quadrado ABCD no quadrado A'B'C'D'.



9. O modelo Phong é uma aproximação da física para produzir renderizações próximas de cenas reais.
 - a. **(0,5 pts)** Descreva quais os vetores que o modelo utiliza para calcular a cor final de um ponto arbitrário. Ilustre com uma figura.
 - b. **(0,5 pts)** O que são reflexões ambiente, difusa e especular ?
10. **(0,5 pts)** O que significa *z-buffer* e para que propósito ele é utilizado ?
11. **(1,0 pts)** O que é uma projeção perspectiva ? Defina uma matriz de transformação que represente tal projeção.
12. **(1,0 pts)** Um triângulo possui normais $\mathbf{n}_0=(1,0,0)$, $\mathbf{n}_1=(0,1,0)$ e $\mathbf{n}_2=(1,1,1)/\sqrt{3}$, nos vértices $\mathbf{P}_0$, $\mathbf{P}_1$, e $\mathbf{P}_2$, respectivamente. Assumindo o *vertex shader* e *fragment shader* abaixo, qual a cor que o *fragment shader* retornará para um pixel localizado na metade da aresta entre $\mathbf{P}_0$ e $\mathbf{P}_1$?

```
attribute vec4 vPos;
attribute vec3 vNormal;
varying vec3 fN;
uniform mat4 MVP;
void main()
{
    fN = vNormal;
    gl_Position = MVP*vPos;
}
```

```
varying vec3 fN;
void main()
{
    gl_FragColor = vec4(fN, 1.0);
}
```