

## Computação Gráfica – Projeto 3

### Objetivo:

Alterar o Projeto 2 para incorporar iluminação ambiente, difusa e especular.

### Requisitos:

1. Um dos objetos externos (i.e., do ambiente externo) deve ter alguma translação envolvida. Caso o Projeto 2 já contenha tal objeto, então ele pode ser reutilizado. Esse objeto transladando será (ou conterá) uma fonte de luz. Para fins de semântica, pode-se adicionar um cubo ou esfera próximo ao objeto para representar a fonte de luz. **A fonte de luz do ambiente externo só deve afetar os objetos do ambiente externo (você precisarão pesquisar como fazer isso).**
2. Dois objetos do ambiente interno atuarão como fontes de luz de cores diferentes (por exemplo, abajur e lâmpada no teto). Se o ambiente interno não tiver dois objetos que semanticamente possam atuar como fontes de luz, deve-se adicionar objetos assim no ambiente (não faz sentido um sofá ser uma fonte de luz, por exemplo, mas é ok assumir que um celular ou uma lanterna em cima dele seja). **As fontes de luz do ambiente interno só devem afetar objetos do ambiente interno (você precisarão pesquisar como fazer isso).**
3. Determine eventos de teclado para ligar e desligar as luzes de forma independente, inclusive a ambiente. Em outras palavras, cada fonte de luz terá seu “interruptor”. Qualquer (fonte de) luz pode ser ligada ou desligada a qualquer momento. **Como esperado, ligar ou desligar uma luz deve ter um efeito sobre um ou mais objetos:** se uma luz for acesa ou apagada e nada mudar na cena, algo está errado.
4. Determine eventos de teclado para incrementar e decrementar a luz ambiente.
5. Determine eventos de teclado para incrementar e decrementar a reflexão difusa.

6. Determine eventos de teclado para incrementar e decrementar a reflexão especular.
7. Todo objeto do seu cenário deve ter seus próprios parâmetros de iluminação difusa e especular, **e não devem ser usados parâmetros prontos advindos de arquivos .mtl.**
8. Não é mais preciso ter os eventos de teclado previstos no Projeto 2 (i.e., para rotação, translação, escala, malha poligonal).

## Critérios de Avaliação:

1. Atendimento aos requisitos.
2. Complexidade da cena e o quanto o resultado final faz sentido / é coerente.
3. Qualidade do código (estruturação e comentários), **que deverá ser submetido via e-disciplinas até o dia 09 de junho.**
4. **Apresentação em sala de aula nos dias 10 e 12 de junho.** Quem não se sentir à vontade para apresentar para todos poderá apresentar apenas para o professor.
5. Apresentações fora da data apenas mediante comprovação (atestado médico, declaração da empresa, etc.)

## Outras informações importantes:

1. **Pode ser feito individualmente ou em dupla.**
2. **Caso você tenha feito o Projeto 2 sozinho e agora fará em dupla, ou caso a dupla tenha mudado, enviar email para [jeanponciano@icmc.usp.br](mailto:jeanponciano@icmc.usp.br) comunicando os integrantes da dupla. Coloque o código da disciplina (SCC0250 ou SCC0650) no assunto.**
3. Pode utilizar qualquer código-base fornecido.
4. Pode-se utilizar, inclusive, outras linguagens de programação, desde que utilize apenas bibliotecas do OpenGL e do sistema de janelas. O uso de outras bibliotecas gráficas não será aceito.

5. Devem ser utilizadas apenas funções do pipeline moderno. No OpenGL, isso significa que as seguintes funções são obsoletas (*deprecated*) e não podem ser utilizadas: `glRotate`, `glTranslate`, `glScale`, `glVertex`, `glColor`, `glLight`, `glMaterial`, `glBegin`, `glEnd`, `glMatrix`, `glMatrixMode`, `glLoadIdentity`, `glPushMatrix`, `glPopMatrix`, `glRect`, `glBitmap`, `glAlphaFunc`, `glNewList`, `glDisplayList`, `glPushAttrib`, `glPopAttrib`, `glVertexPointer`, `glColorPointer`, `glTexCoordPointer`, `glNormalPointer`, `glMatrixMode`, `glCal`.