

Transformări (III).

Reperul de vizualizare (poziția camerei)

Mihai-Sorin Stupariu

Sem. I, 2022 - 2023

Coordonate de modelare. Coordonate de vizualizare

Reperul de vizualizare

Transformarea de vizualizare

Coordonate de modelare și coordonate de vizualizare

► Coordonatele de modelare

Coordonate de modelare și coordonate de vizualizare

► Coordonatele de modelare

- originea O

Coordonate de modelare și coordonate de vizualizare

► Coordonatele de modelare

- originea O
- axele de coordonate Ox, Oy, Oz cu versorii e_1, e_2, e_3

Coordonate de modelare și coordonate de vizualizare

► Coordonatele de modelare

- originea O
- axele de coordonate Ox , Oy , Oz cu versorii e_1 , e_2 , e_3
- implicit, obiectele/primitive (vârfurile) sunt indicate în raport cu acest sistem de coordonate

Coordonate de modelare și coordonate de vizualizare

► Coordonatele de modelare

- originea O
- axele de coordonate Ox, Oy, Oz cu versorii e_1, e_2, e_3
- implicit, obiectele/primitive (vârfurile) sunt indicate în raport cu acest sistem de coordonate

- Apelarea funcției `glm::lookAt()`; are ca efect (implicit) generarea unui nou reper / sistem de coordonate, numite **reper de vizualizare / coordonate de vizualizare**

Coordonate de modelare și coordonate de vizualizare

► Coordonatele de modelare

- originea O
- axele de coordonate Ox, Oy, Oz cu versorii e_1, e_2, e_3
- implicit, obiectele/primitive (vârfurile) sunt indicate în raport cu acest sistem de coordonate

► Apelarea funcției `glm::lookAt()`; are ca efect (implicit) generarea unui nou reper / sistem de coordonate, numite **reper de vizualizare / coordonate de vizualizare**

- originea: P_0 (poziția observatorului)

Coordonate de modelare și coordonate de vizualizare

► Coordonatele de modelare

- originea O
- axele de coordonate Ox, Oy, Oz cu versorii e_1, e_2, e_3
- implicit, obiectele/primitive (vârfurile) sunt indicate în raport cu acest sistem de coordonate

► Apelarea funcției `glm::lookAt()`; are ca efect (implicit) generarea unui nou reper / sistem de coordonate, numite **reper de vizualizare / coordonate de vizualizare**

- originea: P_0 (poziția observatorului)
- axele: date de versorii u, v, n (construiți în continuare)

Funcția `glm::lookAt()`

- Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)
 - Orientarea

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)
 - Orientarea
- ▶ Funcția `glm::lookAt`
`glm::lookAt (x0, y0, z0, xref, yref, zref, Vx, Vy, Vz);`

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)
 - Orientarea
- ▶ Funcția `glm::lookAt`

`glm::lookAt (x0, y0, z0, xref, yref, zref, Vx, Vy, Vz);`

 - (x_0, y_0, z_0) : coordonatele observatorului P_0 în reperul de modelare;

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)
 - Orientarea
- ▶ Funcția `glm::lookAt`

`glm::lookAt (x0, y0, z0, xref, yref, zref, Vx, Vy, Vz);`

 - (x_0, y_0, z_0) : coordonatele observatorului P_0 în reperul de modelare;
 - $(x_{ref}, y_{ref}, z_{ref})$: coordonatele unui punct de referință P_{ref} spre care se uită observatorul;

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)
 - Orientarea
- ▶ Funcția `glm::lookAt`

`glm::lookAt (x0, y0, z0, xref, yref, zref, Vx, Vy, Vz);`

 - (x_0, y_0, z_0) : coordonatele observatorului P_0 în reperul de modelare;
 - $(x_{ref}, y_{ref}, z_{ref})$: coordonatele unui punct de referință P_{ref} spre care se uită observatorul;
 - (V_x, V_y, V_z) : vector care indică verticala din planul de vizualizare

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`;: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)
 - Orientarea
- ▶ Funcția `glm::lookAt`

`glm::lookAt ($x_0, y_0, z_0, x_{ref}, y_{ref}, z_{ref}, V_x, V_y, V_z$);`
 - (x_0, y_0, z_0) : coordonatele observatorului P_0 în reperul de modelare;
 - $(x_{ref}, y_{ref}, z_{ref})$: coordonatele unui punct de referință P_{ref} spre care se uită observatorul;
 - (V_x, V_y, V_z) : vector care indică verticala din planul de vizualizare
- ▶ Implicit: $P_0 = (0, 0, 0)$, $P_{ref} = (0, 0 - 1)$, $V = (0, 1, 0)$

Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`; care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)
 - Orientarea
- ▶ Funcția `glm::lookAt`

`glm::lookAt (x0, y0, z0, xref, yref, zref, Vx, Vy, Vz);`

 - (x_0, y_0, z_0) : coordonatele observatorului P_0 în reperul de modelare;
 - $(x_{ref}, y_{ref}, z_{ref})$: coordonatele unui punct de referință P_{ref} spre care se uită observatorul;
 - (V_x, V_y, V_z) : vector care indică verticala din planul de vizualizare
- ▶ Implicit: $P_0 = (0, 0, 0)$, $P_{ref} = (0, 0 - 1)$, $V = (0, 1, 0)$
- ▶ În continuare: construirea reperului de vizualizare pornind de la argumentele funcției `glm::lookAt()`;

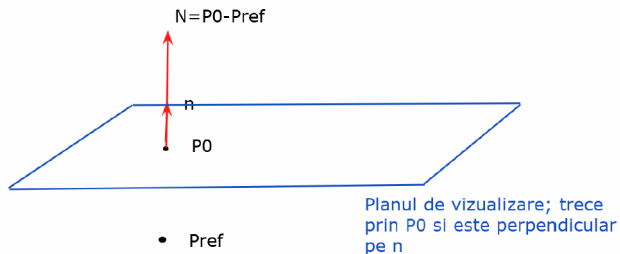
Funcția `glm::lookAt()`

- ▶ Pentru a înțelege funcția `glm::lookAt()`; care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
 - Poziția (coordonatele) observatorului
 - Direcția / Punctul de referință (spre care este îndreptată privirea sau dispozitivul)
 - Orientarea
- ▶ Funcția `glm::lookAt`

`glm::lookAt (x0, y0, z0, xref, yref, zref, Vx, Vy, Vz);`

 - (x_0, y_0, z_0) : coordonatele observatorului P_0 în reperul de modelare;
 - $(x_{ref}, y_{ref}, z_{ref})$: coordonatele unui punct de referință P_{ref} spre care se uită observatorul;
 - (V_x, V_y, V_z) : vector care indică verticala din planul de vizualizare
- ▶ Implicit: $P_0 = (0, 0, 0)$, $P_{ref} = (0, 0 - 1)$, $V = (0, 1, 0)$
- ▶ În continuare: construirea reperului de vizualizare pornind de la argumentele funcției `glm::lookAt()`;
- ▶ Originea reperului: $P_0 = (x_0, y_0, z_0)$; axele date de u, v, n

Reperul de vizualizare - vectorul n



de ce $P_0 - P_{ref}$ și nu $P_{ref} - P_0$?

de ce se împarte la $\|N\|$?

$$N = P_{ref} \xrightarrow{P_0} P_0 = P_0 - P_{ref}; \quad n = \frac{N}{\|N\|}$$

Reperul de vizualizare - vectorii v , u

- ▶ În planul de vizualizare sunt definiți doi vectori u și v care sunt vectorii primelor două axe ale reperului de vizualizare (“orizontala” și “verticala” din planul de vizualizare).

Reperul de vizualizare - vectorii v, u

- ▶ În planul de vizualizare sunt definiți doi vectori u și v care sunt vectorii primelor două axe ale reperului de vizualizare (“orizontală” și “verticală” din planul de vizualizare).
- ▶ **primul versor** u - direcționează orizontală din planul de vizualizare: este perpendicular pe vectorul n (ca să fie inclus în planul de vizualizare) și este perpendicular pe vectorul V indicat în `gluLookAt`

$$u = \frac{V \times n}{\|V\|}$$

Reperul de vizualizare - vectorii v, u

- ▶ În planul de vizualizare sunt definiți doi vectori u și v care sunt vectorii primelor două axe ale reperului de vizualizare (“orizontala” și “verticala” din planul de vizualizare).
- ▶ **primul versor** u - direcționează orizontala din planul de vizualizare: este perpendicular pe vectorul n (ca să fie inclus în planul de vizualizare) și este perpendicular pe vectorul V indicat în `gluLookAt`

$$u = \frac{V \times n}{\|V\|}$$

- ▶ **al doilea versor** v - verticala “reală” din planul de vizualizare

$$v = n \times u$$

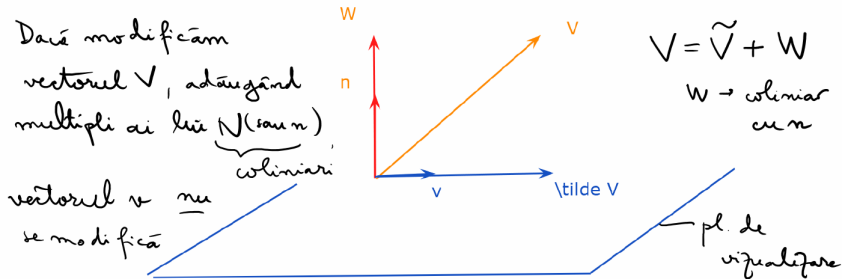
Legătura dintre vectorii V și v

Comentariu/Întrebare: ce legătură există între vectorul V , indicat ca “verticală” în funcția `glm::lookAt ()`; și vectorul v , calculat ca fiind al doilea versor al reperului de vizualizare?

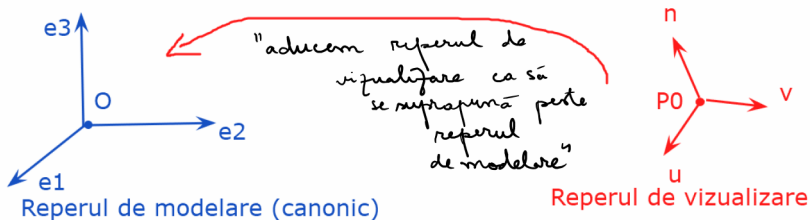
Legătura dintre vectorii V și v

Comentariu/Întrebare: ce legătură există între vectorul V , indicat ca “verticală” în funcția `glm::lookAt ()`; și vectorul v , calculat ca fiind al doilea versor al reperului de vizualizare?

R: Vectorul V se descompune ca suma dintre un vector \tilde{V} (=proiecția lui V pe planul de vizualizare) și un vector W , perpendicular pe planul de vizualizare (coliniar cu n). Are loc relația $v = \frac{\tilde{V}}{\|\tilde{V}\|}$.



Schimbarea reperului ca transformare



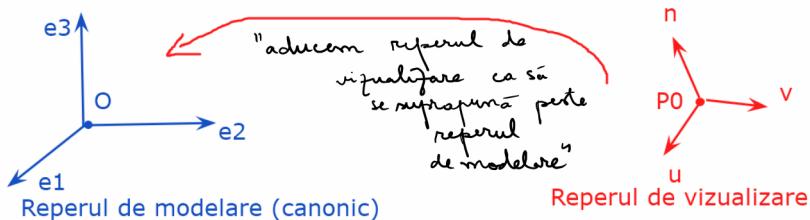
Descrierea transformărilor:

- translație a.i. P_0 să devină originea, adică aplicăm

$T(-x_0, -y_0, -z_0)$ vectorii sunt 1×2 , cu normă 1

- rotație 3D a.i. reperul ortonormat (u, v, n) să se suprapună cu reperul ON (e_1, e_2, e_3)

Schimbarea reperului ca transformare



Descrierea transformărilor:

- translație a.î. P_0 să devină originea, adică aplicăm

$T(-x_0, -y_0, -z_0)$ vectorii sunt 1×2 , cu normă 1

- rotație 3D a.î. reperul ortonormat (u, v, n) să se suprapună cu reperul ON (e_1, e_2, e_3)

Care este matricea asociată?

Matricea asociată schimbării de reper

Matricele asociate celor două transformări:

$$\textcircled{*} \Pi(-x_0, -y_0, -z_0) \rightarrow M_{\Pi(-x_0, -y_0, -z_0)} = \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\textcircled{*}$ rotatia? Pentru început: matricea 3×3 care transformă (e_1, e_2, e_3) în reperul ON (u, v, n) .

$$A = \begin{pmatrix} u_x & v_x & n_x \\ u_y & v_y & n_y \\ u_z & v_z & n_z \end{pmatrix}$$

colonele acestei matrice sunt componentele lui u, v, n în reperul canonic

Matricea asociată schimbării de reper

Matricea A^{-1} este cea care transformă reperul (u, v, n) în reperul canonic.

Obs. $A^t \cdot A = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix} \cdot \begin{pmatrix} u_x & v_x & n_x \\ u_y & v_y & n_y \\ u_z & v_z & n_z \end{pmatrix} =$

$$= \underline{\underline{I}}_3 \text{ (deoarece } (u, v, n) \text{ este reper ON)}$$

Deci: întrucât (u, v, n) este reper ON, matricea A verifică relația $A^t \cdot A = \underline{\underline{I}}_3$ (A s.n. ortogonală) și matricea inversă este $A^{-1} = A^t$.

Matricea asociată schimbării de reper

În final, matricea 4×4 asociată este:

$$M = \underset{(4 \times 4)}{A^t} \cdot M_{\Pi} = \begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow$$

$$\Rightarrow M = \begin{pmatrix} u_x & u_y & u_z & -\langle u, P_0 \rangle \\ v_x & v_y & v_z & -\langle v, P_0 \rangle \\ n_x & n_y & n_z & -\langle n, P_0 \rangle \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

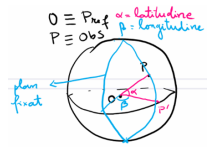
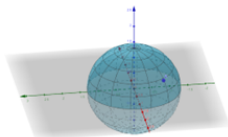
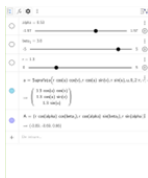
Survolarea unui obiect - codul 07_02_survolare_cub.cpp

- La ce revine a survola un obiect?

Survolarea unui obiect - codul 07_02_survolare_cub.cpp

- La ce revine a survola un obiect?
- Reprezentarea sferei de centru C și rază r

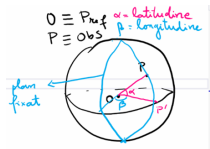
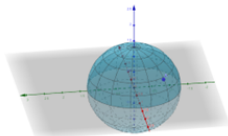
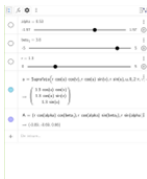
$$\begin{cases} x = C_x + r \cos(\alpha) \cos(\beta) \\ y = C_y + r \cos(\alpha) \sin(\beta) \\ z = C_z + r \sin(\alpha) \end{cases} \quad \alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \beta \in [0, 2\pi]$$



Survolarea unui obiect - codul 07_02_survolare_cub.cpp

- La ce revine a survola un obiect?
- Reprezentarea sferei de centru C și rază r

$$\begin{cases} x = C_x + r \cos(\alpha) \cos(\beta) \\ y = C_y + r \cos(\alpha) \sin(\beta) \\ z = C_z + r \sin(\alpha) \end{cases} \quad \alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \beta \in [0, 2\pi]$$



- Pentru a implementa survolarea, observatorul Obs se deplasează pe o sferă cu centrul în punctul de referință Ref și cu raza dist. În cod dist, alpha, beta sunt variabile.

```
//pozitia observatorului - se deplaseaza pe sfera
Obsx = Refx + dist * cos(alpha) * cos(beta);
Obsy = Refy + dist * cos(alpha) * sin(beta);
Obsz = Refz + dist * sin(alpha);
```