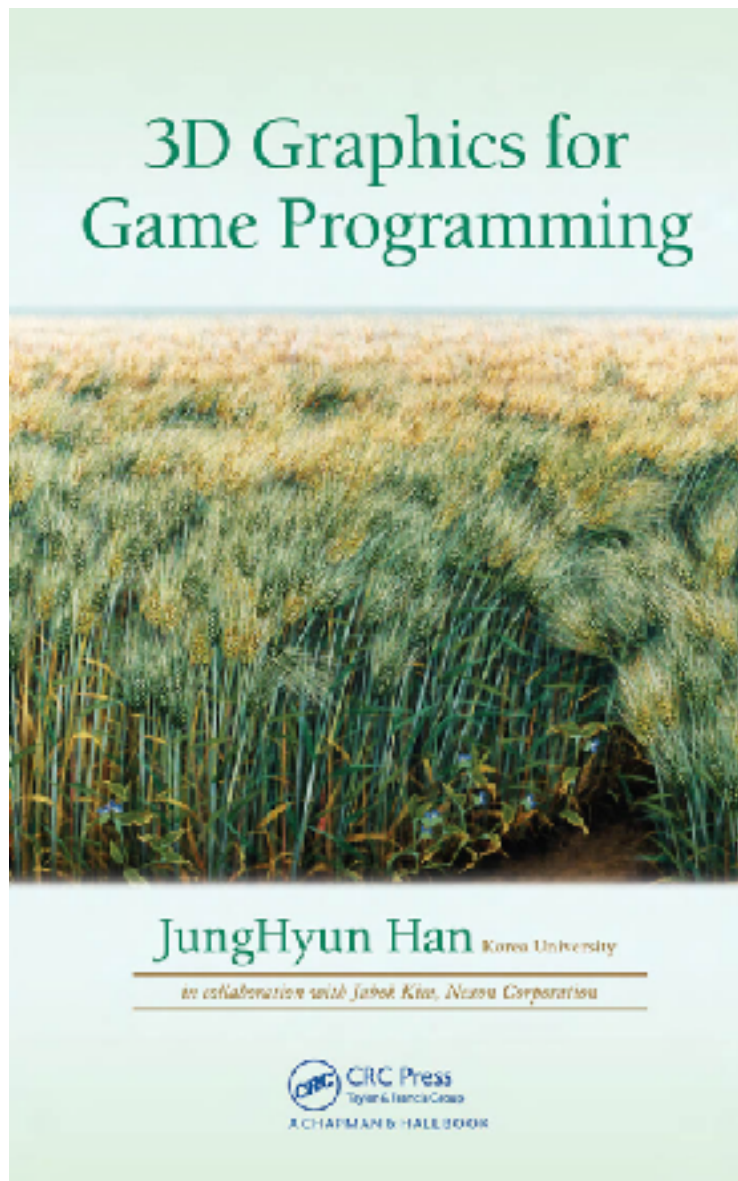


BCA611 Video Oyunları için 3B Grafik

Ders 2

Zümra Kavafoğlu
<https://zumrakavafoglu.github.io/>

Kaynaklar



- JungHyun Han, 3D Graphics for Game Programming, CRC Press, ISBN-13: 978-1-4398-2738-3

<http://media.korea.ac.kr/book>

- OpenGL Red Book
<http://www.glprogramming.com/red/>

- WebGL
<http://learningwebgl.com>

Oyun Üretimi (Grafik)



Oyun üretiminin 3 ana bileşeni

Modelleme ve çevrimdışı(offline) animasyon: Grafik sanatçıları

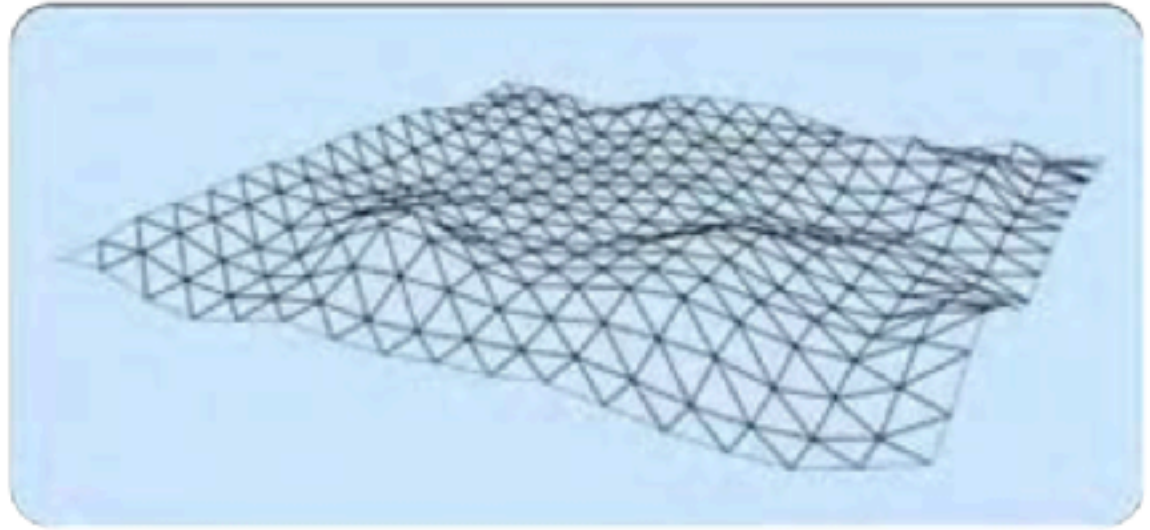
Run-time Animasyon ve Render: Yazılımcılar

Modelleme



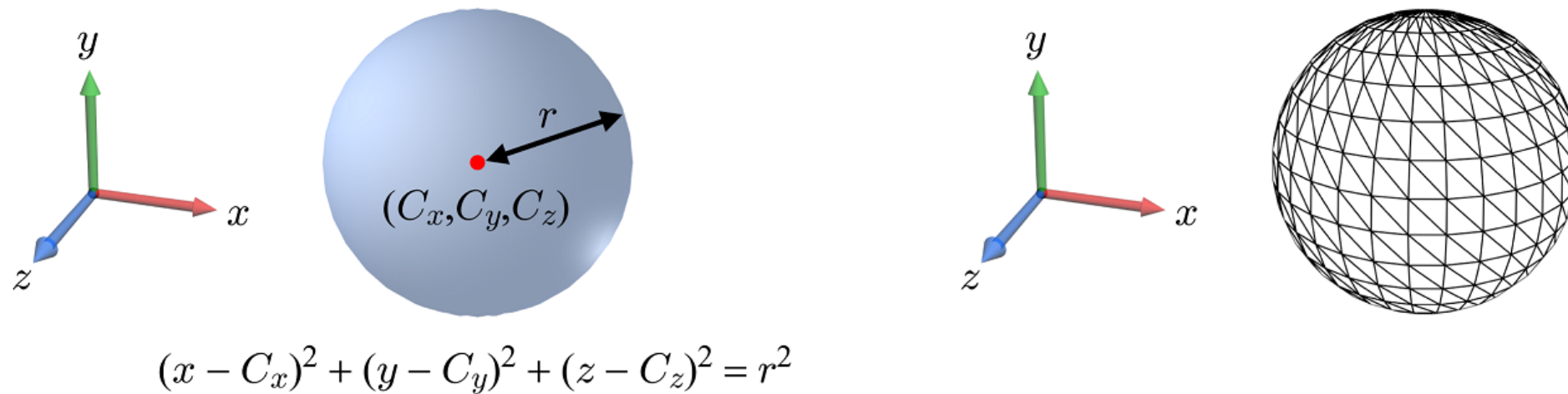
GTA V - Rockstar Games

Modelleme - Poligonal Mesh



Modelleme için en çok kullanılan yöntem nesnenin ya da karakterin yüzeyini poligonlarla modellemektir ve bu modele poligonal mesh denir.

Poligonal Mesh

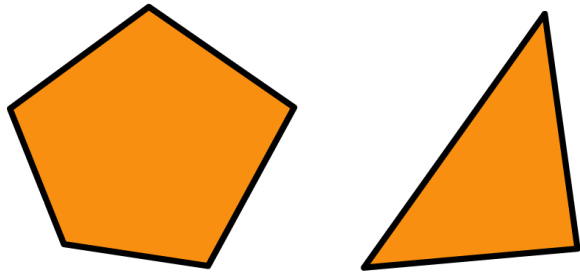


Poligonal mesh gösterimi, kapalı eşitlikle belirlenen yüzeyin birebir aynısı değildir. Mesh köşeleri smooth yüzeyin bir yaklaşımını oluşturur.

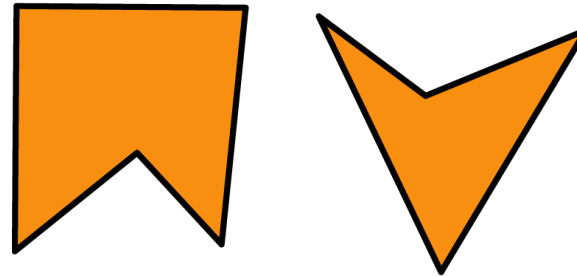
GPU 'lar poligonal mesh 'leri işleyebilmek için optimize edilmişlerdir, bu sebeple gerçek-zamanlı uygulamalarda poligonal mesh tercih edilir.

Poligonal Mesh

OpenGL keyfi sayıda köşelerin sayısına sahip olan poligonları destekler, burada koşul poligonun dışbükey ve düzlemsel olmasıdır.



convex polygons



concave polygons

Direct3D yalnızca üçgenleri destekler. En basit poligon üçgendir ve her zaman konveks(dışbükey) ve düzlemseldir.

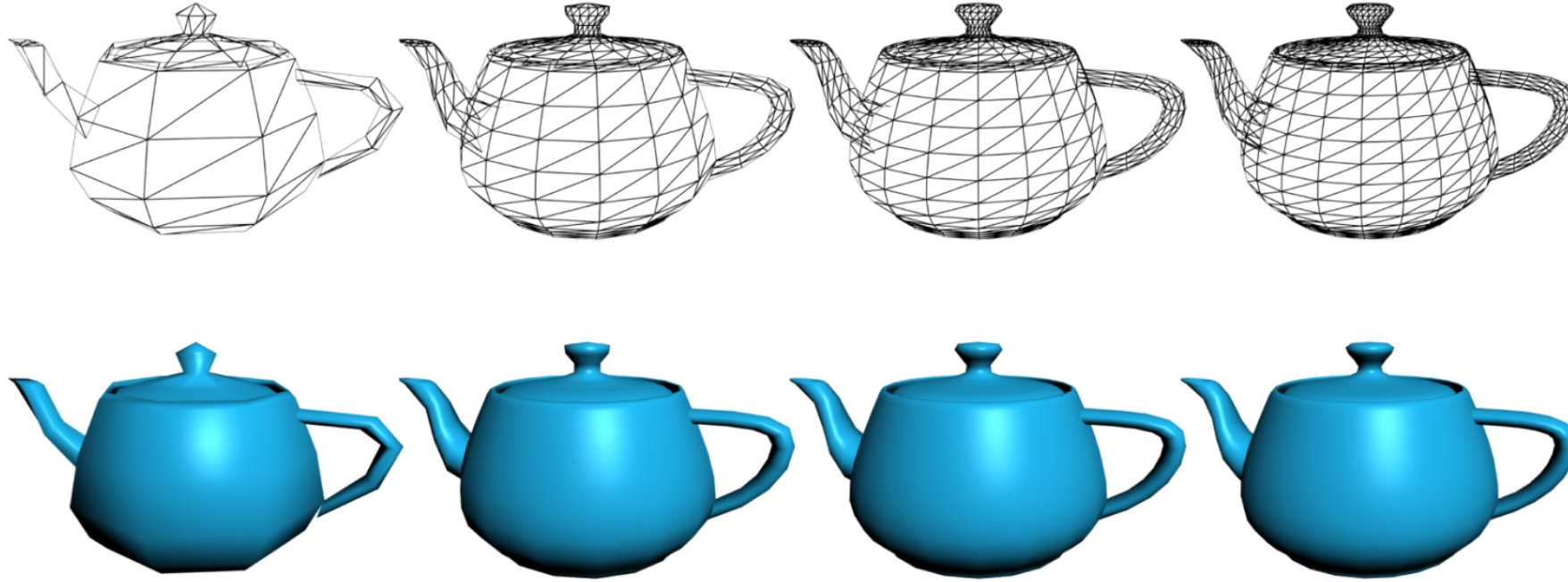


Üçgen Mesh



Dikdörtgen Mesh

Poligonal Mesh



düzeltme(refinement)

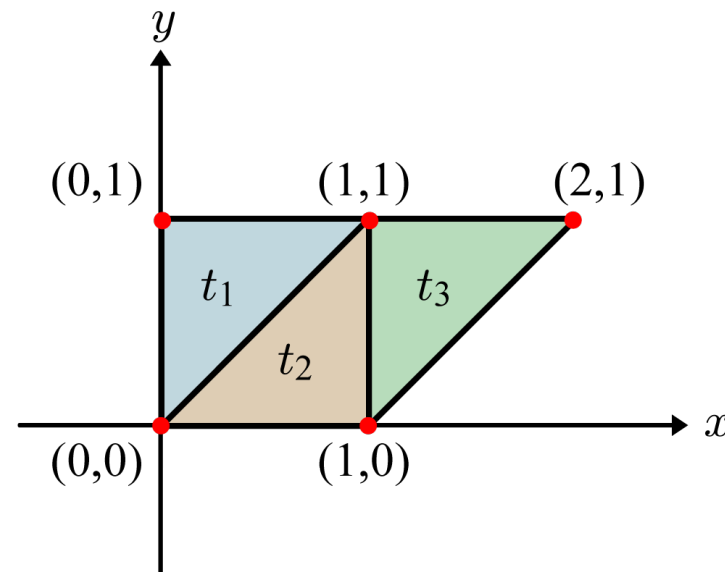
sadeleştirme(simplification)

Doğruluk ve verim arasındaki değiş tokuş: Çözünürlük(üçgen sayısı) arttıkça, mesh orijinal yüzeye daha çok yaklaşır ancak mesh'i işlemek için gerekli süre artar.

Poligonal Mesh Gösterimi - İndekssiz Üçgen Listesi

İndekssiz üçgen listesinde:

- Her bir üçgenin köşeleri doğrusal sırada numaralanır ve köşe belleği (vertex buffer, vertex array(opengl)) adı verilen bellek alanında sırayla saklanır.
- Bu gösterime “üçgen listesi” denir.

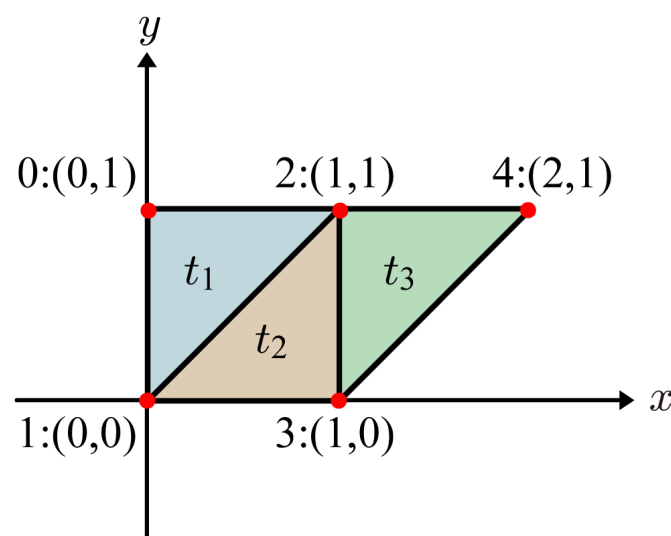


$(0,1)$	t_1
$(0,0)$	
$(1,1)$	
$(0,0)$	t_2
$(1,0)$	
$(1,1)$	
$(1,1)$	t_3
$(1,0)$	
$(2,1)$	

- Bu gösterim verimli değildir çünkü köşe belleği gereksiz veri tutar. Örneğin $(1,1)$ köşesi üç üçgenin ortak köşesidir ve köşe belleğinde üç farklı yerde bulunmaktadır.

Poligonal Mesh Gösterimi - İndeksli Üçgen Listesi

- Üçgensel bir meshte neredeyse her köşe birden çok üçgenin ortak köşesidir. Bu durumda ayrı bir indeks belleği(index buffer) kullanılarak köşe belleği daha kompakt hala getirilebilir. Bu gösterime indeksli üçgen listesi denir.
- Bu gösterimde
 - Her bir köşe, köşe belleğinde yalnızca bir defa bulunur.
 - İndeks belleğinde her bir üçgen için üç indeks depolanır.



	vertex buffer	index buffer
0	(0,1)	0
1	(0,0)	1
2	(1,1)	2
3	(1,0)	1
4	(2,1)	3
		2
		2
		3
		4

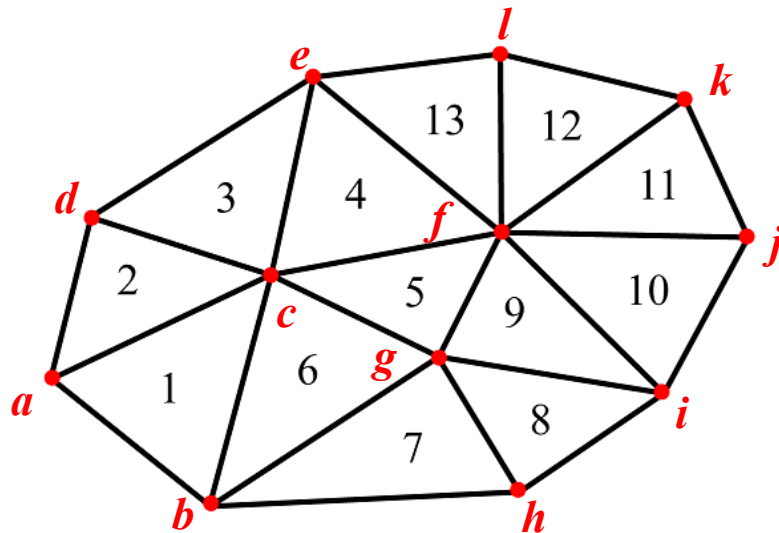
Red dashed lines group the index buffer entries for each triangle: t_1 (indices 0, 1, 2), t_2 (indices 1, 3, 2), and t_3 (indices 2, 3, 4).

Poligonal Mesh Gösterimi - İndeksli Üçgen Listesi

- Köşe belleğinde tutulan köşe verisi yalnızca köşe pozisyonu değildir, normaller, doku koordinatları gibi pek çok başka veriyi de içerir. Dolayısıyla köşe belleğindeki gereksiz veriden kurtulmakla kazanılan depolama alanı, indeks belleği için gerekli olan alandan daha fazladır.
- İndeksli üçgen listesi kullanmak yalnızca depolama bakımından değil, hız bakımından da daha verimlidir. Üçgensel bir mesh render için GPU'ya gönderilir. Her köşe GPU'nun köşe işleme aşaması tarafından dönüştürülür ve bir **dönüşüm-sonrası önbellekte(post-transform cache)** saklanır. Eğer bir üçgen dönüşüm sonrası bellekteki bir köşeye sahipse, o köşe doğrudan önbellekten çekilir, tekrar dönüştürülmesine gerek kalmaz.

Poligonal Mesh Gösterimi - ACMR

- Üçgen başına işlenen ortalama köşe sayısına average cache miss ratio(ACMR) denir ve render performansını ölçmek için kullanılır. ACMR ne kadar küçükse, render o kadar verimlidir.
- Ancak önbelleklerin kapasitesi sınırlıdır. Dolayısıyla üçgenlerin çizdirilme sırası önemlidir.
- Düşük kapasiteli bir önbellek için (örneğin 4), iki sıralamanın karşılaştırmasını yapalım.

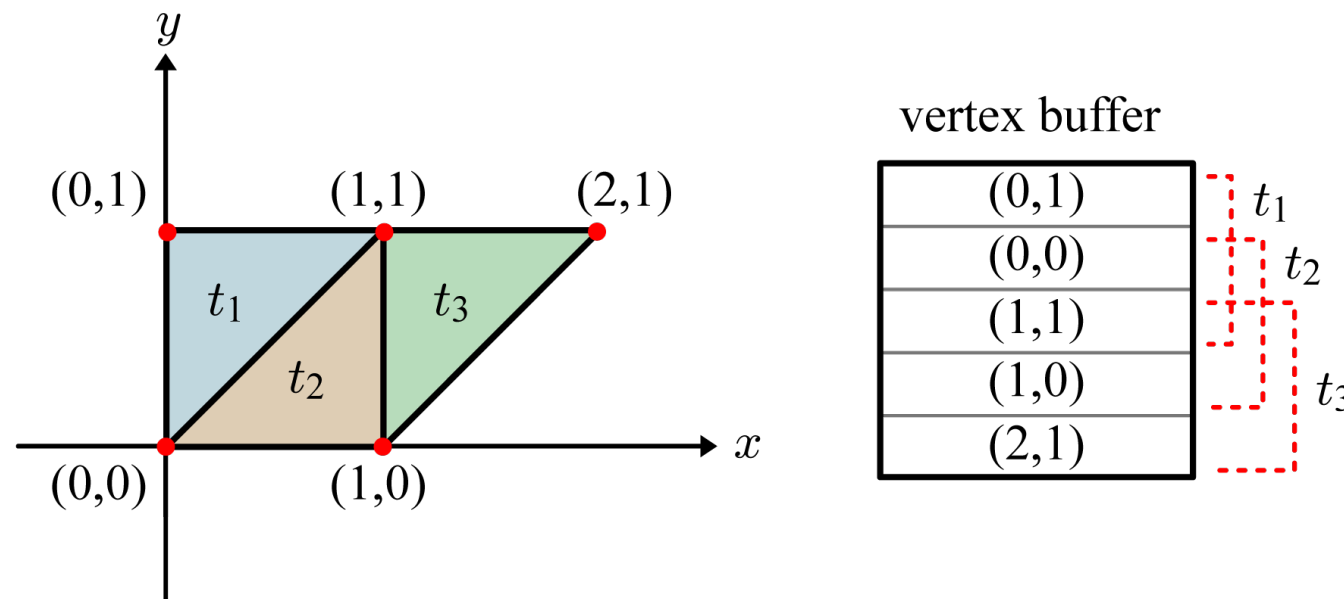


ordering with high locality: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \dots$

ordering with low locality: $1 \rightarrow 8 \rightarrow 11 \rightarrow 3 \rightarrow 10 \rightarrow 13 \dots$

Poligonal Mesh Gösterimi - Üçgen Şeridi(Triangle Strip)

- Her üçgen kendisinden önceki üçgenle iki köşe paylaşır (ilk üçgen hariç)



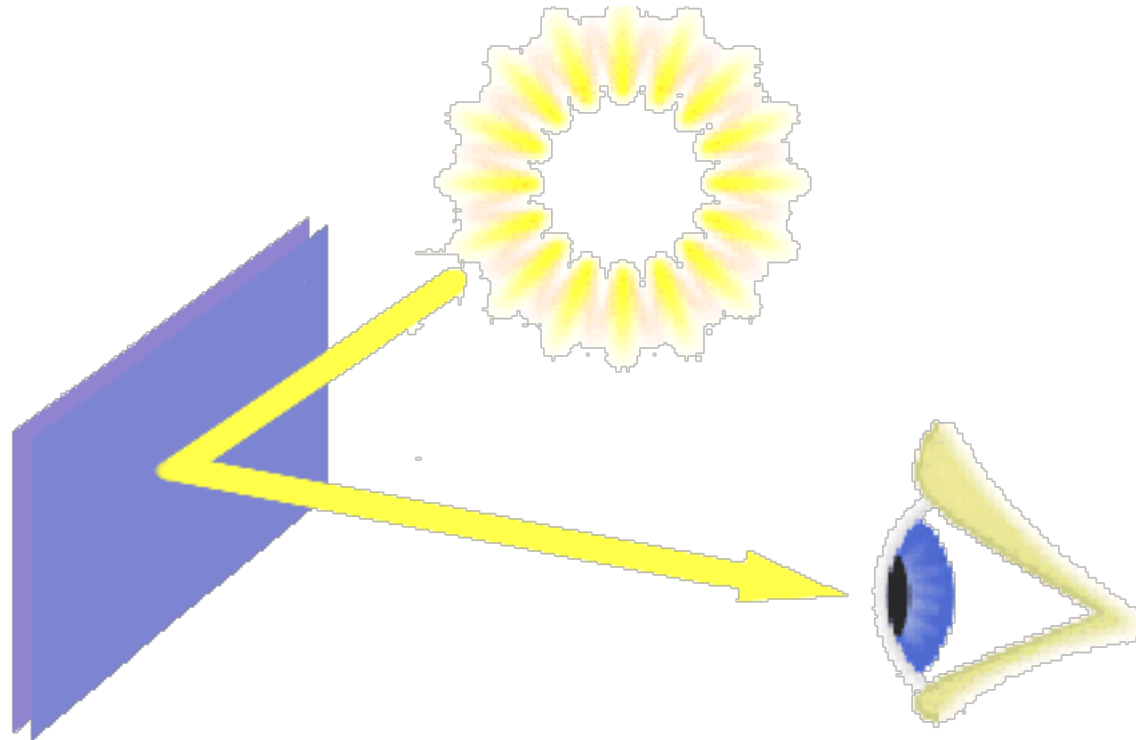
- Köşe belleğinden ilk üç köşenin işlenmesiyle ilk üçgen(t_1) render edilir. Bu üç köşe cache yerleşir. İlk üçgenin son iki köşesi cache 'den alınır (burada (0,0) ve (1,1) köşeleri) ve sadece tek bir köşe köşe belleğinden cache içine çekilir (burada (1,0) köşesi). Benzer şekilde cache 'den ikinci üçgenin son iki köşesi alınır (burada (1,1) ve (1,0) köşeleri) ve üçüncü köşe köşe belleğinden çekilir (burada (2,1) köşesi) .

Poligonal Mesh Gösterimi - Üçgen Şeridi(Triangle Strip)

- Üçgen şeridin render performansını inceleyelim.
- İdeal durumda ilk üçgen için 3 köşe işlenir ve kalan her bir üçgen için bir köşe daha işlenir. O halde n üçgen render etmek için $n+2$ köşe işlenmelidir. Bu durumda $ACMR = (n+2)/n$ olur. n arttıkça $ACMR$ 1'e yakınsar.

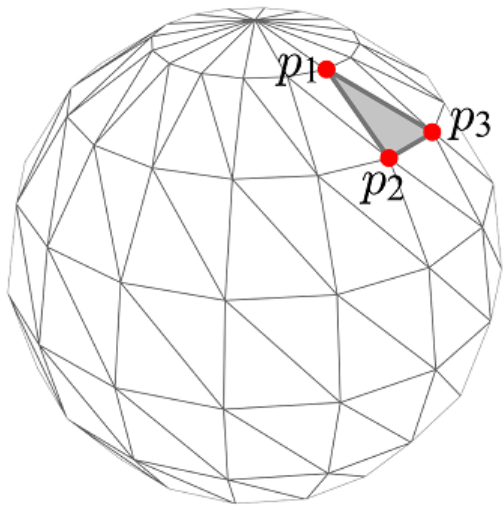
Yüzey Normali

- Gerçek dünyada bir ışık kaynağından yayılan ışık nesne yüzeyine gelir ve nesne yüzeyinden tekrar yansıyarak göze geri döner.
- Işık kaynağıyla nesnelerin yüzeyi arasındaki bu etkileşim bilgisayar grafiğinde taklit edilir.
- Bu benzetime aydınlatma denir ve aydınlatmada yüzey normallerinin önemli bir rolü vardır.

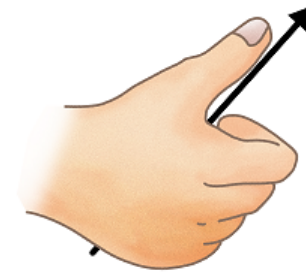
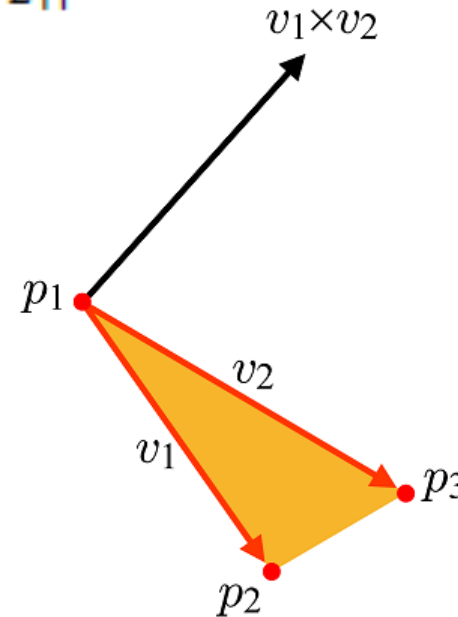


Yüzey Normali

- (p_1, p_2, p_3) üçgenini ele alalım.
- Bu üçgenin normali, üçgenin kenarların birer vektör olarak alınması ve sonra bunların vektörel çarpımlarının alınmasıyla elde edilir.
- p_1 köşesiyle p_2 köşesine bağlayan kenarı v_1 vektörüyle ve p_1 köşesiyle p_3 köşesine bağlayan kenarı v_2 vektörüyle gösterirsek, bu durumda üçgenin normal vektörü

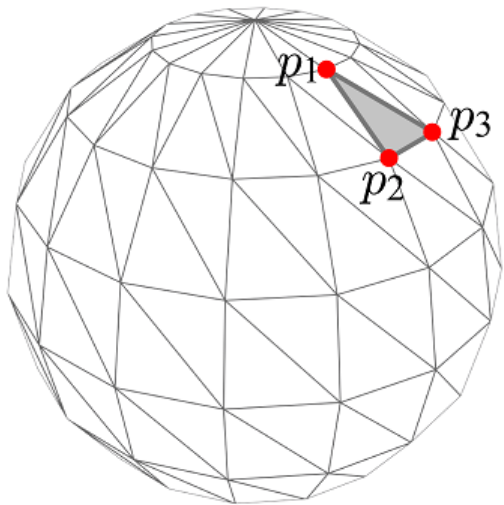


$$\frac{v_1 \times v_2}{\|v_1 \times v_2\|}$$

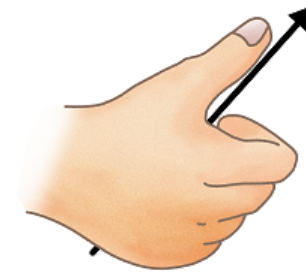
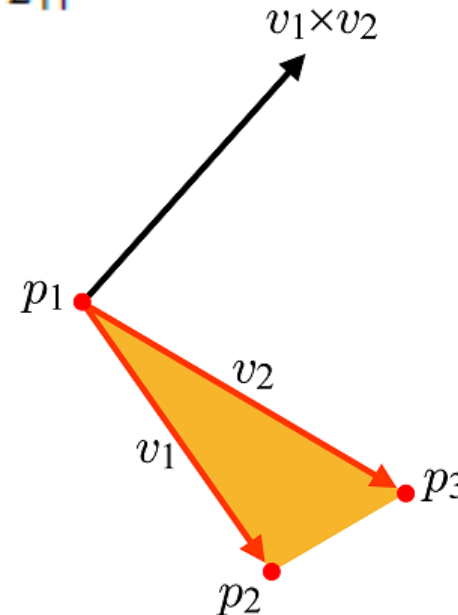


Yüzey Normali

- $v_1 = (x_1 \ y_1 \ z_1)$ ve $v_2 = (x_2 \ y_2 \ z_2)$ ise $v_1 \times v_2 = (y_1 z_2 - z_1 y_2 \ z_1 x_2 - x_1 z_2 \ x_1 y_2 - y_1 x_2)$
- Her normal vektör birim vektör olmalıdır.
- Bir üçgenin birbirlerinin tam tersi yönlerde doğru yönlenmiş iki farklı normal vektörü vardır.

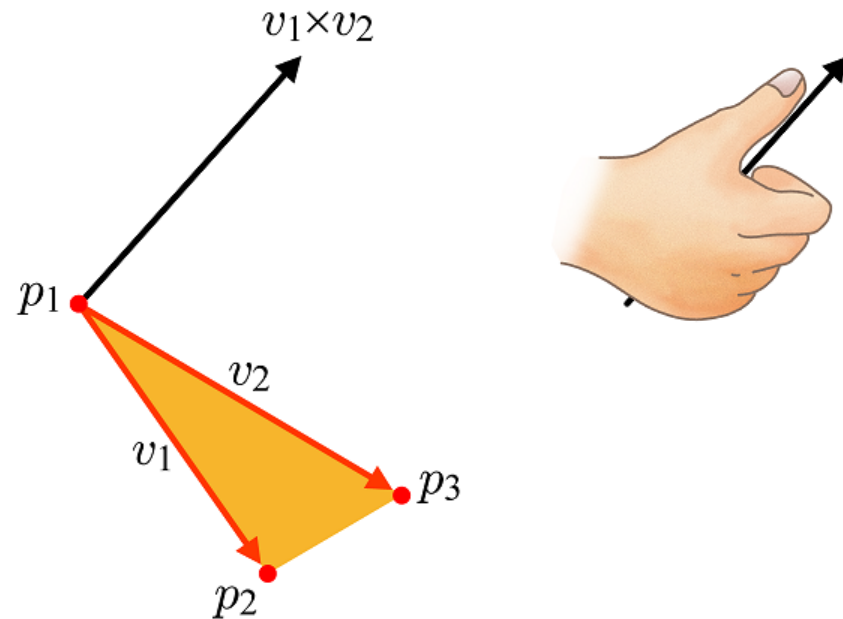
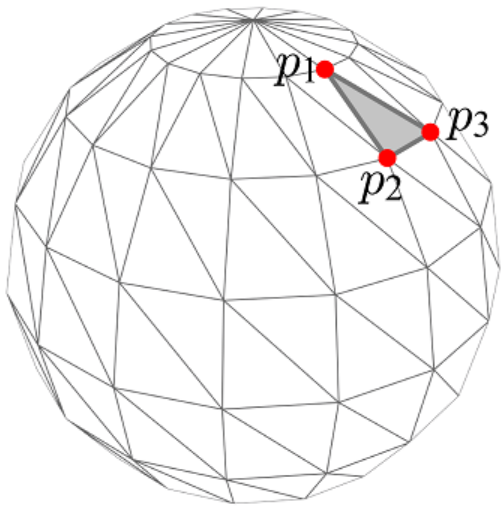


$$\frac{v_1 \times v_2}{\|v_1 \times v_2\|}$$



Yüzey Normali

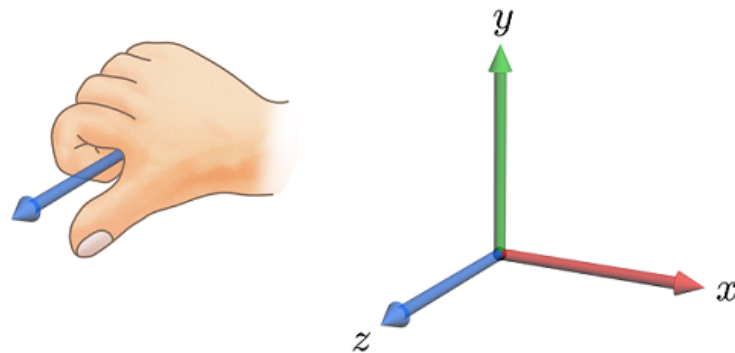
- Normal vektörünün yönü üçgenin köşe sırasına bağlıdır.
- Şekilde gösterilen (p_1, p_2, p_3) sırası saat-yönünün tersi yönde sırayı (CCW) gösterirken, köşelerin (p_1, p_3, p_2) şeklindeki sırası saat yönünde (CW) sırayı gösterir.
- Kabul edilen normal vektörün yüzeyin dış tarafını göstermesidir.



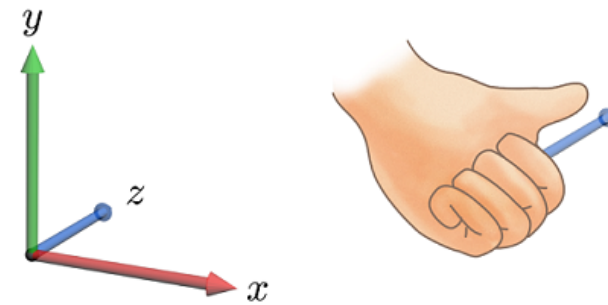
Koordinat Sistemleri ve Köşe Sıralama

Sağ el sistemi(RHS) VS. Sol el sistemi (LHS)

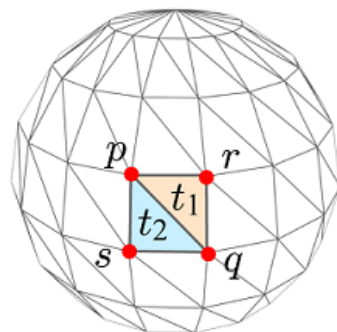
right-hand system (RHS)



left-hand system (LHS)



- Sağ el sisteminde, sağ elin baş parmak dışındaki parmakları x ekseninden y eksenine doğru kıvrılırken, baş parmak z ekseninin pozitif tarafını işaret eder. Sol el sisteminde aynı kural sol ele uygulanır.
- Normallerin nesneden dışarı doğru olması için RHS'de , köşeler CCW(saat yönünün tersine), LHS'de köşeler CW(saat yönünde) sıralanmalıdır.



for RHS

$$t_1 = \langle p, q, r \rangle$$

$$t_2 = \langle s, q, p \rangle$$

for LHS

$$t_1 = \langle p, r, q \rangle$$

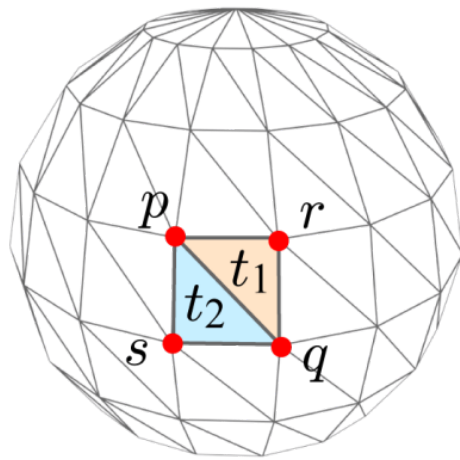
$$t_2 = \langle s, p, q \rangle$$

RHS'den LHS'ye çevirme

- RHS tabanlı bir pakette üretilmiş bir poligon mesh LHS tabanlı bir pakete geçirildiğinde bir çeşit yeniden modelleme yapılmalıdır.
 - Üçgen köşelerinin sıralaması değiştirilmelidir: $(p_1, p_2, p_3) \rightarrow (p_1, p_3, p_2)$
 - z koordinatı ters yöne çevrilmelidir.

Yüzey Normali

- İndeks belleğinde köşeler saat yönünün tersine (CCW) sıralanır.

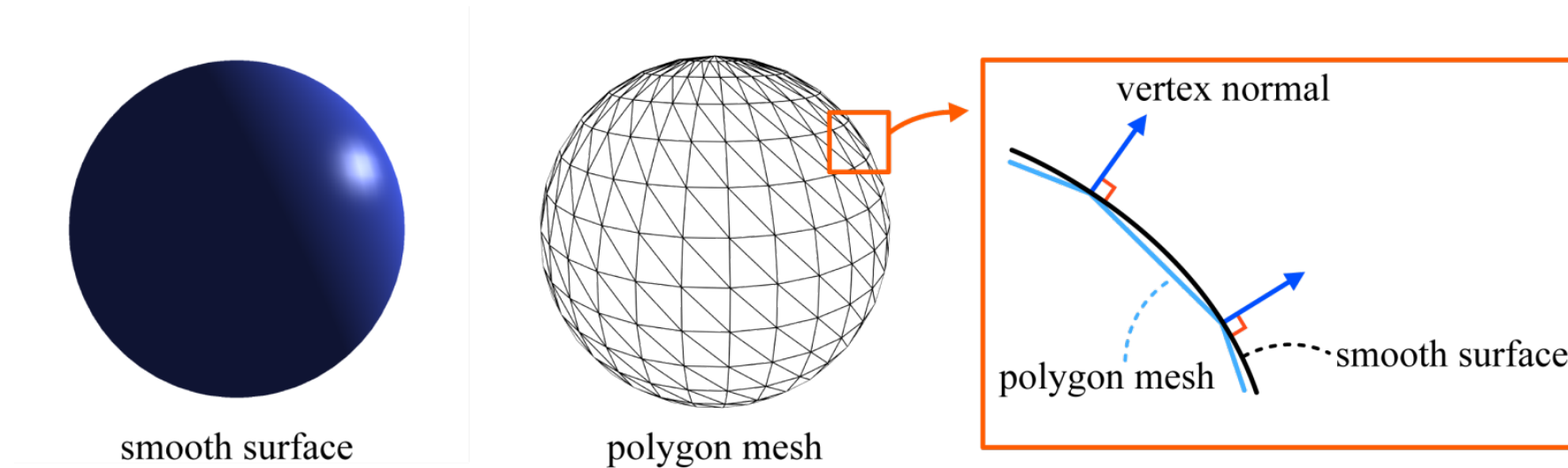


	vertex buffer	index buffer
0	p	0
1	q	1
2	r	2
3	s	3
4	...	1
		0
		...

t_1 (indices 0, 1, 2)
 t_2 (indices 3, 1, 0)

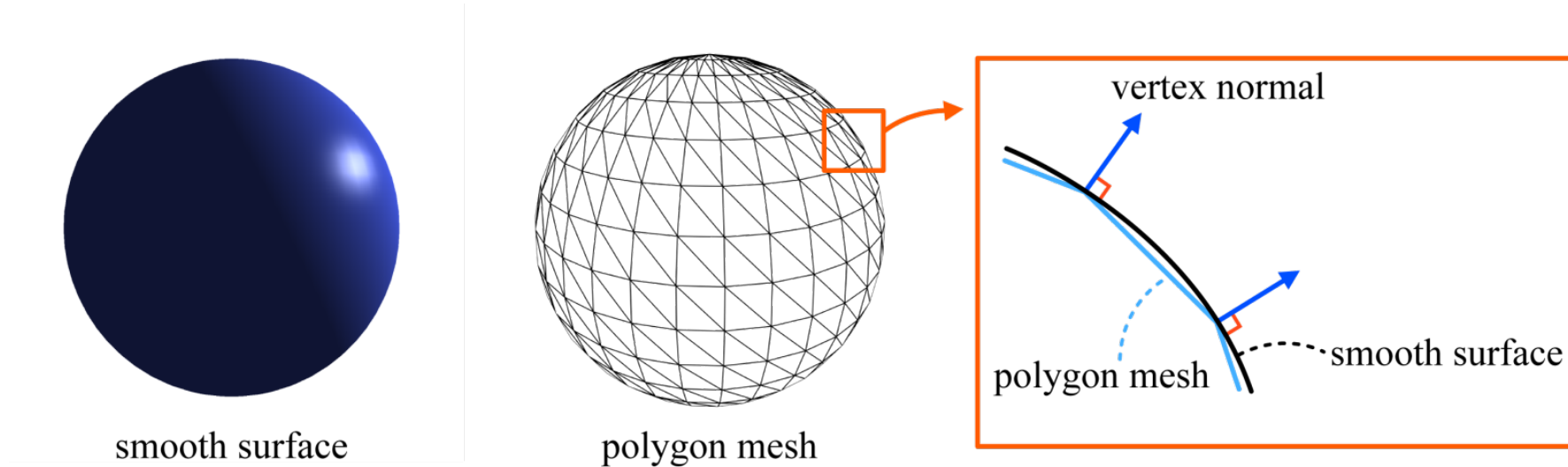
Köşe Normali

- Daha sonra da bahsedeceğimiz gibi esas önemli olan köşe normalidir.
- Poligon mesh smooth bir yüzeyin bir yaklaşımdır. O halde mesh köşeleri smooth yüzeyin üzerinde olduğunu varsaydığımız noktalardır. Dolayısıyla köşe normali, smooth yüzeyin o köşenin bulunduğu noktadaki normaline yaklaşık bir değerde olmalıdır.



Köşe Normali

- Köşe normali genellikle köşeyi paylaşan tüm üçgenlerin normal vektörlerinin ortalaması alınarak bulunur.



- Çoğu zaman köşe normalleri modelleme paketleri tarafından otomatik olarak hesaplanır, köşe belleğinde köşe özelliği(vertex attribute) olarak saklanır ve render aşamasına aktarılır.

Export ve Import

- Export: Verinin başka uygulamalara uygun çıktısını üretme işlemi



- Export yapmak için genellikle host program(yani modeler , örneğin 3ds Max) tarafından çalıştırılan scriptler kullanılır.
- 3ds Max export işlemi için MAXScript script dilini sağlar ve bu dille bir exporter yazılabilir.

Export ve Import

```
Specify Out as the output file
foreach polygon mesh
  Write the number of vertices to Out
  foreach vertex
    Write its positions and normals to Out
  endforeach
  Write the number of polygons to Out
  foreach polygon
    Write its three indices to Out
  endforeach
endforeach
```

```
vertices 530
49.2721 0.809525 85.6266 -0.966742 0.0 0.255752
48.5752 0.809525 88.2606 -0.966824 0.0 0.255444
49.3836 0.809525 89.1386 -0.092052 0.0 0.995754
...
faces 1024
0 5 6
6 1 0
...
```

İndeksli bir üçgen listesini export eden pseudo script kodu

Çıktı text dosyasının bir kısmı

- Bu örnekte export edilen poligon mesh 530 köşeden ve 1024 üçgenden oluşmaktadır.
- Bir köşe 6 float değerden oluşmaktadır: 3 pozisyon için ve 3 normal için.
- Köşe indeksi ilk köşe için 0, ikinci köşe için 1 olarak belirlenir ve aynı şekilde devam eder.
- Bir üçgen 3 indeks tarafından tanımlanır, örneğin ilk üçgen 0'ncı 5'nci ve 6'ncı köşelerden oluşmaktadır.

Export ve Import

- Export edilen poligonal mesh çalışma zamanında özel bir importerla uygulamaya yüklenir.
- Bir indeks üçgen listesiyle temsil edilen poligonal mesh genellikle CPU'da tutulan iki diziye yüklenir: biri köşeler için diğeri indeksler için.
- Bir takım grafik API fonksiyonları bu dizileri GPU'ya göndermek için çağırılır ve GPU poligon meshini render eder.