

Laboratório de Aplicações com Interface Gráfica

(Aulas Práticas)

MIEIC – 2017/2018

Trabalho Prático 2 – Aperfeiçoamento de Técnicas de Utilização de WebGL

O objetivo deste trabalho é introduzir novas técnicas gráficas, como animação e shaders baseados em GLSL ES 1.0 (OpenGL for Embedded Systems' Shading Language). Propõe-se assim a implementação de algumas funcionalidades em código, que possam depois ser exploradas através de uma extensão à linguagem LSX, e à criação de uma cena que as utilize. Este documento descreve as funcionalidades pretendidas, bem como a extensão proposta. O parser de LSX, realizado no TP1, deve ser utilizado e ampliado de forma a suportar as novas funcionalidades solicitadas neste enunciado.

1 Funcionalidades Pretendidas

1.1 Animação

Uma animação neste contexto corresponde a uma transformação geométrica representável em função do tempo. Assim, em intervalos de tempo aproximadamente regulares, a função devolve a função transformação geométrica a aplicar a um objeto.

Nesta fase do trabalho, dever-se-á implementar um conjunto de classes `Animation`, `LinearAnimation` e `CircularAnimation`, de acordo com as sub-seções seguintes.

1.1.1 Classe `Animation`

Implementar a classe `Animation` como classe base (abstrata) para aplicar animações a um objeto.

1.1.2 Classe `LinearAnimation`

a) Criar a classe `LinearAnimation`, derivada de `Animation`, para trajetórias lineares, que permita definir uma animação caracterizada por um vetor de Pontos de Controlo e pela velocidade de movimentação (esta última medida em unidades 3D por segundo)

Exemplo:

Pontos de Controlo = $\{(0,0,0), (1,0,0), (1,1,0)\}$

Velocidade = 10 unidades/segundo

b) O objeto em movimento deve alterar a sua orientação horizontal (x,z), rodando em torno de um eixo vertical, de modo a corrigir a direção quando, de acordo com a trajetória, muda de segmento de reta (ou seja, o movimento é tipo "helicóptero").

1.1.3 Classe CircularAnimation

a) Criar a classe CircularAnimation, derivada de Animation, para trajetórias circulares em planos paralelos a ZX (rotação em torno de um eixo vertical), que permita definir uma animação caracterizada pelo centro e raio de circunferência, ângulo inicial (medido em relação à direção positiva do eixo XX), ângulo de rotação (este último a adicionar ao ângulo inicial), ambos em graus, velocidade linear (esta última correspondendo a um comprimento de arco, medido em unidades 3D, por segundo).

Exemplo:

Centro = (10, 10, 10)

Raio = 5

Ângulo Inicial = 40°

Ângulo de rotação = 20°

Velocidade = 10 unidades/segundo

b) O objeto em movimento deve alterar a sua orientação horizontal (x,z) de acordo com a trajetória, tal como no caso anterior.

1.1.4 Classe BezierAnimation

Criar a classe BezierAnimation para trajetórias no espaço 3D. A animação é caracterizada pelas coordenadas dos quatro pontos de controlo da curva de Bézier correspondente, assim como velocidade linear da mesma. O objeto animado deve manter uma orientação tangencial à trajetória no plano ZX, ou seja, rodando em torno do eixo vertical Y.

NOTA: o comprimento da curva pode ser aproximado pelos dois *convex hull* que resultam da aplicação do primeiro nível do algoritmo de Casteljau.

Exemplo:

P1, P2, P3, P4= {(0,0,0), (1,0,0), (1,1,0), (0,1,0)}

Velocidade = 10 unidades/segundo

1.1.5 Classe ComboAnimation

Implementar uma classe ComboAnimation que permita realizar uma animação constituída por uma sequência de instâncias das classes anteriores, segundo a extensão proposta abaixo para a linguagem LSX.

Um objeto animado desta forma deverá manter a sua horizontalidade (ou seja, manter-se paralelo ao plano XZ), apenas devendo alterar a sua orientação por rotação em torno de um eixo vertical; a orientação do objeto no plano ZX deve ser coerente com a direção e sentido do seu movimento (como p.ex. um helicóptero).

NOTA: uma ComboAnimation não pode conter outra ComboAnimation.

1.1.6 Aplicação em Cenas LSX

Criar uma cena em LSX que inclua objetos animados segundo as classes de animação definidas anteriormente.

No mínimo, deverá incluir duas animações:

a)- Uma animação composta por, pelo menos, dois segmentos de reta (classe LinearAnimation), intermediados por um segmento circular (CircularAnimation).

b)- Uma animação composta por, pelo menos, duas curvas de Bézier (classe BezierAnimation) e um ou mais troços com base em segmentos de reta ou circulares.