Transformări (III).
Reperul de vizualizare (poziția camerei)

Mihai-Sorin Stupariu

Sem. I, 2024 - 2025

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/primitivele (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/primitivele (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/primitivele (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/primitivele (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 $\mathbf{u}, \mathbf{v}, \mathbf{n}$ (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` $(x_0, y_0, z_0, x_{ref}, y_{ref}, z_{ref}, V_x, V_y, V_z)$;

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;

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;

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

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( )

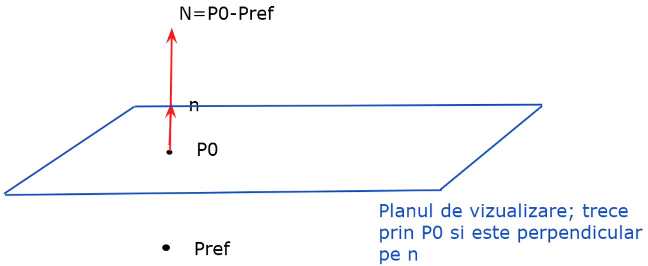
- Pentru a înțelege funcția `glm::lookAt( );`: care sunt elementele geometrice relevante atunci când vorbim despre observarea unei scene 3D? (de exemplu vederea umană sau folosirea unui aparat fotografic / telefon mobil).
  - Poziția (coordonatele) observatorului
  - Direcția / Punctul de referință (spre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

  `glm::lookAt` $(x_0, y_0, z_0, x_{ref}, y_{ref}, z_{ref}, V_x, V_y, V_z)$;
  - (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: $P_0 = (0,0,0), P_{ref} = (0,0-1), V = (0,1,0)$
- În continuare: construirea reperului de vizualizare pornind de la argumentele funcției `glm::lookAt( );`

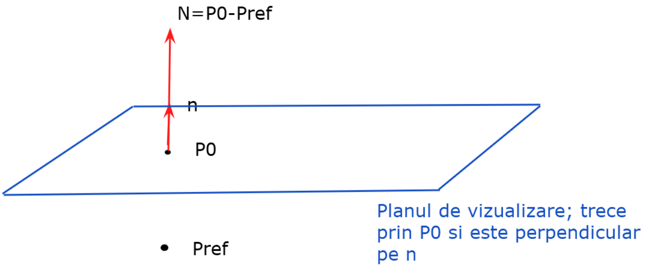
Funcția `glm::lookAt( )`

- 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` $(x_0, y_0, z_0, x_{ref}, y_{ref}, z_{ref}, V_x, V_y, V_z)$;
  - (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: $P_0 = (0, 0, 0), P_{ref} = (0, 0 - 1), V = (0, 1, 0)$
- 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 $\mathbf{u}, \mathbf{v}, \mathbf{n}$

Reperul de vizualizare - vectorul **n**

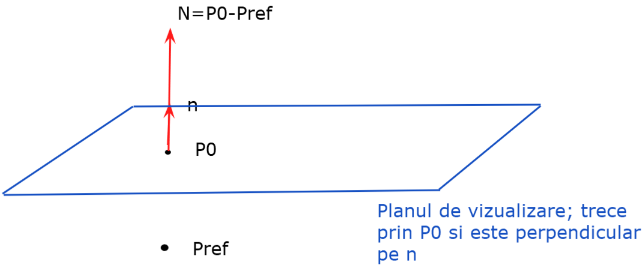


Reperul de vizualizare - vectorul **n**



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

Reperul de vizualizare - vectorul **n**



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

Comentarii: de ce $P_0 - P_{ref}$ și nu $P_{ref} - P_0$? De ce se împarte la $\|N\|$?

Reperul de vizualizare - vectorii $\mathbf{v}, \mathbf{u}$

- 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 $\mathbf{v}, \mathbf{u}$

- 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`

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

Reperul de vizualizare - vectorii $\mathbf{v}, \mathbf{u}$

- 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`

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

- **al doilea versor v** - verticala "reală" din planul de vizualizare

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

Legătura dintre vectorii V și $\mathbf{v}$

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

Legătura dintre vectorii V și $\mathbf{v}$

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

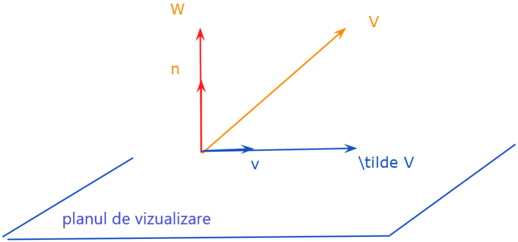
R: Vectorul V se descompune ca suma dintre un vector $\tilde{V}$ (=proiecția lui V pe planul de vizualizare) și un vector W, perpendicular pe planul de vizualizare (coliniar cu $\mathbf{n}$), altfel spus $V = \tilde{V} + W$. Are loc relația $\mathbf{v} = \frac{\tilde{V}}{\|\tilde{V}\|}$.

Legătura dintre vectorii V și $\mathbf{v}$

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

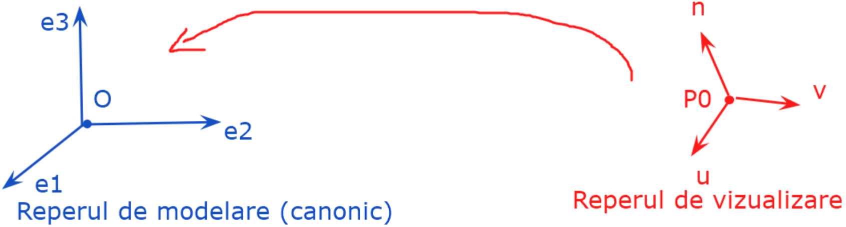
R: Vectorul V se descompune ca suma dintre un vector $\tilde{V}$ (=proiecția lui V pe planul de vizualizare) și un vector W, perpendicular pe planul de vizualizare (coliniar cu $\mathbf{n}$), altfel spus $V = \tilde{V} + W$. Are loc relația $\mathbf{v} = \frac{\tilde{V}}{\|\tilde{V}\|}$.

Obs: Dacă modificăm vectorul V, adăugând multipli ai lui N (sau $\mathbf{n}$, deoarece N și $\mathbf{n}$ sunt coliniari), vectorul $\mathbf{v}$ nu se modifică.



Schimbarea reperului ca transformare

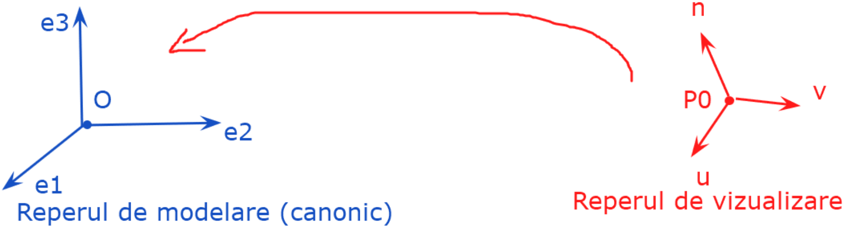
Schimbarea de reper $\leftrightarrow$ Efectuarea unei transformări



"Aducem reperul de vizualizare ca să se suprapună peste reperul de modelare".

Schimbarea reperului ca transformare

Schimbarea de reper $\leftrightarrow$ Efectuarea unei transformări

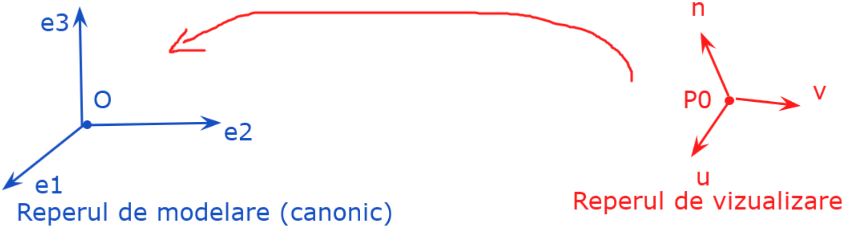


"Aducem reperul de vizualizare ca să se suprapună peste reperul de modelare".

Descrierea transformărilor:

Schimbarea reperului ca transformare

Schimbarea de reper $\leftrightarrow$ Efectuarea unei transformări



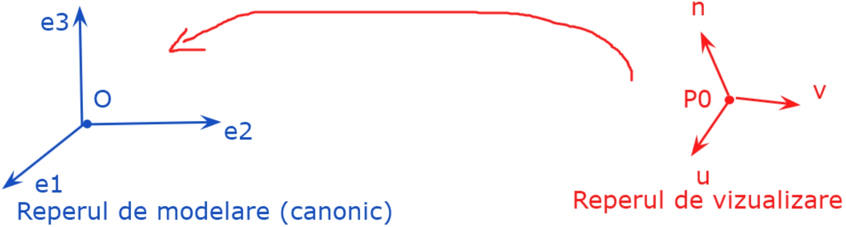
"Aducem reperul de vizualizare ca să se suprapună peste reperul de modelare".

Descrierea transformărilor:

- translatăm astfel încât P_0 să devină originea, adică aplicăm $\mathbf{T}_{(-x_0,-y_0,-z_0)}$

Schimbarea reperului ca transformare

Schimbarea de reper $\leftrightarrow$ Efectuarea unei transformări



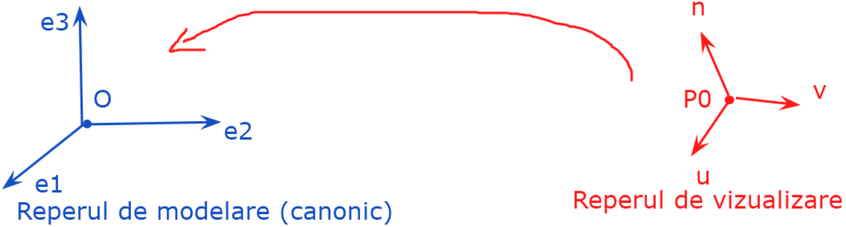
"Aducem reperul de vizualizare ca să se suprapună peste reperul de modelare".

Descrierea transformărilor:

- translatăm astfel încât P_0 să devină originea, adică aplicăm $\mathbf{T}_{(-x_0,-y_0,-z_0)}$
- aplicăm o rotație 3D $\mathbf{R}$ astfel încât reperul ortonormat $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ să se suprapună cu reperul ortonormat (e_1, e_2, e_3) (reperul canonic) - un reper este ortonormat dacă vectorii sunt perpendiculari 2 câte 2 și de normă 1.

Schimbarea reperului ca transformare

Schimbarea de reper $\leftrightarrow$ Efectuarea unei transformări



"Aducem reperul de vizualizare ca să se suprapună peste reperul de modelare".

Descrierea transformărilor:

- translatăm astfel încât P_0 să devină originea, adică aplicăm $\mathbf{T}_{(-x_0,-y_0,-z_0)}$
- aplicăm o rotație 3D $\mathbf{R}$ astfel încât reperul ortonormat $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ să se suprapună cu reperul ortonormat (e_1, e_2, e_3) (reperul canonic) - un reper este ortonormat dacă vectorii sunt perpendiculari 2 câte 2 și de normă 1.

Care este matricea asociată?

Matricea asociată schimbării de reper

Matricele asociate celor două transformări:

- pentru translația $\mathbf{T}_{(-x_0,-y_0,-z_0)}$

$$M_{\mathbf{T}_{(-x_0,-y_0,-z_0)}} = \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matricea asociată schimbării de reper

- pentru rotația **R**:

Matricea asociată schimbării de reper

- pentru rotația $\mathbf{R}$:
  - matricea 3×3 care transformă reperul (e_1, e_2, e_3) în reperul ortonormat $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ este

$$A = \begin{pmatrix} \mathbf{u}_x & \mathbf{v}_x & \mathbf{n}_x \\ \mathbf{u}_y & \mathbf{v}_y & \mathbf{n}_y \\ \mathbf{u}_z & \mathbf{v}_z & \mathbf{n}_z \end{pmatrix}$$

(coloanele acestei matrice sunt componentele lui $\mathbf{u}, \mathbf{v}, \mathbf{n}$ în reperul canonic)

Matricea asociată schimbării de reper

- pentru rotația $\mathbf{R}$:
    - matricea 3×3 care transformă reperul (e_1, e_2, e_3) în reperul ortonormat $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ este

$$A = \begin{pmatrix} \mathbf{u}_x & \mathbf{v}_x & \mathbf{n}_x \\ \mathbf{u}_y & \mathbf{v}_y & \mathbf{n}_y \\ \mathbf{u}_z & \mathbf{v}_z & \mathbf{n}_z \end{pmatrix}$$

      (coloanele acestei matrice sunt componentele lui $\mathbf{u}, \mathbf{v}, \mathbf{n}$ în reperul canonic)
    - matricea care transformă $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ în (e_1, e_2, e_3) este A^{-1}

Matricea asociată schimbării de reper

- pentru rotația $\mathbf{R}$:
  - matricea 3×3 care transformă reperul (e_1, e_2, e_3) în reperul ortonormat $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ este

$$A = \begin{pmatrix} \mathbf{u}_x & \mathbf{v}_x & \mathbf{n}_x \\ \mathbf{u}_y & \mathbf{v}_y & \mathbf{n}_y \\ \mathbf{u}_z & \mathbf{v}_z & \mathbf{n}_z \end{pmatrix}$$

    (coloanele acestei matrice sunt componentele lui $\mathbf{u}, \mathbf{v}, \mathbf{n}$ în reperul canonic)
  - matricea care transformă $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ în (e_1, e_2, e_3) este A^{-1}
  - întrucât $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ este reper ortonormat, A este matrice **ortogonală**, adică matricea A verifică relația $A^t \cdot A = \mathbb{I}_3$ (verificați!), iar matricea inversă este $A^{-1} = A^t$
  - din A^t se construiește (în mod natural) matricea 4×4 asociată rotației, $M_{\mathbf{R}}$

Matricea asociată schimbării de reper

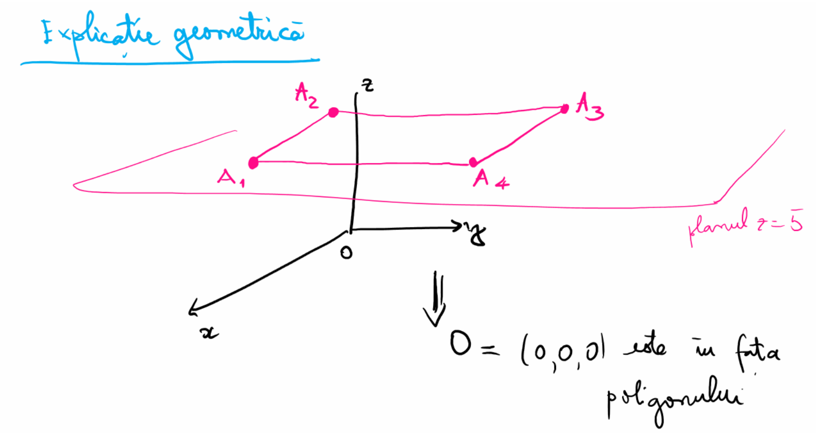
În final, matricea 4×4 asociată transformării de vizualizare este

$$M = M_{\mathbf{R}} \cdot M_{\mathbf{T}_{(-x_0,-y_0,-z_0)}} = \begin{pmatrix} \mathbf{u}_x & \mathbf{u}_y & \mathbf{u}_z & 0 \\ \mathbf{v}_x & \mathbf{v}_y & \mathbf{v}_z & 0 \\ \mathbf{n}_x & \mathbf{n}_y & \mathbf{n}_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} \mathbf{u}_x & \mathbf{u}_y & \mathbf{u}_z & -\langle \mathbf{u}, P_0 \rangle \\ \mathbf{v}_x & \mathbf{v}_y & \mathbf{v}_z & -\langle \mathbf{v}, P_0 \rangle \\ \mathbf{n}_x & \mathbf{n}_y & \mathbf{n}_z & -\langle \mathbf{n}, P_0 \rangle \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Exemplul 1. Cod sursă 06_01_poligoane3D.cpp

$A_1 = (5, -5, 5), A_2 = (-5, -5, 5), A_3 = (-5, 5, 5), A_4 = (5, 5, 5)$



Exemplul 1. Cod sursă 06_01_poligoane3D.cpp

$A_1 = (5, -5, 5), A_2 = (-5, -5, 5), A_3 = (-5, 5, 5), A_4 = (5, 5, 5)$

- scriem ecuația planului sub forma $\underbrace{Ax + By + Cz}_{= \pi(x,y,z)} + D = 0$

- folosim determinantul

$$\begin{array}{l} \\ A_1 \to \\ A_2 \to \\ A_3 \to \end{array} \begin{vmatrix} x & y & z & 1 \\ 5 & -5 & 5 & 1 \\ -5 & -5 & 5 & 1 \\ -5 & 5 & 5 & 1 \end{vmatrix} = \ldots = x \cdot 0 - y \cdot 0 + z \cdot (-100) - (-500)$$

(calculați!)

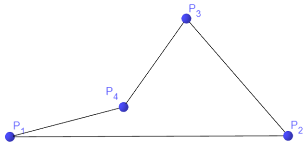
- avem $\pi(x,y,z) = -100z + 500$

- avem $\pi(0,0,0) = 500 > 0$ $\to$ punctul $(0,0,0)$ este în fața poligonului

Exemplul 2. Cod sursă `06_02_poligoane3D_exemplu2.cpp`

Fie punctele $P_1 = (6,2,0), P_2 = (-4,4,8), P_3 = (0,0,8)$ (toate trei situate în planul de ecuație $x + y + z = 8$).

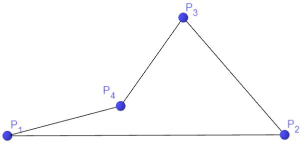
a) Să se aleagă P_4 astfel ca patrulaterul $P_1P_2P_3P_4$ să fie concav.



Exemplul 2. Cod sursă 06_02_poligoane3D_exemplu2.cpp

Fie punctele $P_1 = (6, 2, 0), P_2 = (-4, 4, 8), P_3 = (0, 0, 8)$ (toate trei situate în planul de ecuație $x + y + z = 8$).

a) Să se aleagă P_4 astfel ca patrulaterul $P_1P_2P_3P_4$ să fie concav.



Dacă alegem un punct P_4 în interiorul triunghiului $P_1P_2P_3$ (combinația convexă a punctelor P_1, P_2, P_3, cu coeficienți > 0 cu suma 1), atunci patrulaterul $P_1P_2P_3P_4$ este concav. De exemplu, punctul P_4 poate fi ales ca fiind dat de

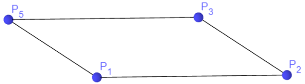
$$P_4 = \frac{1}{2}P_1 + \frac{1}{4}P_2 + \frac{1}{4}P_3$$

(P_4 este mijlocul segmentului $[P_1Q]$, unde Q este mijlocul lui $[P_2P_3]$). Explicit, avem $P_4 = (2, 2, 4)$.

Exemplul 2. Cod sursă06_02_poligoane3D_exemplu2.cpp

Fie punctele $P_1 = (6, 2, 0), P_2 = (-4, 4, 8), P_3 = (0, 0, 8)$ (toate trei situate în planul de ecuație $x + y + z = 8$).

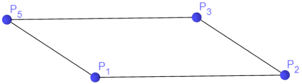
b) Să se aleagă P_5 astfel ca patrulaterul $P_1P_2P_3P_5$ să fie convex.



Exemplul 2. Cod sursă `06_02_poligoane3D_exemplu2.cpp`

Fie punctele $P_1 = (6, 2, 0), P_2 = (-4, 4, 8), P_3 = (0, 0, 8)$ (toate trei situate în planul de ecuație $x + y + z = 8$).

b) Să se aleagă P_5 astfel ca patrulaterul $P_1P_2P_3P_5$ să fie convex.



Alegem P_5 astfel încât $P_1P_2P_3P_5$ să fie un paralelogram. Folosindu-ne de faptul că diagonalele unui paralelogram se taie în părți egale, adică

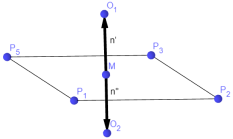
$$\frac{1}{2}P_1 + \frac{1}{2}P_3 = \frac{1}{2}P_2 + \frac{1}{2}P_5,$$

deducem $P_5 = P_1 + P_3 - P_2$, deci $P_5 = (10, -2, 0)$.

Exemplul 2. Cod sursă `06_02_poligoane3D_exemplu2.cpp`

Fie punctele $P_1 = (6, 2, 0), P_2 = (-4, 4, 8), P_3 = (0, 0, 8)$ (toate trei situate în planul de ecuație $x + y + z = 8$).

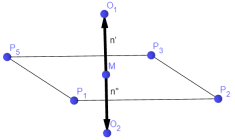
c) Să se aleagă puncte O_1 și O_2 astfel ca poligonul $P_1P_2P_3P_5$ să fie văzut din față, respectiv din spate.



Exemplul 2. Cod sursă `06_02_poligoane3D_exemplu2.cpp`

Fie punctele $P_1 = (6, 2, 0)$, $P_2 = (-4, 4, 8)$, $P_3 = (0, 0, 8)$ (toate trei situate în planul de ecuație $x + y + z = 8$).

c) Să se aleagă puncte O_1 și O_2 astfel ca poligonul $P_1P_2P_3P_5$ să fie văzut din față, respectiv din spate.



Mai întâi calculăm produsul vectorial $\overrightarrow{P_1P_2} \times \overrightarrow{P_2P_3}$, care este egal cu $(32, 32, 32)$. Așadar, vectorul normal la plan este $n = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$.

Alegem convenabil un punct M din plan, un vector n' coliniar și de același sens cu n și un vector n'' coliniar și de sens opus cu n. Concret: $M = (3, 1, 4)$ (mijlocul segmentului $[P_1P_3]$), $n' = (10, 10, 10)$ și $n'' = (-10, -10, -10)$.

Definim O_1 astfel ca $\overrightarrow{MO_1} = n'$, adică $n' = O_1 - M$, așadar $O_1 = (3, 1, 4) + (10, 10, 10) = (13, 11, 14)$.

Analog, definim O_2 ca $\overrightarrow{MO_2} = n''$, deci $O_2 = (3, 1, 4) - (10, 10, 10) = (-7, -9, -6)$.

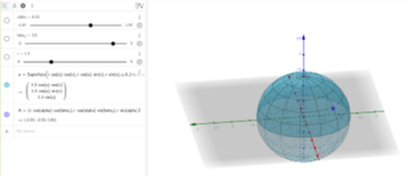
Survolarea unui obiect - codul `07_03_survolare_cub.cpp`

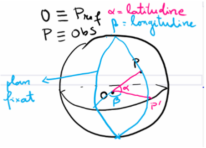
- La ce revine a survola un obiect?

Survolarea unui obiect - codul `07_03_survolare_cub.cpp`

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

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

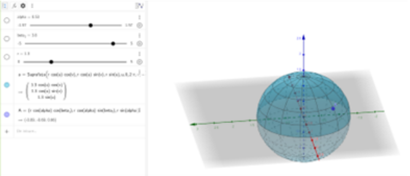


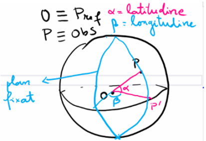


Survolarea unui obiect - codul `07_03_survolare_cub.cpp`

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

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





- 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);
```