

Sahne Yönetimi

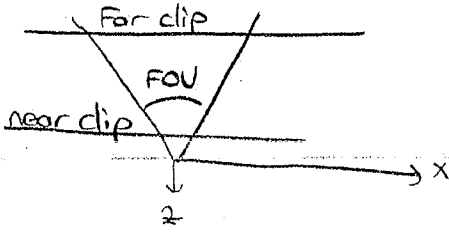
Oyun programlarının temel aşamalarından biridir. Bulunulan sahnedeki karakterlerin modellenmesi, detaylarının verilmesi ve bunların hafızayı yönetim teknikleriyle gerektiği şekilde yüklenmesidir.

- 1) Sahnede gereksiz olan nesne ve karakterlerin o anki harekete göre çıkarılması ilk işlemdir.

### Arkaplan çıkarma (Backface Culling)

Kamera oasına girmeyen nesnelerin çıkarılması işlemidir. Normalde (rendering derinlik katanırma işlemi yavaş bir işlemdir. Bunun üzerine gerçekleştirilecek çıkarma işlemi de bu işlemi yavaşlatabilir.

### Kamera ve Hacim Görüş Açıkları (Camera & View Volumes)



Bilgisayar ortamında nesnenin modellenmesi kamera pozisyonu ve görüş mesafesiyle ilgilidir. 2 boyutta 2gen, 3 boyutta 2gen piram yapıyla yapılır.

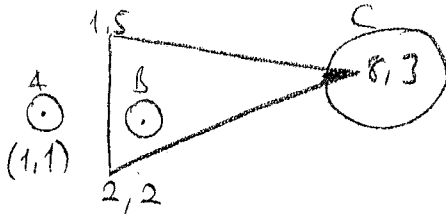
— Bir nesnenin bilgisayar ortamında görülmesinde 3 durum vardır.

- 1) Tamamen görülebilir olanlar,
- 2) Kısmen görülebilir olanlar,
- 3) Tamamen görülmeyen yapılar.

\* Kesit olma işlemi yarı görülebilir nesneler içindir.

— Bir nesnenin alan içinde olup-olmadığı problemi ?

Özet: 2 boyutta bir nesnenin kapalı alan içinde olup-olmadığını bulan program.



Bir nokta diğerinin içinde midir?  
İçindeyse ne kadar içindedir?

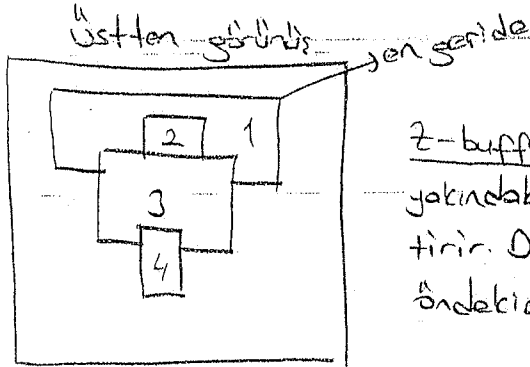
## Poltals (Ortamlar)

1'den fazla sahnesi olan oyunlardaki ortamların oluşturulması aşama aşama gerçekleştirilir. Bir ortamdan bir yada birden fazla ortama geçiş yapılabilir. Bu geçişlerin ortaya koyma tekniğinde graflar (Cizgiler) kullanılır.

## Görülebilir Nesneler ve Kümeler

Bir nesnenin görülebilmesi kamera pozisyonu ile ilgilidir. Nesnenin ne kadarının görüleceğinin ortaya konduğu algoritmalarla bir tane z-buffer algoritmasıdır. Derinliğe göre işlem yapma algoritmasıdır.

(Geriden - öne doğru) Görülebilir yer öndeki nesnelerdir. (Birçok nesneler için)



z-buffer, en gerideki nesneden itibaren en yakındaki nesneye doğru modelleme gerçekleştirir. Dolayısıyla en fazla götüken nesne en öndedir.

## Seviye Detay Kümeleri

Uzaktaki nesnenin detayları oluşturulmaz. Nesneye yaklaştıkça detaylar verilir. İlerleyen - ırgara (mesh) tekniği (Progressive Meshes) kullanılarak detay oluşturulabilir. Bunun için; <sup>işgören</sup>

a) Matematiksel işlemler yapılır. Alt bölmeleme (Subdi-ision) tekniği denir. <sup>işgören</sup> nokta sayısını artırarak yapılır.



b) Alt bölmeleme ile oluşturulmuş objeye yapıları kullanılır.

Data sayısı çok fazla olduğunda; geliştirilmiş, iyileştirilmiş alt bölmeleme algoritmalarına başvurulabilir.

## Çok Çözünürlüklü Modelleme (Multiresolution modelling) (2. yöntem)

21 Şubat

Bu yöntemin temeli uzakteki nesnelerin detaylarını gizlemektir. Örneğin, bir dağ yapısı (terrain), oval, vadi gibi doğal yapıların modellenmesinde yaklaştıkça detayların artırılma tekniğidir.

### NOT: Önemli!

- 1) Sahnedeki önemli bir kavram gölgelerdir. Gelişmiş oyun programlarının kullandığı gölgeleme tekniği "Phong" gölgelemesidir.
- 2) Ortam haritalarının belli bir sırayla sahneye yerleştirilmesi.
- 3) Görüntü işleme teknikleri yardımıyla da görsel efektlerin, diğer efektlerin sahneye dahil edilmesidir.

### NOT Realflow programı

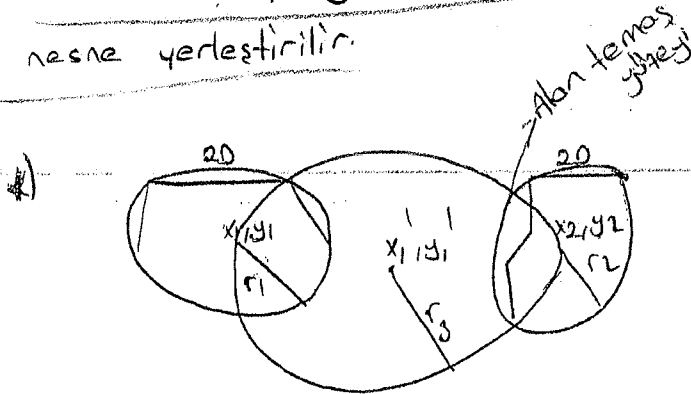


## ? Çarpışma Tespit Algoritmaları ?

Tanım: 2 boyutlu ya da 3 boyutlu ortamda 2 ya da daha fazla nesnenin temas edip, etmediğini bulma işlemidir.

- Oyun motorları içinde "collider" yazılımları ile bu işlem gerçekleştirilebilir.
- Karmasık nesnelerin temas nokta ve yüzeylerinin bulunması problemidir.

1) - 2 boyutlu uzayda temas edip, etmediğini bulmak için nesnelerin tamamını kapsayacak şekilde bilinen bir geometrik şekil içine nesne yerleştirilir.



B sabit, A hareketli.

HAREKET (HIZ, YÖNE)

$t=0$  ,  $x_1, y_1$

$t=1$  ,  $x_1 + \Delta x, y_1 + \Delta y$

- Yörünge kontrolü (nesneler birbirine değişir mi?, değişmiyor mu?)

(+) Avantajı: Hız meselesidir.

(-) Dezavantajı: Gereksiz alanı işgal ettiği için gereksiz temas noktaları içerir.

→ A nesnesi ile B nesnesinin kesişiminin bulunması corpisma tespiti'dir.

→ Corpismanın tespiti hız ve ivme ile ilgilidir.

→ Yapılan işlem, nesnenin tamamını kapsayacak şekilde, basit geometrik şekil içine yerleştirilebilir.

NOT) Nesnenin hızı yada yavaş olması  $\Delta_x$ ,  $\Delta_y$  'yi etkiler.

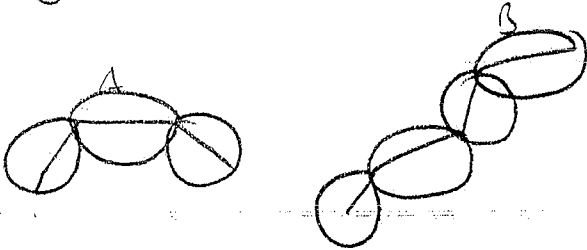
NOT) Çemberdeki sıkıntı gereksiz olanların nesneye dahil edilmesidir. Avantajı hızlı işlem sağlar. (Dikdörtgen içine de alınabilir.)  
Bu da 2 olabilir!

\*\*\*③

a) → parça, parça nesneyi tanımlayan geometrik şekiller kullanmak.

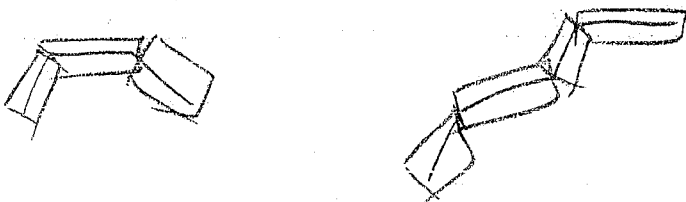
(+) Avantajı: Gereksiz olanlar atılır.

(-) Dezavantajı: Birden fazla çemberin olması ve kontrol edilecek değerlerin sayısının artması.



b) Herbir parçayı dikdörtgen şeklinde göstermek.

Bu yöntem en çok kullanılan yöntemdir. Özellikle oyunlarda ve animasyonlarda en fazla kullanılan yöntemdir.



## 4) IDEALİ:

- ideal, gerçek datanın kendisiyle doğrulan çalışmaktır.
- Fakat, dataların doğruluğundan dolayı utun zaman alabilen durumlar ortaya çıkar.
- 3 boyutlu nesneler polinomlar şeklinde modellenmiş durumda özellikle eğrisel bölgelerde nokta ve yüzey sayısının doğruluğu sıkıntılara yol açar.
- Nesnenin yer değişim miktarının bulunması ve bunun modellenmesi kritiktir.
- +0 anında (ilk pozisyonda), nesnenin durumu kaydedilir.
- +1 anında nesnenin yeni pozisyon bilgisi tutularak işlem yapılır.

## Nesnenin Köşe Koordinatlarının Normalinin Bulunması?

- 2 vektörün vektörel çarpımı bu 2 vektörün oluşturduğu düzleme dik olan vektörü verir. Buna "yüzey normali" denir.

(Vektörel çarpım)  $\vec{A} \times \vec{B} \neq \vec{B} \times \vec{A}$

$\left( \begin{array}{c} \text{Dış} \\ \text{doğru} \\ \text{normal} \\ \text{vektör} \end{array} \right) \quad \left( \begin{array}{c} \text{İçe doğru} \\ \text{normal} \\ \text{vektör} \end{array} \right)$

- Normal vektörleri nesnenin hareket yönünü belirlemede kullanılır.

- 2 vektör arasındaki açı "cosinüs teoremi" ile bulunur. (Skaler çarpım)

$A \cdot B = |A| \cdot |B| \cdot \cos \alpha$

$\cos \alpha = \frac{A \cdot B}{|A| \cdot |B|} = \frac{A_x \cdot B_x + A_y \cdot B_y + A_z \cdot B_z}{\sqrt{A_x^2 + A_y^2 + A_z^2} \cdot \sqrt{B_x^2 + B_y^2 + B_z^2}}$

## Çılgın - Çılgın Kesisi

- Normalde 2 boyutlu ya da 3 boyutlu nesnelerin modellenmesinde yüzey şekli olarak ve yüzey parçası olarak Çılgın ya da dörtgenler kullanılır.
- Özellikle eğrisel yüzeylerde Çılgın yapısını tercih etmek faydalıdır.
- Düz olan bölgelerde Çılgın boyutunu küçültüp, sayıyı arttırmak faydalıdır.

## Çılgın Hareketli Parçalar

2 durum söz konusudur.

→ 1.) Nesnelerden birinin hareket etmesi, diğerinin sabit durması.

\* Bu durumda, hareket eden parçaya göre işlem yapılır.

\* Bu durumda problem; Çılgın, kare ve dörtgen kesisi problemidir.

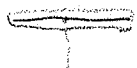
→ 2.) Her iki nesnenin birbirine doğru hareket etmesi problemidir.

\* Hızları eşit ya da farklı olabilir.

## Optimizasyon Stratejileri

NOTS Optimizasyon Stratejilerinin Amacı: Uygun bir veri yapısı kullanarak hızlı bir şekilde işlem yapmaktır. Bunun için çeşitli ağaç yapıları geliştirilmiştir.

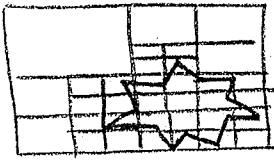
900





## 1) Oc-trees:

- Amacı: Nesne sınırlarını takip edecek şekilde mümkün olduğu kadar küçük dörtgen parçaları şeklinde modellemektir.
- Böylece, sınırdaki temsil edilebilen nokta sayısı azaltılmış olur.
- Nesne sınırlarını takip edecek şekilde büyük bir karenin parçalarına bölünerek sınırlara yerleştirilmesidir.

2) KD Trees: Amacı: Nesne sınırlarını takip edecek şekilde veriyi azaltmaktır.

## 3) BSP Trees:

Amacı: Bu ağaçların amacı, nesnenin alanını basit üçgenlerle tanımlamaktır.

- Çalışma şekli; Nesne kenarındaki çizgilerin birbiriyle kesişecek şekilde düzlem oluşturmalarıdır. Böylece nesne parçaları şeklinde modellenir.

## 4) K - Dops:

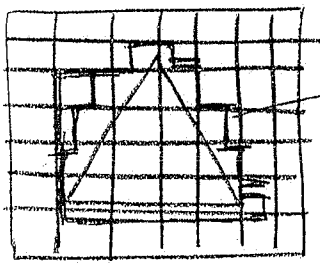
Amacı: Nesne köşelerini kapsayacak şekilde çokgen yapısı oluşturmaktır.

- Böylece, nesnenin kendisi basit bir çokgen olarak tespit edilir.
- Fazla olanlar bu algorithma da oluşabilir.

### 5) Uniform Grids (Yayılı İzgara) Kullanmak

Amaç Nesnenin tamamını kapsayacak şekilde eşit aralıklı ve eşit boyutlu izgara kullanmaktır.

→ Grid boyutu ne kadar küçük olursa o kadar düşük nesne sınırı elde edilmiş olur.



→ Bu şekilde görür nesneyi

### 6) Nokta Gıptı Azaltma İşlemi?

Görüşme tespitiinde dikkat edilecek diğer bir işlem veriyi azaltmaktır.

Amaç; Hızlı bir şekilde işlem yapmaktır.

Fizik Simülasyonu

Gerçek dünyadaki değerlerin fiziksel etkileri bilgisayar ortamına cesitli değişkenler ve fonksiyonlar olarak aktarıldığı yapılar "simülasyonlar (benzetimler)" dir.

Kinematik: Herhangi bir nesneye uygulanan kuvvet gösteril edilerek, bu nesnedeki hareket çalışmalarını bilimidir.

Dinamik: Fiziksel hareket çalışma disiplini.

1) Forward (İleri) Dinamik: Uygulanan kuvvetin sonucu olarak hareketin hesaplanmasıdır. (Hedef yok, sadece kuvvet var)

2) Inverse (Ters) Dinamik: Gerçekli hareketin sağlanabilmesi için uygulanması gereken kuvvet miktarının hesaplanması disiplini. Özellikle simülasyon ve animasyon işlemlerinde daha çok kullanılır.

Paracocıklar: Büyük boyutlu nesnelerle işlem yapmak matematiksel açıdan sıkıntılıdır. Bunun yerine 2 ve 3 boyutlu uzayda daha küçük parçalı nesnelerle çalışılır.

Yaygın olarak 2 boyutta cember, 3 boyutta küre yapısı kullanılır.

Bunun nedeni, orijin noktası ve yarıçapının bilinmesinin yeterli olmasıdır.

→ Her bir parçacığın 3 değeri vardır.

→ Potansiyon:  $x$

→ Hız:  $v = dx/dt$

→ İvme:  $a = dv/dt$

→ Hızlanma Değeri:  $a = a_0$

$$v = \int a \cdot dt = a_0 \cdot t + v_0$$

$$x = \int v \cdot dt = \frac{1}{2} \cdot a_0 \cdot t^2 + v_0 \cdot t + x_0$$

NOT  $v_0 = 0$  ise parçacık,  $(x_0)$  başlangıç noktasındadır.

→ Kütle:  $m$

→ Momentum:  $p = m \cdot v$

→ Kuvvet:  $f = dp/dt = m \cdot (dv/dt) = m \cdot a$

$$\boxed{f = m \cdot a}$$

→ YERÇEKİMİ KUVVETİ ?

→  $\boxed{f = m \cdot g} \rightarrow g = -9.8 \text{ m/s}^2$  olarak kabul edilir.

ama Dünyanın her yerinde yerçekimi kuvveti aynı değildir.

Kutuplarda en fazla, Ekvator'da en azdır.

→ Sürdürmenin gözetildiği ortamlarda yerçekimi kuvveti uygulamalara dahil edilmelidir.

Kuvvet Gözetleri

1) Yay (Sönümleyici) Kuvveti: Aşağı - yukarı, sağa - sola hareket eden yayların hareketi ve simülasyonu ile ilgilidir.

Gerçekimsiz ortamda yayda sadece belli katsayılarla göre basit hareket beklenir. Bu katsayı k'sabitidir.

2) Sürtünme Kuvveti: Hareketi engelleyen kuvvettir.

Statik ve Dinamik türleri vardır.

Gerçekçi olan dinamik sürtünme kuvvetidir.

Özellikle çarpışma ve sonrasında nesnelerin hareketinin modellenmesinde kritiktir.

NOT) Oyun programlamada çarpışma ortamlarını, kuvvetlerin vektör ya da impuls (birim darbe) şeklinde olduğu kabul edilir.

Bu darbelerin belli zaman aralıklarıyla nesneye etki ettiği düşünülür.

NOT) Kuvvetin sürekliliği nesne üzerindeki etkisi, momentumu etkiler.

## KATI NESNELER

→ Aalet Meselesi :

- ↳ Nesnelerin fiziksel yapısından kaynaklanan iç enerjiye "alet" denir.
- ↳ Nesnenin kendi iç dinamiklerinden kaynaklanan bir tür enerjisidir.
- ↳ Aalet, nesnenin başlangıç enerjisine, başlangıçta uygulanan kuvvete bağlıdır. Ve bu durum nesnenin hem acısal hem de düzlemsel momentumunu etkiler.
- Özellikle yuvarlanan nesneler için önemlidir, değerlidir.

## Katı Nesnelerin Garpisması

2 türü vardır: 1) sız konusudur.

1.) 2+ yönlere hareket.

2.) Kırılmalar.

\* Nesneyi farklı boyutlarda küçük parçalara bölerek modellemek esastır. Böylece kırılma anında her bir parçaya etki edecek kuvvete göre bir dağılım, yere sacılma işlemi olmaktadır.