

Atividade de laboratório 3

P.A.E. Diego Cintra e Fábio Felix
diegocintra@usp.br, f_diasfabio@usp.br

07 de junho de 2018

As atividades descritas a seguir devem seguir as seguintes restrições:

- Todas devem ser implementadas **individualmente**;
- As atividades devem ser implementadas utilizando a API da OpenGL, sendo as bibliotecas `gl`, `glu`, `glut`, `glew` e `OpenCV` as únicas que podem ser utilizadas.
- As linguagens permitidas são `C` e `C++`.
- Para submissão, aqueles que optarem por utilizar Windows devem compactar todo o código-fonte como um arquivo “.zip”, incluindo executável. Os que optarem por sistemas operacionais baseados em UNIX também devem enviar todo o código-fonte compactado, acompanhado de um `Makefile`.

Atividade 1

Questões propostas

1. Insira no mínimo um objeto na cena que possa ser texturizado. Além desse(s) objeto(s) a cena também deve possuir textura. As texturas devem ser imagens carregadas externamente, com o auxílio de uma biblioteca de manipulação de imagem. Para esta etapa do trabalho pode ser utilizada qualquer biblioteca disponível ou até mesmo código implementado pelo(s) aluno(s) para carregamento das imagens. [Recomenda-se fortemente a utilização da biblioteca OpenCV para o carregamento, como nos testes apresentados nos slides da aula.](#)

- Para a textura do objeto, lembre-se que ele **deve** ser descrito com primitivas e suas normais. Portanto, funções como `glutSolidSphere` e `glutWireCube` podem não apresentar correto funcionamento. Utilize funções fornecidas pela biblioteca GLU, como demonstrado nos slides da aula e no exemplo a seguir:

```
1  ...
2  GLUquadric *sphereQuadric;
3  sphereQuadric = gluNewQuadric();
4  gluQuadricTexture(sphereQuadric, GL_TRUE);
5  gluSphere(sphereQuadric, 0.3, 32, 32);
6  ...
```

- Para a textura da cena, deve-se utilizar o conceito de *skybox*¹ descrito nos slides da aula. Para essa textura, utilize somente *cube map* para definição dos parâmetros (cujas constantes são definidas por `GL_TEXTURE_CUBE_MAP`).
 - A escolha das imagens que serão utilizadas como texturas é livre.
2. Permita com que a câmera rotacione em torno dos eixos X, Y e Z, da seguinte maneira: a tecla ‘x’ gira a câmera em torno do eixo X de maneira horária; já ‘X’, de maneira anti-horária. A tecla ‘y’ gira a câmera em torno do eixo Y de maneira horária; já ‘Y’, de maneira anti-horária. Por fim, a tecla ‘z’ gira a câmera em torno do eixo Z de maneira horária; já ‘Z’, de maneira anti-horária.

Obs.: É obrigatória a utilização da projeção perspectiva nessa atividade. O *lookat* deve ser configurado de maneira a apresentar tanto o skybox quanto os objetos desenhados na cena.

¹[https://en.wikipedia.org/wiki/Skybox_\(video_games\)](https://en.wikipedia.org/wiki/Skybox_(video_games))