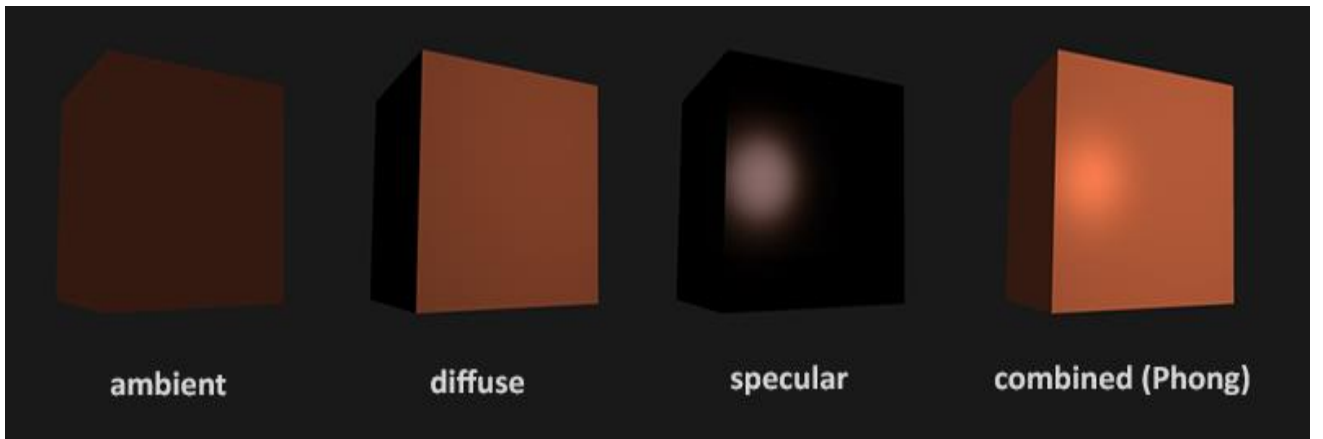


Basit Işıklandırma

Gerçek dünyadaki ışıklandırma son derece karmaşıktır ve çok fazla faktöre bağlıdır, bu da sahip olduğumuz sınırlı işlem gücüyle hesaplayamayacağımız bir şeydir. Bu nedenle OpenGL' de ışıklandırma, işlenmesi çok daha kolay olan ve nispeten benzer görünen basitleştirilmiş modeller kullanan gerçeklik yaklaşımlarına dayanmaktadır. Bu ışıklandırma modelleri, anlayabileceğimiz gibi ışığın fiziğine dayanmaktadır. Bu modellerden birine **Phong aydınlatma modeli** denir. Phong aydınlatma modeli 3 bileşenden oluşur: ortam (*ing. ambient*), dağınık (*ing. diffuse*) ve düzgün (*ing. specular*) ışıklandırma. Aşağıda bu ışıklandırma bileşenlerinin kendi başlarına ve karma olarak nasıl göründüklerini görebilirsiniz:



- Ortam Işıklandırması : Karanlık olduğunda bile, dünyanın bir yerinde (ay, uzak bir ışık) genellikle halen bir miktar bulunur, bu yüzden nesneler neredeyse hiçbir zaman tamamen karanlık değildir. Bunu benzetimlemek için, nesneye her zaman biraz renk veren bir ortam aydınlatma sabiti kullanıyoruz.
- Dağınık Işıklandırma : Bir ışık nesnesinin bir nesne üzerindeki yönlü etkisini benzetimler. Bu, ışıklandırma modelinin görsel olarak en önemli bileşenidir. Bir nesnenin bir kısmı ışık kaynağına ne kadar çok bakarsa o kadar parlak olur.
- Düzgün Işıklandırma : Parlak nesnelerde görünen ışığın, parlak noktasını benzetimler. Düzgün vurgular ışığın rengine, nesnenin renginden daha yatkındır.

Görsel olarak ilginç sahneler oluşturmak için en azından bu 3 ışıklandırma bileşenini benzetimlemek istiyoruz. En basit olanla başlayacağız: **ortam ışıklandırması**.

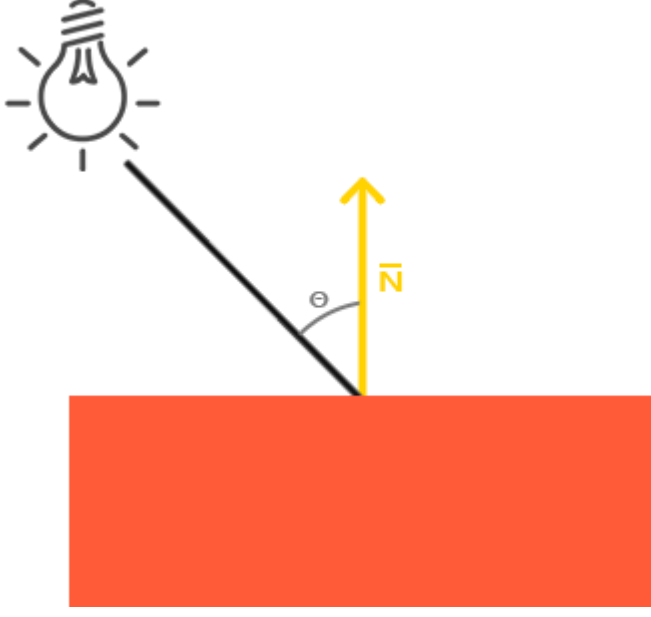
Ortam Işıklandırması

Işık genellikle tek bir ışık kaynağından gelmez, ancak anında görünür olmadıklarında bile, etrafımıza dağılmış birçok ışık kaynağından oluşmaktadır. Işığın özelliklerinden biri, doğrudan görünmeyen noktalara ulaşarak birçok yöne dağılabilmesi ve sıçrayabilmesidir; böylece ışık diğer yüzeylere *yansıyabilir* ve bir nesnenin ışıklandırılması üzerinde dolaylı bir etkiye sahip olabilir. Bunu dikkate alan algoritmalara **global aydınlanma** (ing.global illumination) algoritmaları denir, ancak bunların hesaplanması karmaşık ve maliyetlidir.

Karmaşık ve pahalı algoritmaların hayranları olmadığımızdan, çok basit bir küresel aydınlatma modeli, yani **ortam ışıklandırması** kullanarak başlayacağız. Nesnenin parçalarının nihai sonuç rengine eklediğimiz küçük bir sabit (ışık) rengi kullanıyoruz, böylece doğrudan bir ışık kaynağı olmasa bile her zaman etrafa saçılan bir ışık varmış gibi görünmesini sağlıyoruz.

Dağınık Işıklandırma

Ortam ışıklandırması tek başına en ilginç sonuçları üretmez, ancak dağınık ışıklandırma nesne üzerinde önemli bir görsel etki vermeye başlayacaktır. Dağınık ışıklandırma, nesneye, parçaları bir ışık kaynağından gelen ışık ışınlarına ne kadar yakınsa, daha fazla parlaklık verir. Dağınık ışıklandırmayı daha iyi anlamak için aşağıdaki görsele bir göz atalım:



Solda, nesnemizin tek bir parçasını hedef alan bir ışık ışını olan bir ışık kaynağı görüyoruz. Işık ışınının parçaya hangi açıda dokunduğunu ölçmemiz gerekir. Işık ışını nesnenin yüzeyine dikse, ışık en büyük etkiyi yaratacaktır. Işık ışını ve parça arasındaki açıyı ölçmek için, **normal vektör** adı verilen bir vektör kullanırız, yani parçanın yüzeyine dik bir vektör (burada sarı bir ok olarak gösteriliyor). İki vektör arasındaki açı ise iç çarpım ile kolayca hesaplanabilir.

[Dönüşümler](#) konusundan, iki birim vektör arasındaki açı ne kadar düşük olursa, iç çarpımın 1 değerine o kadar meyilli olduğunu hatırlayabilirsiniz. Her iki vektör arasındaki açı 90 derece olduğunda, iç çarpım 0 olur. Aynısı θ için de geçerlidir: θ büyüdükçe, ışığın parçanın rengi üzerindeki etkisi azalır.

Her iki vektör arasındaki açının (sadece) kosinüsünü elde etmek için birim vektörlerle (uzunluğu 1 birim olan vektörler) çalışacağımızı unutmayın, bu nedenle tüm vektörlerin normalleştirildiğinden emin olmamız gerekir, aksi takdirde iç çarpım kosinüsten daha fazla olarak geri döner.

Sonuçta elde edilen iç çarpım, ışığın parçanın rengi üzerindeki etkisini hesaplamak için kullanabileceğimiz bir skaler değer döndürerek ışığa doğru yönlerine göre farklı ışıklandırılmış parçalara neden olur.

Peki, dağınık ışıklandırmayı hesaplamak için neye ihtiyacımız var:

- Normal vektörü: köşe noktasının yüzeyine dik olan bir vektör.
- Yönlendirilmiş ışık ışını: Işığın konumu ile parçanın konumu arasındaki fark vektörü olan bir yön vektörüdür. Bu ışık ışını hesaplamak için ışığın konum vektörüne ve parçanın konum vektörüne ihtiyacımız vardır.

Normal vektörleri

Bir normal vektörü, bir köşe noktasının yüzeyine dik olan bir (birim) vektördür. Bir köşe noktası kendi başına yüzeye sahip olmadığından (uzayda sadece tek bir nokta olduğundan), köşe noktasının yüzeyini bulmak için çevresindeki köşeleri kullanarak normal bir vektör elde ederiz. Çapraz çarpım kullanarak küpün tüm köşeleri için normal vektörleri hesaplamak için küçük bir hile kullanabiliriz, ancak 3- boyutlu küp karmaşık bir şekil olmadığından bunları köşe verilerine el ile ekleyebiliriz. Normallerin aslında her bir düzlemin yüzeyine dik vektörler olduğunu hayal etmeye çalışın (bir küp 6 düzlemden oluşur).

Materyaller

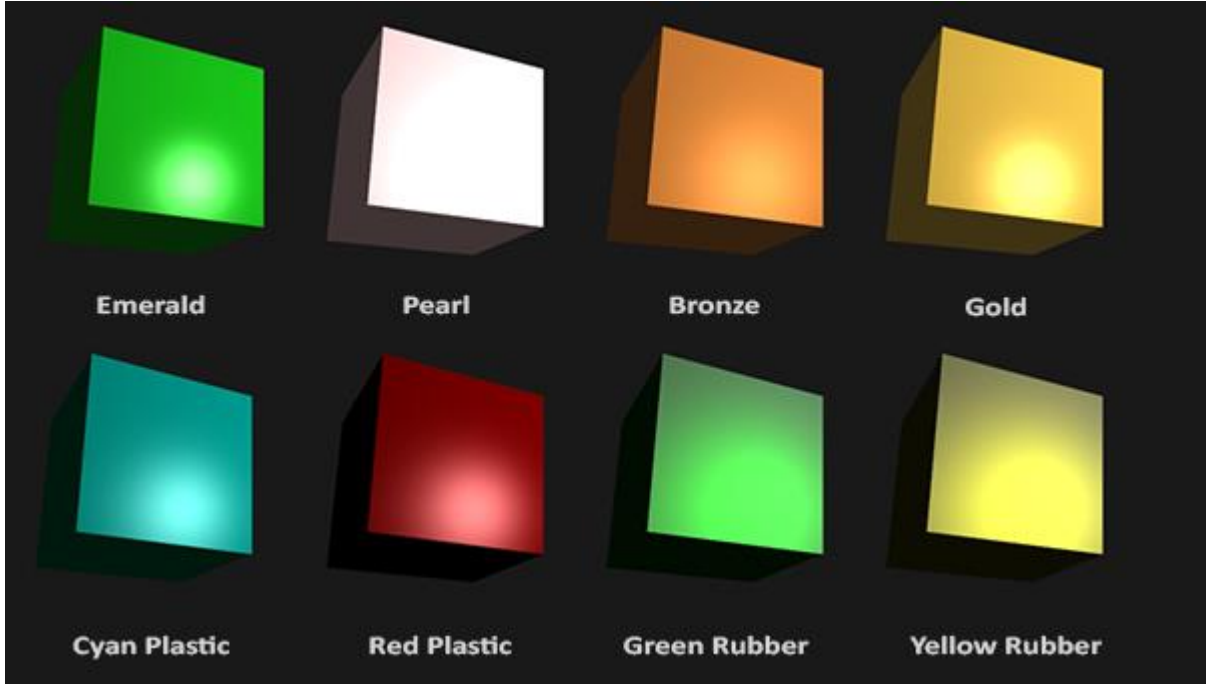
Gerçek dünyada her nesnenin ışığa farklı bir tepkisi vardır. Örneğin çelik nesneler genellikle kil vazodan daha parlaktır ve ahşap bir kap, çelik bir kap gibi ışığa tepki vermez. Bazı nesneler ışığı çok fazla dağılmadan yansıtır, bu da küçük aynasal vurgulara neden olur ve diğerleri çok fazla saçılarak vurguyu daha büyük bir yarıçapta verir. OpenGL' de birkaç nesne türünü benzetimlemek istiyorsak, her yüzeye özgü *materyal* özelliklerini tanımlamamız gerekir.

Bir yüzeyi tanımlarken, üç aydınlatma bileşeninin her biri için bir materyal rengi tanımlayabiliriz: ortam, dağınık ve yansıtıcı aydınlatma. Bileşenlerin her biri için bir renk belirleyerek, yüzeyin renk çıktısı üzerinde ince taneli kontrole sahip oluruz. Şimdi bu üç renge bir parlaklık bileşeni ekleyip ihtiyacımız olan tüm materyal özelliklerine sahip oluruz.

Phong aydınlatmasının bileşenlerinin her biri için bir renk vektörü tanımlıyoruz. ***ambient*** materyali vektörü, yüzeyin ortam aydınlatması altında hangi rengi yansıttığını tanımlar; bu genellikle yüzeyin rengiyle aynıdır. ***diffuse*** materyal vektörü, dağınık aydınlatma altındaki yüzeyin rengini tanımlar. Yaygın renk (tıpkı ortam aydınlatması gibi) istenen yüzeyin rengine ayarlanır. ***specular*** materyal vektörü, yüzeydeki yansıyan vurgunun rengini ayarlar (veya

hatta muhtemelen yüzeye özgü bir rengi yansıtır). Son olarak *shininess* , yansıyan açık tonun saçılımını/ yarıçapını etkiler.

Bir nesnenin materyalini tanımlayan bu 4 bileşenle birçok gerçek dünya materyalini benzetimleyebiliriz. devernay.free.fr adresinde bulunan tablo, dış dünyada bulunan gerçek materyalleri benzetimleyen materyal özelliklerinin bir listesini gösterir. Aşağıdaki görsel, bu gerçek dünya materyallerinden birkaçının küpümüz üzerindeki etkisini göstermektedir:



Gördüğünüz gibi, bir yüzeyin materyal özelliklerini doğru bir şekilde belirlemek, nesneye ilişkin sahip olduğumuz algıyı değiştiriyor. Etkiler açıkça fark edilir, ancak daha gerçekçi sonuçlar için küpü daha karmaşık bir şeyle değiştirmemiz gerekecek.

Bir nesne için doğru materyal ayarlarını bulmak, çok fazla deneyim gerektiren zorlu bir başarıdır. Bir nesnenin görsel kalitesini yanlış yerleştirilmiş bir materyal ile tamamen yok etmek o kadar da rastlanmadık bir durum değildir.

Sözlük

- Renk vektörü: kırmızı, yeşil ve mavi bileşenlerin (RGB olarak kısaltılmıştır) bir kombinasyonu ile gerçek dünyadaki renklerin çoğunu gösteren bir vektördür. Bir nesnenin rengi, bir nesnenin emmediği yansıtılan renk bileşenleridir.
- Phong ışıklandırma modeli: bir ortam, dağınık ve düzgün bileşeni hesaplayarak gerçek dünya ışıklandırmasına yaklaşmak için bir model.
- Ortam ışıklandırması: her bir nesneye küçük bir parlaklık vererek küresel ışıklandırmanın yaklaştırılmasıdır, böylece nesneler doğrudan ışıklandırılmadıkça tamamen karanlık değildir.
- Dağınık gölgeleme: daha güçlü hale gelen ışıklandırma, bir köşe / parça bir ışık kaynağına hizalanır. Açılarını hesaplamak için normal vektörleri kullanır.
- Normal vektörü: bir yüzeye dik olan bir birim vektör.
- Normal matrisi: öteleme olmadan model (veya model görünümü) matrisi olan 3x3 bir matris. Aynı zamanda, düzensiz ölçekleme uygularken normal vektörleri doğru yönde bakacak şekilde (ters- devrik) değiştirilir. Aksi takdirde, düzensiz ölçekleme kullanıldığında normal vektörler bozulur.
- Düzgün ışıklandırma: izleyici, bir ışık kaynağının bir yüzeydeki yansımasına ne kadar yakın bakıyorsa, düzgün bir vurgu ayarlar. İzleyicinin yönüne, ışığın yönüne ve vurgulamanın saçılma miktarını belirleyen bir parlaklık değerine bağlıdır.
- Phong gölgeleme: parça gölgelendiricide uygulanan Phong aydınlatma modeli.
- Gouraud gölgeleme: köşe nokta gölgelendiricide uygulanan Phong aydınlatma modeli. Az sayıda köşe kullanırken gözle görülür eserler üretir. Görsel kalite kaybı için verimlilik kazanır.
- GLSL struct: gölgelendirici değişkenleri için bir konteyner görevi gören C benzeri bir yapıdır. Çoğunlukla giriş, çıkış ve uniform'ları düzenlemek için kullanılır.
- Materyal: bir nesnenin yansıttığı ortam, dağınık ve düzgün renk. Bunlar bir nesnenin sahip olduğu renkleri belirler.
- Işık (özellikler): bir ışığın ortam, dağınık ve düzgün yoğunluğudur. Bunlar herhangi bir renk değerini alabilir ve her bir Phong bileşeni için bir ışık kaynağının hangi renk/ yoğunlukta parladığını tanımlayabilir.
- Dağınık eşleme: parça başına dağınık rengi ayarlayan bir doku imgesi.
- Düzgün eşleme: parça başına düzgün yoğunluğu/ rengi ayarlayan bir doku eşlemedir. Sadece bir nesnenin belirli alanlarında düzgün vurgulara izin verir.

- Yönlü ışık: sadece tek bir yönü olan bir ışık kaynağıdır. Tüm ışık ışınlarının paralel görüldüğü ve yön vektörünün tüm sahne boyunca aynı kaldığı etkisine sahip sonsuz bir mesafede olacak şekilde modellenmiştir.
- Nokta ışık: uzaktan sönen ışık ile bir sahnede bir konuma sahip bir ışık kaynağı.
- Sönümlenme: spot ışıklarda ve spot ışıklarda kullanılan mesafe boyunca yoğunluğunu azaltma işlemi.
- Spot: belirli bir yönde bir koni tarafından tanımlanan bir ışık kaynağı.
- Fener: izleyicinin bakış açısından konumlandırılmış bir spot ışığı.
- GLSL uniform dizisi: tekdüze değerler dizisidir. Dinamik olarak tahsis edilememesi dışında, bir C dizisi gibi çalışır.

Kaynaklar: <https://cgtranslators.gitbook.io/opengl-ogrenin/isiklandirma/basit-isiklandirma>

<https://cgtranslators.gitbook.io/opengl-ogrenin/isiklandirma/materyaller>