

20.11.2019  
Görsel Bilgi

- canvas'ı da kare yapmıştık.
- farklı aspect ratio istiyorsak görebileceğimiz alanın koordinatlarını da ona göre değiştirebiliriz.
- Bu bunlardan matrisleri elde ediyoruz. (Dönüşüm matrisi). Bu matrisleri birim verdiğimizde dörtgenler prizmasını default'a değiştirebiliriz.

$$S_x = \frac{2}{\text{right} - \text{left}} \Rightarrow x \text{ 'teki scale miktarı.}$$

- translation da görebiliriz. Her zaman origin merkezinde olmayabilir.

$$\left( \frac{\text{right} + \text{left}}{2}, \frac{\text{top} + \text{bottom}}{2}, -\frac{\text{far} - \text{near}}{2} \right) \Rightarrow \text{merkez}$$

- Bu volume'un merkezini origine taşımak için translate yaparız.

### Perspective Projection in WebGL

- camera position is COP. Our camera is at the origin.
- Projection plane is at  $z=d$  where  $d < 0$  (negative  $z$ )

$$\frac{x_p}{d} = \frac{x}{z}$$

$$\frac{y_p}{d} = \frac{y}{z}$$

unit now,

 $\rightarrow (x, y, z, 1)$ 

- Son satır 0 0 0 1 oluyordu. w value can be different from 1.

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

$$(x, y, z, 1) \rightarrow (x, y, z, z/d)$$

w'ya bölüyoruz ki koordinatları elde ederiz.

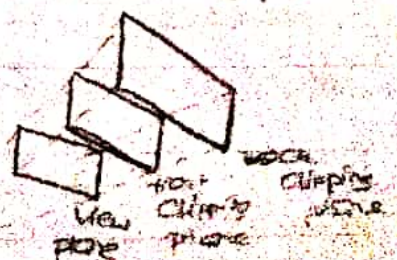
- frustum (left, right, bottom, top, near, far)

or  
perspective (fovy, aspect, near, far)özellikle  
pers.  
yapabiliriz.

angle  
in y  
direction

aspect  
ratio

planes



\* frustum  $\rightarrow$  orthonormal bir set. Bu setin kökeni prizma ve küre. 3D bir cisim verdiğimizde orthonormal.

\* perspective  $\rightarrow$  perspective ne kadar COP den uzak olursa, o kadar geniş bir alanı kapsar. Yani orthonormal bir set.



20.11.2019  
Çarşamba

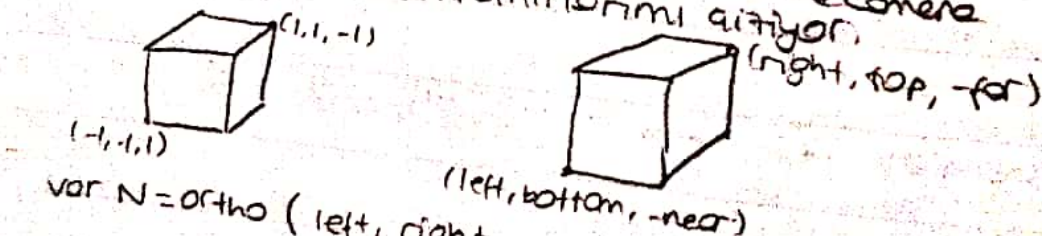
- Normalite  $\rightarrow$  1 br vektör oluşturmak için.
- $u, v, n \rightarrow x, y, z$  for the camera.
- Eski  $x, y, z$ 'den yeni (kameraya göre) olan koordinatlar  $(u, v, n)$  geçmek istiyoruz.
- $n = at-eye \Rightarrow$  kameranın baktığı doğrultuyu temsil eder.
- Normalite  $n$
- $u = \frac{up \times n}{|up \times n|}$   $v_1 \times v_2 \rightarrow$  ikisine de düz bir vektör elde ederiz.
- $v = \frac{n \times u}{|n \times u|} \Rightarrow$  correct up vector
- Genelde up vektörünü  $(0, 1, 0)$  veriyoruz. Tam olarak doğru değil.  
 $(0, -1, 0) \rightarrow$  Baş aşağı
- $\Rightarrow$  Bu yüzden up ve n arasındaki açı  $90^\circ$ 'den farklı olabilir.
- $v$  doğru up vektörüdür.
- Lookat, WebGL'de yok. rotasyonlar farklı olarak yazılır.

### Orthogonal Projection in WebGL

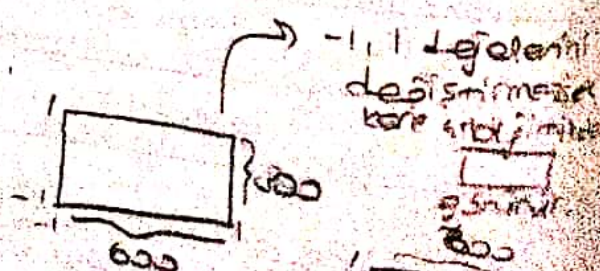
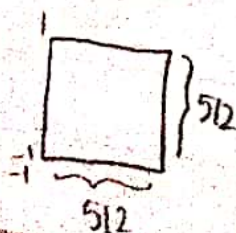
- The default projection.

```
var N = ortho (-1.0, 1.0,
               -1.0, 1.0,
               -1.0, 1.0);
```

- near, -far  $\Rightarrow$  this amount away from the camera
- Orthogonal yolların sınırlarını getiriyor.



```
var N = ortho (left, right,
               bottom, top,
               near, far);
```



- 1, 1 değerlerini  
değiştirmişsek  
kare artık birimle  
gösterilir.

\* frustum  
dört  
geniştir  
\* perspektive



COM337

### Stack-Based Traversal in Application Code

20.11.2019  
Garsamba

- figure'u çektirmeden MV matrix'i değiştirirsek figure'u değiştirmiş oluruz. → Nishenin tamamını kaldıracağız diye düşünüyor.

$M_0$   
drawTorso

$M = M_0 M_h$   
drawHead

→ önceki M'e ihtiyacımız olacak.

$M = M_0 \cdot M_{h0}$

↳ üst kolun Torso'ya göre nasıl konumlandırıldığı

- $M_0$  olunca ihtiyacımız kalmayacak. (rollback yapmayı)
- Stack kullanırız. (LIFO). Son depere tekerrür ulaşmamız gerektiği durumlarda stack kullanılır.
- Matris stack'li var. Buna matrislerin kopyalarını koyuyoruz.
- Transformations mixed with pushed and pop.

var modelViewMatrix = mat4(1); ⇒ Identity matrix  
var mvStack = [];

mvStack.push(modelViewMatrix) → copy and push it.

şey tarot en basta copy alınmalı.

↑ pop call

↓ önce push

COM477

20.11.2019  
Garsamba

RISK EVALUATION  
QUALITATIVE & QUANTITATIVE RISK ASSESSMENT  
METHODS



20.11.2019  
Görüşme

## HIERARCHICAL MODELING & CONSTRUCTIVE SOLID GEOMETRY

- Önce subpartları, sonra bu subpartların nasıl bir arada geldiğini tanımlama.  $\Rightarrow$  defining a complex object or system - easier way

### Basic Modeling Concepts

- model: Representation of a system.
- Generic model = model diyince yanında bunu kast ediyorsun.

### System Representation

- and  $\rightarrow$  asıl end point'in bir kopyası, = instance
- Class'tan objeler oluştururken instance oluşturmuş olursun.
- Solid modeling  $\rightarrow$  an object like chair vs. Not extremely flexible. Parçaların nasıl bir araya geldiğini tanımlamak için Procedure'lar (fonksiyon) kullanılır.

### Symbol Hierarchies

- Many models can be organized as a hierarchy of symbols.

### Hierarchical Models

1. which direction
  2. speed of movement
- Sarının hareketi tekerelerin hareketinden ayrılmalı. Bu yüzden bu şekli çok tercih etmeyiz.
  - Bunu kullanmak daha kolay ve daha kolay, aynı olmaları kullanırsak orantı kolunu üst kolundan aynı hedeflere ulaşma hareketlerinde hatalar yapabiliriz.
  - Hepsi Torso'ya bağlıdır. Torso'yu hareket ettirince bacak da hareket edecektir.
  - Veri yapıları kullanarak ve fonksiyonları organize ederiz. Önce hayali yapıyı oluşturuyoruz.

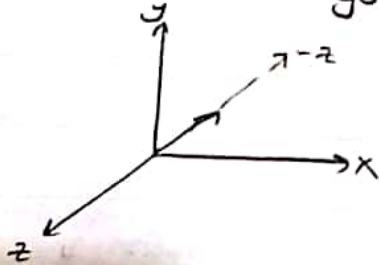


COM337

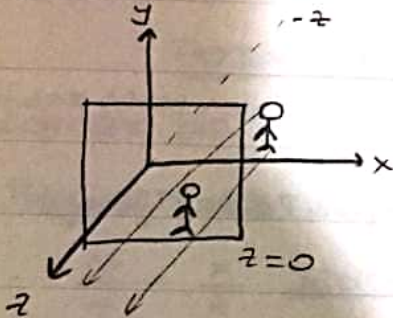
## Arranging the camera

20.11.2019  
Garsamba

- Default olarak orthogonal - parallel.
- kameranın  $-z$  yönüne bakıyor.



- Aslında kamerayı nereye koyduğumuz önemli değil. Çünkü perspektifteki gibi bir noktada kesilmiyor.



- kamerayı cisimden uzaklaştırmak  
cisim kameradan uzaklaştırmak
- MV matrix'i değiştireceğiz. İstedığımız transformationlar için  
eye ve at'i belirledikten  
sonra rotate edebiliriz.
- lookat (eye, at, up)

↓  
- where it's looking at  
- where the camera is located  
- kameranın durduğu yer

↓  
- kameranın hangi yöne bakacağı  
- looking towards the at position  
- hangi doğrultuda bakılıyor?  
- Bakışın bir noktası.

- lookat'i belirlemek için kameranın üstü.

- eye for translation
- at & up → how much we rotate