**Para cada uma das projeções, quais os valores que ficam nas matrizes mModel, mView e mProjection?**

**mModel**

Matriz identidade em todas as projeções (mat4()).

**mView**

**Projeções ortogonais**

|  |  |  |
| --- | --- | --- |
| Alçado Principal | Planta | Alçado Lateral Direito |
|  |  |  |
| mat4() | rotateX(90) | rotateY(-90) |

**Projeções axonométricas**

Isometria: axonometricMatrix(30,30)

Dimetria: axonometricMatrix(42,7)

Trimetria: axonometricMatrix(54.27, 23.27)

**Projeções Oblíquas**

|  |  |
| --- | --- |
| Cavaleira | Gabinete |
|  |  |
| obliqueMatrix(1, 45) | obliqueMatrix(0.5, 45) |

**Projeção perspetiva:** perspectiveMatrix(d)

**mProjection**

A matriz é a mesma em todas as projeções:

mult(ortho(-1\*aspectRatio, 1\*aspectRatio, -1, 1, 10, -10), scalem(zoom, zoom, 1))

**Onde/Como fazem o ajuste da escala (manipulada com o scroll)?**

Para este fim, criamos um event listener para scroll no canvas, de modo que este chame a função zoomCanvas(e). Esta função determina o valor do zoom que será usado para fazer o ajuste da escala no canvas. Na função updateCanvas(), o ajuste de escala é feito multiplicando a matriz ortogonal pela matriz de escala, com o valor do zoom corrente em x e y, na mProjection.

**Onde/Como tratam de efetuar o ajuste para que não haja deformação quando se redimensiona a janela?**

Mais uma vez, criamos um event listener para redimensionar a janela, de modo a chamar a função updateCanvas(). Nesta função, define-se o comprimento do canvas igual ao comprimento da janela e a largura igual a 60% da largura da janela. Depois da atribuição dos tamanhos do canvas, calcula-se o aspect ratio desse. O aspect ratio será usado como valor para os parâmetros left e right da matriz ortogonal.

**Quais os limites que definiram para cada um dos parâmetros de cada projeção?**

|  |  |  |
| --- | --- | --- |
| Axonométrica | Oblíqua | Perspetiva |
| A: 0 -> 360 | l: 0 -> 5 | d: 2 -> 5 |
| B: 0 -> 360 | α: 0 -> 360 |  |

**Observações:**

Neste projeto, apesar de não ser necessário, decidimos criar um programa só para a Superfície Quádrica, tendo em conta a queda de desempenho considerável num dos nossos computadores, além de ser a opção mais eficiente. O código do respetivo vertex shader está incluido no ficheiro html.