LookAt

2019-2020

Fernando Birra

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

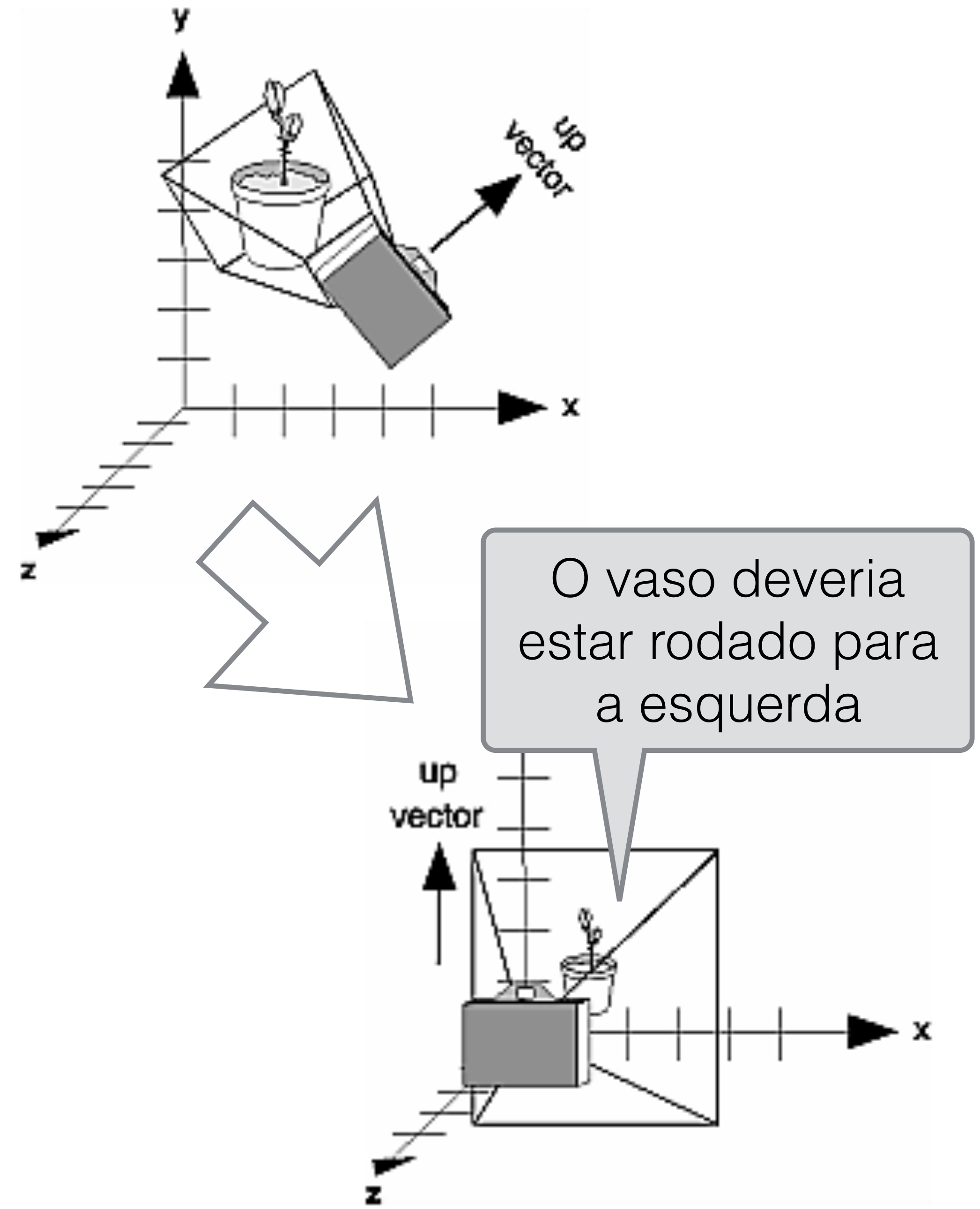
Problema

- Nas (maior parte das) matrizes de projeção estudadas assume-se que o plano de projeção é o plano Z=0, e que o observador está a olhar para o lado negativo do eixo Z.

- Como conciliar isso com uma orientação arbitrária da câmara?

Solução

- Transformar coerentemente (da mesma forma) todos os objetos da cena e 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.

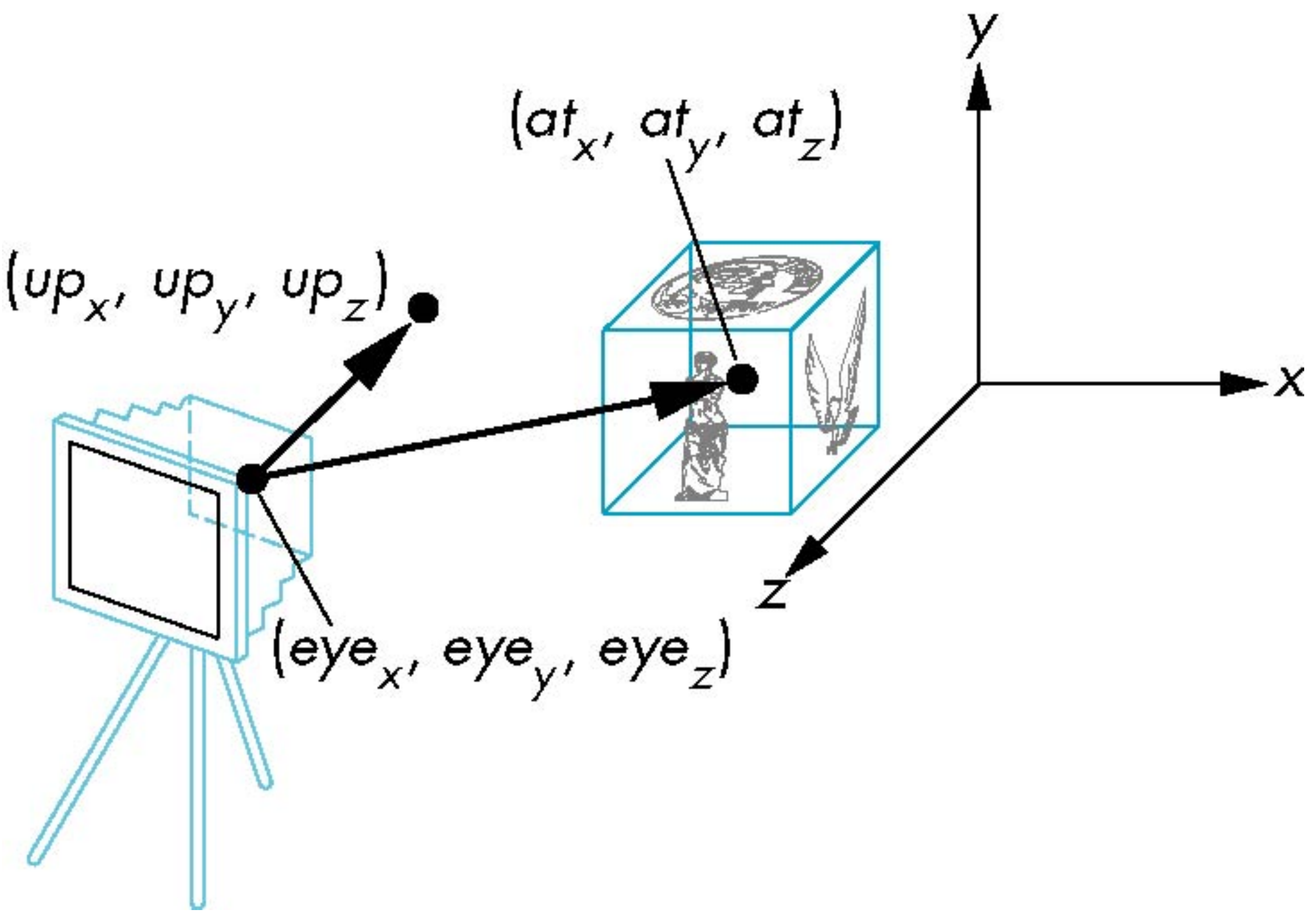


lookAt()

- 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

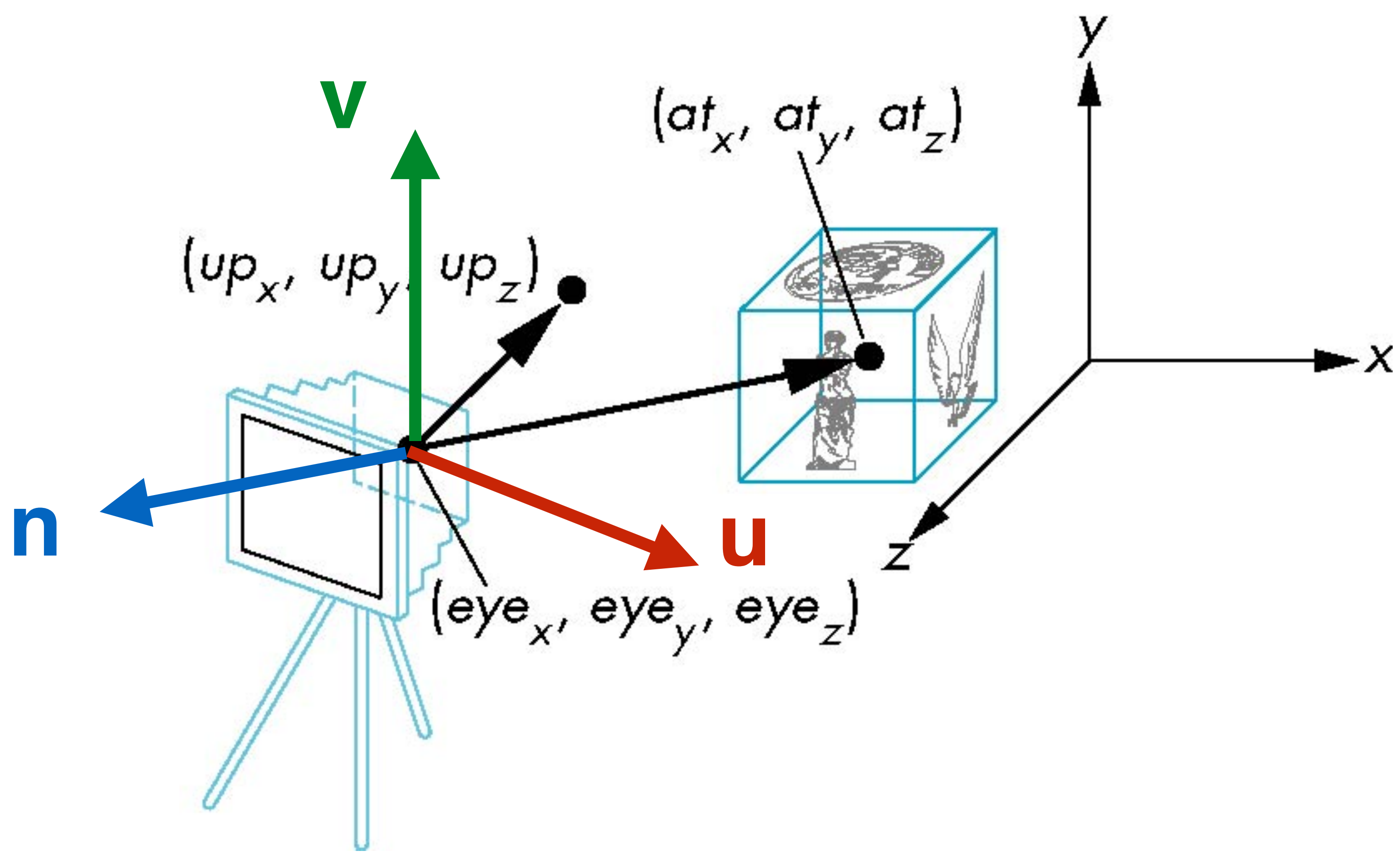
  $T(-eye_x, -eye_y, -eye_z)$

- 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: $R\ T(-eye_x, -eye_y, -eye_z)$

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

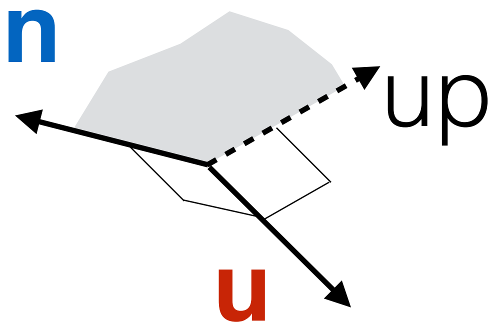
$$\mathbf{n} = (eye - at) \,/\, \|eye - at\|$$

Determinação de u, v e n

- O vetor up poderá, por mero acaso, ser logo perpendicular a **n** e, nesse caso, **v** estaria logo 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**.

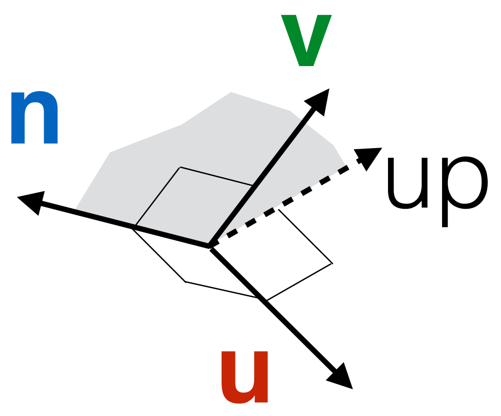
$$\mathbf{u} = (up \times \mathbf{n}) / ||up \times \mathbf{n}||$$



Determinação de u, v e n

- Finalmente, o vetor **v** pode agora ser facilmente determinado por forma a ser perpendicular quer a **u**, quer a **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:
  - **u** com x; **v** com y e **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 **u**,**v**,**n** (com a origem em comum). Assim:
  - $R\mathbf{u} = (1,0,0,0)^T$; $R\mathbf{v} = (0,1,0,0)^T$; $R\mathbf{n} = (0,0,1,0)^T$

Dedução de R

- R**u** = $(1,0,0,0)^T$; R**v** = $(0,1,0,0)^T$; R**n** = $(0,0,1,0)^T$

Sendo R uma matriz de rotação, com a forma ao lado, resulta que:

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

$[r_{11},r_{12},r_{13},0]$**u** = 1 $[r_{11},r_{12},r_{13},0]$**v** = 0 $[r_{11},r_{12},r_{13},0]$**n** = 0

$[r_{21},r_{22},r_{23},0]$**u** = 0 $[r_{21},r_{22},r_{23},0]$**v** = 1 $[r_{21},r_{22},r_{23},0]$**n** = 0

$[r_{31},r_{32},r_{33},0]$**u** = 0 $[r_{31},r_{32},r_{33},0]$**v** = 0 $[r_{31},r_{32},r_{33},0]$**n** = 1

Dedução de R

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

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

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

$\downarrow$ $\quad$ $\downarrow$ $\quad$ $\downarrow$

$(r_{11},r_{12},r_{13}) \parallel (u_x,u_y,u_z)$ $\quad$ $(r_{11},r_{12},r_{13}) \perp (v_x,v_y,v_z)$ $\quad$ $(r_{11},r_{12},r_{13}) \perp (n_x,n_y,n_z)$

$(r_{21},r_{22},r_{23}) \perp (u_x,u_y,u_z)$ $\quad$ $(r_{21},r_{22},r_{23}) \parallel (v_x,v_y,v_z)$ $\quad$ $(r_{21},r_{22},r_{23}) \perp (n_x,n_y,n_z)$

$(r_{31},r_{32},r_{33}) \perp (u_x,u_y,u_z)$ $\quad$ $(r_{31},r_{32},r_{33}) \perp (v_x,v_y,v_z)$ $\quad$ $(r_{31},r_{32},r_{33}) \parallel (n_x,n_y,n_z)$

Conclusão

- Estando determinada a matriz R:

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- A matriz final construída pela função lookAt() será:

R T(-eye$_x$, -eye$_y$, -eye$_z$):

Conclusão

R T(-eye$_x$, -eye$_y$, -eye$_z$):

$$\begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -eye_x \\ 0 & 1 & 0 & -eye_y \\ 0 & 0 & 1 & -eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Desenvolvendo:

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

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):

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