Câmera no WebGL
Uma versão de Implementação

Posicionamento

- Realizado pela descrição dos vetores Direção e UP, e pelo ponto Posição (EYE)
  - Usados na matriz de lookAt
    - Está implementada na biblioteca glMatrix

DIR, UP e EYE

- EYE (Posição) → PONTO que especifica a posição da câmera no sistema Global
- UP → VETOR que aponta para a direção para cima da câmera
- DIR → VETOR que mostra a direção da câmera, ou seja, para onde ela está olhando
- ***TODOS*** os vetores devem ser normalizados e mantidos ortográficos entre si

Mapeamento para a Matriz de LookAt

- Parâmetros da matriz de lookAt
  - EYE → posição dos olhos
  - CENTER → ponto onde os olhos estão olhando
  - UP → onde está a parte de cima da câmera
- Mapeando na nossa descrição:
  - EYE = EYE
  - CENTER = EYE + DIR
  - UP = UP

Manipulação dos Vetores

- Dois tipos de movimentos
  - Translação
    - Altera a posição da câmera sem alterar a sua direção e/ou orientação
      - Somente o ponto EYE é alterado
  - Rotação
    - Altera a direção e/ou orientação, sem alterar a posição
      - Somente os vetores UP e DIR são alterados

Movimento de Translação

- Subir ou descer → caminha na direção do vetor UP
  - EYE = EYE ± UP
- Ir para frente ou para trás → caminha na direção do vetor NORMAL
  - EYE = EYE ± DIR

Movimento de Translação (2)

- Esquerda ou direita, caminha na direção do vetor RIGHT
  - Vetor RIGHT = DIR x UP
  - EYE = EYE ± RIGHT

Movimento de Rotação

- Rotação em torno do vetor DIR
  - Quaternion formado por ÂNGULO vetor DIR
    - Aplica-se o operador rotação no vetor UP
- Rotação em torno do vetor UP
  - Quaternion formado por ÂNGULO vetor UP
    - Aplica-se o operador rotação no vetor DIR

Movimento de Rotação (2)

- Rotação em torno do vetor RIGHT
  - Lembrando que: RIGHT = DIR x UP
  - Quaternion: ÂNGULO vetor RIGHT
    - Aplica-se o operador rotação tanto no vetor UP, quanto no vetor DIR

Matriz MVP

- MVP → Model, View, Projection
  - Define as transformações do sistema local para o sistema unitário
    - Model → transformação do local para o global
    - View → matriz de lookAt
    - Projection → projeção
    - Transformação → multiplicação dada por:
      - T = P * V * M
  - Normalmente implementada no shader Vertex

Vertex 2.1

```
#version 120

attribute vec3 position;

uniform   mat4 model;
uniform   mat4 view;
uniform   mat4 projection;

void main()
{
        gl_Position = projection * view * model * vec4(position, 1.f);
}
```