

Projeto (Etapa 3)

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

23 de maio de 2018

As atividades descritas a seguir devem seguir as seguintes restrições:

- Todas devem ser implementadas em grupos de **três ou quatro** alunos;
- 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**. Em ambos os casos, um arquivo com extensão “.txt” deve ser incluso, especificando os nomes dos integrantes do grupo.
- Caso haja plágio ou trabalhos muito similares, os grupos envolvidos receberão nota 0.
- O local de entrega deve ser feito na página da disciplina (<https://edisiplinas.usp.br/course/view.php?id=61213>), em uma atividade cujo título será “Trabalho prático 3”.

Contextualização do trabalho

Na última etapa do trabalho prático, foi requisitada a construção de um ator tridimensional com a forma de aranha e, ao utilizar as setas direcionais do teclado, esse ator deveria ser rotacionado e deslocado, sendo esse deslocamento somente para frente ou para trás, e a rotação em torno do próprio eixo. Para fixação dos conteúdos vistos até então, expandiremos o projeto prático adicionando texturas na cena.

Questões propostas

1. Implemente um ator em uma cena que tenha a forma de uma aranha. Esse ator deve ser constituído de cefalotórax, abdômen e oito pernas.
 - Pode-se aproximar as formas dos olhos, cefalotórax e abdômen com o uso de esferas.
 - As pernas podem ser aproximadas com o uso de retângulos ou linhas. Para as pernas, é necessário **no mínimo** uma articulação, visando simular a diferença de movimentação no membro.
2. Utilize a primitiva `GL_QUADS` para desenhar um “chão” na cena, sendo esse o local por onde a aranha pode se movimentar.
3. Permita com que, ao utilizar das setas do teclado, a aranha se movimente de maneira correspondente. No caso das setas para cima e para baixo, a aranha deve se deslocar ou para frente ou para trás, seguindo o eixo que ela está direcionada. Já com as setas da esquerda e da direita, a aranha deve rotacionar em torno do seu próprio eixo.
 - O movimento da aranha deve respeitar algumas restrições, definidas na primeira etapa deste projeto prático e duplicadas a seguir - para apoio na compreensão da movimentação de uma aranha, recomenda-se a visualização deste vídeo: https://www.youtube.com/watch?time_continue=8&v=dE2QPYPKju04.
 - Tomando como base a Figura 1, divide-se aqui os estados de movimento da aranha em três.

- $P1$ caracteriza o estado quando ela está parada, e nenhuma perna se encontra em movimento. Tem-se duas pernas mais compridas, denominadas “pernas externas”, e as pernas mais curtas, chamadas de “pernas internas”. As externas são responsáveis pelo impulso do cefalotórax e abdômen da aranha para frente, sendo as mais internas utilizadas para controle e sustentação de seu corpo.
- $P2$ indica o estado em que suas quatro pernas do lado direito do cefalotórax estão impulsionando o corpo para frente, similar a quando bípedes estendem a perna direita para se moverem. Percebe-se que as duas pernas mais “para fora” se estendem, ao passo que as duas pernas “para dentro” estão indo de encontro uma com a outra. Do lado esquerdo do cefalotórax, porém, a situação é inversa: as duas pernas mais externas estão se contraindo, indo de encontro às duas pernas mais internas que estão se estendendo.
- O movimento descrito em $P3$ é exatamente o mesmo de $P2$, porém acontecendo de maneira inversa - as pernas externas do lado direito estão se contraindo, ao passo que as externas do lado esquerdo estão se estendendo.

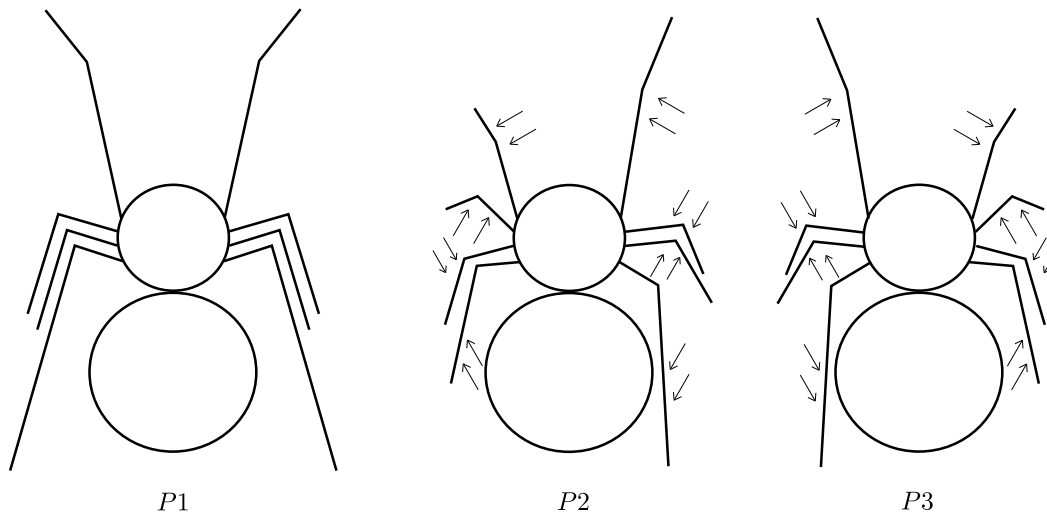


Figura 1: Visão de cima do corpo de uma aranha (simplificado). $P1$ representa o estado da aranha sem qualquer movimento. Já $P2$ e $P3$ representam os movimentos de quando as quatro pernas da direita e esquerda movem o corpo, respectivamente.

4. Implemente duas janelas para visualização da cena, definindo dois *viewports* distintos, sendo que nas duas, a câmera pode ser “posicionada” em qualquer lugar, como por exemplo nas coordenadas (3.0, 2.0, 10.0).
5. Manipule a função `gluLookAt(...)` de maneira a **sempre** manter a aranha no centro da porção visível da cena. Isso significa que a câmera deve ter como ponto focal central a aranha, em todas as janelas do programa.
6. Insira três texturas na cena, sendo que uma deve ser a textura do cefalotórax e abdômen da aranha, a outra do “chão” que a aranha caminha, e outra ao redor da cena. As texturas devem ser imagens carregadas externamente, com o auxílio de uma biblioteca de manipulação de imagem. Para esta etapa do trabalho, é permitida somente a utilização da biblioteca `OpenCV`, ou qualquer outro manipulador de imagem **implementado** pelo grupo.
 - Para a textura do corpo da aranha, utilize a função `gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)` para o desenho do cefalotórax e abdômen, e utilize ou uma textura 2D, 3D ou um *cube map* para definição dos parâmetros (cuja constantes são definidas, respectivamente, por `GL_TEXTURE_2D`, `GL_TEXTURE_3D` e `GL_TEXTURE_CUBE_MAP`).
 - Para a textura do “chão”, utilize ou uma textura 2D, 3D ou um *cube map* para definição dos parâmetros (cuja constantes são definidas, respectivamente, por `GL_TEXTURE_2D`, `GL_TEXTURE_3D` e `GL_TEXTURE_CUBE_MAP`).
 - Para a textura do céu, deve-se utilizar o conceito de *skybox*¹, em que seis imagens distintas devem ser dispostas de maneira a formar um cubo. Para essa textura, utilize somente *cube map* para definição dos parâmetros (cuja constantes são definidas por `GL_TEXTURE_CUBE_MAP`).
 - A escolha das imagens que serão utilizadas como texturas é livre.

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

7. Utilize o conceito de *distance fogging* ² para implementar uma simples neblina, utilizando a função `glFog(GLenum pname, TYPE *params)`, sem a necessidade de utilizar *shaders*. A cor da neblina deve ser cinza (com $RGBA = 0.5, 0.5, 0.5, 1.0$) e seu fator de densidade exponencial (`GL_EXP`). A neblina deve ser ativadaa pelo usuário ao pressionar a tecla 'n'.
8. O grupo também pode utilizar uma ou mais fontes de iluminação para a cena (OPCIONAL).

²https://en.wikipedia.org/wiki/Distance_fog