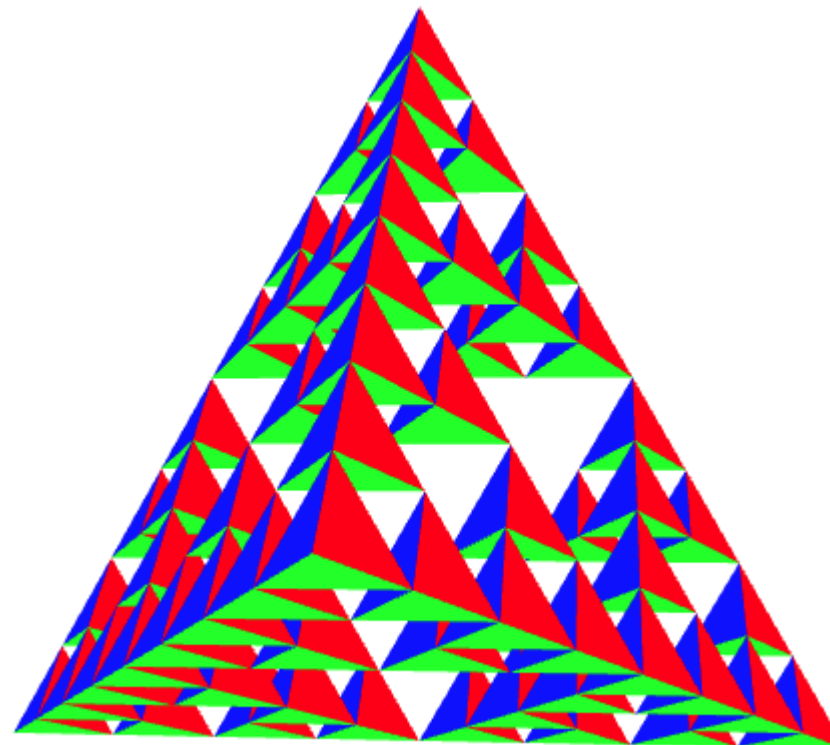


TÖL105M TÖLVUGRAFÍK

Fyrirlestur 12: Ofanvarp

Hjálmtyr Hafsteinsson
Haust 2024



- Sýnidæmi um `lookAt`-fallið
- Ofanvarp (*projection*)
 - Hornrétt (*orthogonal*)
 - Fjarvíddar (*perspective*)
- Útfærsla í WebGL
 - Fallið `ortho()`
 - Fallið `perspective()`

aukaefni

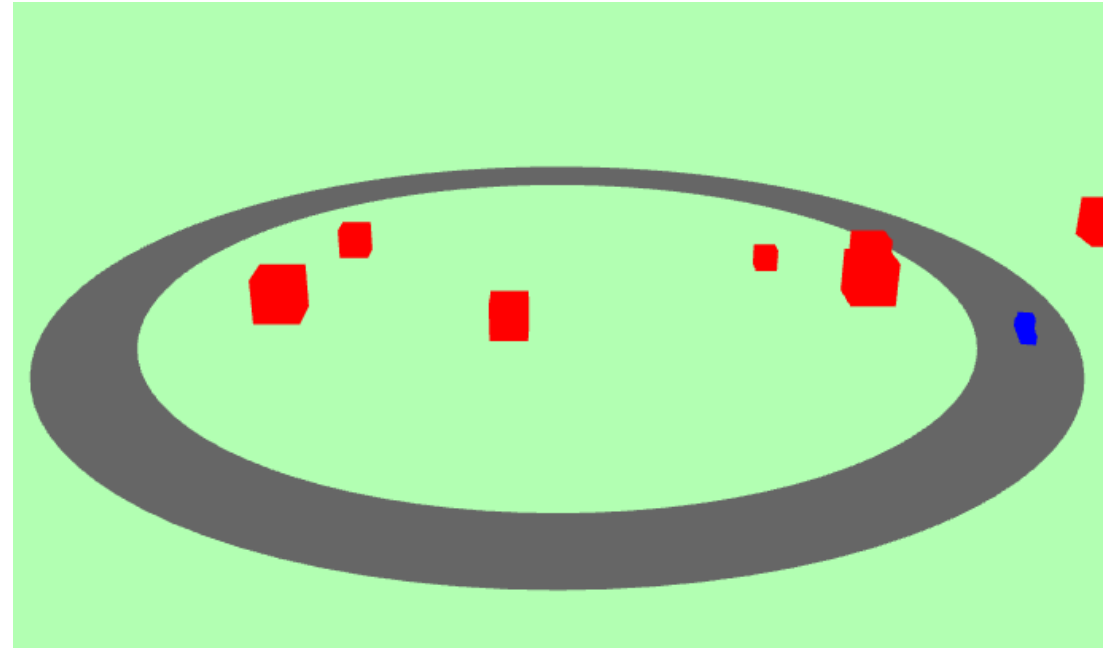
5.4, 5.5

5.6, 5.7

Sýniforrit: viewpoints

- Setur upp hringlaga braut, hús og bíl sem keyrir eftir brautinni
- Skilgreinir mörg sjónarhorn:
 - Ofan frá
 - Innan úr hringnum
 - Innan úr bílnum
 - Fyrir ofan og aftan bílinn
 - Til hliðar við bílinn
 - ...

Öll sjónarhornin eru forrituð með `lookAt`-fallinu og mismunandi röð á teikningu hluta



Brautin liggur í xy-sléttunni, +z er upp
Miðja brautarinnar er með radíus 100.0
(innri radíus 90.0, ytri radíus 110.0)

Fjarlægt og kyrrt sjónarhorn – 1

```
function render() {  
    ...  
    var mv = mat4();  
    switch( view ) {  
        case 1:  
            mv = lookAt( vec3(250.0, 0.0, 100.0+height),  
                        vec3(0.0, 0.0, 0.0),  
                        vec3(0.0, 0.0, 1.0) );  
            drawScenery( mv );  
            mv = mult( mv, translate( carXPos, carYPos, 0.0 ) );  
            mv = mult( mv, rotateZ( carDirection ) );  
            drawCar( mv );  
            break;  
        ...  
    }
```

Auga stað sett
ofarlega og til hliðar

Uppvigur er
+z-ásinn

Bíll fluttur á réttan stað

Bíl er snúið um z-ás

```
carXPos = TRACK_RADIUS * Math.sin(radians(carDirection));  
carYPos = TRACK_RADIUS * Math.cos(radians(carDirection));
```

Bíllinn keyrir réttisælis um brautina

Horft á bíl innan úr hring – 2

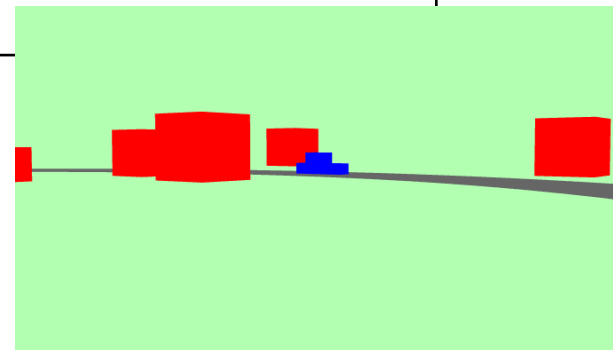
Auga nú inni í hringnum
og horfir á bílinn, sem
hreyfist um hringinn

Hringurinn er í xy-sléttunni,
rádius á innri brún hringa er 90.0

```
function render() {  
    ...  
    var mv = mat4();  
    switch( view ) {  
        case 2:  
            mv = lookAt( vec3(75.0, 0.0, 5.0+height),  
                        vec3(carXPos, carYPos, 0.0),  
                        vec3(0.0, 0.0, 1.0) );  
            drawScenery( mv );  
            mv = mult( mv, translate(carXPos, carYPos, 0.0) );  
            mv = mult( mv, rotateZ( carDirection ) );  
            drawCar( mv );  
            break;  
        ...  
    }
```

Upphafleg hæð er 5.0, en hægt
að breyta með örvalyklum

(carXPos, carYPos) eru hnit bílsins í xy-sléttunni



Horft innan úr bíl – 4

Bíllinn er nú í miðju hnitakerfisins

Snúa öllu í kringum bílinn

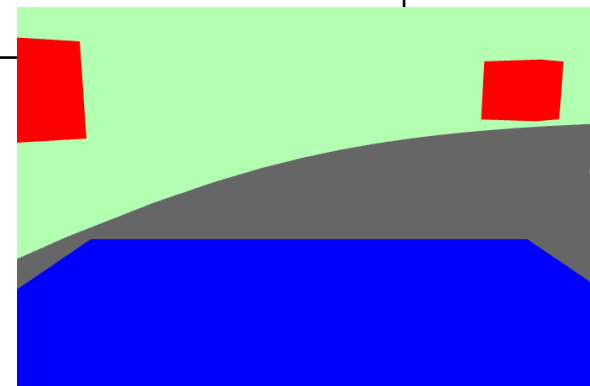
Hliðra öllu að bílnum

Teikna líkan

```
function render() {  
    ...  
    var mv = mat4();  
    switch( view ) {  
        case 4:  
            mv = lookAt( vec3(-3.0, 0.0, 5.0+height),  
                        vec3(12.0, 0.0, 2.0+height),  
                        vec3(0.0, 0.0, 1.0 ) );  
            drawCar( mv );  
            mv = mult( mv, rotateZ( -carDirection ) );  
            mv = mult( mv, translate(-carXPos, -carYPos, 0.0)  
            );  
            drawScenery( mv );  
            break;  
        ...  
    }
```

Í líkanahnitum liggur bíllinn eftir x-ás

Augað er aðeins fyrir aftan miðju bílsins og horfir beint fram á við (eftir x-ás).
Ef augað hækkað þá hækkar áhorfspunktur líka



Innan úr bíl á eitt hús – 5

Vinnur á mótí
snúningi bílsins

Horfa innan úr bíl á
staðinn (40, 120)

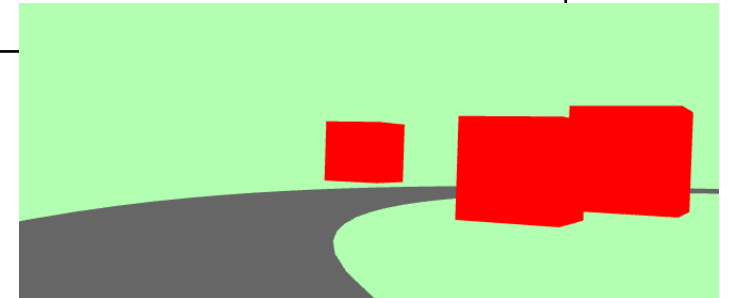
Snúa öllu í kringum bílinn

Hliðra öllu að bílnum

Teikna líkan

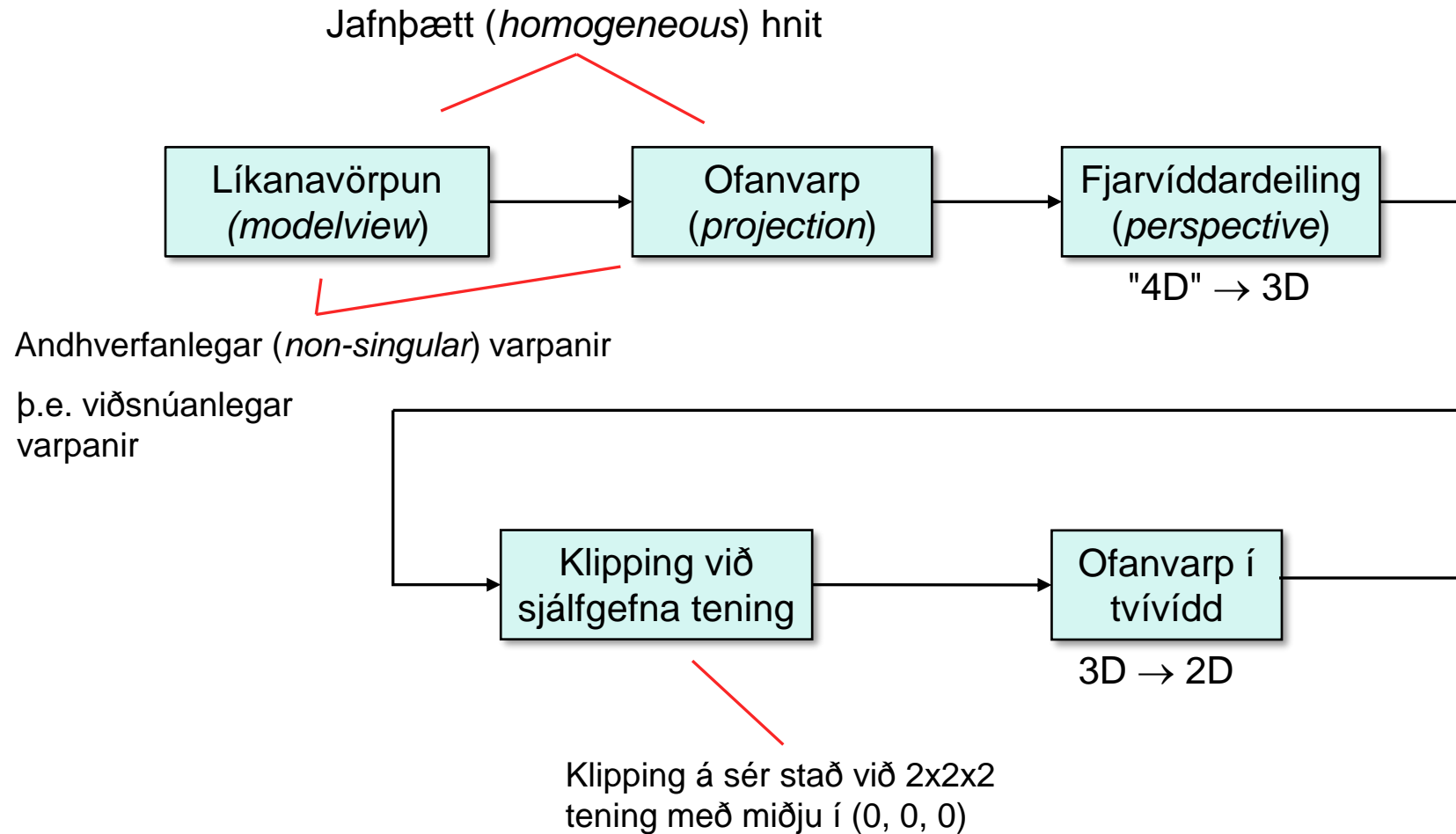
```
function render() {  
  ...  
  var mv = mat4();  
  switch( view ) {  
    case 5:  
      mv = rotateY( carDirection );  
      mv = mult(mv, lookAt(vec3(3.0, 0.0, 5.0+height),  
                           vec3(40.0-carXPos, 120.0-carYPos, 0.0),  
                           vec3(0.0, 0.0, 1.0) ) );  
  
      drawCar( mv );  
      mv = mult( mv, rotateZ( -carDirection ) );  
      mv = mult( mv, translate(-carXPos, -carYPos, 0.0) );  
      drawScenery( mv );  
      break;  
    ...  
  }  
}
```

Áhorfspunkturinn breytist sífellt, því
bíllinn færir, en húsið er kyrrt

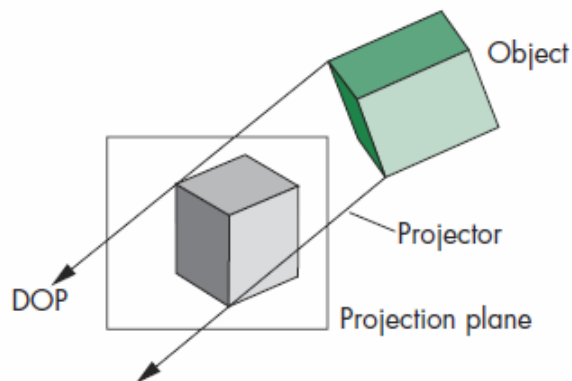


1. Hvaða **lookAt**-skipun myndi horfa beint niður á miðju brautarinnar úr hæðinni 200 í forritinu **viewpoints**?
2. Hvaða viðföng í **ortho**-fallið myndu skilgreina staðalsjónrúmið (*canonical view volume*)?
3. Hvers vegna þarf ekki *fovx* (*field of view in x-direction*) í **perspective**-fallinu?

Ofanvarp í grafíkþípu

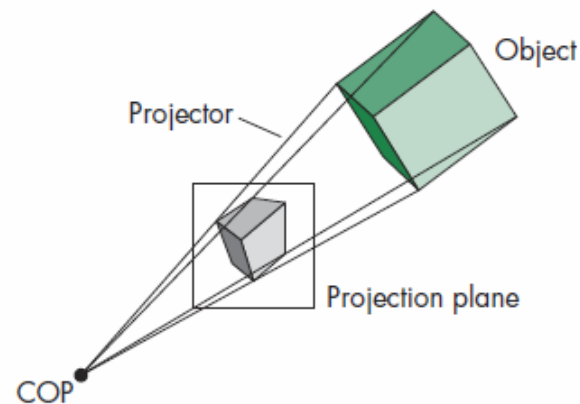


- Erum að varpa þrívíðum hlutum á tvívíða sjónsléttu



Hornrétt (*parallel*)

Hefur sjónstefnu (*direction of projection*),
jafngildir brennipunkti óendanlega langt í burtu
Lengd lína á sjónsléttu er óháð fjarlægð
þeirra frá sjónsléttunni

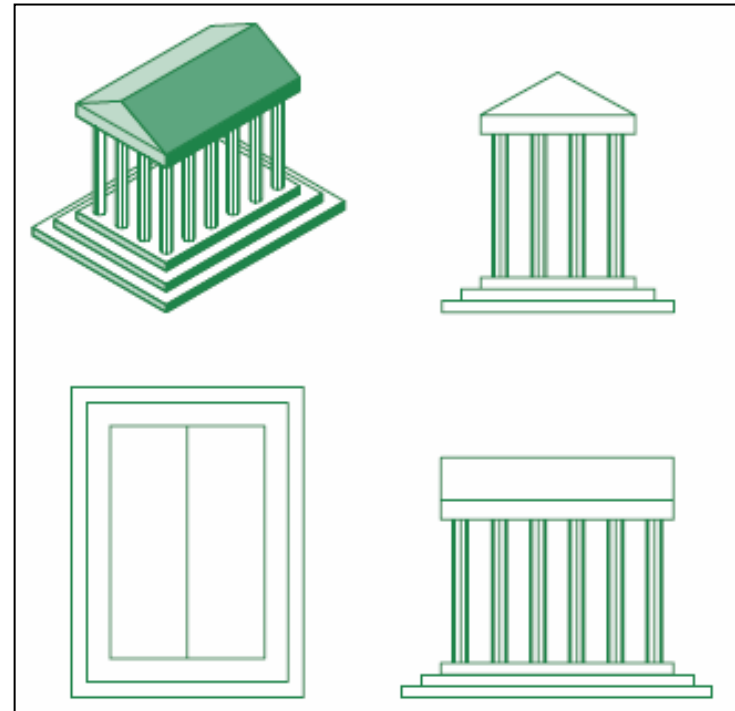


Fjarvíddar (*perspective*)

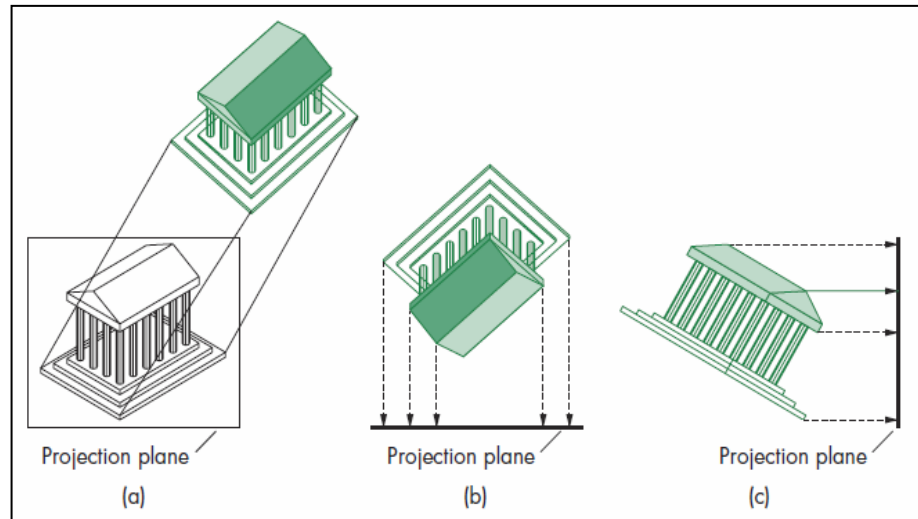
Hefur brennipunkt (*center of projection*)
bakvið sjónsléttu. Fjarlægð hans frá sjónsléttu
ákvarðar sjónhornsbjögun
Fjarlægð lína frá sjónsléttu gerir þær minni
á sjónsléttunni

- Í arkitektateikningum verður að vinna með hornrétt ofanvarp
 - Verður að vera hægt að mæla hversu stór tiltekinn hluti húss á að vera
- Venjulegur notandi fær betri mynd af húsi með fjarvíddarofanvarpi
- Notum því oft fjögur ofanvörp:
 - Hornrétt eftir öllum ásum (3)
 - Fjarvíddar (1)

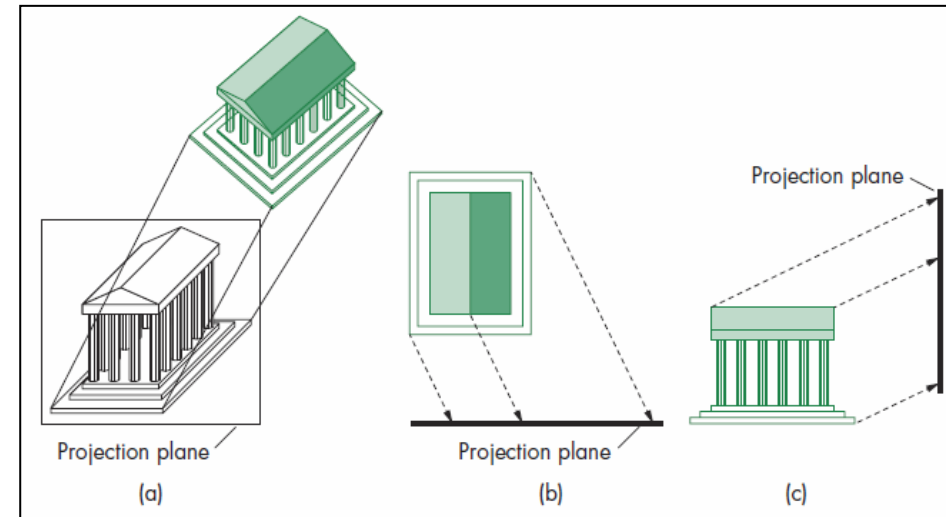
Dæmigert í
CAD forritum



- Samsíða ofanvörp þurfa ekki endilega að vera samsíða hnitakerfisásunum

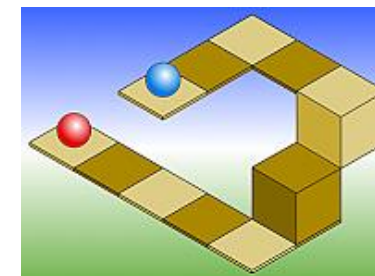


Hnitrétt (*axonometric*) ofanvörp



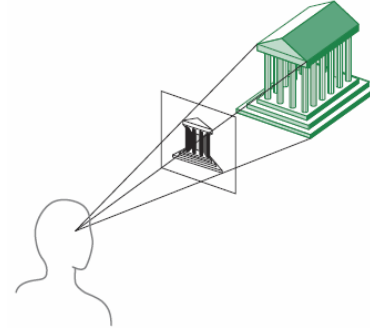
Skásett (*oblique*) ofanvörp

Öll þessi ofanvörp varðveita
fjarlægðir og horn



Getum fengið
furðumyndir

- Eðlilegasta ofanvarpið
 - Okkur finnast hlutir minnka með aukinni fjarlægð
- Listmálarar hafa skilgreint nokkrar gerðir
 - Ræðst af fjölda hverfipunkta (*vanishing points*)



(a)
Þrír

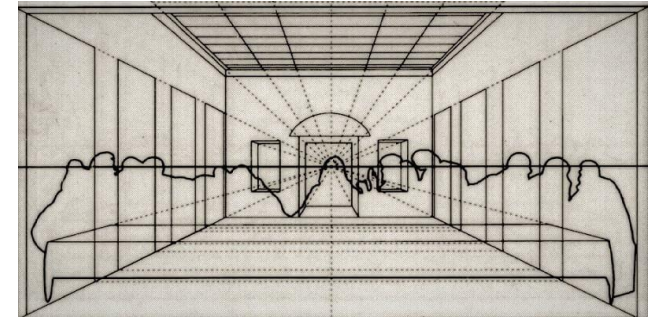


(b)
Tveir



(c)
Einn

Stærðfræðilega allt jafngilt
(ræðst af afstöðu hlutar)



- Sjálfgefið að nota fylkið

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

fáum þá: $\mathbf{p}_p = \mathbf{M}\mathbf{p}$

eða:

$$\begin{aligned} x_p &= x \\ y_p &= y \\ z_p &= 0 \\ w_p &= 1 \end{aligned}$$

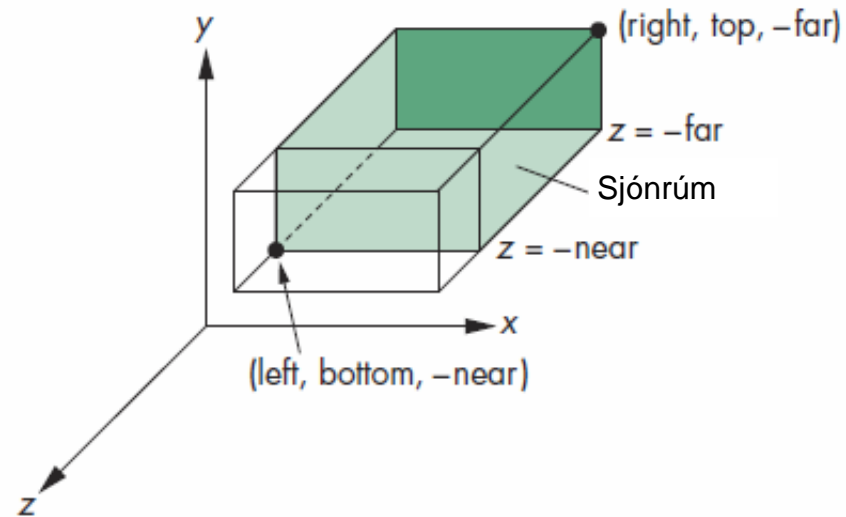
Ofanvarpssléttan er þá $z=0$

Skilgreining samsíða ofanvarps

- Fallið `ortho(. .)` í `MV.js` skilgreinir samsíða ofanvarp og sjónrúm

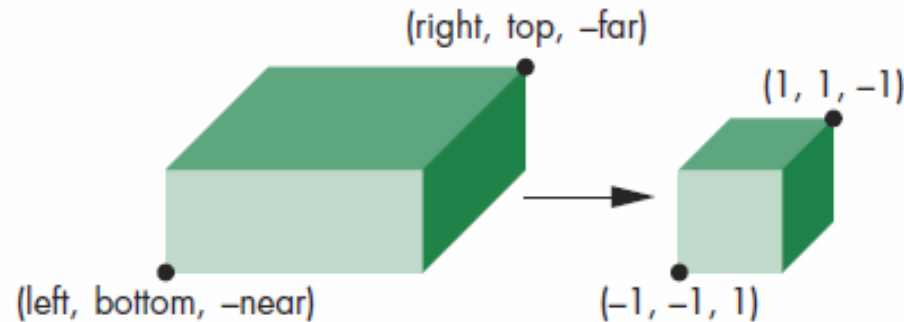
`ortho(left, right, bottom, top, near, far)`

Gildin `near` og `far` eru fjarlægð frá áhorfanda (í núllpunkti)



- Finna vörpun sem breytir skilgreinda sjónrúmi yfir í staðalsjónrúm
 - Staðalsjónrúm (*canonical view volume*) er 2x2x2 teningur

`ortho(left, right, bottom, top, near, far)`



- Samsett úr tveimur vörpunum:

- Færa miðju kassans í núllpunkt

$$T(-(left+right)/2, -(top+bottom)/2, (far+near)/2)$$

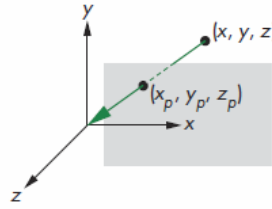
- Kvarða til að lengd hliðanna sé 2

$$S(2/(left-right), 2/(top-bottom), 2/(near-far))$$

$$P = ST = \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & -\frac{right+left}{right-left} \\ 0 & \frac{2}{top-bottom} & 0 & -\frac{top+bottom}{top-bottom} \\ 0 & 0 & \frac{2}{near-far} & \frac{far+near}{far-near} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

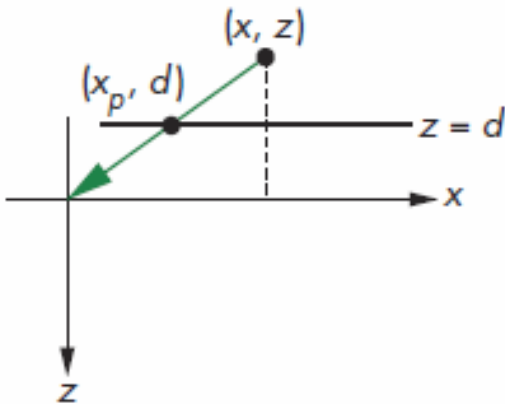
1. Hvaða `lookAt`-skipun myndi horfa beint niður á miðju brautarinnar úr hæðinni 200 í forritinu `viewpoints`?
2. Hvaða viðföng í `ortho`-fallið myndu skilgreina staðalsjónrúmið (*canonical view volume*)?
3. Hvers vegna þarf ekki *fovx* (*field of view in x-direction*) í `perspective`-fallinu?

Fjarvíddarofanvarp

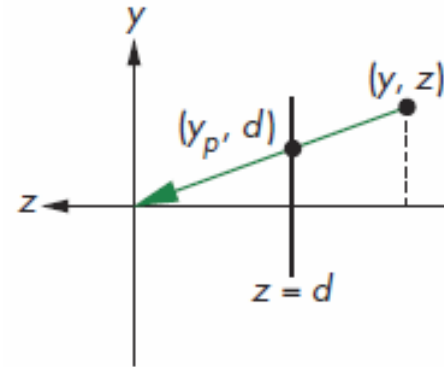


- Brennipunktur (miðja ofanvarps) er $(0, 0, 0)$ og ofanvarpssléttan er $z = d$ ($d < 0$)

Horft ofan frá:



Horft frá hlið:



Gefur jöfnurnar:

$$x_p = \frac{x}{z/d} \quad y_p = \frac{y}{z/d} \quad z_p = d$$

- Nýtum okkur jafnpætt hnit til að einfalda fylkið

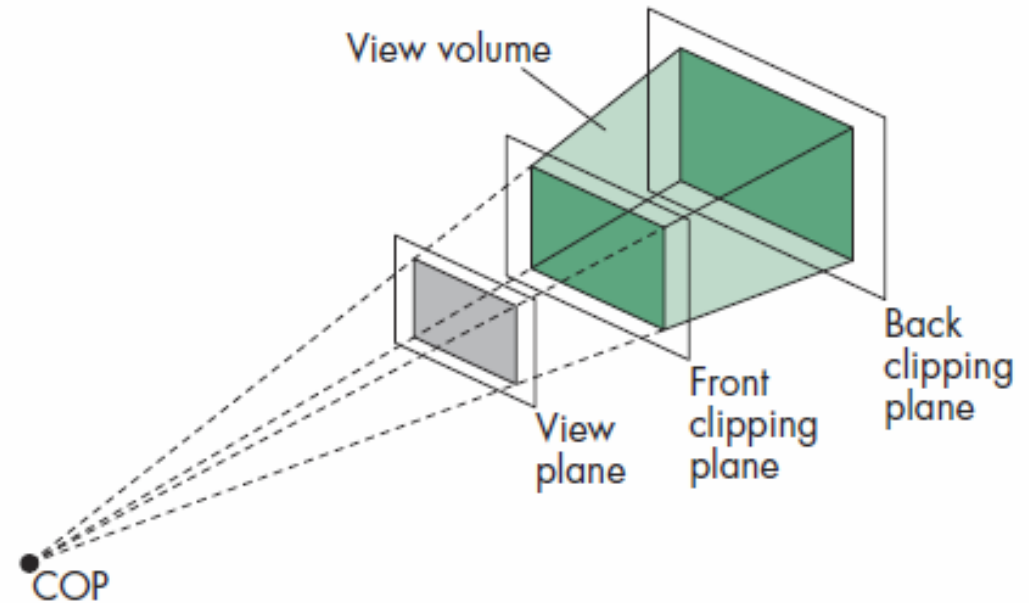
$$\mathbf{q} = \mathbf{M}\mathbf{p} \quad \text{þar sem} \quad \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{og} \quad \mathbf{q} = \begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix}$$

en nú er jafnpætta hnitið $w \neq 1$, svo við deilum í gegn með því og fáum:

$$\mathbf{q}' = \begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{bmatrix}$$

sem er nákvæmlega það sem við vildum fá

- Sjónrúm í fjarvídd (*perspective*) verður afklipptur píramídi (*frustum*)
 - Myndast á milli fremri og aftari klippisléttu (*clipping plane*)
- Getum skilgreint þennan afklippta píramíta beint með falli úr **MV.js**

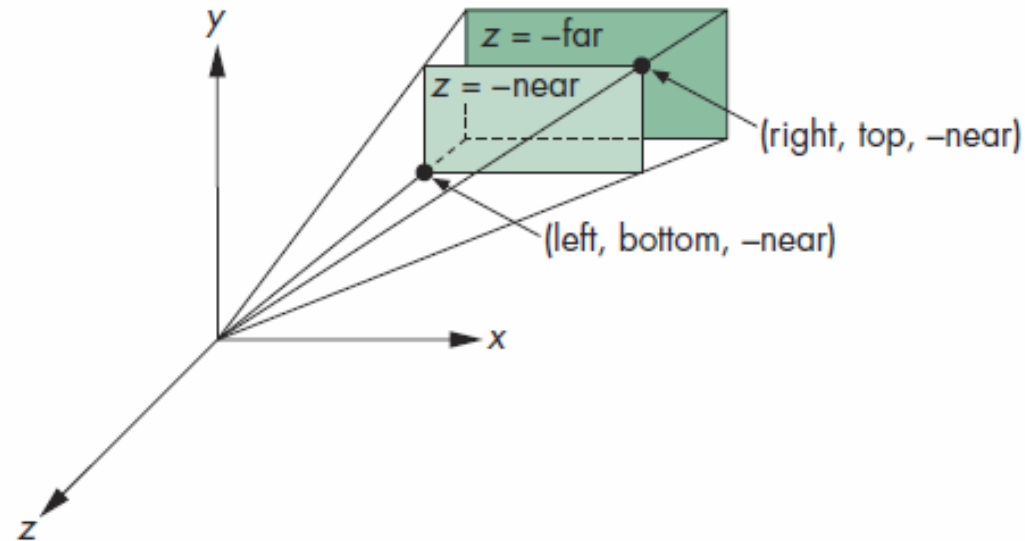


Sjá [demó](#) frá Cornell

`frustum(left, right, bottom, top, near, far)`

Eftirlíking á falli úr
"gamla" OpenGL

Mjög svipað og skilgreining
sjónrúms í hornréttu ofanvarpi



Vandamál: Erfitt að fá það ofanvarp sem óskað er eftir.
Framsetningin er ekki mjög rökrétt

Betra fjarvíddarfall í WebGL

`perspective(fovy, aspect, near, far)`

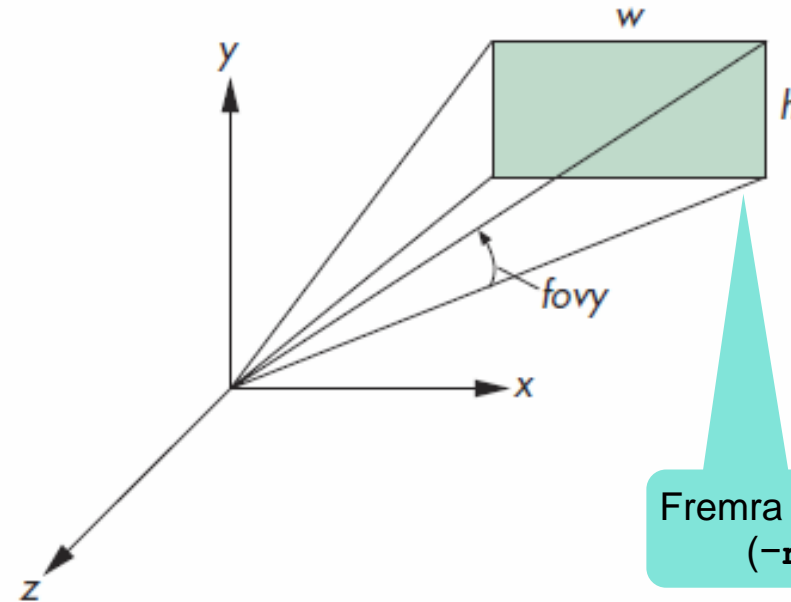
Eftirlíking á falli úr
"gamla" OpenGL

fovy: sjónhorn í y-átt
fovy: *field of view in y-direction*

aspect = w/h

near: fjarlægð fremri klippisléttu frá núllpunkti

far: fjarlægð aftari klippisléttu frá núllpunkti



Fremra klippislétta
(-near)

- Sendum bæði líkanafylki og ofanvarpsfylki til hnútalitara

```
uniform mat4 modelViewMat;  
uniform mat4 projectionMat;  
  
void main() {  
    fColor = vColor;  
    gl_Position = projectionMat * modelViewMat * vPosition;  
}
```

Skilgreinum og sendum ofanvarpsfylki oftast í upphafi forrits,
þ.e. í fallinu `init()`

```
projMatrixLoc = gl.getUniformLocation( program, "projectionMat" );  
projMatrix = perspective(fovy, aspect, near, far);  
gl.uniformMatrix4fv( projMatrixLoc, false, flatten(projMatrix)
```

Sjá: [cube-ortho](#)

Sjá: [cube-persp](#)

1. Hvaða `lookAt`-skipun myndi horfa beint niður á miðju brautarinnar úr hæðinni 200 í forritinu `viewpoints`?
2. Hvaða viðföng í `ortho`-fallið myndu skilgreina staðalsjónrúmið (*canonical view volume*)?
3. Hvers vegna þarf ekki *fovx* (*field of view in x-direction*) í **perspective**-fallinu?