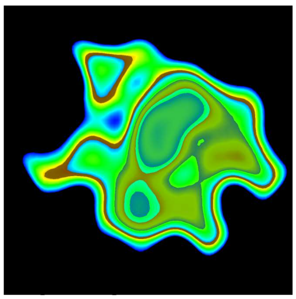
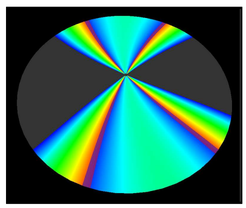
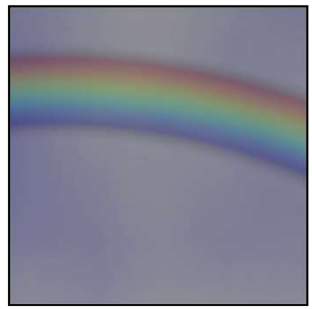
Spectral Effects: Chromatic Refraction and Wavelength Interference

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

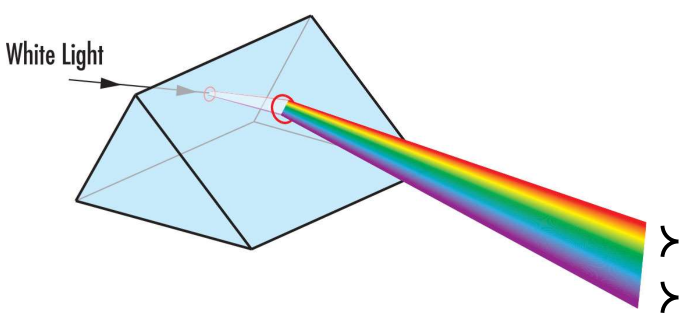
Oregon State University
Mike Bailey
mjb@cs.oregonstate.edu







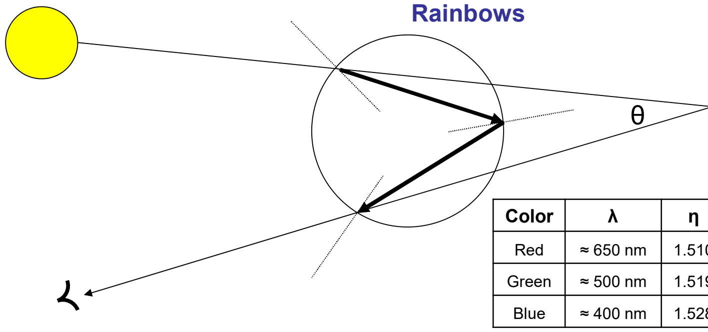
Each Wavelength of Light Has a Slightly Different Index of Refraction so that each Wavelength Bends Differently in a Prism



http://www.edmundoptics.com

Different colors are seen in different places

Rainbows



Color	λ	η	Θ	cosΘ	ΘΘ
Red	≈ 650 nm	1.510	42°	0.743	50.0°
Green	≈ 500 nm	1.519	41°	0.755	51.5°
Blue	≈ 400 nm	1.528	40°	0.766	53.0°

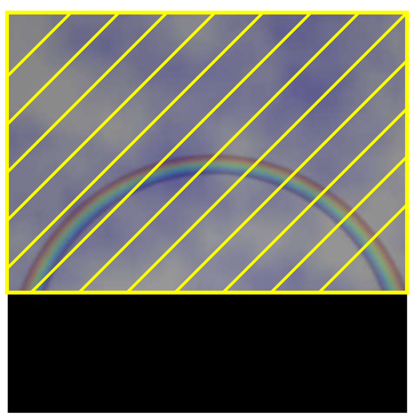
Primary Rainbow → Θ

Secondary Rainbow → ΘΘ

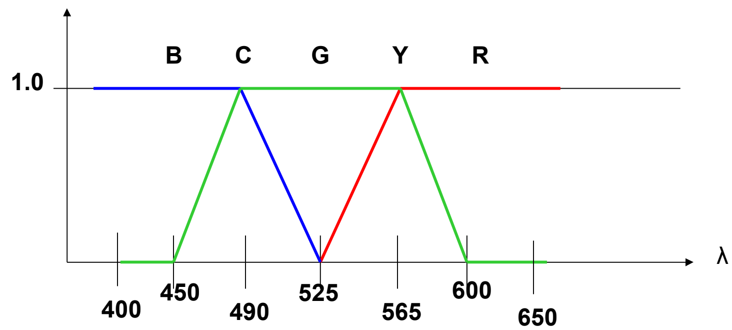
Rainbow Strategy

Or anything else, really. You just need a large “fragment-generator”.

1. Draw one big quadrilateral across the scene
2. Anywhere that .7400 ≤ cos(Θ) ≤ .7700, paint a color
3. Otherwise, discard.



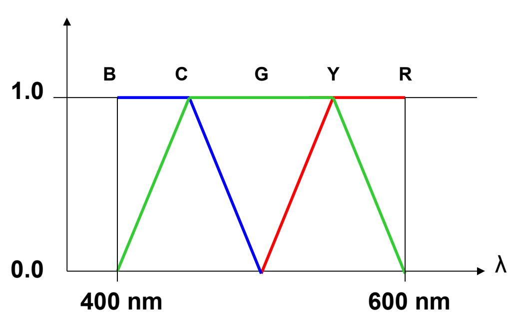
Spectral Colors




```
float
Pulse( float min, float max, float tol, float t )
{
   float a = min - tol;
   float b = min + tol;

   float c = max - tol;
   float d = max + tol;

   return  smoothstep(a,b,t) - smoothstep(c,d,t);
}
```

```
vec3 SunDirection = vec3( 0., SunY, 10. );
vec3 PtToSun = normalize( SunDirection );
vec3 PtToEye = normalize( vec3(0.,0.,0.) - ECposition );
float costheta = dot( PtToEye, PtToSun );

float R = Pulse( .7400, .7490, Tol, costheta );
float G = Pulse( .7490, .7605, Tol, costheta );
float B = Pulse( .7605, .7700, Tol, costheta );
```


Spectral Colors




```
float t = ( λ – 400. ) / ( 600. – 400. );     // 0. to 1.
vec3 rgb = Rainbow( t );
```

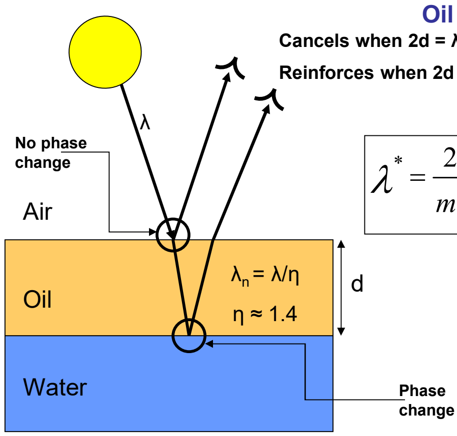
Changing the Range [0.,1.] to Rainbow Colors


```
vec3
Rainbow( float t )
{
      t = clamp( t, 0., 1. );
      vec3 rgb = vec3( 0., 0., 0. );

      // b -> c
      if( t >= 0. )
      {
            // rgb.r = 0.;
            rgb.g = 4. * ( t - (0./4.) );
            rgb.b = 1.;
      }

      // c -> g
      if( t >= (1./4.) )
      {
            // rgb.r = 0.;
            rgb.g = 1.;
            rgb.b = 1. - 4. * ( t - (1./4.) );
      }

      // g -> y
      if( t >= (2./4.) )
      {
            rgb.r = 4. * ( t - (2./4.) );
            rgb.g = 1.;
            // rgb.b = 0.;
      }

      // y -> r
      if( t >= (3./4.) )
      {
            rgb.r = 1.;
            rgb.g = 1. - 4. * ( t - (3./4.) );
            // rgb.b = 0.;
      }

      return rgb;
}
```

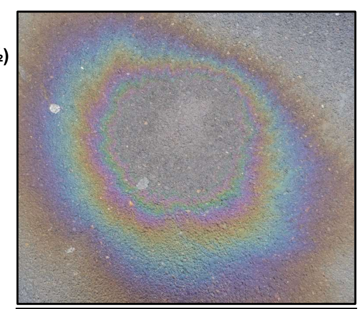

Oil Slicks

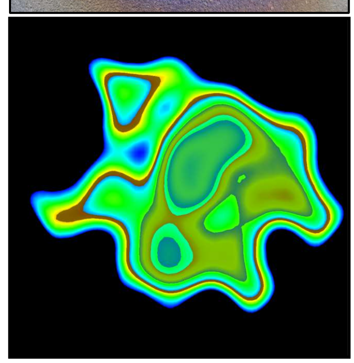
Cancels when 2d = $\lambda_n \cdot (m)$

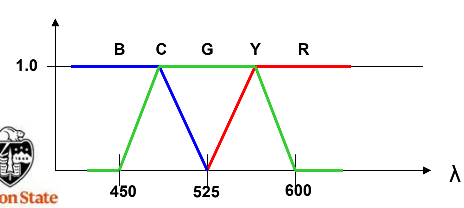
Reinforces when 2d = $\lambda_n \cdot (m+½)$



$$\lambda^* = \frac{2d\eta}{m+\frac{1}{2}}$$

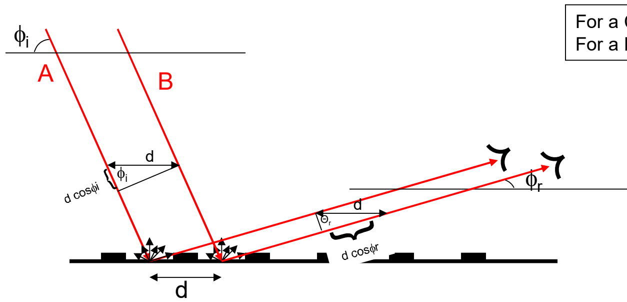


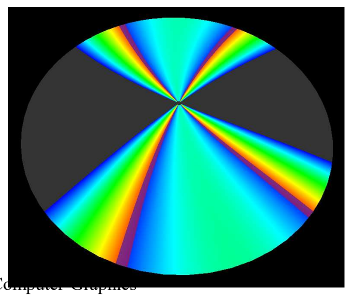




Diffraction Gratings

For a CD, d = 1600 nm
For a DVD, d = 740 nm



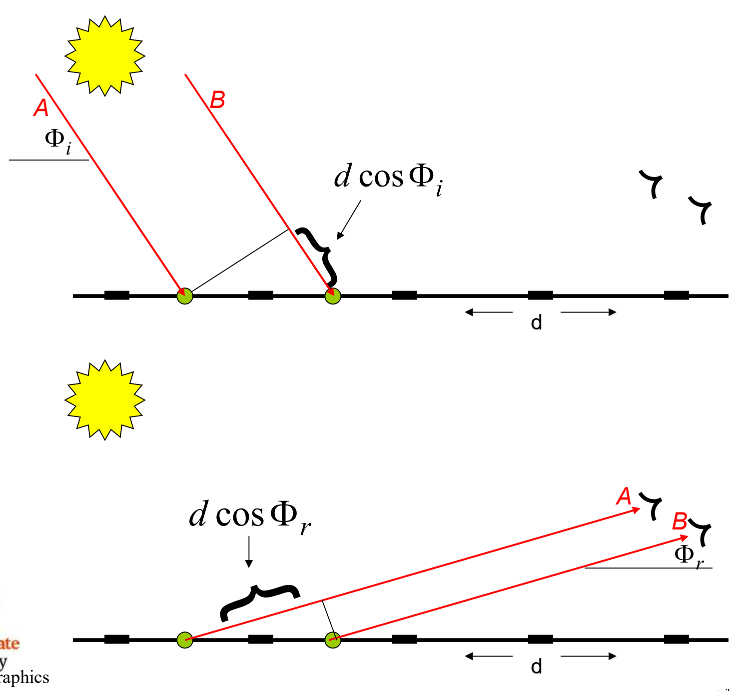


On the way in, Ray A travels *dcos(ϕ_i)* ***less*** than Ray B does.

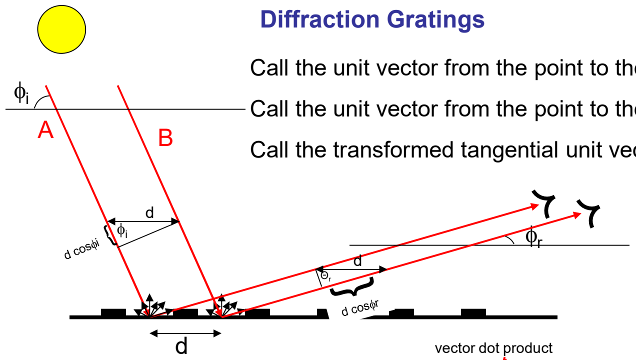
On the way out, Ray A travels *dcos(ϕ_r)* ***more*** than Ray B does.

So, wavelengths reinforce when abs[dcos(ϕ_i) - dcos(ϕ_r)] is a multiple of the wavelength = mλ

$$\lambda^* = d \cdot | \cos(\phi_i) - \cos(\phi_r) | \; / \; m$$



Diffraction Gratings



Call the unit vector from the point to the light **ToLight**.

Call the unit vector from the point to the eye **ToEye.**

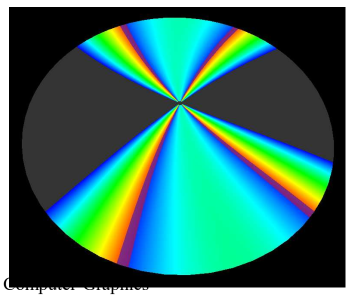
Call the transformed tangential unit vector **Tangent.**

vector dot product

Then, $\cos(\phi_i)$ is *ToLight•Tangent*

And, $\cos(\phi_r)$ is *ToEye•(-Tangent)*

So that $\cos(\phi i) - \cos(\phi r)$ is: *Tangent•(ToLight + ToEye)*



$$\lambda^* = d \cdot | \cos(\phi_i) - \cos(\phi_r) | \;/\; m$$