

Modelação Hierárquica (versão 2)

Modelação e visualização duma carro de combate a incêndios

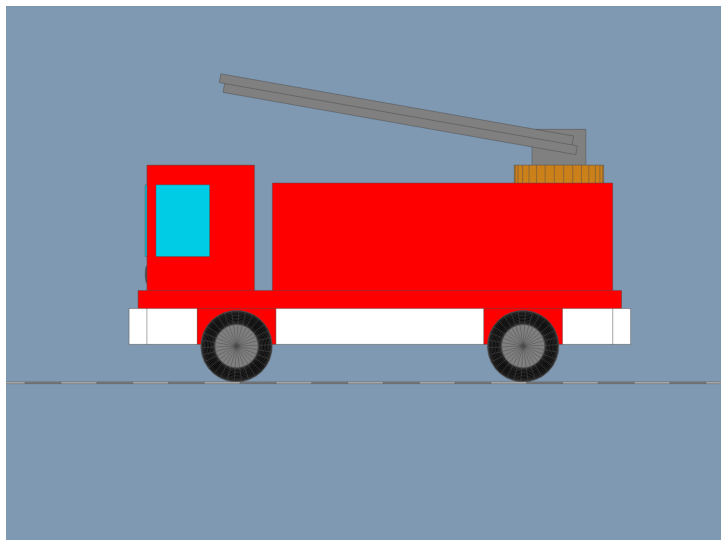


Fig. 1 - Alçado Principal

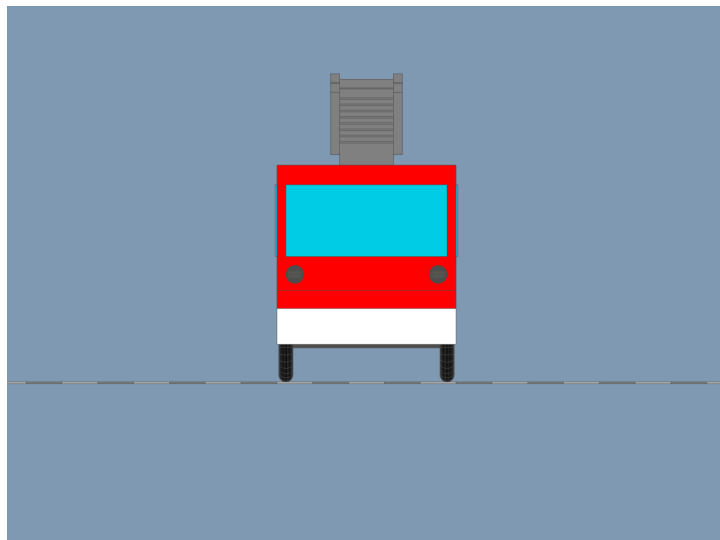


Fig. 2 - Alçado Lateral Esquerdo

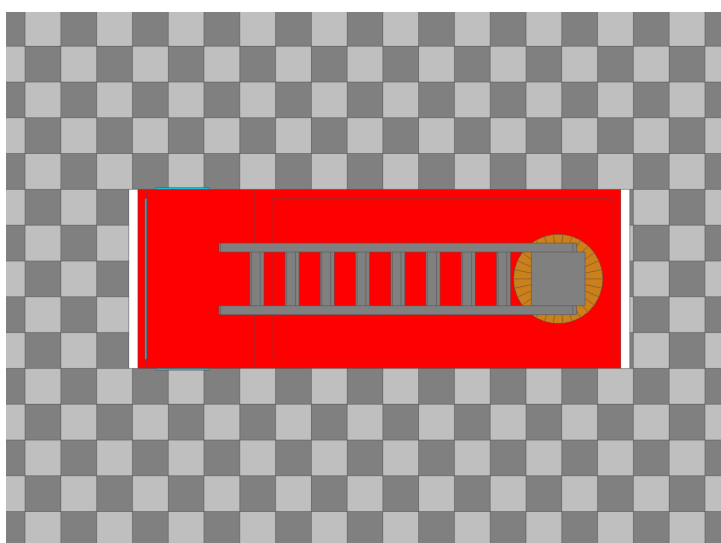


Fig. 3 - Planta

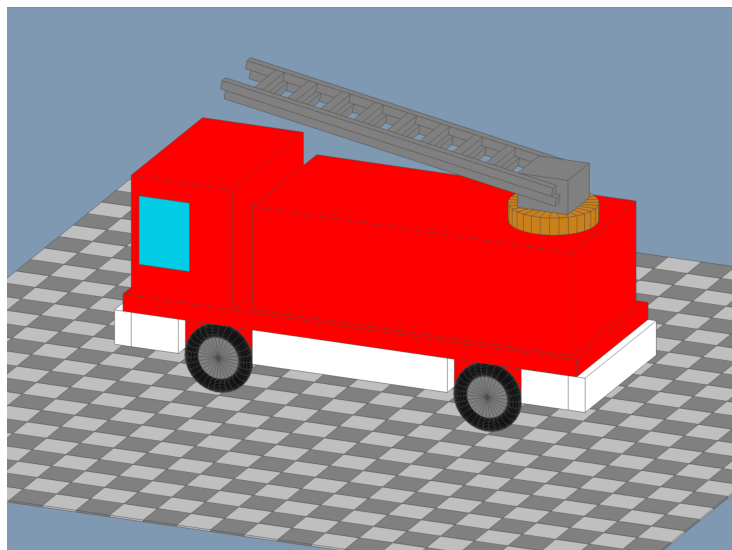


Fig. 4 - Projeção Axonométrica

Objetivo

Neste trabalho pretende-se desenvolver uma aplicação interativa que permita manipular e visualizar um carro de combate a incêndios, semelhante ao representado nas Figuras 1-4.

A iteração é efetuada quase exclusivamente recorrendo ao teclado e, na Figura 5, são apresentados os comandos que a aplicação deverá suportar. Estes comandos estão divididos em:

- comandos de manipulação da carro ('w', 's', 'i', 'k', 'j', 'l', 'a', 'd')

- comandos para escolher a projeção e alternar entre 1/4 vistas ('1', '2', '3', '4' e '0')
- comandos para alterar os parâmetros, θ e γ , da projeção axonométrica (teclas de cursor)
- comando para mudar o modo de visualização, entre os modos wireframe e sólido (' ') e fazer reset ao fator de zoom ('r').

Keys:

- 'h' - Toggle this panel
- '0' : Toggle 1/4 views
- '1' : Front view
- '2' : Left view
- '3' : Top view
- '4' : Axonometric view
- ' ' : Toggle wireframe/solid
- 'q' : Rotate ladder CCW
- 'e' : Rotate ladder CW
- 'w' : Raise ladder
- 's' : Lower ladder
- 'o' : Extend ladder
- 'p' : Reduce ladder
- 'a' : Move forward
- 'd' : Move backward
- 'r' : Reset view params
- 'ArrowLeft' : Increase theta
- 'ArrowRight' : Decrease theta
- 'ArrowUp' : Increase gamma
- 'ArrowDown' : Decrease gamma

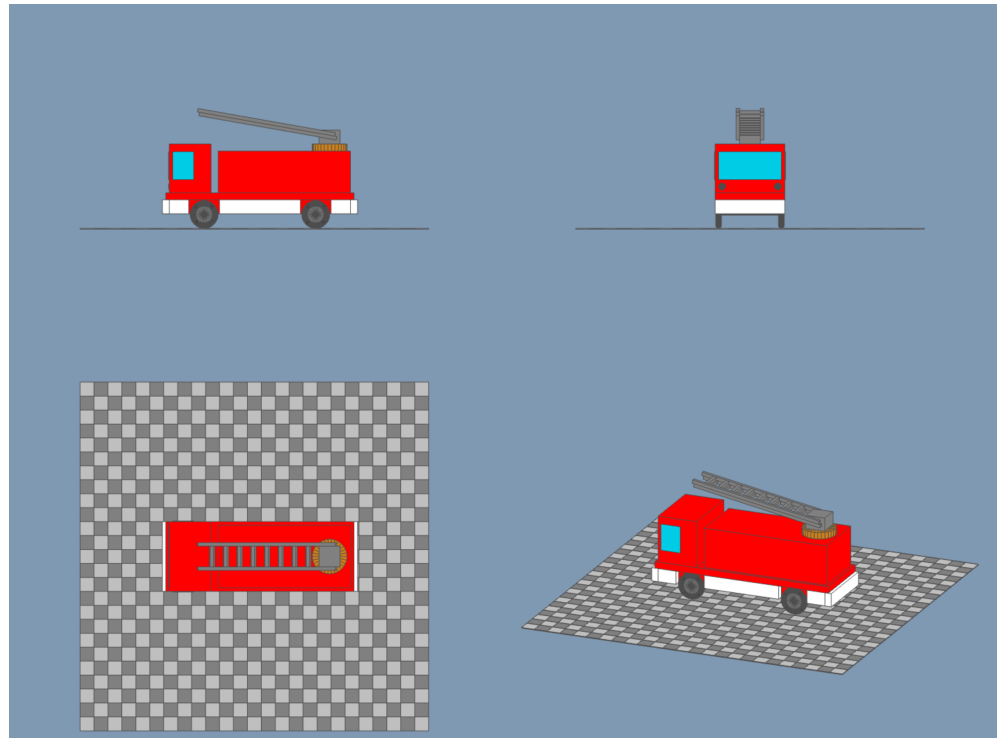


Fig.5 - Comandos do programa

Fig. 6 - Projeções múltiplas

Adicionalmente, o utilizador deverá poder usar a roda do rato para ampliar/reduzir a imagem, mantendo o centro. A visualização do carro não deverá sofrer deformação aquando do redimensionamento da janela por parte do utilizador. A aplicação deverá também garantir que existe alguma forma de visualizar o carro por completo.

Para além do carro, a aplicação deverá desenhar o chão ($y=0$) recorrendo a uma disposição de cubos, de cores alternadas, em xadrez.

Quando o utilizador escolhe 4 vistas em simultâneo, a disposição das mesmas deverá obedecer ao método europeu de projeções múltiplas, como se pode ver na Figura 6. O efeito consegue-se desenhando a cena 4 vezes, uma em cada visor, estando o canvas dividido em 4 visores.

Detalhes técnicos

Modelação do carro

A modelação do carro deverá exibir uma hierarquia de elementos que agreguem as diferentes partes do mesmo. Tanto quanto possível deverá definir constantes que determinam características chave, tais como as dimensões de certas partes do modelo. Alguns exemplos (e são apenas exemplos, sem qualquer carácter vinculativo) poderiam ser: o número de rodas atrás, a altura da cabina, a largura da zona de carga, o número de degraus de cada escada, etc. O carro é de desenho e dimensões livres, embora tenha que conter os seguintes elementos:

- Uma escada que pode ser estendida ou recolhida, composta por dois elementos, com degraus (comandos 'o' e 'p')
- A escada do ponto anterior poderá ainda ser regulada na sua inclinação (comandos 'w' e 's')
- A escada estará montada numa plataforma, colocada no topo do carro, a qual poderá rodar em ambos os sentidos (comandos 'q' e 'e')
- O carro deverá ser composto por uma cabine e uma zona de carga.
- O carro deverá ter 4 rodas, as quais poderão rodar, em função do deslocamento aplicada ao carro (comandos 'a' e 'd')
- No total, o carro deverá possuir um número mínimo de 10 primitivas. O exemplo apresentado contém muitas mais...

Há **duas** opções para a implementação do desenho do carro (isto não se aplica ao desenho do chão)

- Depois de desenhar/construir o grafo de cena, gerar o código correspondente, tal como feito nas aulas práticas e nos exemplos das aulas teóricas.
- Depois de desenhar/construir o grafo, criar um objeto javascript, em árvore, que representa esse mesmo grafo e implementar uma função capaz de o varrer e efetuar o desenho das primitivas.

Para a segunda opção deverão ser considerados nós do tipo:

- ramo/nó interno com transformações e com descendentes
- ramo/nó terminal com transformações e com uma primitiva

O grafo terá obrigatoriamente na raiz um ramo/nó do tipo a). Nesta opção, deverá considerar a hipótese de os ramos/nós possuírem um nome, possibilitando assim a escrita duma função que,

dado o nome do ramo/nó, devolva a referência para o respetivo objeto. Assim, poderão facilmente ser implementados os ‘handlers’ de eventos que irão alterar os parâmetros das transformações do grafo.

O desenho do chão será efetuado sem uma representação em memória do respetivo grafo, escrevendo diretamente o código, tal como se fez nas aulas práticas.

Avaliação

A avaliação do trabalho está definida da seguinte forma:

- Modelação do carro com os elementos básicos referidos e restantes funcionalidades: 17 valores, dos quais 2 valores são atribuídos à capacidade do modelo ser parametrizável manipulando as constantes que determinam o seu aspeto.
- Colocação de elementos decorativos adicionais (piscas, vidros, espelhos retrovisores, pára-choques, pirilampo, etc.): 3 valores