

Aula prática 5

Objetivos

- Construir um programa interactivo que permita instanciar primitivas.
- Modelar cada primitiva através duma transformação de instanciação (T.R.S) definida pelo utilizador.

Exercício 5.1

Pretende-se construir uma aplicação interativa que permita ao utilizador instanciar um objeto repetidas vezes, podendo aplicar, à última instância do objeto, uma transformação de instanciação, representada pela sequência de transformações elementares (T.R.S).

A aplicação deverá disponibilizar os seguintes elementos de interface:

- slider para ajustar os valores da translação (em X e em Y, sendo a componente Z sempre 0);
- slider para ajustar o valor da rotação em torno de Z;
- slider para ajustar o valor da escala (em X e em Y, sendo a componente Z sempre 1).

Os limites dos valores a aplicar aos sliders serão:

- Translação: [-1,1]
- Escala: [0.05,2]
- Rotação: [-180,180]

Os sliders aplicar-se-ão à CTM (current transformation matrix) aplicada à primitiva corrente.

Existe ainda um botão que permite guardar definitivamente a CTM e associá-la à primitiva que estava a ser editada, sem que se possa voltar atrás. De forma automática, instancia-se uma nova primitiva, com valores por omissão para as transformações, por forma a que a CTM seja a matriz identidade. Não esquecer atualizar os valores dos elementos da interface (sliders) para o seu valor refletir o estado atual.

Em cada momento, o programa deverá ser capaz de redesenhar todas as primitivas que foram guardadas, bem como a primitiva em edição.

Tarefas:

- Para começar, adapte um dos exemplos de trabalhos anteriores que desenhe uma primitiva (quadrado ou triângulo) no canvas.
- Acrescente no *vertex shader* uma variável uniform que corresponde à CTM.
- Inclua no ficheiro html os sliders para controlar a transformação corrente, bem como o botão para acrescentar de forma definitiva uma primitiva à lista de primitivas.
- Declare no javascript um vetor ou um dicionário que servirá para guardar a CTM associada a cada primitiva. Uma vez que a primitiva neste exemplo é única, não há necessidade de qualquer informação adicional, para lá da própria transformação de instanciação.
- Modifique a função `render()` para desenhar todas as primitivas guardadas, bem como a primitiva que está a ser editada no momento.
- Acrescente no seu programa as *callbacks* para tratar os eventos gerados pelos *sliders*. Sugestão: organize o seu programa por forma a que em resposta ao *slider* apenas tenha que alterar informação de estado (alterar valores de variáveis).

Exercícios propostos

TPC:

Exercício 5.2 - Acrescente as seguintes funcionalidades:

- um botão para fazer reset à transformação corrente.
- um botão para fazer reset ao programa, voltando ao estado inicial.

Exercício 5.3 - Experimente construir a CTM fazendo variar a ordem das transformações. Teste a aplicação e interprete o que vai acontecendo à medida que faz variar as transformações elementares.

Exercício 5.4 - Imagine que se quer dar ao utilizador a possibilidade de editar as transformações que foram aplicadas aos objetos entretanto criados. Para tal, o utilizador seleccionaria uma instância apontando no ecrã e fazendo click sobre ela. O programa deveria reagir ajustando os valores dos sliders para os valores que foram usados quando a instância foi armazenada.

Que alterações deveria efetuar na sua aplicação? Não as programe, basta que pense detalhadamente no que seria necessário alterar.