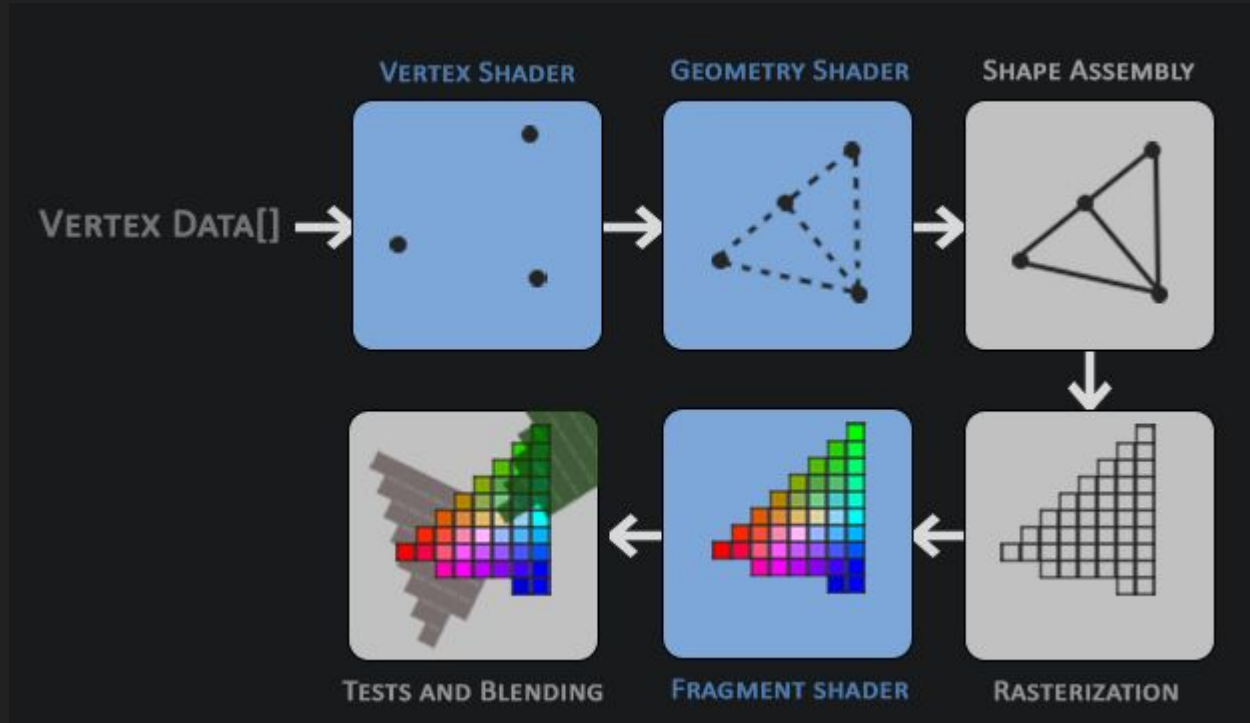


Cvičení 4

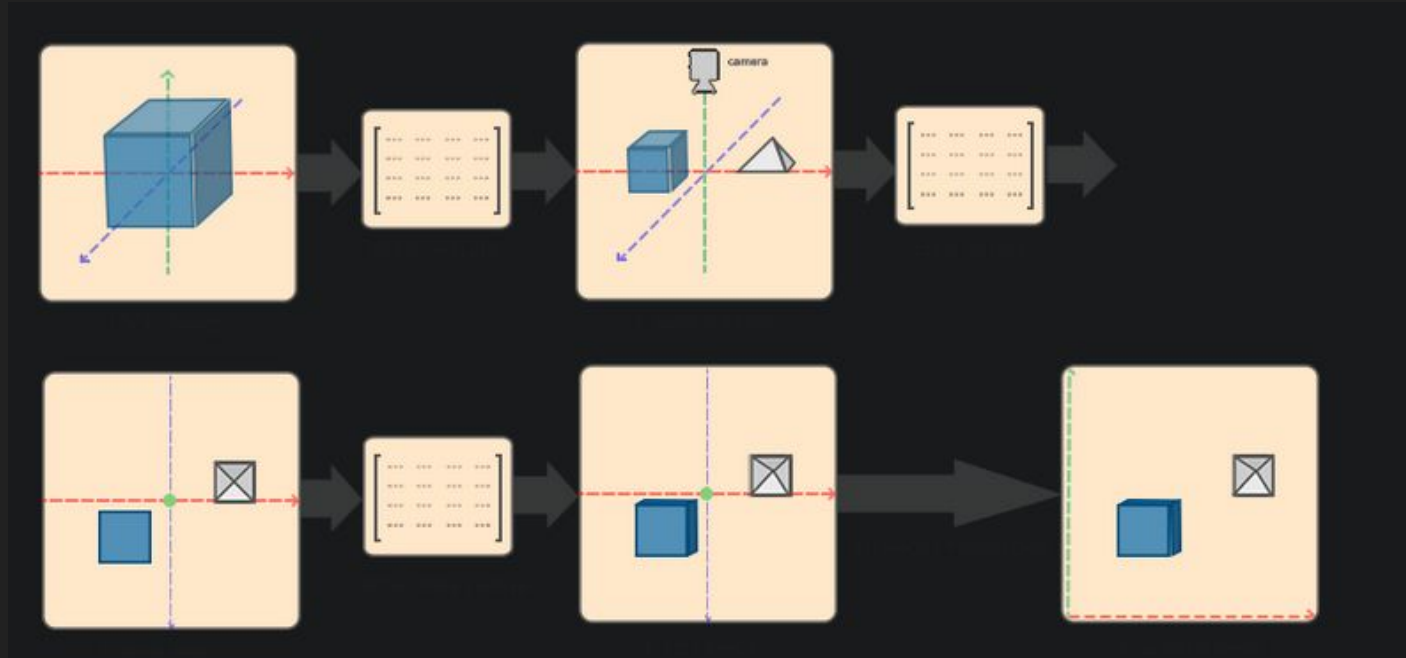
Nefotorealistické renderování

Rendering pipeline



Rendering pipeline

$\text{gl_Position} = \text{projection} * \text{view} * \text{model} * \text{vec4}(\text{aPos}, 1.0)$



Phong lighting

Ambient lighting - konstantní osvětlení v celé scéně

Diffuse lighting - model osvětlen světlem z určitého směru, nejvýraznější část

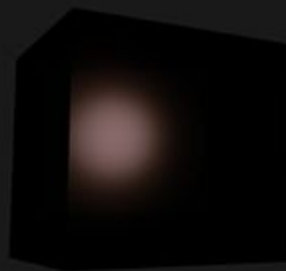
Specular lighting - simuluje odlesky



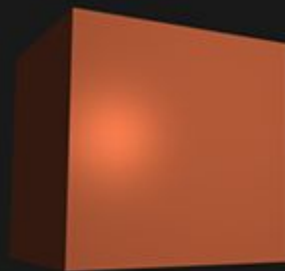
ambient



diffuse



specular



combined (Phong)

Ambient lighting

Malé hodnoty, stejné v celé scéně

=> konstantní osvětlení simulující roztroušené/odražené světlo

$\text{ambient} = \text{ambientStrength} * \text{lightColor}$

$\text{result} = \text{ambient} * \text{objectColor}$



Transformace do world space

Pro výpočet osvětlení musí být objekt i světlo ve stejném souřadnicovém systému

$$\text{pos} = \text{aPos} * \text{model}$$

Transformace normál povrchu

- Vektor nemá homogenní souřadnici ($w=0$)
- Transformujeme pomocí “normal matrix” = transpozice inverze levé horní 3x3 části matice model

$$\text{normal} = \text{aNormal} * \text{mat3}(\text{transpose}(\text{inverse}(\text{model})))$$

Diffuse lighting

Části objektu nakloněny směrem ke světlu => světlejší

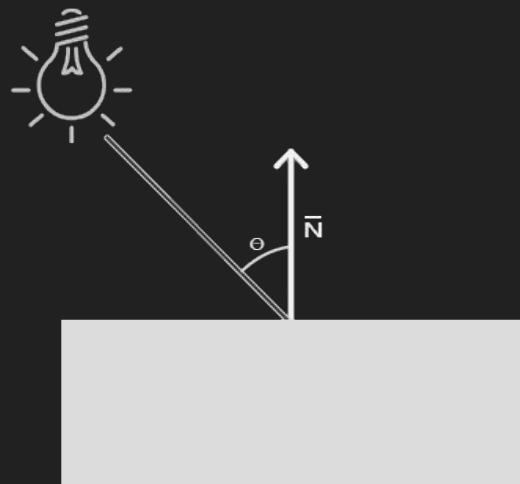
Úhel který fragment svírá s dopadajícími světelnými paprsky

- Úhel mezi normálou povrchu a vektorem ke světlu
- Vypočten jako skalární součin (Dot product)
 - Paralelní vektory => +1, kolmé vektory => 0

$\text{diff} = \max(\text{dot}(\text{norm}, \text{lightDir}), 0.0)$

$\text{diffuse} = \text{diff} * \text{lightColor}$

$\text{result} = (\text{ambient} + \text{diffuse}) * \text{objectColor}$



Specular lighting

Simuluje odlesky světla

“Kolik světla se odrazí do oka pozorovatele”

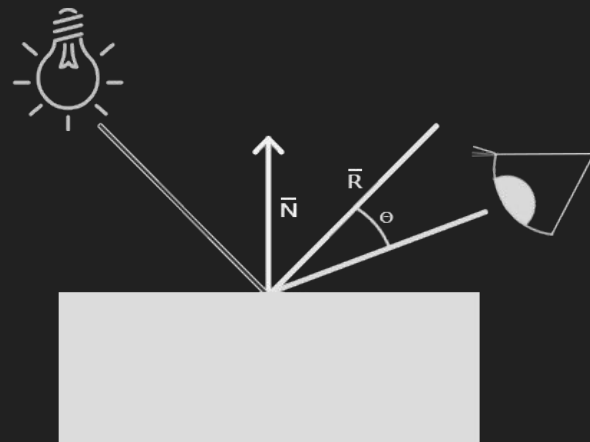
- Úhel mezi odraženým světlem a vektorem k pozorovateli

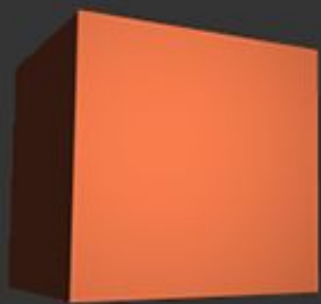
Jeho síla závisí na odrazivosti povrchu

```
spec = pow(max(dot(viewDir, reflectDir), 0.0), 32)
```

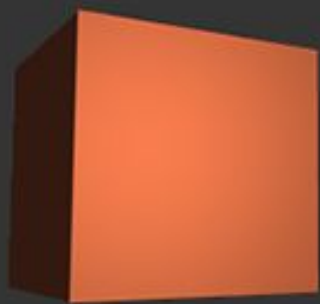
```
specular = specularStrength * spec * lightColor
```

```
result = (ambient + diffuse + specular) * objectColor
```

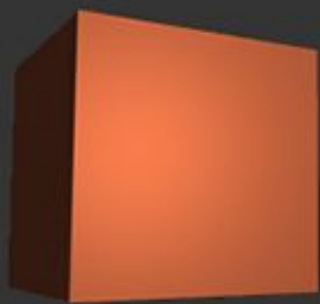




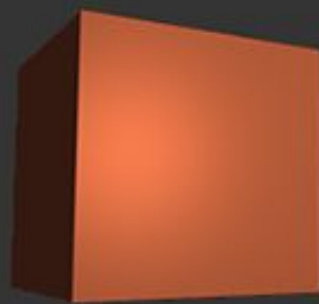
2



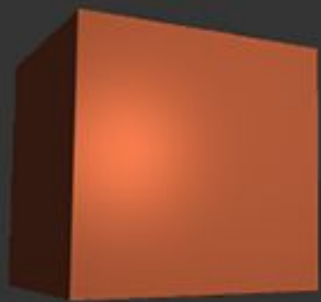
4



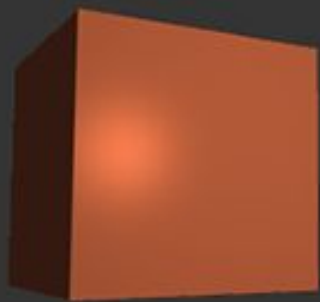
8



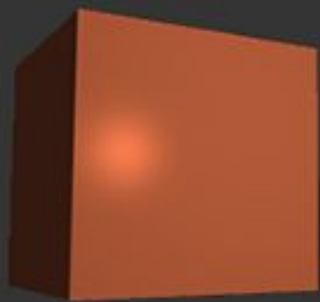
16



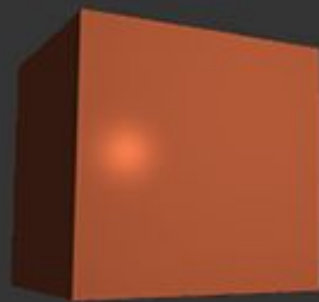
32



64



128



256

Cell shading

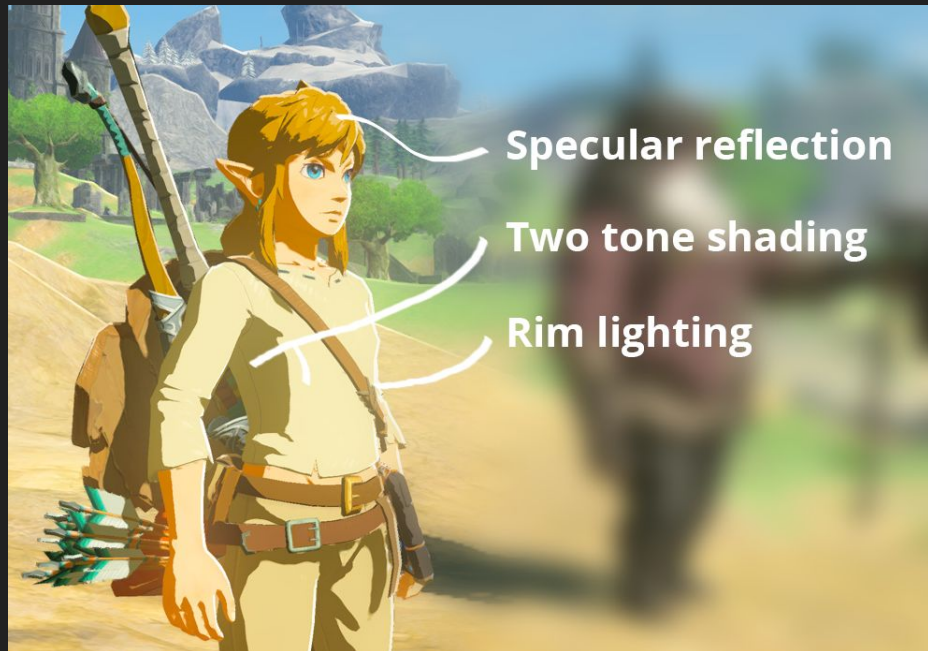
Omezený počet úrovní jasu

Upravíme diffuse lighting

$\text{toonLVL} = 4$

$\text{toonScale} = 1.0 / \text{toonLVL}$

$\text{diff} = \text{ceil}(\text{diff} * \text{toonLVL}) * \text{toonScale}$



Rim lighting

Osvětlení na hranách objektu

```
rimFactor = max(1.0 - dot(norm, -viewDir), 0.0)
```

```
rimFactor = pow(rimFactor, 3)
```

Barva osvětlení např. konstanta

```
rimColor = vec3(1.0, 0.0, 0.0) * rimFactor
```