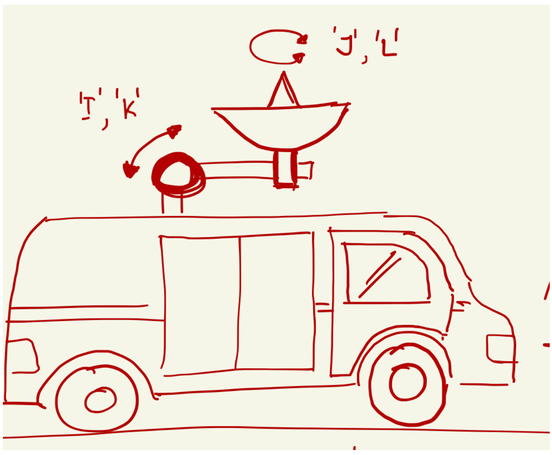
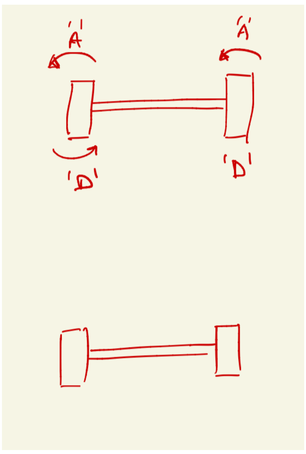
Projeto 2: Modelação hierárquica

Change Log:

- 02/Nov/2018 @ 02h30 - Publicação do enunciado.





Objetivo

Neste projeto vamos programar uma aplicação que nos permitirá visualizar e controlar uma carrinha de comunicações (aquelas que dão suporte a emissões televisivas de exteriores). A carrinha irá constituir um modelo hierárquico 3D, representado por um grafo de cena. O modelo será constituído **exclusivamente** a partir dos objetos primitivos disponibilizados nas aulas, com a versão atualizada aqui. A carrinha deverá permitir ao utilizador o controlo:

- Do seu avanço/recuo, usando as teclas 'W' e 'S'. A carrinha avança e recua sempre na sua direção ao longo do comprimento. Se as rodas não estiverem alinhadas, o programa deverá fazê-lo automaticamente antes de avançar ou recuar.
- Da rotação das suas rodas da frente, usando as teclas 'A' e 'D'. Esta rotação só terá efeito com a carrinha parada.
- Da elevação do braço que suporta a antena parabólica (modelada através dum objeto do tipo parabolóide que será da vossa responsabilidade modelar), usando as teclas 'I' e 'K'
- Da rotação do braço que suporta a antena parabólica (devendo esta rodar por estar presa ao referido braço), usando as teclas 'J' e 'L'

Para além da representação da carrinha deverá ainda desenhar o solo (sugere-se a utilização de vários cubos encostados uns aos outros, em grelha). Durante o deslocamento da carrinha as rodas deverão rodar sem derrapar.

Todos os objetos serão desenhados em malha de arame (wireframe), devendo o programa desenhar as várias componentes da carrinha com cores diferentes, para se distinguirem bem.

A cena deverá poder ser observada, em cada momento, por uma de 4 câmaras, com projeção ortogonal, pré-definidas:

- Vista de topo, tecla '1'
- Vista lateral, tecla '2'
- Vista de frente, tecla '3'
- Vista duma direção arbitrária, à vossa escolha, tecla '0'

Modos de visualização dos objetos

A visualização das primitivas será efetuada em dois modos distintos:

- Malha de arame — apenas são visíveis as arestas correspondentes às faces, interligado os vértices, mas as cores de cada componente deverão permitir a sua fácil distinção. Por exemplo, as rodas poderão ser pretas, as jantes, cinzentas, a carroçaria branca, etc.
- A visualização em malha de arame deverá igualmente alternar entre a visualização dos objetos com cor fixa ou mapeando e interpolando os valores absolutos dos vetores normal em cada vértice. Para a comutação entre estes dois modos deverá usar-se a barra de espaços.

Detalhes técnicos

Variáveis (conjunto mínimo) a passar para os shaders

Cada modelo enviará para o vertex shader os seguintes atributos:

- vPosition — posição do vértice;
- vNormal — `vec3` descrevendo o vetor perpendicular da superfície naquele vértice.

Para além destes atributos, o vertex shader deverá ainda receber as seguintes matrizes (`uniform mat4`):

- mProjection — tratará do volume de visualização do tipo de projeção selecionada.
- mModelView —Composição das matrizes de modelação e de orientação da câmara na seguinte forma: mModelView = mView * mModel. A matriz mView tratará de orientar o objeto, de acordo com a projeção escolhida, de modo a que a direção de projeção seja a do eixo Z (com recurso à função lookAt). A matriz mModel será distinta para cada primitiva, de acordo com o percurso efetuado no grafo de cena.

O fragment shader deverá receber a cor com que um determinado objeto será desenhado.

Regras e Informação Adicional

Composição dos grupos

Os trabalhos práticos deverão ser realizados por grupos de 2 alunos dum mesmo turno prático. A entrega do trabalho a título individual terá que ser devidamente justificada e autorizada pelo respectivo docente do turno prático.

Entrega

O trabalho será entregue via Google Classroom, usando a classe do respectivo turno prático até às **23h59 de 15 de Novembro de 2020**.

Ambos os alunos do grupo terão que efetuar a **submissão do trabalho**.

Na pasta Common apenas deverão estar ficheiros a que todos os alunos têm acesso, nomeadamente as bibliotecas fornecidas (MV.js, webgl-utils, initShaders), bem como os objetos fornecidos (sphere.js, cylinder.js, etc.), inalterados. Os restantes ficheiros deverão estar todos localizados na pasta onde se encontra o .html e o .js principal da aplicação, incluindo a vossa implementação do objeto parabolóide.

Assim, apenas deverão entregar os ficheiros localizados na mesma pasta onde se encontra o .html e o .js da aplicação. Na pasta deverá ainda constar um PDF com o grafo de cena, devendo ser usada a notação seguida nas aulas teóricas.

Importante!

Apenas poderão usar os objetos fornecidos nas aulas (e o vossa parabolóide).

Avaliação

Os trabalhos serão avaliados pelo respetivo docente das aulas práticas e discutidos com os respetivos alunos em data a definir oportunamente.