Proiekzio-aldaketak. Kamara.

<a.soroa@si.ehu.es>

EHU

2006ko martxoak 21

Ikusmena eta proiekzioa

- Gure begiek 3Dko mundua 2Dra bihurtzen dute
- Horretarako, *proiekzio* bat egiten dugu
- Kamararen kudeaketak bi zati ditu
  - Bista-aldaketa: kamararen posizio zein norabidea
  - Proiekzio-aldaketa: 3Dtik 2Dra pasa
- Koordenatu homogeneoak erabiliko ditugu
- Kamara anima daiteke aldaketa hauekin jokatuz

Bista-aldaketa

- Kamararen posizioak eta norabideak erreferentzia-sistema bat osatzen dute.
- Behar da jakin kamararen jatorria $\mathbf{E}$, eskuinera doan bektorea $\mathbf{R}$, gora doan bektore unitarioa $\mathbf{U}$, eta atzera begiratzen duen bektore unitarioa $\mathbf{D}$.
- Hiru bektore horiek erreferentzia-sistema ortogonala osatu behar dute
  - Unitarioak
  - Elkarrekiko elkarzutak
- Izan dadin 3×3 dimentsioko $\mathbf{B}$ biraketa-matrizea, zeinaren zutabeak $\{\mathbf{R}, \mathbf{U}, \mathbf{D}\}$ bektoreak diren:

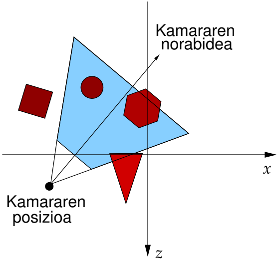
$$\mathbf{B} = \begin{pmatrix} R_x & U_x & D_x \\ R_y & U_y & D_y \\ R_z & U_z & D_z \end{pmatrix}$$

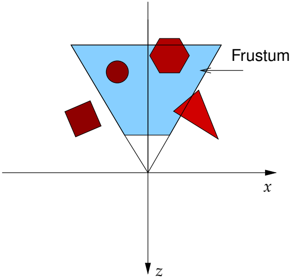
Bista-aldaketa

- Bista-aldaketak kamara jatorrian kokatzen du, eta bere $\{\mathbf{R}, \mathbf{U}, \mathbf{D}\}$ bektoreak munduko erreferentzia-sistemekin bat etorrarazten ditu (z ardatza behera begiratuz).
- $\mathbf{M}_b$ bista-aldaketa:

$$\mathbf{M}_b = \begin{pmatrix} \mathbf{B}^T & -\mathbf{B}^T\mathbf{E} \\ 0 & 1 \end{pmatrix}$$

- *Rigid body transformation*





Bista-aldaketa

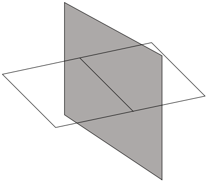
- Behin bista-aldaketa eragin ondoren, kamarak honako parametro hauek ditu normalean:

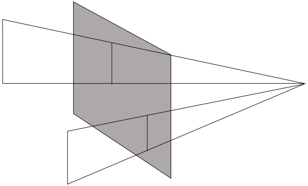
$$
\begin{aligned}
\mathbf{M}_b\mathbf{E} &= (0,0,0) \\
\mathbf{M}_b\mathbf{R} &= (1,0,0) \\
\mathbf{M}_b\mathbf{U} &= (0,1,0) \\
\mathbf{M}_b\mathbf{D} &= (0,0,-1)
\end{aligned}
$$

- Arazoa: nola zehaztu $\{\mathbf{R},\mathbf{U},\mathbf{D}\}$?

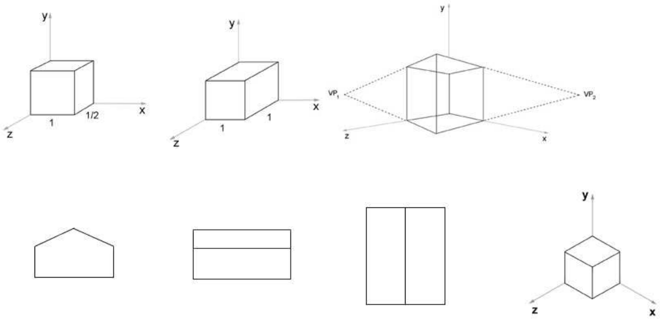
Proiekzio-aldaketak

- 3Dko eszena bat 2Dko pantaila batean irudikatu behar da.
  - Proiekzioa
- Proiekzio mota anitz dago:
  - Proiekzio paraleloak
    - Proiekzio-planoa
    - Proiekzio-norabidea (paraleloa da usuena)
  - Perspektiba proiekzioa
    - Proiekzio-planoa
    - Proiekzio-zentroa
    - Objektu urrunak txikiagoak ikusten dira





Proiekzio motak



- Cabinet, Cavalier, 2 proiekzio-puntuko perspektiba
  Frontea, alboa, altxaera, isometrikoa

http://www.mtsu.edu/ csjudy/planeview3D/tutorial.html

Proiekzio motak

- Guk bi proiekzio mota ikusiko ditugu
  - Proiekzio ortografikoa (paraleloa)
    - Arkitektura
    - CAD/CAM
  - Perspektiba proiekzioa
    - Gizakiok ikusten duguna
    - Gehien erabiltzen dena

Proiekzio ortografikoa

- Zuzen paraleloen proiekzioak paraleloak dira ere
- Adibidez, $z = 0$ planuan proiekzio ortografikoa lortzeko:

$$\mathbf{P}_O = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Arazoa: alderantzizkorik ez du
- (x, y, z) eta $(x, y, -z)$ puntuek proiekzio bera jasotzen dute.

Proiekzio ortografikoa

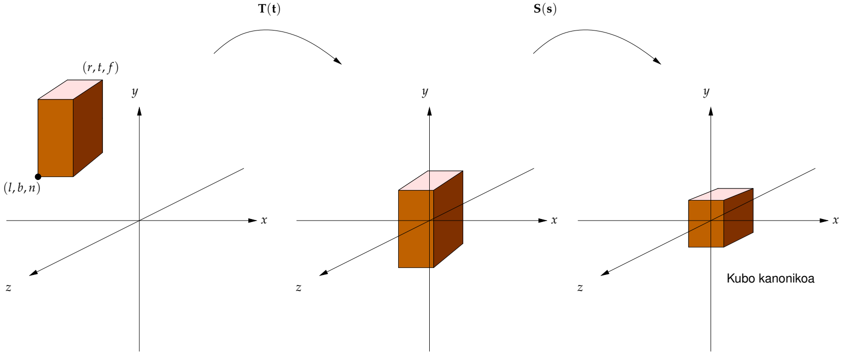
- Beste modu bat definitzeko:
- (l, r, b, t, n, f) 6-koteak munduko ardatzekiko lerrokaturik (*AABB*, “Axis Aligned Bounding Bok”) dagoen kutxa bat definitzen du.
- Proiekzioak kutxa hori kubo kanonikora bihurtzen du

$$\mathbf{P}_O = \mathbf{S}(\mathbf{s})\mathbf{T}(\mathbf{t}) = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & -\frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Proiekzio ortografikoa

- $\mathbf{P}_O = \mathbf{S}(\mathbf{s})\mathbf{T}(\mathbf{t})$
  - $\mathbf{t} = (-(r+l)/2, -(t+b)/2, -(f+n)/2$
  - $\mathbf{s} = (2/(r-l), 2/(t-b), 2/(f-n))$
- $\mathbf{P}_O^{-1} = \mathbf{T}(-\mathbf{t})\mathbf{S}((r-l)/2, (t-b)/2, (f-n)/2)$



Proiekzio ortografikoa

- Kontuan izan
  - $n > f$, z ardatza negatiboan ikusten ari garelako
- Kubo kanonikoa
  - Bi puntuz defintzen da: $(-1, -1, -1)$ eta $(1, 1, 1)$
  - *Bista-bolumen kanonikoa* definitzen du (“canonical view volume”)
  - Bolumen honen barneko koordenatuei *dispositiboaren koordenatu normalizatuak* (“normalized device coordinates”) esaten zaie
  - Mugaketa (“clipping”) oso azkarra da bolumen honetan (hardware)

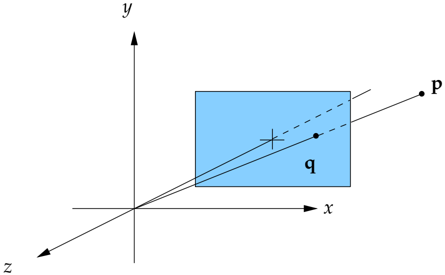
Proiekzio ortografikoa OpenGL-n

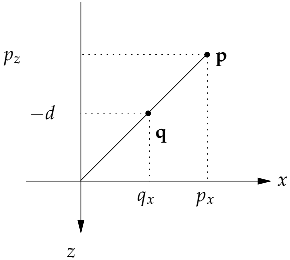
```
void glOrtho(   GLdouble left,
                GLdouble right,
                GLdouble bottom,
                GLdouble top,
                GLdouble near,
                GLdouble far )
```

Perspektiba proiekzioa

- Zuzen paraleloen proiekzioak ez dute zertan zuzenak izan behar
- Zuzen proiektatuek puntu batera hurbiltzen dira
- Urruneko objektuak txikiago ikusten dira
- Gehien erabilitako proiekzio mota

Perspektiba proiekzioa





- $z = -d, d > 0$ planura proiektatuko dugu. $\mathbf{q}$ puntua $\mathbf{p}$ puntuaren proiekzioa da
- Munduko erreferentzia-sistema
- Antzeko triangeluen legea:

$$\frac{q_x}{p_x} = \frac{-d}{p_z} \Rightarrow q_x = -d\frac{p_x}{p_z}$$

Perspektiba proiekzioa

- $q_x = -d\frac{p_x}{p_z}$, $q_y = -d\frac{p_y}{p_z}$, $q_z = -d$

$$\mathbf{P}_p = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{pmatrix}$$

$$\mathbf{q} = \mathbf{P}_p\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \\ -p_z/d \end{pmatrix} = \begin{pmatrix} -dp_x/p_z \\ -dp_y/p_z \\ -d \\ 1 \end{pmatrix}$$

- Arazoa: Berriro ere, $\mathbf{P}_p$-k ez du alderantzizkorik

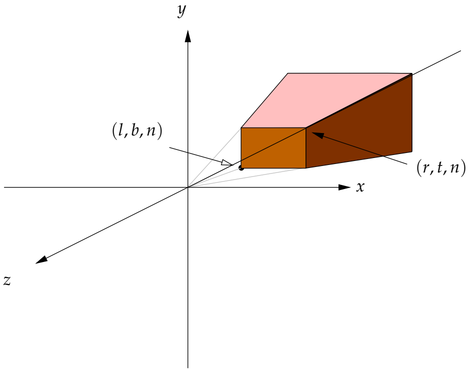
Perspektiba proiekzioa

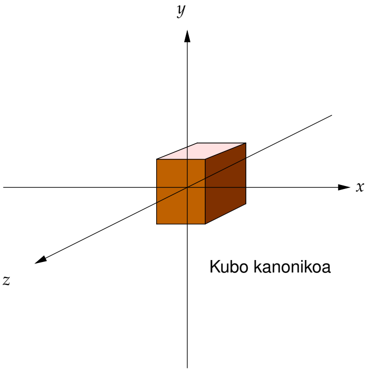
- Bista-bolumen kanonikora mapatuko dugu proiekzioa
- Perspekribaten bolumena piramide moztua da
- Piramidea definitzeko: (l, r, b, t, n, f)

$$\mathbf{P}_p = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Perspektiba proiekzioa

- Proiekzioaren ondoren, koordenatuak $[1,-1]^2$ daude, kubo kanonikoaren barrukoak baitira. Horrela, $\mathbf{q} = \mathbf{P}_p\mathbf{p}$ bada, orduan $-1 \leq q_i \leq +1 \quad \forall i \in \{x, y, z\}$





Perspektiba proiekzioa. Sakonera-koordenatu aldatua

- x eta y koordenatuak interpolazio-linealaren bidez aldatzen dira, baina z koordenatua ez

$$\begin{aligned} x' &= f(x) = \frac{2}{r-l}(x - \frac{(r+l)z}{2n}) \\ y' &= g(y) = \frac{2}{t-b}(y - \frac{(t+b)z}{2n}) \\ z' &= h(\frac{1}{z}) = -\frac{2f}{f-n}(1 - \frac{n}{z}) \end{aligned}$$

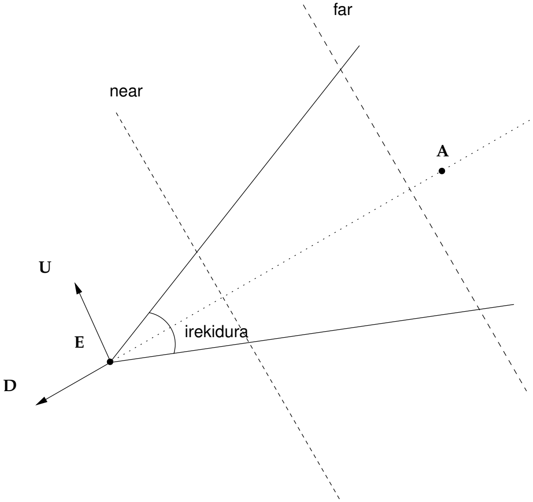
- Proiektatu ondoren, zenbait interpolazio egiten da:
  - normalak
  - testura-koordenatuak
  - ...
- Kontu handia izan behar da z koordenatuko interpolazioarekin
  - Perspektiba interpolazioa
- Praktikan, OpenGL arduratzen da guztiaz

Perspektiba proiekzioa OpenGL-n

```
void glFrustum(   GLdouble left,
                  GLdouble right,
                  GLdouble bottom,
                  GLdouble top,
                  GLdouble zNear,
                  GLdouble zFar)
```

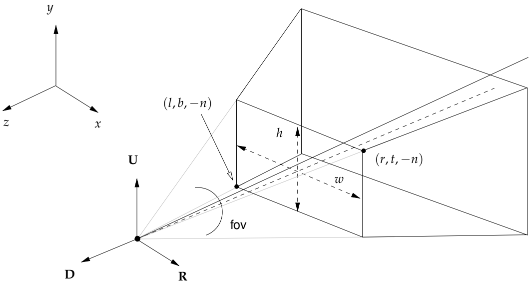
- Praktikan, askotan $l = -r$ eta $b = -t$, beraz pantaila laukizuzena aukeratuz.
- Bi parametro erabili ohi dira perspektiba zehazteko: irekidura (*fov*) eta tamaina-ratioa (*aspect ratio*)

Irekidura-angelua



Field of view

- Piramidearen bolumena adierazten du.
- GLU-k bi parametro eskatzen ditu: fov_y eta $\frac{w}{h}$



Field of view

- **fov**: piramidearen ezker/eskuin eta goiko/beheko planoen arteko angelua (normalean berdina)
- $\phi = 2\arctan(\frac{h}{2n})$

  ϕ fov (y norabidean)
  h altuera
  n plano gertuen distantzia
- Piramidearen zabalera adierazten du.

Planu gertuaren posizioa

- **fov** jakinda (normalean 45–60 gradu inguruan) eta (h) pantailaren altuera jakinda *near* distantzia kalkula daiteke (n)

$$n = \frac{h}{2\tan(\frac{\phi}{2})}$$

- Kamarak objektu bat begiratzen badu
  - Kamara $\mathbf{V}$n kokatuta
  - ϕ ikeridura y norabidean (*fovy*)
  - $\mathbf{A}$n kokatutako objektu bat begiratzen
  - r erradioko esfera batek estaltzen du objektua

$$n = \| A - V \| = \frac{r}{\sin\frac{\phi}{2}}$$

Planu gertuaren posizioa

- Normalean, planu gertua (“Near plane”, n) finkoa da
  - Begietatik gertuegi badago, Z-buffer eta perspektibek doitasuna galtzen dute $\rightarrow$ teilakamenduak
  - Begietatik urruntzen bada: objektu asko ikusmen-puntu eta planoaren artean gera daitezke

gluPerspective

```
void gluPerspective(    GLdouble fovy,
                        GLdouble aspect,
                        GLdouble zNear,
                        GLdouble zFar )
```

- **fovy**: “field of view”, *y* ($\mathbf{U}$) norabidean
- **aspect** (*aspect ratio*): pantailaren zabalera (w) eta altueraren (h) arteko ratioa, $\frac{w}{h}$. *fovy* eta *aspect* jakinda, *x* norabideko “field of view” kalkula daiteke (*fovx*).
- **zNear**: *near plane* distantzia (beti positiboa)
- **zFar**: *far plane* distantzia (beti positiboa)

gluPerspective

- (*fovy, aspect, zNear, zFar*) emanda:
  - Izan bedi $f = \cot(\frac{\text{fovy}}{2}) = \frac{1}{\tan(\frac{\text{fovy}}{2})}$
  - Perspektiba matrizea horrela osatuko da:

$$\mathbf{P}_p = \begin{pmatrix} \frac{f}{\text{aspect}} & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & \frac{\text{zNear}+\text{zFar}}{\text{zNear}-\text{zFar}} & \frac{2\text{zNear}*\text{zFar}}{\text{zNear}-\text{zFar}} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Piramide moztuaren zehaztapena

- Izan bitez fov_y irekidura y norabidean, $\frac{w}{h}$ tamaina-ratioa, n planu gertuaren distantzia eta f atzeko planuaren distantzia ezagunak badira, nola kalkulatu (l, r, b, t) ?
- Kontuan izan behar den erlazioa:

$$\frac{r-l}{t-b} = \frac{w}{h}$$

- Kalukuluak eginez:

$$\begin{aligned} t &= n \tan \frac{\text{fov}_y}{2} \\ b &= -t \\ r &= \frac{w}{h} t \\ l &= -r \end{aligned}$$

Frustum planoen zehaztapena

- (Gribb & Hartmann, 2001) lanean oinarrituta
- Proiekzio eta bista-aldaketa matrizeak hartzen ditu kontuan
- Izan bedi $\mathbf{P}_{o|p}$ proiekzio-matrizea, $\mathbf{M}_b$ bista-aldaketa, eta $\mathbf{M} = \mathbf{P}_{o|p}\mathbf{M}_b$
- $\mathbf{P} = (x, y, z, 1)^T$ puntua beste $\mathbf{T} = \mathbf{MP} = (x', y', z', w')^T$ batean aldatuko da
- $\mathbf{T}$ puntuaren laugarren koordenatua, w' ez da 1 izango, proiekzioa dela medio.
  - koordenatu guztiak w' koordenatuarekin zatitu
- Izan bedi $\mathbf{U} = (u_x, u_y, z_y)$, non: $\left\{ \begin{array}{lcl} u_x & = & \frac{x'}{w'} \\ u_y & = & \frac{y'}{w'} \\ u_z & = & \frac{z'}{w'} \end{array} \right.$

Frustum planoen zehaztapena

- $\mathbf{U}$ puntua dispositiboaren koordenatu normalizatuan dago
- Kamarak ikusiko du baldin puntua 1 unitateko kuboaren barruan dagoen (*clipping cube*), alegia, $-1 \leq u_i \leq 1,\ i \in x, y, z$
- Azter dezagun kuboko ezker planoaren eskuin aldea, non $-1 \leq u_x$

$$-1 \leq u_x \Leftrightarrow -1 \leq \frac{x'}{w'} \Leftrightarrow x' + w' \geq 0 \Leftrightarrow$$
$$(\mathbf{m}_0 \cdot \mathbf{P}) + (\mathbf{m}_3 \cdot \mathbf{P}) \geq 0 \Leftrightarrow (\mathbf{m}_0 + \mathbf{m}_3) \cdot \mathbf{P} \geq 0$$

  non $\mathbf{m}_0$ eta $\mathbf{m}_3$ $\mathbf{M}$ matrizearen lehenengo eta laugarren lerroak diren, hurrenez hurren.
- $(\mathbf{m}_0 + \mathbf{m}_3) \cdot \mathbf{P} \geq 0$ espresioak *frustum*-aren ezker planoa adierazten du

$$\mathbf{n}X = d, \text{ non } \begin{cases} n_x &= m_{00} + m_{30} \\ n_y &= m_{01} + m_{31} \\ n_z &= m_{02} + m_{32} \\ d &= m_{33} - m_{03} \end{cases}$$

Frustum planoen zehaztapena

- Normalaren keinua aldatu behar da, kanpora apunta dezan. Horrela, frustum-aren 6 planoak (l,r,b,t,n,f) dira:

$l\begin{cases} n_x = -m_{30} - m_{00} \\ n_y = -m_{31} - m_{01} \\ n_z = -m_{32} - m_{02} \\ d = m_{33} + m_{03} \end{cases}$	$r\begin{cases} n_x = -m_{30} + m_{00} \\ n_y = -m_{31} + m_{01} \\ n_z = -m_{32} + m_{02} \\ d = m_{33} - m_{03} \end{cases}$
$b\begin{cases} n_x = -m_{30} - m_{10} \\ n_y = -m_{31} - m_{11} \\ n_z = -m_{32} - m_{12} \\ d = m_{33} + m_{13} \end{cases}$	$t\begin{cases} n_x = -m_{30} + m_{10} \\ n_y = -m_{31} + m_{11} \\ n_z = -m_{32} + m_{12} \\ d = m_{33} - m_{13} \end{cases}$
$n\begin{cases} n_x = -m_{30} - m_{20} \\ n_y = -m_{31} - m_{21} \\ n_z = -m_{32} - m_{22} \\ d = m_{33} + m_{23} \end{cases}$	$f\begin{cases} n_x = -m_{30} + m_{20} \\ n_y = -m_{31} + m_{21} \\ n_z = -m_{32} + m_{22} \\ d = m_{33} - m_{23} \end{cases}$

Kamararen parametroak/Datu-egitura

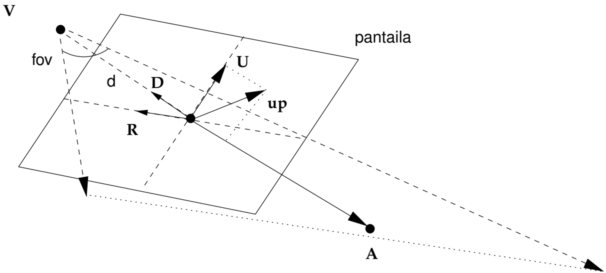
Adieraziko dugu:

- Kokapen eta orientazioa (munduko erref. sistema)
    - $\mathbf{V}$ kokapena
    - $\mathbf{A}$ begiratzen den puntua
    - $\mathbf{up}$ gora doan bektorea
        - Batzuetan unitarioa izen behar du
        - Ez du zertan $\mathbf{VA}$-rekiko elkarzuta izan behar (baina ezin du paraleloa izan)
    - Kamararen z ardatza
        - *Near plane* d_n
        - *Far plane* d_f
    - Sistemarekiko independenteak
        - Irekidura
        - *Aspect ratio* zabalera/altuera $a = \frac{w}{h}$
- Proiekzio mota (ortogonala/perspektiba)

Look at

- $\mathbf{V}, \mathbf{A}, \mathbf{up}$ emanik, erref-sistema horrela kalkulatzen da:
  Izan bedi $\mathbf{F} = \frac{\mathbf{V}-\mathbf{A}}{\|\mathbf{V}-\mathbf{A}\|}$ bektore unitarioa, $\mathbf{A}$tik $\mathbf{V}$ra doana. Orduan

$$
\begin{aligned}
\mathbf{R} &= \frac{\mathbf{up}}{\| \mathbf{up} \|} \times \mathbf{F} \\
\mathbf{U} &= \mathbf{F} \times \mathbf{R} \\
\mathbf{D} &= \mathbf{F}
\end{aligned}
$$



gluLookAt

```
void gluLookAt( GLdouble eyeX,
                GLdouble eyeY,
                GLdouble eyeZ,
                GLdouble centerX,
                GLdouble centerY,
                GLdouble centerZ,
                GLdouble upX,
                GLdouble upY,
                GLdouble upZ )
```

Laburbilduz

Pipeline

- Modeloaren sistematik munduko sistemara $\mathbf{M}_L$
- Munduko sistematik kamararen sistemara
  - Bista-aldaketa: $\mathbf{M}_b$
- Kubo normalizatura: $\mathbf{P}_{o|p}$
- *Frustum culling* (ikusiko dugu)
- Pantailaren koordenatu-sistemara
  - Laukiraketa: $\mathbf{M}_v$

$$\mathbf{P}' = \mathbf{M}_v \mathbf{P}_{o|p} \mathbf{M}_b \mathbf{M}_L \mathbf{P}$$

Kamararen kontrola

- Era anitz dago kamara mugitzeko
  - Aztertu (*examine, circle, arc*)
  - Kamara mugitu (*crane, dolly*)
  - Zoom
  - Biraketa
    - Kamara finkoa (*pan, tilt*)
    - Kamara hegoan, hegazkinak (*yaw, heading, pitch*)
  - Paseatu
  - Objektu bat jarraitu
  - . . .

Kamararen kontrola

- Kamararen egoera erabat finkatzen da bere datu-egiturak une batean dituen parametroekin
- Erabiltzailearen interfazeak zein ataza zehazten duen
- Kamara nola mugitzen den, hots, nola erabaki kamararen egoera berria
  - Uneko egoera
  - Erabiltzailearen elkareragitea

Aztertu

- Kamara bat $\mathbf{A}$ zentroa duen esfera baten gainean mugitzea
- Erabiltzaileak fokua bat aukeratzen du (arreta-puntua)

Aztertu

Arc over-under

- Meridianoaren arabera mugitu (poloen artean)
  - $\mathbf{V}$ eta $\mathbf{up}$ biratu $\mathbf{e}$ ardatzearekiko, zeina munduko XZ planoaren paraleloa den, $\mathbf{VA}$rekiko elkarzuta eta $\mathbf{A}$tik pasatzen den.
  - Kamara polo batetik gertuegi badago $\longrightarrow$ singularitatea

Arc left-right

- Paraleloen arabera mugitu
  - $\mathbf{V}$ eta $\mathbf{up}$ biratu $\mathbf{e}$ ardatzarekiko, zeina munduko Y ardatzarekiko paraleloa den eta $\mathbf{A}$tik pasatzen den.

Zenbat biratu ?

- Faktore konstantea
- Faktore dinamikoa (adib. saguarekin biratzen bada)

Kamararen leku aldaketa

- $\mathbf{V}$ eta $\mathbf{A}$ puntuak mugitu

$$\begin{aligned} \mathbf{V}' &= \mathbf{T}(\mathbf{t})\mathbf{V} \\ \mathbf{A}' &= \mathbf{T}(\mathbf{t})\mathbf{A} \end{aligned}$$

- Zein erreferentzia-sistema erabili ?
- Kamararena
    - *crane*: $\mathbf{R}$ ardatzan zehar (ezker/eskuin): $\mathbf{t} = \pm \mathrm{step} \cdot \mathbf{R}$
    - *dolly*: $\mathbf{U}$ ardatzan zehar (gora/behera): $\mathbf{t} = \pm \mathrm{step} \cdot \mathbf{U}$
    - *truck*: $\mathbf{D}$ ardatzan zehar (aurrera/atzera): $\mathbf{t} = \pm \mathrm{step} \cdot \mathbf{D}$
- Hegan egitearen baliokidea
    - Objektu bat lurrean egonda, eta behera begira, lurraren behera joango litzateke
    - Ezker-eskuinera begiratzea, baina aurrera joan

Kamararen leku aldaketa

- $\mathbf{V}$ eta $\mathbf{A}$ puntuak mugitu

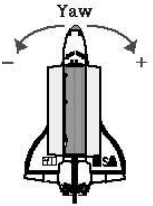
$$\begin{aligned} \mathbf{V}' &= \mathbf{T}(\mathbf{t})\mathbf{V} \\ \mathbf{A}' &= \mathbf{T}(\mathbf{t})\mathbf{A} \end{aligned}$$

- Zein erreferentzia-sistema erabili ?
- Munduarena
  - x ardatzan zehar (ezker/eskuin): $\mathbf{t} = (\pm\text{step}, 0, 0)^T$
  - y ardatzan zehar (gora/behera): $\mathbf{t} = (0, \pm\text{step}, 0)^T$
  - z ardatzan zehar (aurrera/atzera): $\mathbf{t} = (0, 0, \pm\text{step})^T$
- Arazoak
  - Lurra ez bada erregularra!

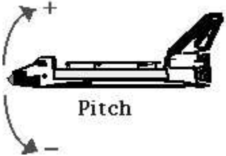
Dolly eta Zoom

- *Dolly* aldaketak (aurreratzea, atzeratzea) *zoom* bat bezala ikus daiteke. Baina ez dira berdinak.
- *Dolly*k ez du sakoneraren efektua lortzen, perspektiba ez baita aldatzen.
- *Zoom* lortzeko “field of view” irekidura aldatu behar da.
  - Angelu handiek (70 80) distortsioak sortzen dute
- Beste modu bat *zoom* egiteko: $\mathbf{A}$ puntua $\mathbf{V}$ra hurbildu
  - Baina ez gehiegi (singularitatea)

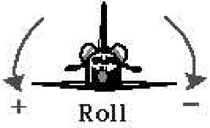
Kamararen biraketa



Heading,yaw: $\mathbf{A}' = \mathbf{R_U}(\alpha)\mathbf{A}$,
$\mathbf{up}' = \mathbf{R_U}(\alpha)\mathbf{up}$



pitch: $\mathbf{A}' = \mathbf{R_R}(\alpha)\mathbf{A}$,
$\mathbf{up}' = \mathbf{R_R}(\alpha)\mathbf{up}$



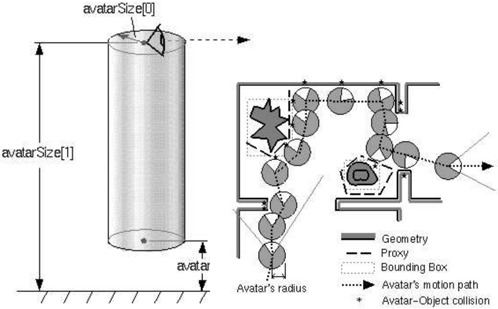
roll: $\mathbf{A}' = \mathbf{R_D}(-\alpha)\mathbf{A}$,
$\mathbf{up}' = \mathbf{R_D}(-\alpha)\mathbf{up}$

http://liftoff.msfc.nasa.gov/academy/rocket_sci/shuttle/attitude/pyr.html

Avatar

- Hiru parametro:
  - Begiaren altuera (lurrarekiko)
    - Kamara bertan kokatzen da
  - Belaunaren altuera
    - Objektu baten kontra joatean, objektuaren gainean igo daitekeen edo ez erabakitzeko
  - Erradioa
    - Kolisioak
- Zilindro bat bezala modelatzen da (belaunetik burura doana)

Avatar


avatarSize[0]
avatarSize[1]
avatar
Avatar's radius
Geometry
Proxy
Bounding Box
Avatar's motion path
Avatar-Object collision