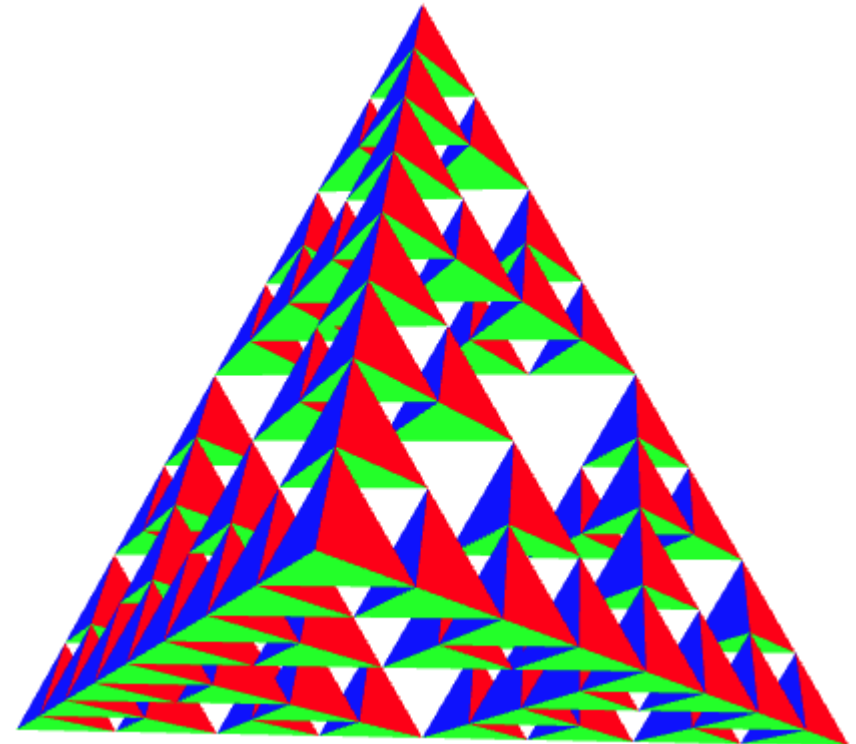


TÖL105M TÖLVUGRAFÍK

Fyrirlestur 11: Sjónvörpun

Hjálmtyr Hafsteinsson
Haust 2024



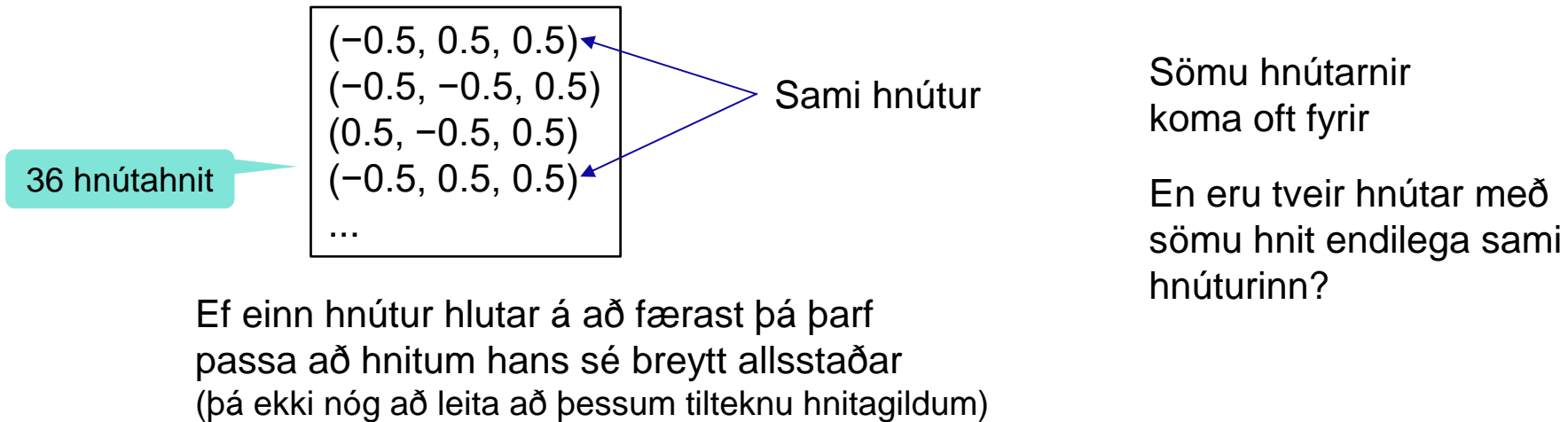
- Önnur framsetning á teningi
 - `gl.drawElements`
- Sýnidæmi um samsettar varpanir
 - Bókstafur
 - Sólkerfi
- Sjónvörpun
 - Staðsetning auga
 - Sjónrúm

4.6

4.10, 4.11
aukaefni

5.1 - 5.3

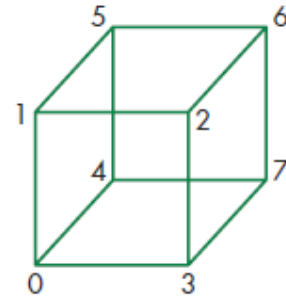
- Höfum skilgreint hluti með því að gefa upp hnit allra þríhyrninga sem mynda þá
 - Teningur skilgreindur með 12 þríhyrningum, eða með 36 hnúta-hnitum



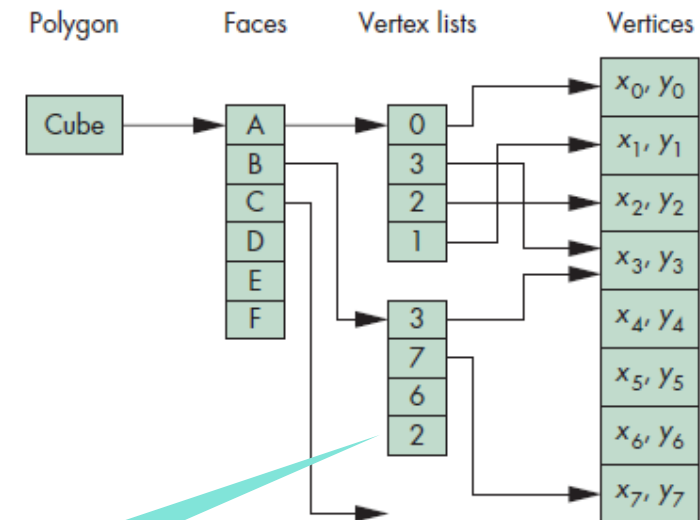
- Við skilgreiningu líkana er betra að skipta þeim upp í:
 - Rúmfræði (*geometry*) – staðsetningu einstakra hnúta
 - Grannfræði (*topology*) – hvernig hnútar eru tengdir í hliðar

- Skilgreinum þá tvö fylki:

- Hnit hnútanna
 - Teningurinn hefur 8 hnúta
- Hvaða hnútar mynda hvern þríhyrning
 - Teningurinn hefur 12 þríhyrninga



Breytum þá hniti
hnútar á einum stað



Listi yfir hnúta
í hverri hlið

Hnit hvers hnútar

Teningur á nýju formi

- Skilgreinum tening með 8 hnútum og 12 þríhyrningum

```
var vertices = [  
    vec3( -0.5, -0.5,  0.5 ),  
    vec3( -0.5,  0.5,  0.5 ),  
    vec3(  0.5,  0.5,  0.5 ),  
    vec3(  0.5, -0.5,  0.5 ),  
    vec3( -0.5, -0.5, -0.5 ),  
    vec3( -0.5,  0.5, -0.5 ),  
    vec3(  0.5,  0.5, -0.5 ),  
    vec3(  0.5, -0.5, -0.5 )  
];
```

Hnútar

Sjá: [cube-elem](#)

```
var indices = [  
    1, 0, 3,  
    3, 2, 1,  
    2, 3, 7,  
    7, 6, 2,  
    3, 0, 4,  
    4, 7, 3,  
    6, 5, 1,  
    1, 2, 6,  
    4, 5, 6,  
    6, 7, 4,  
    5, 4, 0,  
    0, 1, 5  
];
```

Þríhyrningar

- Senda vísa í sérstakt vísaminni á GPU:

```
var iBuffer = gl.createBuffer();  
gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, iBuffer);  
gl.bufferData(gl.ELEMENT_ARRAY_BUFFER, new Uint8Array(indices), gl.STATIC_DRAW);
```

Önnur gerð af minnissvæði

Vísarnir eru jákvæðar 8-bita heiltölur

Notum svo fallið `gl.drawElements` til að teikna, í stað `gl.drawArrays`

```
function render() {  
  ...  
  gl.drawElements( gl.TRIANGLES, numVertices, gl.UNSIGNED_BYTE, 0 );  
  ...  
}
```

Fjöldi hnúta

Tag vísa

Upphafsstaður
í vísafylki

1. Nefnið einn kost og einn ókost við að aðgreina rúmfræði og grannfræði þegar þrívíðir hlutir eru skilgreindir (þ.e. tilgreina hnit hnúta sér og tengingu þeirra í þríhyrninga sér).
2. Mars hefur fylgitunglin Fóbos og Deimos. Teiknið upp vörpunartré með sólinni, jörðinni, tunglinu, Mars og Fóbos og Deimos.
3. Að hvaða leyti er *mouse-look* takmarkaðra en *roll-pitch-yaw*?

Sýnidæmi: Bókstafurinn H



- Búum til bókstafinn **H** í prívídd úr þremur teningum

Vinstri leggurinn

Hægri leggurinn

Miðjustöngin

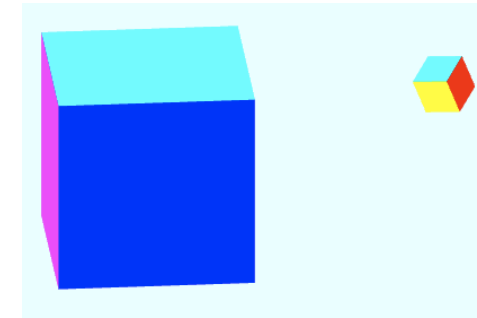
```
function render() {  
    ...  
    mv = mat4();  
    mv1 = mult( mv, translate( -0.3, 0.0, 0.0 ) );  
    mv1 = mult( mv1, scalem( 0.1, 1.0, 0.1 ) );  
    gl.uniformMatrix4fv(matrixLoc, false, flatten(mv1));  
    gl.drawArrays( gl.TRIANGLES, 0, numVertices );  
  
    mv1 = mult( mv, translate( 0.3, 0.0, 0.0 ) );  
    mv1 = mult( mv1, scalem( 0.1, 1.0, 0.1 ) );  
    gl.uniformMatrix4fv(matrixLoc, false, flatten(mv1));  
    gl.drawArrays( gl.TRIANGLES, 0, numVertices );  
  
    mv1 = mult( mv, scalem( 0.6, 0.1, 0.1 ) );  
    gl.uniformMatrix4fv(matrixLoc, false, flatten(mv1));  
    gl.drawArrays( gl.TRIANGLES, 0, numVertices );  
    requestAnimationFrame( render );  
}
```

Hverjum hluta er varpað með **TRS** vörpun (reynðar án **R** í þessum tilfellum!)

Sjá: [letterH](#)

Sýnidæmi: "Sólkerfið"

- Jörð sem snýst í kringum sólu og sjálfa sig
 - Bæði teningar!



Sólin er bara kvörðuð

```
function render() {  
    ...  
    mv = mat4();  
  
    // teikna "sólina"  
    mv = mult( mv, scalem( 0.5, 0.5, 0.5 ) );  
    gl.uniformMatrix4fv(matrixLoc, false, flatten(mv));  
    gl.drawArrays( gl.TRIANGLES, 0, numVertices );  
  
    // teikna "jörðina"  
    mv = mult( mv, rotateY( rotYear ) );  
    mv = mult( mv, translate( 1.8, 0.0, 0.0 ) );  
    mv = mult( mv, rotateZ( earthTilt ) );  
    mv = mult( mv, rotateY( rotDay ) );  
    mv = mult( mv, scalem( 0.2, 0.2, 0.2 ) );  
    gl.uniformMatrix4fv(matrixLoc, false, flatten(mv));  
    gl.drawArrays( gl.TRIANGLES, 0, numVertices );  
    requestAnimationFrame( render );  
}
```

Snúa í kringum sólu

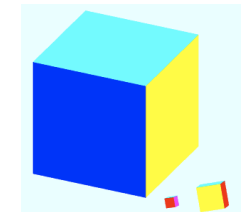
Færa frá sólu

Halla jörðinni

Snúa um sjálfa sig

Jörðin kvörðuð um
0.2*0.5

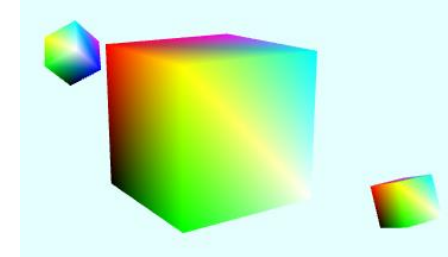
[Bæta við tungli:](#)



Sjá: [solkerfi](#)

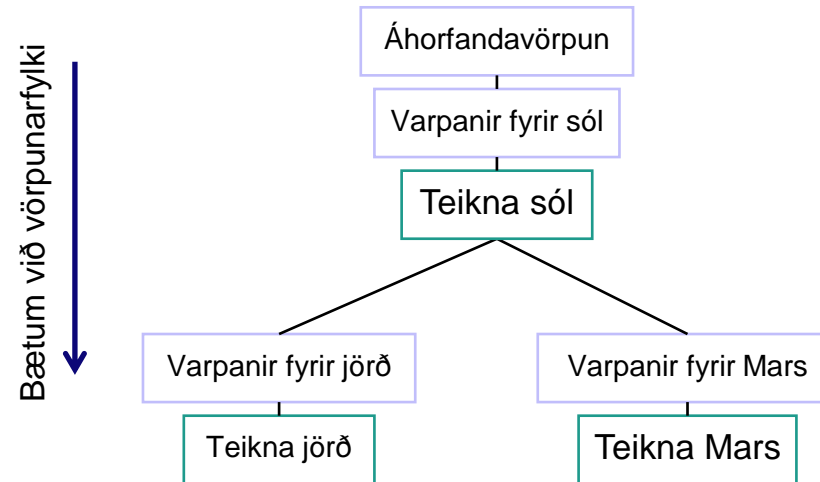
Sýnidæmi: Bæta Mars við

- Jörð og Mars snúast bæði í kringum sól
 - Passa að varpanir fyrir jörð hafi ekki áhrif á Mars
 - Notaðar breytur eða hlaða (*stack*) fyrir vörpunarfylki



Þegar jörð teiknuð:
Áhorfendavörpun*Sólvörpun*Jörðvörpun

Þegar Mars teiknaður:
Áhorfendavörpun*Sólvörpun*Marsvörpun



Sjá: [solkerfi-mars](#)

Notar `gl.drawElements` og hlaða fyrir vörpunarfylki

1. Nefnið einn kost og einn ókost við að aðgreina rúmfræði og grannfræði þegar þrívíðir hlutir eru skilgreindir (þ.e. tilgreina hnit hnúta sér og tengingu þeirra í þríhyrninga sér).
2. Mars hefur fylgitunglin Fóbos og Deimos. Teiknið upp vörpunartré með sólinni, jörðinni, tunglinu, Mars og Fóbos og Deimos.
3. Að hvaða leyti er *mouse-look* takmarkaðra en *roll-pitch-yaw*?

Áhorf (*viewing*)

- Til að skilgreina áhorfanda í WebGL þarf:
 - Staðsetja áhorfandann
 - Setja líkanafylki (*model-view*)
 - Velja "linsu" (þ.e. ákveða bjögun)
 - Setja ofanvarpsfylki (*projection*)
 - Ákveða hvað sést (hve langt í burtu)
 - Setja sjónrúm (*view volume*)

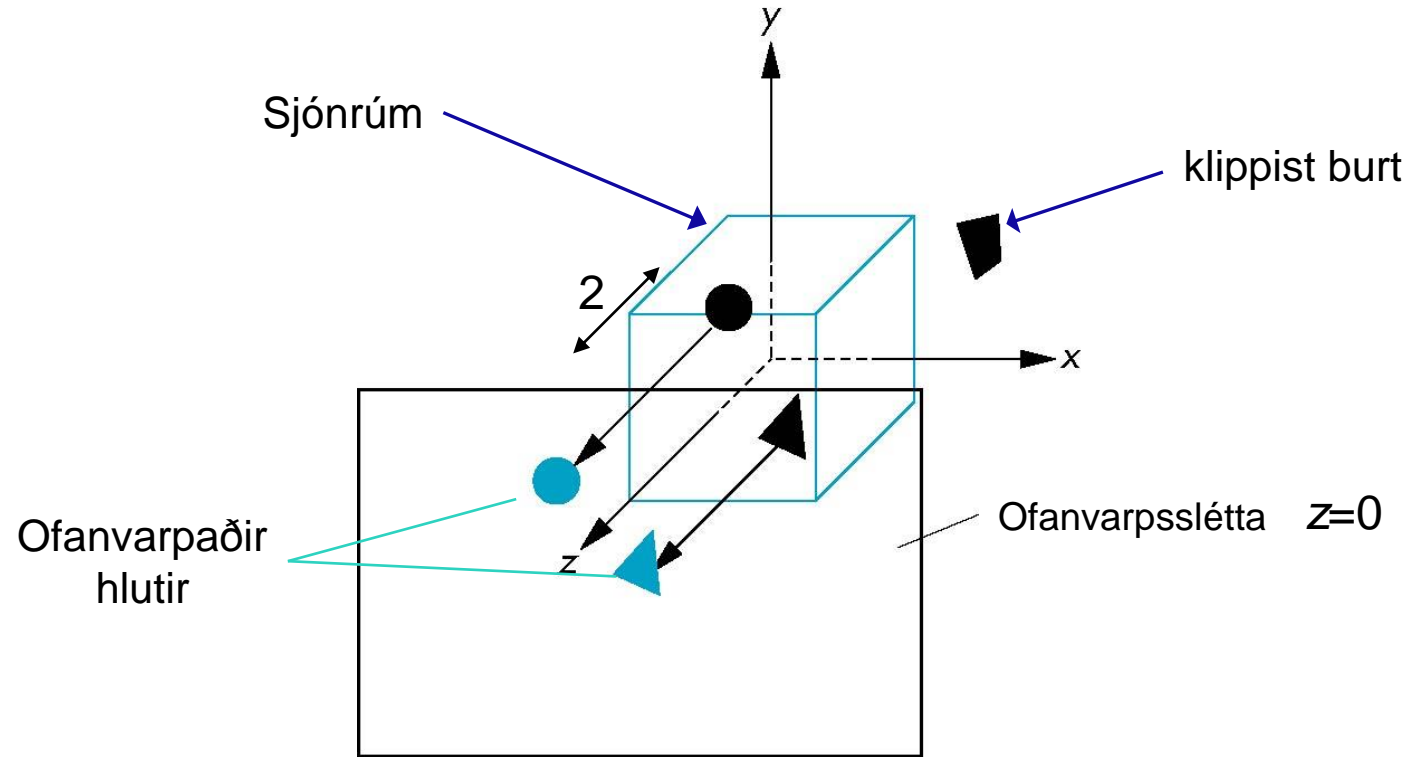
Sjálfgefin gildi:

Staðsetning í $(0, 0, 0)$
Horft niður eftir $-z$ -ás

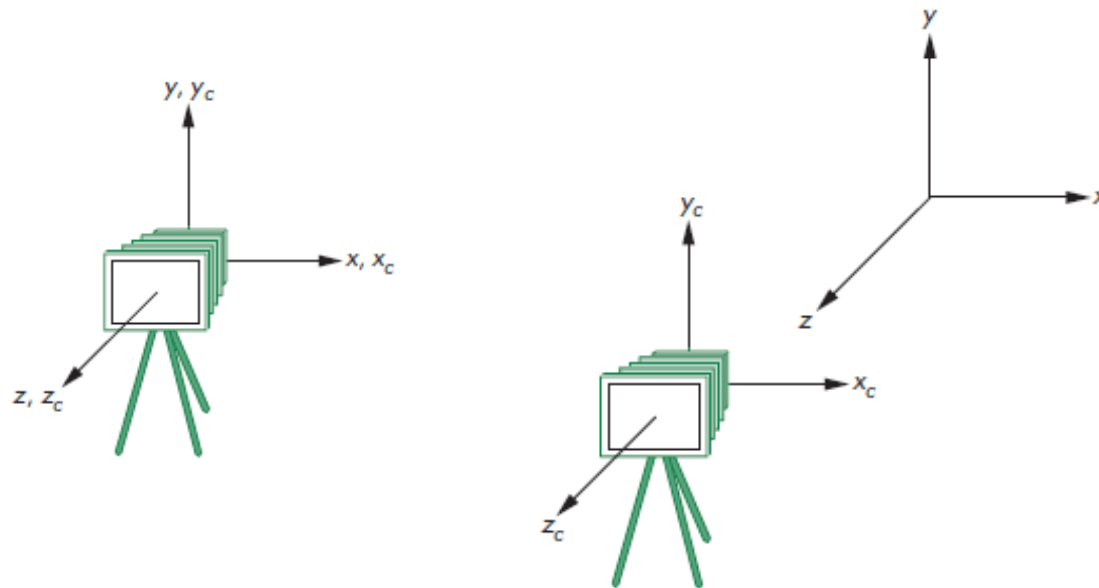
Ofanvarpsfylki er I
(einingarfylkið)

Kassi með miðju í $(0, 0, 0)$ og
hornpunkta $(-1, -1, -1)$ og $(1, 1, 1)$

Sjálfgefið áhorf



- Getum staðsett áhorfanda með venjulegri hliðrun
 - Áhorfandinn er upphaflega í $(0, 0, 0)$
 - Viljum færa hann í $(0, 0, 5)$



Til baka um 5
"skref" eftir z-ás

En þetta er jafngilt því að hliðra
öllum öðrum hlutum líkansins
aftur um 5 skref eftir z-ás!

Síðasta vörpunin er þá: $T(0, 0, -5)$

Einföld staðsetning áhorfanda

- Síðasta vörpunin er staðsetning áhorfanda
 - Hún er því efsta vörpunin í **render**-fallinu

Færa allt aftur um 5.0 skref eftir z-ás

Snúa öllu um 30° rangsælis um x-ás

Ýmsar varpanir líkans

```
function render() {  
    ...  
    var mv = mat4();  
    mv = mult( mv, translate(0.0, 0.0, -5.0));  
    mv = mult( mv, rotateX(30));  
    ...  
    ...  
}
```

Hliðrun áhorfanda
um 5 skref

jafngilt

Hliðrun líkans
um -5 skref

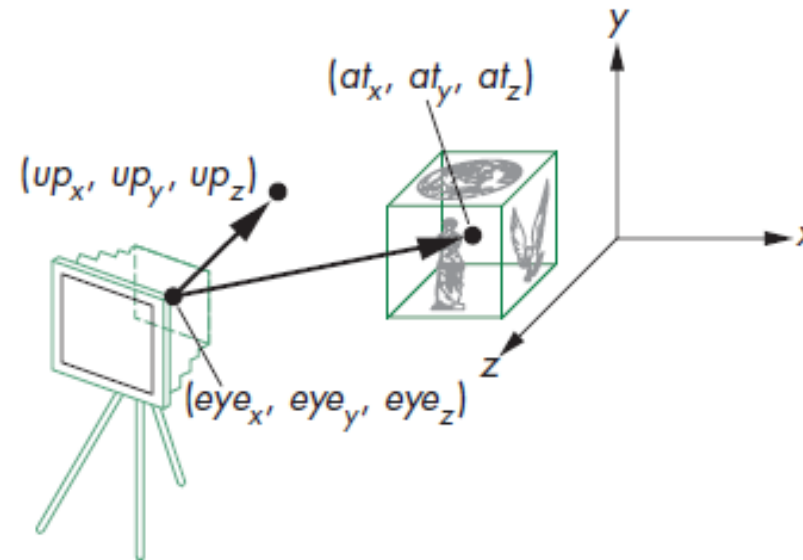
Snúningur áhorfanda
um 30°réttisælis

jafngilt

Snúningur líkans
um 30°rangsælis

- Gamla OpenGL hafði fallið `gluLookAt` til að staðsetja áhorfanda á einfaldan hátt
 - Í `MV.js` er fallið `lookAt` sem hermir eftir því
 - Fallið `lookAt` skilar vörpunarfylki sem verður síðasta vörpunin (þ.e. fyrsta margföldunin!)

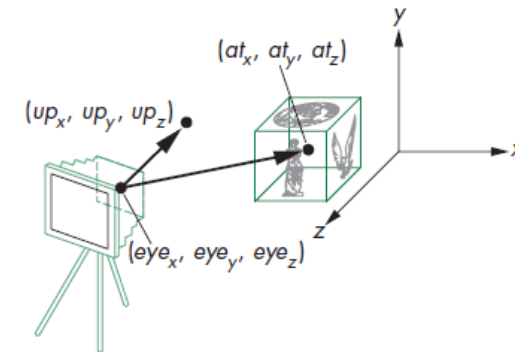
`lookAt(eye, at, up)`



Notkun á lookAt

`lookAt(eye, at, up)`

- **eye** er staðsetning áhorfanda í heimshnitum (punktur)
- **at** er staðsetning áhorfspunkts (punktur)
- **up** gefur halla áhorfanda (vigur)



Leyfum notanda að
breyta z-hniti áhorfanda

Horft á núllpunkt

Enginn halli

```
function render() {  
    ...  
    var mv = lookAt( vec3(0.0, 0.0, zDist),  
                     vec3(0.0, 0.0, 0.0),  
                     vec3(0.0, 1.0, 0.0) );  
    ...  
    ...  
}
```

Finum áhorfsvigur (*view vector*) með
 $\mathbf{v} = \mathbf{at} - \mathbf{eye}$

Höfum þá tvo óháða vigra fyrir
sjónhnitakerfi, \mathbf{v} og \mathbf{up}

Fáum hornréttan vigur á þá
með krossfeldi: $\mathbf{n} = \mathbf{v} \times \mathbf{up}$

En \mathbf{v} og \mathbf{up} eru ekki endilega hornréttir,
svo við búum til nýjan "uppvigur" með
 $\mathbf{u} = \mathbf{n} \times \mathbf{v}$

Snúum \mathbf{v} við (horfa niður eftir honum)

Fylkið sem vigrarnir \mathbf{n} , \mathbf{u} og \mathbf{v} mynda
breytir úr heimshnitum í sjónhnit

```
function lookAt( eye, at, up ) {  
    ...  
    var v = normalize( subtract(at, eye) );  
    var n = normalize( cross(v, up) );  
    var u = normalize( cross(n, v) );  
  
    v = negate( v );  
  
    var result = mat4(  
        vec4( n, -dot(n, eye) ),  
        vec4( u, -dot(u, eye) ),  
        vec4( v, -dot(v, eye) ),  
        vec4()  
    );  
  
    return result;  
}
```

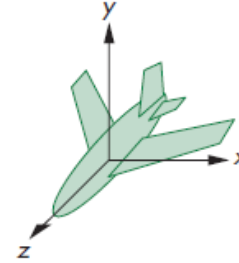
Vantar bara hliðrun í punktinn **eye**. Gerum það með
translate(-eye) (sett í fylkið sem punktfeldi **eye** við hverja línu)

Aðrar staðsetningaaðferðir

- **lookAt**-viðmótið er ekki eina leiðin til að staðsetja áhorfanda

- *roll, pitch, yaw*

- Eins og notað er í flugvélum



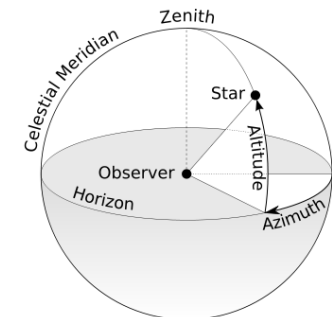
- *mouse-look*

- Hreyfing með W-A-S-D lyklum, en músarhreyfing stýrir stefnu og hvert horft er



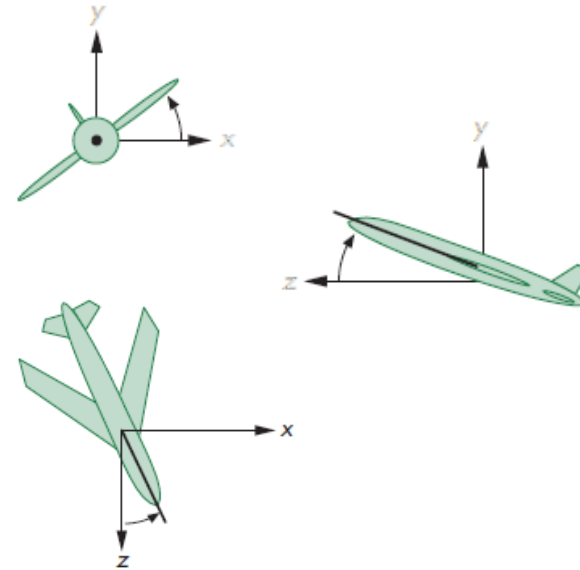
- *elevation, azimuth, twist*

- Notað í stjörnufræði til að staðsetja hluti



roll, pitch, yaw

- *roll* (hliðarhalli) – snúningur um z-ás
- *pitch* (halli) – snúningur um x-ás
- *yaw* (stefna) – snúningur um y-ás



Getum útfært fallið:

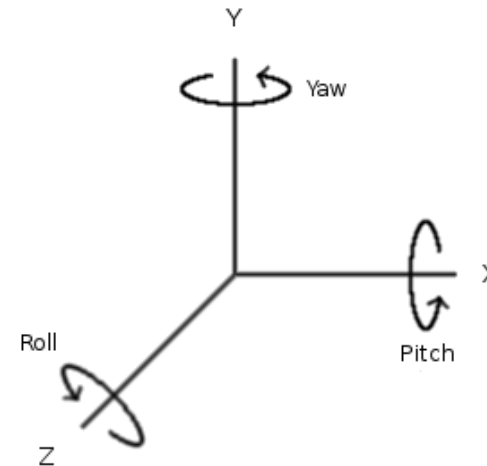
`rpvView(eye, r, p, y)`

Fær staðsetningu (**`eye`**), og snúning
um ásana þrjá:

$$-90 \leq r \leq 90$$

$$-90 \leq p \leq 90$$

$$0 \leq y \leq 360$$



1. Nefnið einn kost og einn ókost við að aðgreina rúmfræði og grannfræði þegar þrívíðir hlutir eru skilgreindir (þ.e. tilgreina hnit hnúta sér og tengingu þeirra í þríhyrninga sér).
2. Mars hefur fylgitunglin Fóbos og Deimos. Teiknið upp vörpunartré með sólinni, jörðinni, tunglinu, Mars og Fóbos og Deimos.
3. Að hvaða leyti er *mouse-look* takmarkaðra en *roll-pitch-yaw*?