Referencial da Câmara e Referencial do Mundo

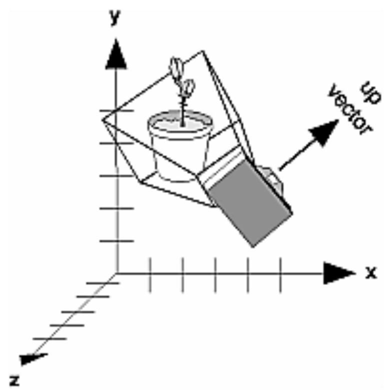
LookAt (World to Camera)

Objetivos

- Saber implementar uma orientação genérica da câmara
- Conhecer a função lookAt() da biblioteca MV.js
- Deduzir a matriz que transforma do referencial do mundo para o referencial da câmara

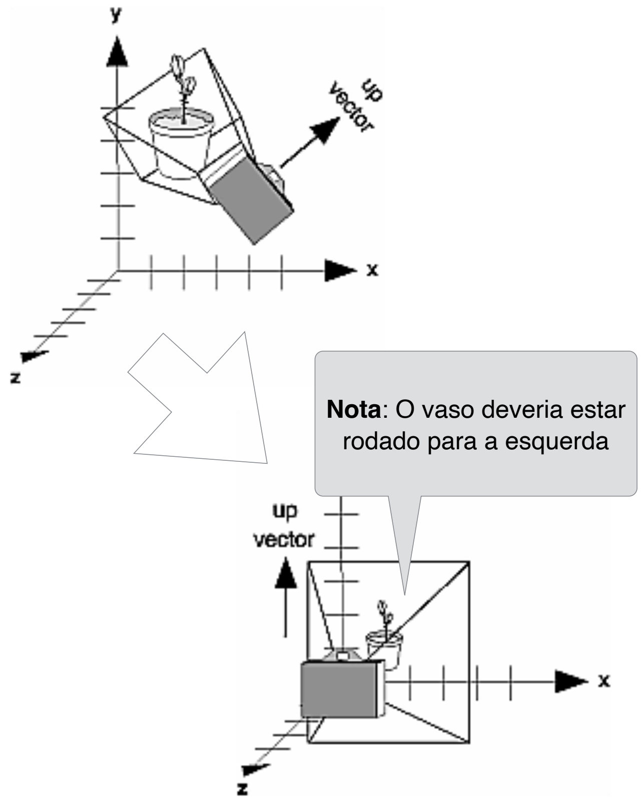
Problema

- Imaginemos que se pretende projetar uma cena como se fosse captada por uma câmara arbitrariamente posicionada e orientada.
- Como deduzir a transformação de World Coordinates para Camera Coordinates ou Eye Coordinates?
- Para ajudar a perceber a solução, vamos imaginar que de seguida, queremos fazer uma projeção perspetiva usando M_{per} (câmara na origem, apontada para $-z$ e direção vertical coincidente com y).



Solução

- Transformar coerentemente (da mesma forma) todos os objetos da cena, incluindo a câmara de tal modo que a câmara passe a estar na origem, apontada para o lado negativo do eixo z e com a direção vertical coincidente com o eixo y.
- Pode fazer-se em 2 passos:
  - Colocar a câmara na origem
  - Rodar por forma a alinhar o referencial da câmara com o referencial do mundo

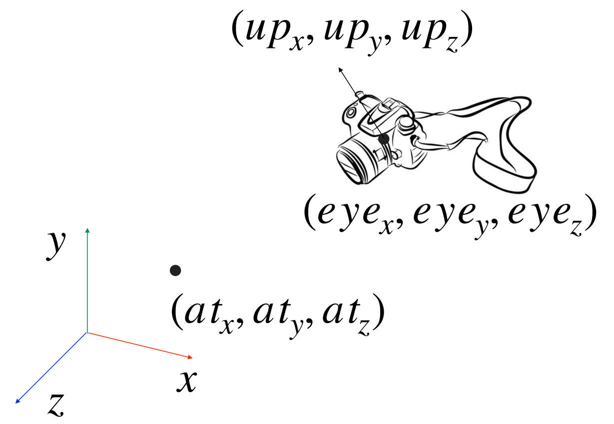


lookAt()

- A biblioteca `MV.js` oferece uma função que permite especificar uma orientação genérica para a câmara:

```
lookAt(eye, at, up)
```

- eye, at e up são fornecidos em coordenadas do mundo (**W**orld **C**oordinates)



lookAt: estratégia para implementação

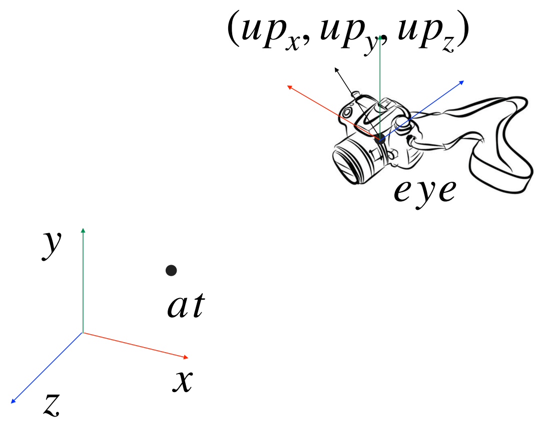
- Estabelecer um referencial ortonormado para a câmara
- Mover o ponto eye para a origem do referencial do mundo

$$T(-eye)$$

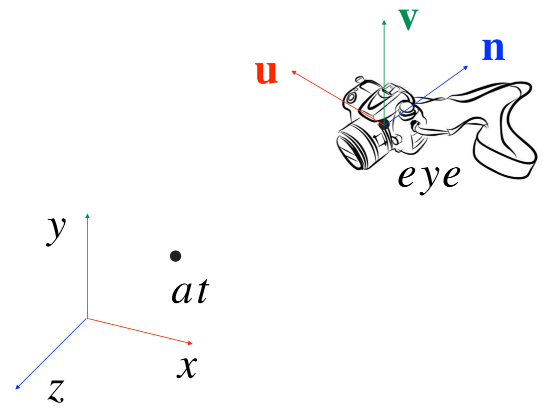
- Rodar de forma a que o referencial da câmara se alinhe com o referencial do mundo

Seja R a matriz respetiva

- A transformação final será: $R \cdot T(-eye)$

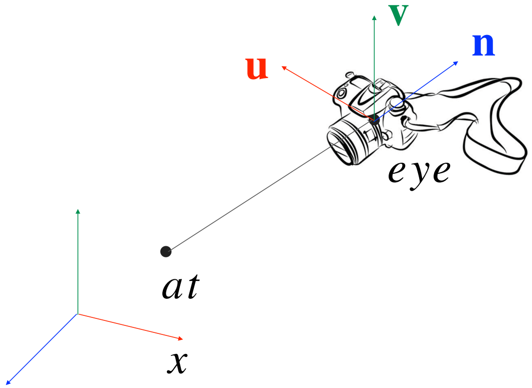


Referencial da câmara



Determinação de u, v e n

- **u**, **v** e **n** são perpendiculares entre si e formam um sistema de coordenadas direito.
- **n** está alinhado com o vetor que define a direção para onde a câmara está apontada, mas com o sentido inverso

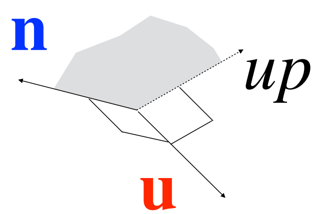


$$\mathbf{n} = \frac{eye - at}{\|eye - at\|}$$

Determinação de u, v e n

- O vetor *up* poderá, por mero acaso, ser logo perpendicular a **n** e, nesse caso, **v** estaria determinado. Não estando isso garantido, é preferível determinar primeiro o vetor **u**.
- **u** representa a direção horizontal da câmara e será perpendicular ao plano definido pelos vetores up e **n**.

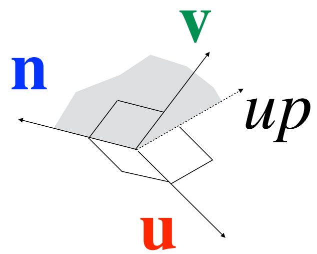
$$\mathbf{u} = \frac{up \times \mathbf{n}}{\|up \times \mathbf{n}\|}$$



Determinação de u, v e n

- Finalmente, o vetor $\mathbf{v}$ pode agora ser facilmente determinado por forma a ser perpendicular quer a $\mathbf{u}$, quer a $\mathbf{n}$:

$$\mathbf{v} = \mathbf{n} \times \mathbf{u}$$



Dedução de R

- A matriz R, será responsável por orientar o referencial da câmara (entretanto com a origem partilhada com o referencial do mundo via a translação entretanto efetuada), por forma a fazer coincidir os seguintes pares de eixos:
  - $\mathbf{u}$ com x
  - $\mathbf{v}$ com y
  - $\mathbf{n}$ com z

- Outra forma de interpretar o papel desta matriz é pensar que ela transforma pontos e vetores no referencial do mundo para o referencial $\mathbf{u}$,$\mathbf{v}$,$\mathbf{n}$ (com a origem em comum). Assim:
  - $R \cdot \mathbf{u} = [1 \quad 0 \quad 0 \quad 0]^T$
  - $R \cdot \mathbf{v} = [0 \quad 1 \quad 0 \quad 0]^T$
  - $R \cdot \mathbf{n} = [0 \quad 0 \quad 1 \quad 0]^T$

Dedução de R

$$R \cdot \mathbf{u} = [1 \quad 0 \quad 0 \quad 0]^T, \quad R \cdot \mathbf{v} = [0 \quad 1 \quad 0 \quad 0]^T, \quad R \cdot \mathbf{n} = [0 \quad 0 \quad 1 \quad 0]^T$$

Sendo R uma matriz de rotação, com a forma:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Resulta:

$$[r_{11} \quad r_{12} \quad r_{13} \quad 0] \cdot \mathbf{u} = 1 \qquad [r_{11} \quad r_{12} \quad r_{13} \quad 0] \cdot \mathbf{v} = 0 \qquad [r_{11} \quad r_{12} \quad r_{13} \quad 0] \cdot \mathbf{n} = 0$$

$$[r_{21} \quad r_{22} \quad r_{23} \quad 0] \cdot \mathbf{u} = 0 \qquad [r_{21} \quad r_{22} \quad r_{23} \quad 0] \cdot \mathbf{v} = 1 \qquad [r_{21} \quad r_{22} \quad r_{23} \quad 0] \cdot \mathbf{n} = 0$$

$$[r_{31} \quad r_{32} \quad r_{33} \quad 0] \cdot \mathbf{u} = 0 \qquad [r_{31} \quad r_{32} \quad r_{33} \quad 0] \cdot \mathbf{v} = 0 \qquad [r_{31} \quad r_{32} \quad r_{33} \quad 0] \cdot \mathbf{n} = 1$$

Dedução de R

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \end{bmatrix} \cdot \mathbf{u} = 1 \qquad \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \end{bmatrix} \cdot \mathbf{v} = 0 \qquad \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \end{bmatrix} \cdot \mathbf{n} = 0$$

$$\begin{bmatrix} r_{21} & r_{22} & r_{23} & 0 \end{bmatrix} \cdot \mathbf{u} = 0 \qquad \begin{bmatrix} r_{21} & r_{22} & r_{23} & 0 \end{bmatrix} \cdot \mathbf{v} = 1 \qquad \begin{bmatrix} r_{21} & r_{22} & r_{23} & 0 \end{bmatrix} \cdot \mathbf{n} = 0$$

$$\begin{bmatrix} r_{31} & r_{32} & r_{33} & 0 \end{bmatrix} \cdot \mathbf{u} = 0 \qquad \begin{bmatrix} r_{31} & r_{32} & r_{33} & 0 \end{bmatrix} \cdot \mathbf{v} = 0 \qquad \begin{bmatrix} r_{31} & r_{32} & r_{33} & 0 \end{bmatrix} \cdot \mathbf{n} = 1$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \end{bmatrix} \parallel \begin{bmatrix} u_x & u_y & u_z \end{bmatrix} \qquad \begin{bmatrix} r_{11} & r_{12} & r_{13} \end{bmatrix} \perp \begin{bmatrix} v_x & v_y & v_z \end{bmatrix} \qquad \begin{bmatrix} r_{11} & r_{12} & r_{13} \end{bmatrix} \perp \begin{bmatrix} u_x & u_y & uz \end{bmatrix}$$

$$\begin{bmatrix} r_{21} & r_{22} & r_{23} \end{bmatrix} \perp \begin{bmatrix} u_x & u_y & u_z \end{bmatrix} \qquad \begin{bmatrix} r_{21} & r_{22} & r_{23} \end{bmatrix} \parallel \begin{bmatrix} v_x & v_y & v_z \end{bmatrix} \qquad \begin{bmatrix} r_{21} & r_{22} & r_{23} \end{bmatrix} \perp \begin{bmatrix} u_x & u_y & uz \end{bmatrix}$$

$$\begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} \perp \begin{bmatrix} u_x & u_y & u_z \end{bmatrix} \qquad \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} \perp \begin{bmatrix} v_x & v_y & v_z \end{bmatrix} \qquad \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} \parallel \begin{bmatrix} u_x & u_y & uz \end{bmatrix}$$

Dedução de R

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \end{bmatrix} \cdot \mathbf{u} = 1 \qquad \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \end{bmatrix} \cdot \mathbf{v} = 0 \qquad \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \end{bmatrix} \cdot \mathbf{n} = 0$$

$$\begin{bmatrix} r_{21} & r_{22} & r_{23} & 0 \end{bmatrix} \cdot \mathbf{u} = 0 \qquad \begin{bmatrix} r_{21} & r_{22} & r_{23} & 0 \end{bmatrix} \cdot \mathbf{v} = 1 \qquad \begin{bmatrix} r_{21} & r_{22} & r_{23} & 0 \end{bmatrix} \cdot \mathbf{n} = 0$$

$$\begin{bmatrix} r_{31} & r_{32} & r_{33} & 0 \end{bmatrix} \cdot \mathbf{u} = 0 \qquad \begin{bmatrix} r_{31} & r_{32} & r_{33} & 0 \end{bmatrix} \cdot \mathbf{v} = 0 \qquad \begin{bmatrix} r_{31} & r_{32} & r_{33} & 0 \end{bmatrix} \cdot \mathbf{n} = 1$$

$$\Downarrow$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}$$

$$\begin{bmatrix} r_{21} & r_{22} & r_{23} \end{bmatrix} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}$$

$$\begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}$$

Conclusão

Desenvolvendo $R \cdot T(-eye)$:

$$\begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -eye_x \\ 0 & 1 & 0 & -eye_y \\ 0 & 0 & 1 & -eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z & -eye \cdot u \\ v_x & v_y & v_z & -eye \cdot v \\ n_x & n_y & n_z & -eye \cdot n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz criada pela função lookAt: transforma coordenadas do mundo em coordenadas da câmara

lookAt()

- A matriz M_{view} devolvida por lookAt() deverá ser aplicada após as transformações de modelação M_{model}, que convertem os objetos primitivos em instâncias na cena e antes da projeção, M_{proj}

$$P' = M_{proj} \cdot M_{view} \cdot M_{model} \cdot P$$

- As projeções axonométricas poderão, em alternativa, ser substituídas pela projeção ortogonal no plano XY, com recurso a lookAt.
- Exemplo (isometria):

lookAt($\underbrace{\text{vec3(1,1,1)}}_{\boxed{eye}}$, $\underbrace{\text{vec3(0,0,0)}}_{\boxed{at}}$, $\underbrace{\text{vec3(0,1,0)}}_{\boxed{up}}$)