

Transformări (III).

Reperul de vizualizare (poziția camerei)

Mihai-Sorin Stupariu

Sem. I, 2021 - 2022

Generalități

Remember - utilizarea celor 4 coordonate

Support teoretic: ⚠️ OpenGL ia în considerare coordonate omogene

(i) sunt utilizate 4 coordonate (4 componente),
semnificativitatea celei de-a 4^a coordonate:

- Vârfuri

vârful (punctul) 3D

$$(x_1, x_2, x_3) \mapsto [x_1 : x_2 : x_3 : 1] \left(= [x_1 : x_2 : x_3 : 1] \right) \quad \forall \alpha \neq 0$$

de exemplu $(1, 2, 5) \mapsto [1 : 2 : 5 : 1] = [3 : 6 : 15 : 3]$ etc.

- Directii

dreaptă / direcție descrisă de (v_1, v_2, v_3)

$$(v_1, v_2, v_3) \mapsto [v_1 : v_2 : v_3 : 0] \left(= [v_1 : v_2 : v_3 : 0] \right) \quad \forall \alpha \neq 0$$

de exemplu: $(1, 1, 0) \mapsto [1 : 1 : 0 : 0] = [2 : 2 : 0 : 0]$ etc.

Remember - utilizarea celor 4 coordonate

(ii) Unei aplicații afine f (translație, scalare, rotație, etc.) îi este asociată o matrice 4×4 , notată M_f .

De exemplu: translația de vector $(1, 5, 4)$.

$$M_f = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

↓ pe ultima linie: $(0, 0, 0, 1)$.
 "este păstrată poziția dreptei de la infinit"

Remember - aplicarea unei transformări

Observații :

- (i) cu convențiile introduse : dacă aplicăm unui vârf de coordonate omogene

$$\xi \equiv \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_0 \end{pmatrix} \left(= \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \right) \triangleq \text{vector coloană}$$

transformarea f dată de matricea M_f (4×4),
rezultatul va fi vectorul coloană matr.

$M_f \cdot \xi$ → apare efectiv în shader | mat 4 f ;
gl_Position = f * in_Position;
 (cu mulțirea la stânga cu M_f)

Remember - compunerea transformărilor

(ii) Fie f_1, f_2 transformări cu matrice M_{f_1}, M_{f_2} .

Ordinea aplicării: f_1 , apoi f_2 . (altfel spus, folosind compunerea, aplicăm $f_2 \circ f_1$). Matriceal avem:

$$\xi \mapsto M_{f_2}(M_{f_1} \cdot \xi) = (M_{f_2} \cdot M_{f_1}) \cdot \xi, \text{ deci}$$

$$\boxed{M_{f_2 \circ f_1} = M_{f_2} \cdot M_{f_1}}$$

compunerea transformărilor
(aplicărilor)

↕
înmulțirea matricelor

(ne șader sau în progr. princ.)

Afirmatia se generalizează pentru n transformări.

Remember - compunerea transformărilor

⚠️ Atenție la ordinea transformărilor.

Exemple

$m_1 = \text{glm::translate}(\dots);$

$m_2 = \text{glm::scale}(\dots);$

$m_3 = \text{glm::rotate}(\dots);$

$m = (m_1 * m_2) * m_3;$

în shader: $\text{gl_Position} = m * \text{in_Position};$

ordinea aplicării: rotația, scalarea, translația.

Tipuri de transformări

- le-am
discutat
- transformări de modelare
(aplicate primitivelor din scenă,
de exemplu translații, rotații, etc.)
 - transformări de vizualizare
 - transformări de proiecție

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 (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: $P_0 = (0, 0, 0)$, $P_{ref} = (0, 0 - 1)$, $V = (0, 1, 0)$

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()`;

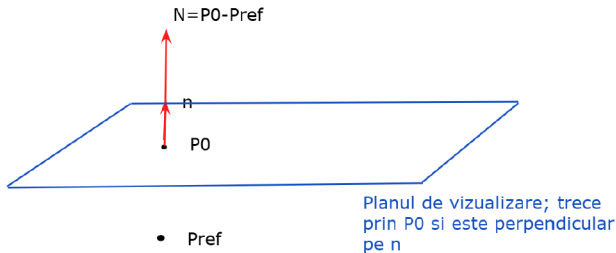
Funcția `glm::lookAt()`

- ▶ 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 $P_0 - P_{ref}$ în nu $P_{ref} - P_0$?

de ce se împarte la $\|N\|$?

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

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 (“orizontală” și “verticală” 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 (“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`

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

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

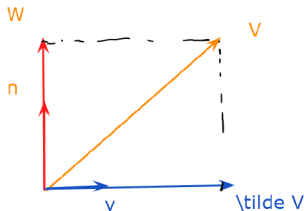
Comentariu/Întrebare: ce legătură există între vectorul V , indicat ca “verticală” în funcția `glm::lookAt ()`; și vectorul v , calculat ca fiind al doilea versor al reperului de vizualizare?

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 $v = \frac{\tilde{V}}{\|\tilde{V}\|}$.

Dacă modificăm
vectorul V ,
adăugând multipli
ai lui N (sau n),
vectorul v nu coliniar
se modifică

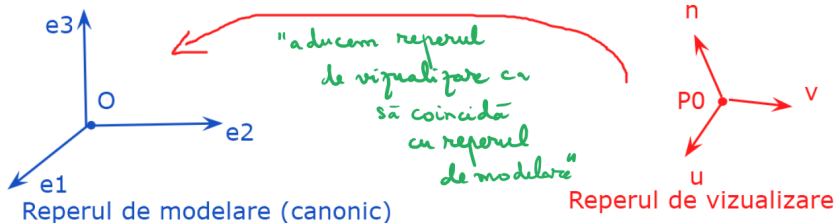


$$V = \tilde{V} + W$$

$v \rightarrow \text{coliniar cu } \tilde{V}$

Schimbarea reperului ca transformare

Schimbarea de reper \leftrightarrow Efectuarea unei transformări



Descrierea transformărilor:

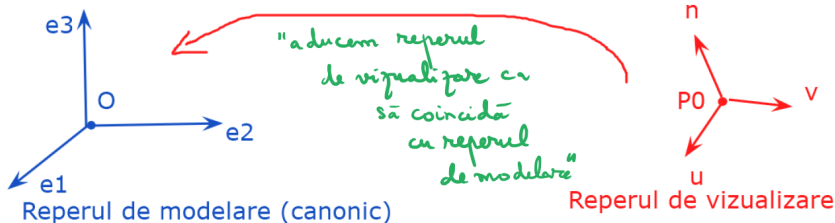
- translație a.î. P_0 să devină originea, adică aplicăm

$T(-x_0, -y_0, -z_0)$ vectorii sunt \perp 2×2 , au normă 1

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

Schimbarea reperului ca transformare

Schimbarea de reper \leftrightarrow Efectuarea unei transformări



Descrierea transformărilor:

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

$T(-x_0, -y_0, -z_0)$ vectorii sunt \perp 2×2 , au normă 1

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

Care este matricea asociată?

Matricea asociată schimbării de reper

Matricele asociate celor 2 transformări:

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

(*) rotatie? Pentru început: matricea 3×3
care transformă (e_1, e_2, e_3) în reperul (u, v, n)

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

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

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

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

Matricea asociată schimbării de reper

În final: matricea 4×4 căutată este.

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

(4x4)

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