

# THREE.JS – OSVETĽOVANIE, SVETELNÉ ZDROJE, MATERIÁLY

doc. Ing. Branislav Sobota, PhD.

Ing. Marián Hudák, Ing. Lenka Bubeňková

Katedra počítačov a informatiky, FEI TU v Košiciach

C 08

© 2024

# CIELE CVIČENIA

- Three.js – osvetľovanie pojmy a príprava
- Three.js - práca so štandardnými druhmi svetelných zdrojov a materiálov.
- Three.js - implementácia vlastných realistických materiálov.

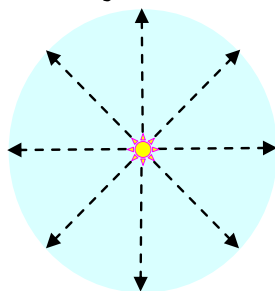
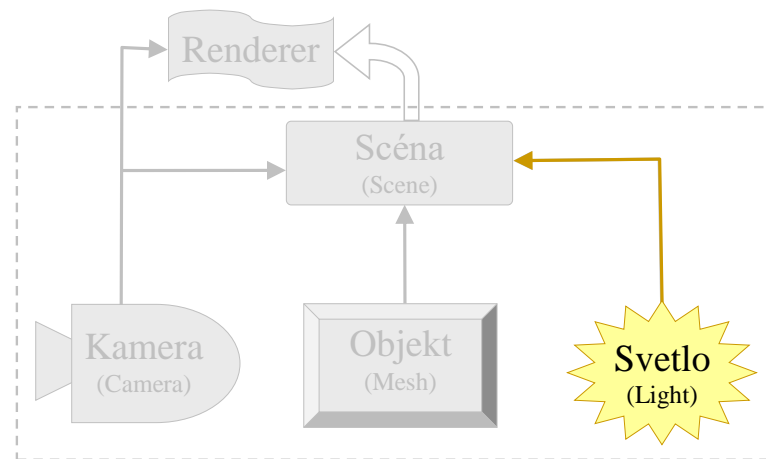


# 1. *THREE.JS* – PRÍPRAVA BALÍČKA

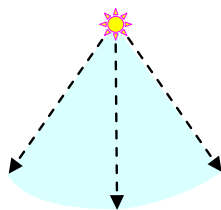
- **Úloha:** Stiahnite si balíček „**Threejs\_Osvetlovanie\_a\_materialy.zip**“ z portálu Moodle KPI a predmetu Počítačova grafika.
- Obsah balíčka skopírujte do vášho projektu aby štruktúra vyzerala nasledovne:
  - WebGL\_getStart
    - > css
    - > js
    - >> threejs
    - >> ThreeEffects.js
    - models
    - > texture
    - index.html

# 1. THREE.JS – OSVETĽOVANIE - POJMY A PRÍPRAVA

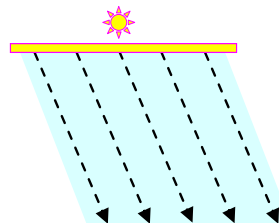
- **Osvetľovanie v Three.js** je nadstavbou nad možnosťami osvetľovania vo WebGL.
- **Závisí od :**
  - typu svetelného prvku/zdroja a
  - druhu používaného materiálu.
- **Jednotlivé typy** základných **svetelných zdrojov** sú ekvivaletné svetelnými zdrojmi vo WebGL.



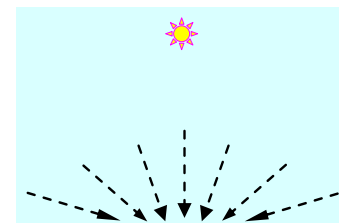
Bodové svetlo  
(point light)



Reflektorové svetlo  
(spot light)



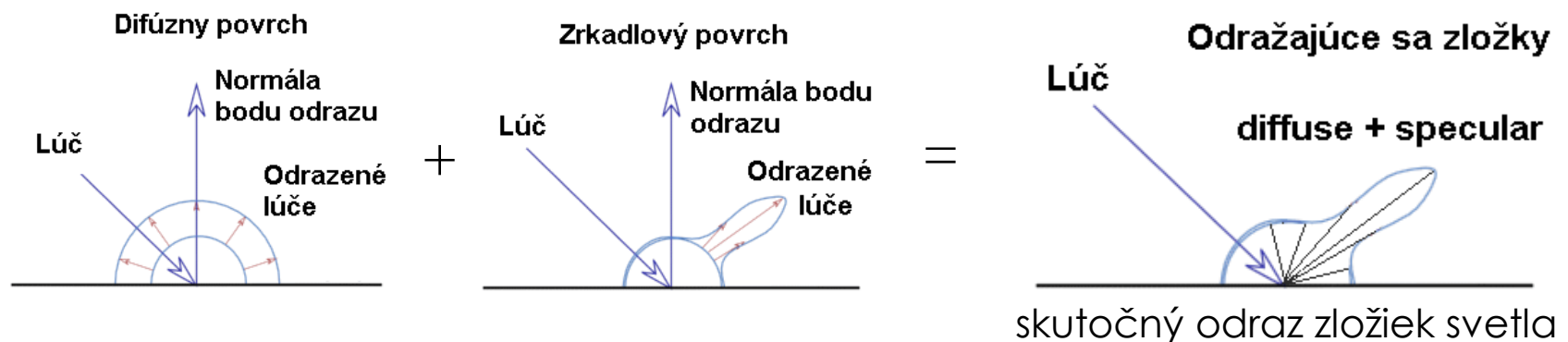
Smerové svetlo  
(directional light)



Okolité (ambientné) svetlo  
(ambient light)

# 1. THREE.JS – ZLOŽKY SVETLA A OSVETĽOVACÍ MODEL

- **Základné zložky svetla sú:**
  - Ambientna zložka (ambient light)
  - Difúzna zložka (Diffuse)
  - Zrkadlová zložka (Specular)
  - ... (napr. odrazená, lomená zložka)
- **Osvetľovací model** definuje aké zložky a akým spôsobom vytvárajú výsledný svetelný lúč (napr. Phongov osvetľovací model)

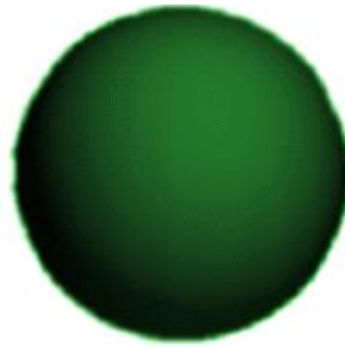


# 1. *THREE.JS* – MATERIÁLY

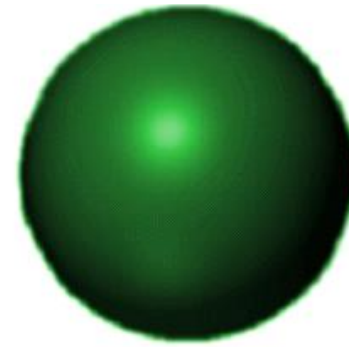
- Druhy základných materiálov v *Three.js* sú:
  - základný materiál (*MeshBasicMaterial*),
  - materiál s podporou **Lambertovho** osvetľovacieho modelu (*MeshLambertMaterial*) a
  - materiál s podporou **Phongovho** osvetľovacieho modelu (*MeshPhongMaterial*)



základný materiál  
(*MeshBasicMaterial*)



materiál s Lambertovým  
osvetľovacím modelom  
(*MeshLambertMaterial*)



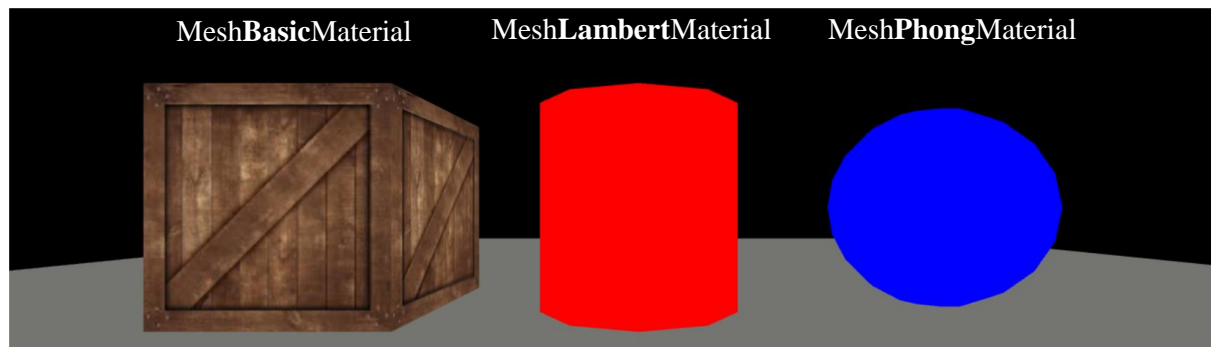
materiál s Phongovým  
osvetľovacím modelom  
(*MeshPhongMaterial*)

## 2. *THREE.JS* – PRÍPRAVA SCÉNY A JEJ AMBIENTNÉHO OSVETLENIA

- **Úloha:** precvičte prácu s ambientným svetlom.
1. Otvorte si skript „**ThreeEffects.js**“ a html súbor „**index.html**“ vo vašom vývojovom prostredí.
  2. V HTML súbore „**index.html**“ *prepíšte cestu predošlého skriptu z „**ThreeScene.js**“ cestou pre skript „**ThreeEffects.js**“.*

```
<script src="js/ThreeEffects.js"></script>
```

3. Spustíte si vizualizáciu vo webovom prehliadači.



V metóde **addLights()** je implementované *ambientné svetlo* (okolité svetlo) bielej farby.

## 2. THREE.JS - IMPLEMENTÁCIA OBJEKTU BODOVÉHO SVETELNÉHO ZDROJA

- **Úloha:** precvičte prácu s bodovým svetlom.

1. V metóde **addLights()** zakomentujte implementáciu ambientného svetla.
2. Následne v metóde vytvorte **objekt bodového svetelného zdroja** :

```
var pointLight = new THREE.PointLight('rgb(255,255,255)',2, 100);
pointLight.position.set( 1, 2, 2 );
scene.add( pointLight );
```

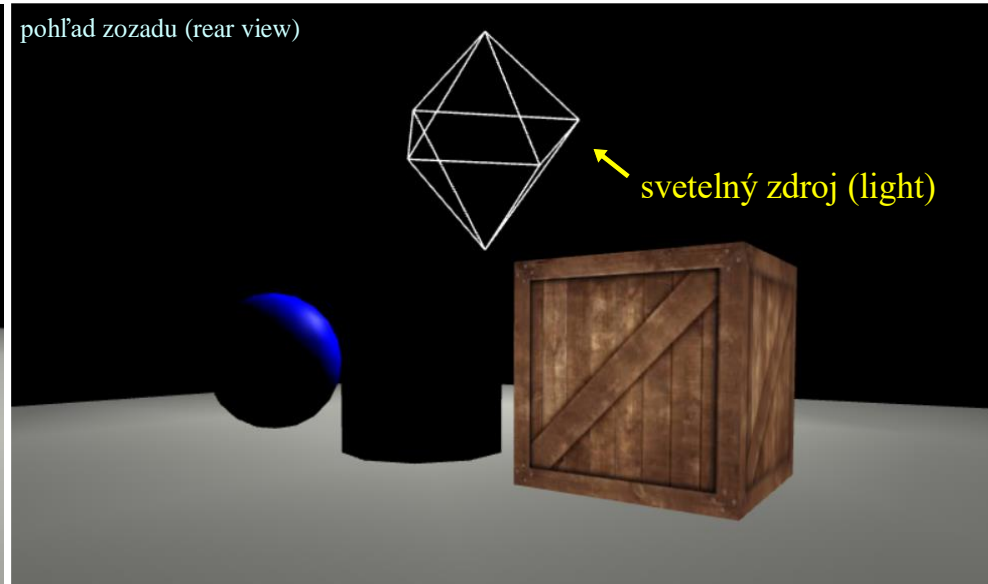
3. Aby bolo ohraničenie objektu bodového svetelného zdroja viditeľné, dodatočne použite v implementácii kód:

```
var sphereSize = 1;
var pointLightHelper = new THREE.PointLightHelper( pointLight, sphereSize );
scene.add( pointLightHelper );
```



## 2. THREE.JS - IMPLEMENTÁCIA OBJEKTU BODOVÉHO SVETELNÉHO ZDROJA

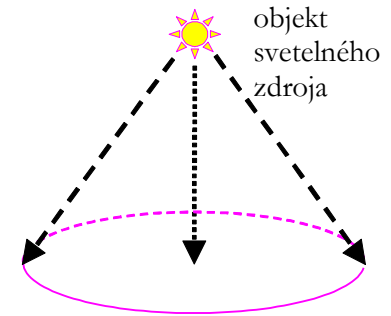
- **Úloha:** precvičte prácu s bodovým svetlom.
4. Spustíte si vizualizáciu vo webovom prehliadači.  
Bodové svetlo sa vizuálne prejavuje pri používaní všetkých druhov materiálov okrem **MeshBasicMaterial**.



## 2. THREE.JS - IMPLEMENTÁCIA OBJEKTU REFLEKTOROVÉHO SVETELNÉHO ZDROJA

- **Úloha:** precvičte prácu s reflektorovým svetlom.
1. V metóde **addLights()** zakomentujte implementáciu bodového svetla.
  2. Následne v metóde vytvorte **objekt reflektorového svetelného zdroja** :

```
var spotlight = new THREE.SpotLight('rgb(255,255,255)');
spotlight.angle = Math.PI/6;
spotlight.position.set(1, 3, 2);
spotlight.intensity = 2;
scene.add(spotlight);
```



3. Aby bolo ohraničenie objektu reflektorového svetelného zdroja viditeľné, dodatočne použite v implementácii kód:

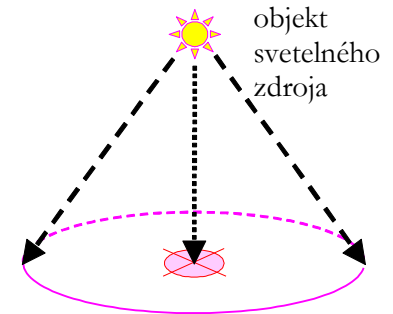
```
var spotlightHelper = new THREE.SpotLightHelper( spotlight );
scene.add( spotlightHelper );
```

## 2. THREE.JS - IMPLEMENTÁCIA OBJEKTU REFLEKTOROVÉHO SVETELNÉHO ZDROJA

- **Úloha:** precvičte prácu s reflektorovým svetlom.

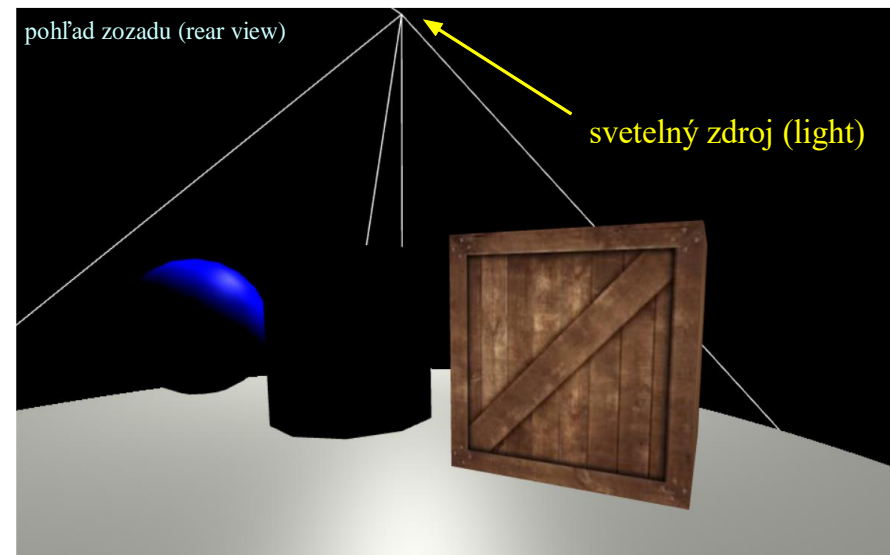
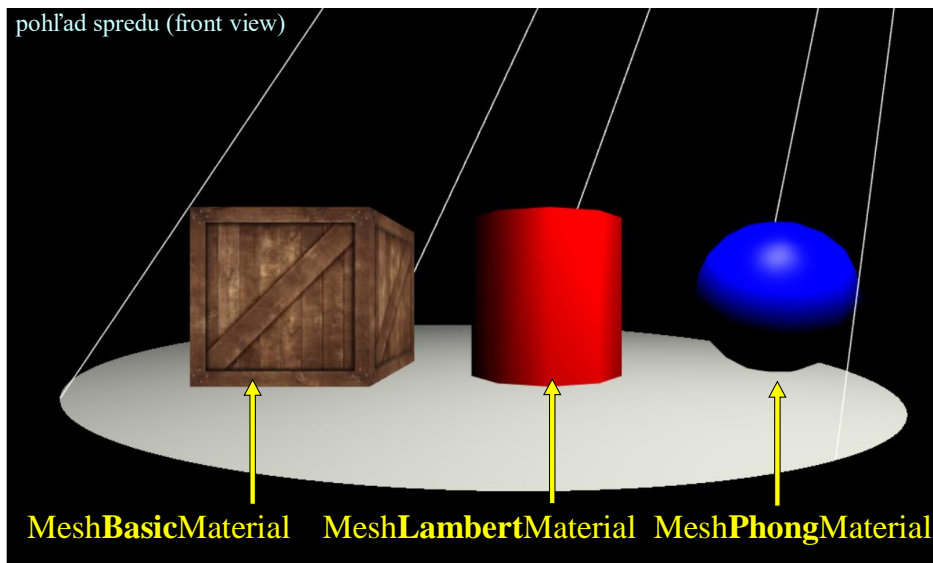
4. **Lúč reflektorového svetla je možné smerovať na určený cieľ**, vytvorený bod v trojrozmernom prostredí. Skúste v metóde **addLights()** implementovať nasledujúci skript :

```
var lightTarget = new THREE.Object3D();
lightTarget.position.set(0,0,0);
scene.add(lightTarget);
spotlight.target = lightTarget;
```



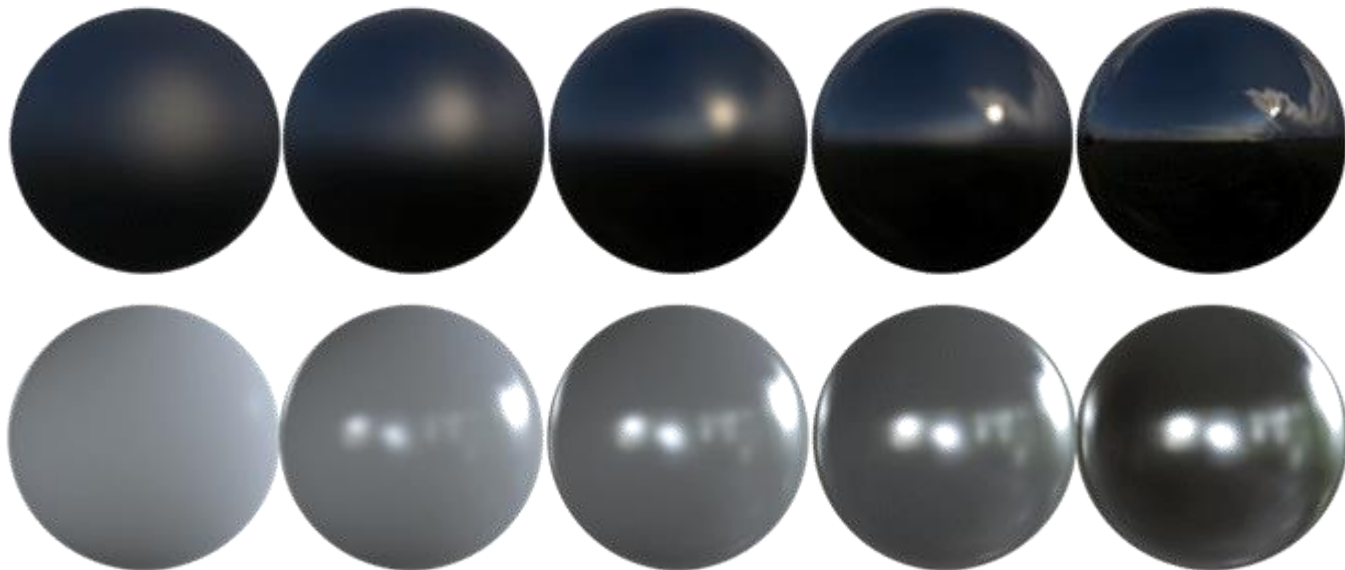
## 2. THREE.JS - IMPLEMENTÁCIA OBJEKTU REFLEKTOROVÉHO SVETELNÉHO ZDROJA

- **Úloha:** precvičte prácu s reflektorovým svetlom.
5. Spustíte si vizualizáciu vo webovom prehliadači.  
Reflektorové svetlo sa vizuálne prejavuje pri používaní všetkých druhov materiálov okrem **MeshBasicMaterial**.



### 3. THREE.JS – IMPLEMENTÁCIA VLASTNÝCH REALISTICKÝCH MATERIÁLOV

- **Three.js – realistický materiál.**
- Pri implementácii vlastného realistického materiálu **v Three.js je dôležitá najmä príprava materiálov**, ktoré reprezentujú špecifickú fyzikálnu vlastnosť objektu: lesk a odraz dopadajúceho svetla. Tieto vlastnosti sa používajú pri **Physically Based Rendering (PBR)** a sú využívané pre realistický vnem.



### 3. THREE.JS - IMPLEMENTÁCIA VLASTNÉHO MATERIÁLU S POUŽITÍM MESHSTANDARDMATERIAL

- **Úloha:** implementujte vlastný realistický materiál (**MeshStandardMaterial**) na objekt kocky.
- 1. V metóde **addLights()** implementujte nasledujúci **objekt reflektorového svetelného zdroja:**

```
var spotlight = new THREE.SpotLight('rgb(0,140,255)');
spotlight.angle = Math.PI/6;
spotlight.position.set(-2.5, 2, 2);
spotlight.intensity = 2;
scene.add(spotlight);
var spotlightHelper = new THREE.SpotLightHelper( spotlight );
scene.add( spotlightHelper );
var lightTarget = new THREE.Object3D();
lightTarget.position.set(0,0,0);
scene.add(lightTarget);
spotlight.target = lightTarget;
```

### 3. THREE.JS - IMPLEMENTÁCIA VLASTNÉHO MATERIÁLU S POUŽITÍM MESHSTANDARDMATERIAL

- **Úloha:** implementujte vlastný realistický materiál (**MeshStandardMaterial**) na objekt kocky.
- 2. V metóde **addObjects()** implementujte vlastný typ materiálu, ktorý priradíte objektu cube:

```
var materialCube = new THREE.MeshStandardMaterial( {
    map: textureCube,
    side: THREE.DoubleSide,
    roughness : 0.12,
    metalness: 0.65} );
```

- 3. V metóde **render()** nastavte objektu **cube** animáciu rotácie okolo osi x :

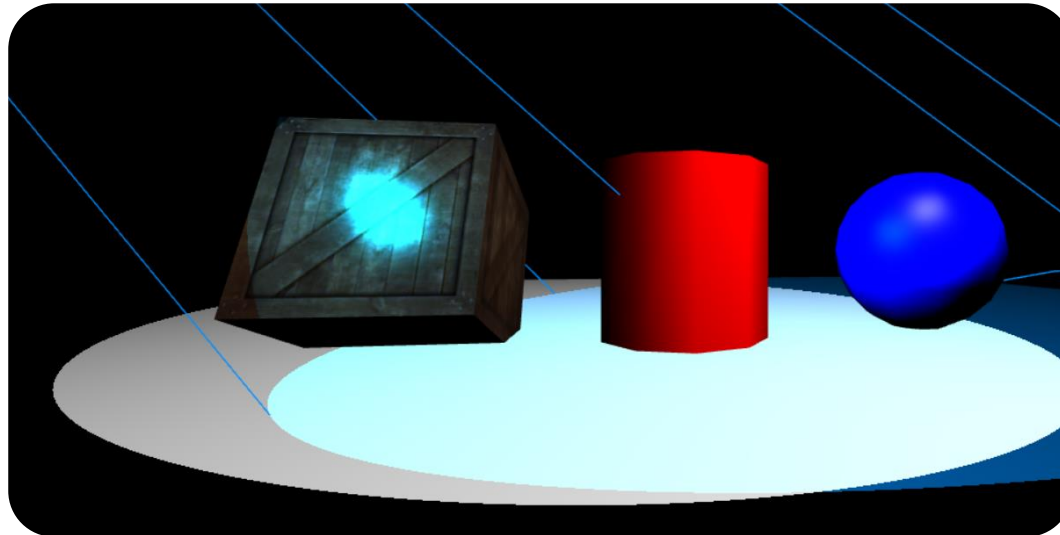
```
cube.rotation.x += 0.01;
```



- Všimnite si zvýraznené parametre materiálu **roughness** (drsnosť) a **metalness** (metalická vlastnosť materiálu).
- Vyskúšajte pracovať s hodnotami parametrov **roughness** a **metalness**, **všímajte si zmeny vlastností materiálu**.
- Odkaz na opis **THREE.[MeshStandardMaterial](#)**

### 3. THREE.JS - IMPLEMENTÁCIA VLASTNÉHO MATERIÁLU S POUŽITÍM MESHSTANDARDMATERIAL

- **Úloha:** implementujte vlastný realistický materiál (**MeshStandardMaterial**) na objekt kocky.
4. Následne otestujte implementáciu vo webovom prehliadači.





### 3. THREE.JS - IMPLEMENTÁCIA VLASTNÉHO MATERIÁLU S POUŽITÍM MESHSTANDARDMATERIAL

- **Úloha:** Pre lepšie zdôraznenie vlastného materiálu implementujte vlastné reflektorové svetlo.

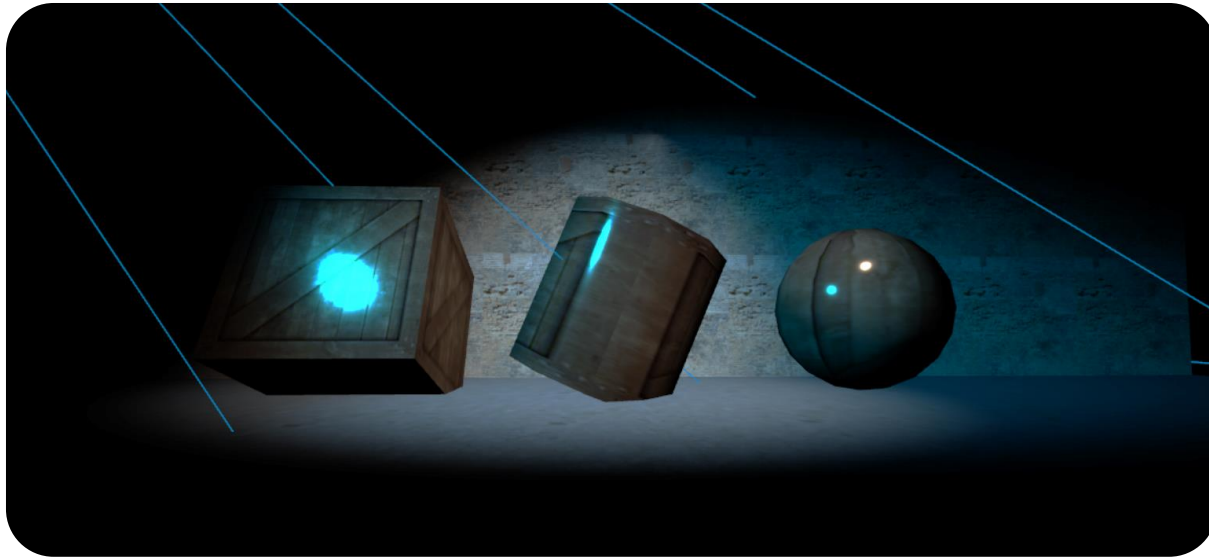
5. Vyskúšajte zameniť rôzne parametre reflektorového svetla:

```
SpotLight( color      : Integer, intensity : Float,
            distance  : Float,  angle      : Radians,
            penumbra   : Float,  decay      : Float )
```

- **color** – farba svetelného lúča,
- **intensity** – intenzita svietivosti,
- **distance** – maximálna vzdialenosť dopadu svetelného lúča,
- **angle** – uhol disperzie, pod ktorým sa lúče šíria do okolia, štandardná hodnota =  $\text{Math.PI}/2$ .
- **penumbra** - ostrosť svetelných lúčov pri dopade na plochy ostatných objektov,
- **decay** – úroveň stmievania svetelného lúča od zdroja svetelného objektu po cieľ dopadu.

### 3. THREE.JS - IMPLEMENTÁCIA VLASTNÉHO MATERIÁLU S POUŽITÍM MESHSTANDARDMATERIAL

- **Úloha:** Pre lepšie zdôraznenie vlastného materiálu odskúšajte vlastné reflektorové svetlo.
6. Vyskúšajte zameniť rôzne parametre reflektorového svetla:



# DOPLŇUJÚCE ÚLOHY

1. Podľa predchádzajúceho cvičenia implementujte na **svetelný zdroj bodového svetla** možnosť **ovládania jeho polohy** aspoň v jednej osi **pomocou klávesnice**. Riadiace klávesy si zvolte.
2. Reimplementujte **vlastný materiál** s použitím textúry **loga LIRKIS**.
3. Podľa predchádzajúceho cvičenia **implementujte objekt**, ktorého **pohyb bude ovládateľný vstupom z klávesnice**. Následne implementujte reflektorové **svetlo**, ktoré **bude objekt sledovať**.



# ÚLOHY NA SAMOSTATNÉ RIEŠENIE

1. Podľa predchádzajúceho cvičenia k objektu s kolesami naviažte objekt reflektorového svetla a pohybujte s ním zároveň s objektom.
2. Overte si vizuálny efekt ak na jedno teleso pôsobia proti sebe dve reflektorové svetlá pri existencii jedného základného bodového svetelného zdroja a bez neho. Experimentujte zároveň s úrovňou ambientného svetla.
3. Implementujte v scéne efekt modrého rotačného policajného majáku (postačuje svetelný zdroj, bez príslušného objektu majáku).



# Q & A

[branislav.sobota@tuke.sk](mailto:branislav.sobota@tuke.sk)  
[lenka.bubenkova@tuke.sk](mailto:lenka.bubenkova@tuke.sk)

Katedra počítačov a informatiky, FEI TU v Košiciach

© 2024