

2. GRAFİK-ANİMASYON İLE İLGİLİ DONANIM BİRİMLERİ

2.1. Ekran Kartı

Bilgisayarın görüntü özellikleri kullanılan ekran kartlarıyla sınırlıdır. Geçmişten günümüze kadar çok değişik özellikte ekran kartları geliştirilmiştir. Başlangıçta ekrandaki piksellerin adreslenmesi için bir standart yoktu ve üreticilerle programcılar görüntü açısından sorun yaşıyorlardı. Ekran kartı üreticileri bir araya geldiler ve VESA (Video Electronics Standards Association) adında video protokollerini standartlaştırdılar. Ekran kartları görüntü arayüzleri standartlarına göre aşağıda listelenmiştir.

2.1.1. Ekran Kartı Türleri

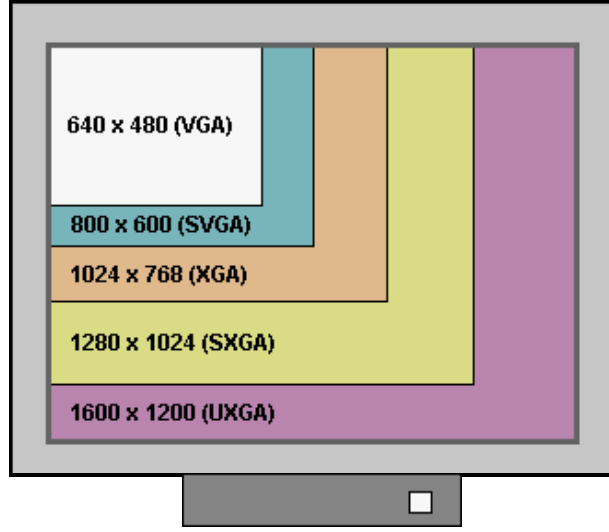
MDA (Hercules) Ekran Kartı : Açılımı Monochrome Display Adapter dir. İlk IBM firması tarafından 1981 yılında üretilmiştir. Gösterebileceği yazı karakterlerinin boyutları önceden belirlenmiştir. Grafik ve renk özelliği yoktu ve sadece 80 kolona 25 satırlık bir ekranda düz metinleri görüntüleyebiliyordu.

CGA Ekran Kartı : Açılımı Color Graphic Adapter (Renkli Grafik Adaptörü) dir. İlk grafik özellikli ekran kartıdır. 1980'li yıllarda üretilmiştir. 320x240 çözünürlüğündeki bir ekranda 16 renk üretilebiliyordu fakat aynı anda bunlardan sadece 4 tanesini kullanabiliyordu. 640x200'lük bir yüksek çözünürlük modunda sadece 2 renk gösterilebiliyordu.

EGA Ekran Kartı : Açılımı Enhanced Graphics Adapter dir. 640x350 piksel çözünürlükte 64 renkten aynı anda 16 tanesi kullanılabiliyordu.

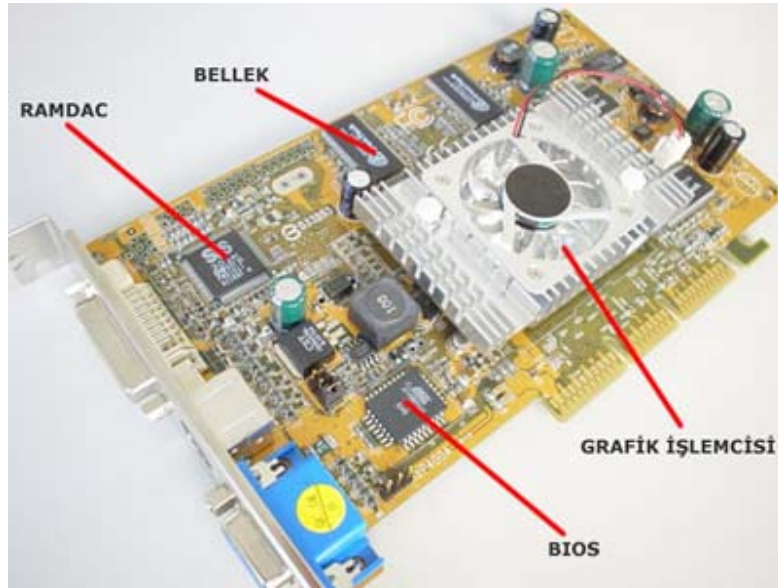
VGA: 1987'de IBM tarafından üretilmiştir ve VGA masaüstü için standart olmuştur. VGA ile 262144 renklilik bir paletten seçilen 256 renk aynı anda kullanılabiliyordu. 640x480'lik standart çözünürlükte aynı anda 16 renk gösterilebiliyordu. Ayrıca 64 renk gri tonlama ile siyah beyaz monitörlerde renk simülasyonu yapabiliyordu.

SVGA: Super VGA ilk kartlardan güncel kartlara kadar çok fazla kartı kapsayan geniş bir standarttır. SVGA ile birlikte ekran kartları için aygıt sürücüsü kavramı ortaya çıkmış ve kartların yanında verilen sürücülerle kartların tüm özelliklerinin işletim sistemi tarafından kullanılabilmesi sağlanmıştır. Karta ve üreticiye bağlı olarak SVGA ile milyonlarca renk değişik çözünürlükleri desteklenmektedir. SVGA ile 800x600 çözünürlüğe ulaşılmış, SVGA'dan sonra IBM XGA ile 1024x768 çözünürlüğe geçilmiş ve sonraki basamak olan 1280x1024'e de bir VESA standardı olan SXGA ile geçilmiştir. En son olarak da UXGA ile de 1600x1200 çözünürlüğe geçilmiştir.



Şekil 2.1.1.1. Ekran Kartı Çözünürlük Görüntüsü

2.1.2. Ekran Kartı Bileşenleri ve Özellikleri



Şekil 2.1.2.1. En temel bileşenleriyle bir ekran kartı

Ekran kartında bulunan çipset performansı, çözünürlüğü, renk sayısını ve hız gibi özellikleri sağlar. Bellek ise doğrudan ekran kartının sunduğu renk sayısı ile ilgilidir. İlk VGA kartlar 16,32,64,128KB gibi belleklere sahipken, bugünkü kartlar 8,16,32,64,128MB olarak üretilmektedirler. Bir ekran kartında aranacak en önemli özellikler çözünürlüğün ve belleğin yüksek olmasıdır. Ekran kartlarının birer BIOS'ları vardır. Burada ekran kartının çalışma parametreleri, temel sistem fontları kayıtlıdır. Ayrıca bu BIOS sistem açılırken ekran kartına ve onun belleğine de küçük bir test yapar.

2.2. 3D Ekran Kartları

3D üç boyutlu görüntü demektir. Bilgisayarda grafikleri oluşturup, ekrana yansıtmak ve bunu gerçek zamanlı olarak uygulayabilmek için büyük işlemci gücüne sahip ekran kartlarına ihtiyaç vardır. Örneğin bir bilgisayar oyununda saniyede 30 kare bellekten alınarak görüntülenecekse 3D kart belleğin yükünü alarak görüntüleri sorunsuz işleyecektir. 3B uygulamalar için kullanılan ekran kartları diğerlerine göre daha pahalıdır. Bir 3B görüntüsü 3 temel adımda oluşturulur:

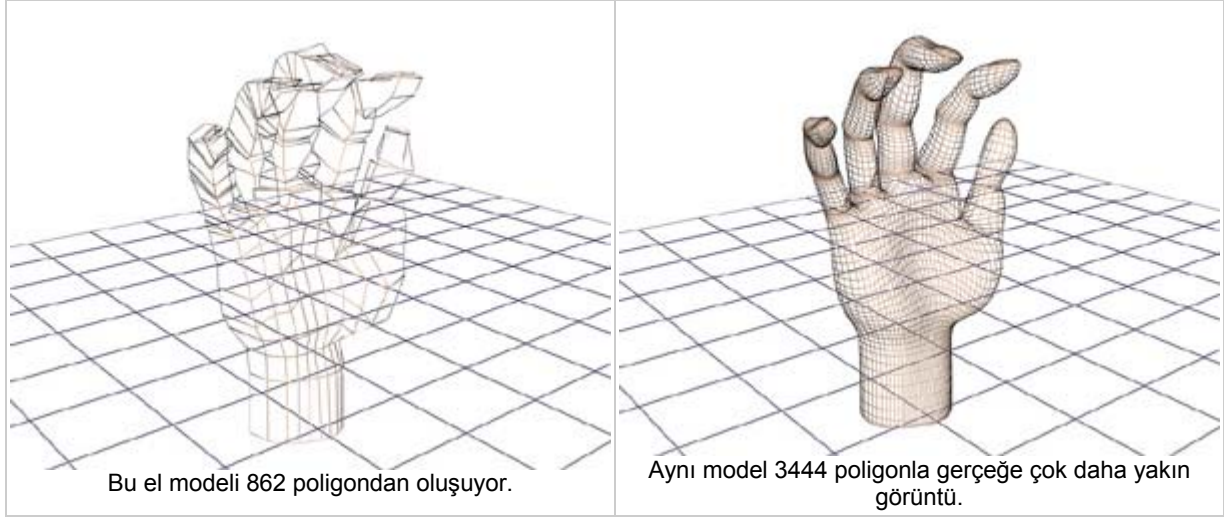
1. Sanal bir 3B ortam yaratılır
2. Ekranda bu ortamın hangi bölümünün gösterileceğine karar verilir.
3. Görüntüyü mümkün olduğu kadar gerçeğe yakın gösterebilmek için her pikselin nasıl görüneceği belirlenir.

Sanal bir 3B ortamı sadece o ortamın bir resmini belirleyemez. Bir masanın üzerinde duran bir eli 3B ortamında tasarlayalım. Masanın sert olduğu el ile dokunulduğunda anlaşılabilir. El ile masaya vurulduğunda da kırılmaz veya el masanın içinden geçemez. Bu ortamın ne kadar çok resmine bakılırsa bakılsın masanın sertliği ve ele vereceği tepki sadece o resimlerle anlaşılmaz. Sanal 3B ortamlar da böyledir. Bu ortamlardaki nesnelere sentetiktir, bütün özellikleri onlara yazılım yoluyla verilir. Programcılar sanal bir 3B dünya tasarlarlarken büyük bir özenle bütün bu detaylara dikkat ederler ve bu işler için özel araçlar kullanırlar.

Belirli bir zamanda oluşturulan 3B görüntünün ancak belirli bir bölümü ekranda gösterilir. Ekrandaki görüntü nesnenin nasıl tanımlandığına, sizin nereye gitmek istediğinize ve nereye baktığınıza göre değişir. Hangi yöne hareket ederseniz edin etrafınızdaki sanal dünya o an bulunduğunuz pozisyonu ve nereye baktığınızı değerlendirerek ekranda ne görmeniz gerektiğine karar verir. Bu farklı sahneler de kendi içlerinde tutarlı olmalıdırlar. Örneğin bir nesne ona baktığınız her açıdan ve uzaklıktan aynı yükseklikteymiş hissi vermelidir. 3. adıma geçmeden önce sabit bir görüntünün nasıl oluşturulduğuna bakıp sonra da bir 3B görüntünün nasıl hareket kazandığına bakacağız.

2.2.1. 3D Şekiller

3B nesnelere ilk başta wireframe denen bir yapı ile oluşturulurlar. Şeklin iskeleti de diyebileceğimiz bu tel örgü en basit haliyle nesnenin şeklini belirler. Wireframe denen bu yapı için bir yüzey tanımlanması şarttır.



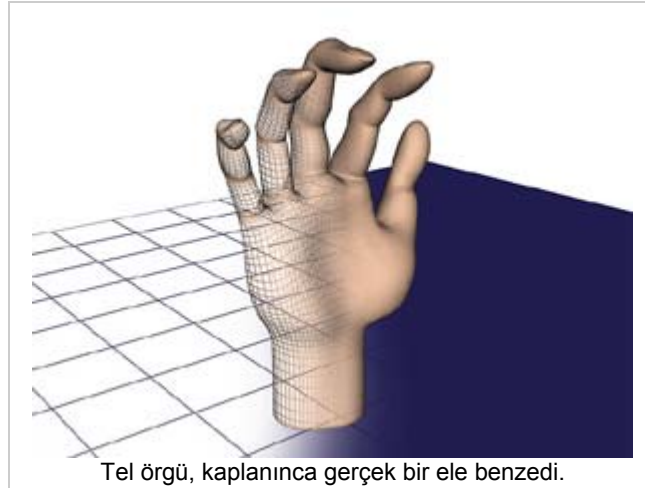
Şekil 2.2.1.1. Çeşitli poligonlardan oluşan 3D şekiller

2.2.2. Yüzey Kaplamaları

Sanal bir 3B ortamda nesnelere elleme şansımız olmadığından onların hakkında sadece onlara bakarak fikir edinebiliriz. Bu yüzden sanal 3B ortamlarda nesnelere dış görünüşleri çok önemlidir. Dış görünüşü şunlar belirler:

- **Renk:** Nesnenin rengi.
- **Kaplama:** Tel örgünün üzerine yapılan kaplamayla nesnenin yüzeyi düz, çizgili veya girintili çıkıntılı görünebilir.
- **Yansımalar:** Nesneye etkiyen ışığa ve etrafındaki diğer nesnelere göre cismin üzerinde yansımalar oluşturulur.

Bir nesneyi gerçek gibi göstermek için bu üç özellik de dengeli bir biçimde nesnenin değişik yüzeylerine uygulanmalıdır. Örneğin bir 3B ortamda bir klavyeyle bir masa ışığı aynı oranda yansıtmaz. Bu üç parametreyi değiştirerek nesnelere sert veya yumuşak hissi verilebilir.



Şekil 2.2.2.1. Yüzey kaplanmış 3D görüntü

2.2.3. Lighting (Işıklandırma)

Karanlık bir odaya girdiğimizde ışığı açarız ve ışık kaynağından her yöne doğru yayılan ışık sayesinde odadaki bütün nesnelerin görüntüsü değişir. Bu ışığın odaya nasıl yayıldığını düşünmeyiz ama 3B grafiklerle uğraşanlar bunu düşünmek zorundalar. Tel örgüleri kaplayan kaplamalar (texturelar) bir yerden aydınlatılmalıdırlar. Ray tracing denilen bir yöntemle ışık ışınlarının alacağı yol çizilir ve bu ışınlar çarptıkları nesnelere farklı yoğunluk ve açılarla yansır. Çoklu ışık kaynaklarını düşündüğümüzde bu hesaplamalar oldukça karmaşık bir hal alabilir.

Işıklandırma cisme ağırlık ve katılık etkisi verirken en çok kullanılan iki efektte önemli rol oynar: Shading ve gölgeler. Shading, bir nesne üzerindeki parlayan ışığın bir tarafında diğer tarafından daha güçlü olmasıdır. Ancak shading sayesinde bir top yuvarlak veya buruşmuş bir battaniye yumuşak görünebilir. Parlaklıktaki bu fark nesnelere derinlik, uzunluk ve genişlik kazandırır.

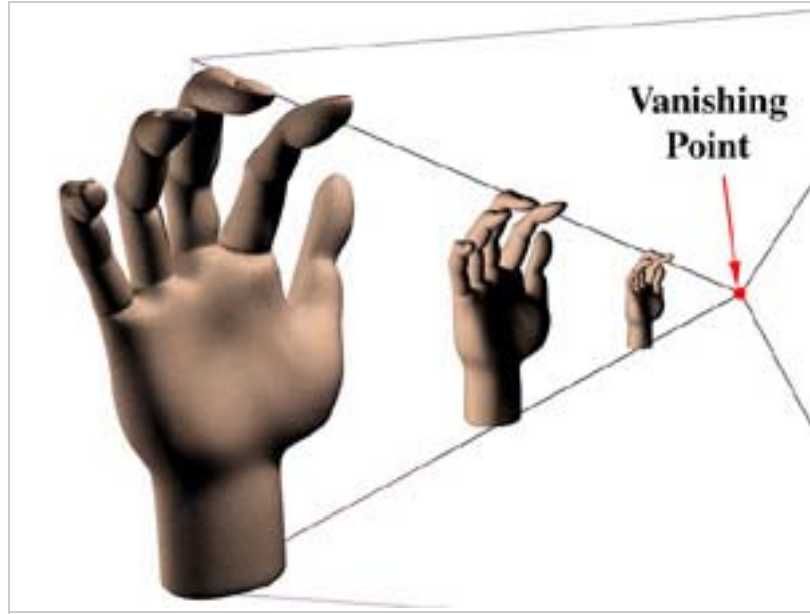


Şekil 2.2.3.1. Işıklandırılmış 3D görüntü

Katı nesnelere üzerlerinden ışık parladığında gölgeler yaratırlar. Gözlerimiz gerçek nesnelere görmeye alışık olduğundan ekranda gölge gördüğümüz zaman matematiksel olarak üretilmiş şekillere değil de bir pencereden gerçek bir dünyaya bakıyormuş gibi hissederiz.

2.2.4. Perspektif

Perspektif kulağa biraz teknik gelebilir ama günlük yaşamımızda çok sık gördüğümüz bir etkidir. Bir yolun kenarında durup ufuk çizgisine doğru baktığınızda yolun iki kenarı da birleşiyormuş gibi görünür. Yol kenarında ağaçlar varsa da bu ağaçlar birleşme noktasına yaklaştıkça da daha küçük görünür. Nesnelere bir noktada birleşiyormuş gibi görünmesini sağlayan bu efekt perspektiftir. Değişik çeşitleri vardır fakat 3B çizimlerde genelde tek noktalı perspektif kullanılır.



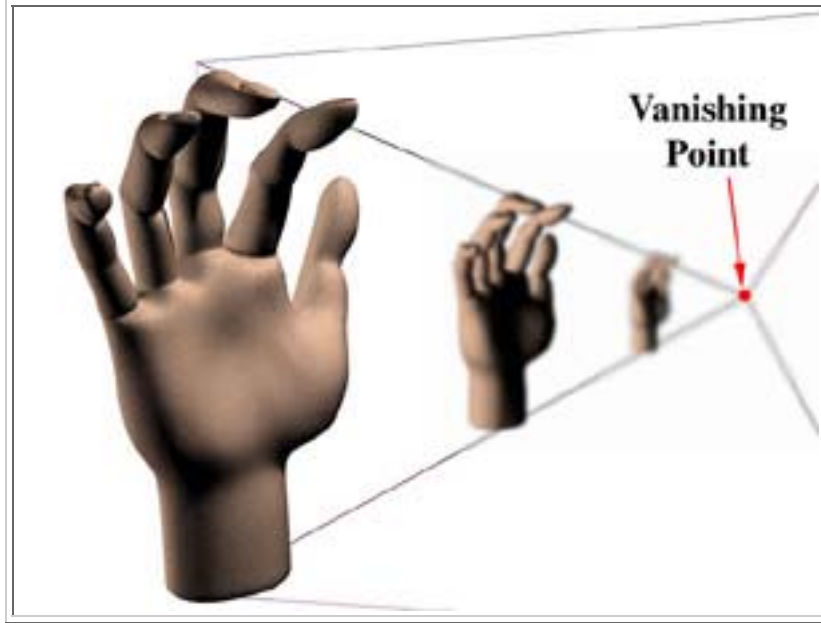
Şekil 2.2.4.1. Perspektif uygulanmış 3D görüntü

Şekildeki eller ayrı duruyor fakat çoğu sahnede nesnelere birbirlerinin önünde dururlar ve birbirlerini kısmen kapatırlar. Bu durumda bunların büyüklüklerinin hesaplanması dışında hangisinin önde olduğu da bilinmelidir. Bunun için Z Buffering denilen teknik kullanılır. Z buffera her poligon için bir sayı atanır ve bu sayı o poligona sahip nesnenin sahnenin ön tarafına yakınlığını belirler. Öneğin 16 bitlik bir Z bufferekrana en yakın poligon için -32768 ve en uzak poligon için de 32767 değerlerini atar.

Gerçekte bir nesnenin arkasındaki diğer nesnelere göremediğimiz için ne görüyor olmamız gerektiğini düşünmeyiz. Sanal 3B ortamlarda da bu sıkça olur ve çok düz bir mantıkla çözülür. Nesnelere yaratıldıkça x ve y ekseninde aynı değere sahip olanlarının Z bufferdaki değerleri karşılaştırılır ve en düşük Z değerine sahip nesne tamamen görüntülenir. Daha yüksek Z değerindekilerinse tamamı görüntülenen nesneyle kesişen bölgeleri görüntülenmez. Nesnelere tamamen oluşturulmadan önce Z değerleri belirlendiği için görünmeyecek bölgeler tamamen hesaplanmaz ve bu da performansı artırır.

2.2.5. Derinlik (Depth of Field)

Yol ve ağaçlar örneğimizi hatırlayalım ve o örnekte oluşabilecek başka bir ilginç olayı düşünelim. Yakınındaki bir ağaca bakarsanız uzaktaki ağaçların netliklerini kaybettiğini görürsünüz. Filmlerde ve bilgisayar ortamında sık kullanılan bu efekt iki amaca hizmet eder. İlki sahnedeki derinlik hissini güçlendirmektir. İkincisi ise dikkatinizi bir nesneye çekmektir.



Şekil 2.2.5.1. Derinlik uygulanmış 3D görüntü

2.2.6. Anti-Aliasing

Bu teknik de gözü aldatarak görüntünün doğal görünmesini amaçlar. Dijital görüntü sistemleri aşağıya ve yukarıya doğru düz çizgiler çizmekte son derece başarılıdırlar fakat iş eğrilere ve çapraz çizgileri çizmeye gelince basamak efekti oluşur ve çizgilerin kenarları yumuşak değil de daha çok bir merdiven gibi gözükür. İşte bu noktada devreye anti-aliasing girer ve çizginin kenarlarındaki piksellere onlara yakın gir tonlardaki renklerle shading uygulayarak kenarları biraz bulanıklaştırır. Bu sayede basamak efekti ortadan kaybolmuş gibi gözükür. Anti-aliasing de doğru pikselleri için doğru renkleri seçmek de başka bir karmaşık işlemdir ve sisteme oldukça yük bindirir.





Şekil 2.2.6.1. Anti-Aliasing uygulanmış 3D görüntü

2.2.7. Görüntüleri Hareketlendirme

Durağan 3B sahnelerini yarattıktan sonra bunları hareketlendirelim. Şu ana kadar anlattığımız işlemlerin hiçbiri donanımı yaratılan bu durağan görüntülere hareket kazandırmaktan daha fazla zorlayamaz. Üçgenlerden ve poligonlardan oluşan tel örgülerimizi hareket ettirmek için ekrandaki her piksel saniyede belirli sayıda hesaplaması yapılmalıdır.

Yüksek çözünürlük denince akla en az 1024x768 gelir. Bu çözünürlükte 786.432 adet piksel kullanılır, her piksel için 32 bit renk kullanıldığında 25.165.824 bit sadece durağan görüntü için gereklidir. Görüntünün 60 FPS hızda çalışması için her saniye 1.509.949.440 bit veri aktarılmalıdır ve bu sadece görüntüyü ekrana yansıtmak için yeterlidir. Bunun yanında bilgisayar görüntü içeriğini, renkleri, şekilleri, ışıkları ve diğer efektleri de hesaplamak zorundadır. Bütün bunlar görüntü işlemcilerinin çok hızlı gelişmesine sebep oluyor çünkü CPU'nun alabileceği her türlü yardıma ihtiyacı var.

2.2.8. Transform (Dönüşüm) İşlemleri

Durağan görüntüler dönüşüm denen matematiksel bir işlem sonucunda hareket kazanırlar. Bakış açımızı her değiştirdiğimizde bir dönüşüm olur. Bir arabanın bize yaklaştıkça daha büyük görünmesi gibi, büyüklüğün her değişiminde bir dönüşüm olur. Bir 3B oyunun her karesinde kullanılan dönüşüm işlemine matematiksel olarak şu şekildedir:

Dönüşümde ilk etapta sanal dünyamızı tanımlayan önemli değişkenler kullanılır:

- $X = 758$ – baktığımız sanal dünyanın yüksekliği
- $Y = 1024$ – bu sanal dünyanın genişliği
- $Z = 2$ – bu da sanal dünyamızın derinliği
- S_x - sanal dünyaya baktığımız pencerenin yüksekliği
- S_y – pencerenin genişliği
- S_z = hangi nesnelerin diğerlerinin önünde görüldüğünü belirten derinlik değişkeni
- $D = .75$ – gözümüzle sanal dünyamıza açılan pencere arasındaki uzaklık

Not : Transform işlemleri ve aşağıda anlatılan formüller sadece bilgi amacıyla verilmiş olup sınavda bu konu ile ilgili soru sorulmayacaktır.

Öncelikle sanal 3B dünyamıza açılan pencerelerimizin genişliği hesaplanır:

$$\begin{aligned} \frac{sx}{d} &= \frac{X}{Z} ; & \frac{sy}{d} &= \frac{Y}{Z} \\ sx &= \frac{768 \cdot .75}{2} = \frac{768}{2/.75} & sy &= \frac{1024 \cdot .75}{2} = \frac{1024}{2/.75} \\ sx &= 288 & sy &= 384 \end{aligned}$$

Daha sonra perspektif dönüşümü yapılır, bu aşamada yeni değişkenler de işin içine girer:

$$\begin{aligned} \omega &= 22 \text{ -- toplam görüş alanı} \\ z_n &= 1 \text{ -- sanal dünyamızı göremediğimiz bize en yakın nokta} \\ z_f &= 30 \text{ -- sanal dünyamızı göremediğimiz bize en uzak nokta} \end{aligned}$$

$$\begin{aligned} s &= \sin\left(\frac{\omega}{2}\right) \\ s &= 0.19 \\ c &= \cos\left(\frac{\omega}{2}\right) \\ c &= .9816 \\ Q &= \frac{s}{1 - z_n/z_f} \\ Q &= 0.1966 \end{aligned}$$

Sonunda (X, Y, Z, 1.0) noktası aşağıdaki işlemciler sonucunda (X', Y', Z', W') noktasına dönüşür:

$$\begin{aligned} X' &= X \cdot c \\ X' &= 753.8688 \\ Y' &= Y \cdot c \\ Y' &= 1005.1584 \\ W' &= s \cdot c \\ W' &= 0.1865 \\ Z' &= Q \cdot Z - Q \cdot z_n \\ &= Q(Z - z_n) \\ &= \frac{s}{1 - z_n/z_f} (Z - z_n) \\ &= s \cdot z_f \frac{(Z - z_n)}{z_f - z_n} \\ Z' &= .1966 \end{aligned}$$

Görüntü ekrana yansıtılmadan önce son bir dönüşüm daha yapılmalıdır, bu kadarı bile bu işlemin karmaşıklığı hakkında size fikir vermiştir. Üstelik bütün bu işlemler tek bir vektör, yani basit bir çizgi için. Aynı işlemlerin görüntüyü oluşturan bütün nesnelere saniyede 60 kere uygulandığını düşünün...