***лекція 4.***

***Слайд 1***

***Тіні***

Рендеринг тіней вимагає великих обчислювальних потужностей, і з цієї причини за замовчуванням тіні відключено в Three.js. Однак увімкнути їх дуже просто.

Для накладання тіней на сцені потрібно попередньо «включити» їх у рендері (***prime04-00.html***):

renderer.shadowMap.enabled = true;

renderer.shadowMap.type = THREE.PCFSoftShadowMap; // фільтрує карти тіней з

// використанням алгоритму процентної фільтрації (PCF) з кращими м'якими

// тінями, особливо при використанні карт тіней з низькою роздільною здатністю.

Далі необхідно увімкнути утворення тіней біля джерела світла

directionalLight.castShadow = true;:

var directionalLight = New THREE.DirectionalLight( 0xffffff );

directionalLight.position.set(-10, 100, 0);

directionalLight.castShadow = true; // Якщо встановлено значення true, це

//світло відкидає тіні.

directionalLight.shadowCameraNear = 2; // Тіньова камера перспективи

// біля параметра. Типово - 50 .

directionalLight.shadowCameraFar = 200; // Далекий параметр усіченої

/ / Піраміди камери перспективи.

// Визначення усіченої області тіньової камери,

// яка використовується для розрахунку карти тіней джерела світла.

directionalLight.shadowCameraLeft = -50;

directionalLight.shadowCameraRight = 50;

directionalLight.shadowCameraTop = 50;

directionalLight.shadowCameraBottom = -50;

directionalLight.distance = 0;

directionalLight.intensity = 0.5;

directionalLight.shadowMapHeight = 1024; // Висота текстури карти тіней у пікселях.

/ / За замовчуванням - 512 .

directionalLight.shadowMapWidth = 1024; // Ширина текстури карти тіней у пікселях.

// За умовчанням - 512.

scene.add(directionalLight);

***Слайд 2***

Також властивість castShadow потрібно включити для тіл, від яких потрібно відкидання тіней.

var cubeGeometry = New THREE.CubeGeometry(4,4,4);

var cubeMaterial = new THREE.MeshBasicMaterial({color: 0xff0000});

var cube = New THREE.Mesh(cubeGeometry, cubeMaterial);

cube.position.x = -4;

cube.position.y = 10;

cube.position.z = 0;

cube.castShadow = true;

scene.add(cube);

Щоб об'єкти сцени сприймали тіні, потрібно включити у них властивість receiveShadow:

var planeGeometry\_2 = New THREE.PlaneGeometry(60,60,1,1);

var planeMaterial\_2 = new THREE.MeshPhongMaterial( { color: 0xff00ff, side: THREE.DoubleSide });

var plane2 = New THREE.Mesh(planeGeometry\_2,planeMaterial\_2);

plane2.rotation.x= Math.PI/2;;

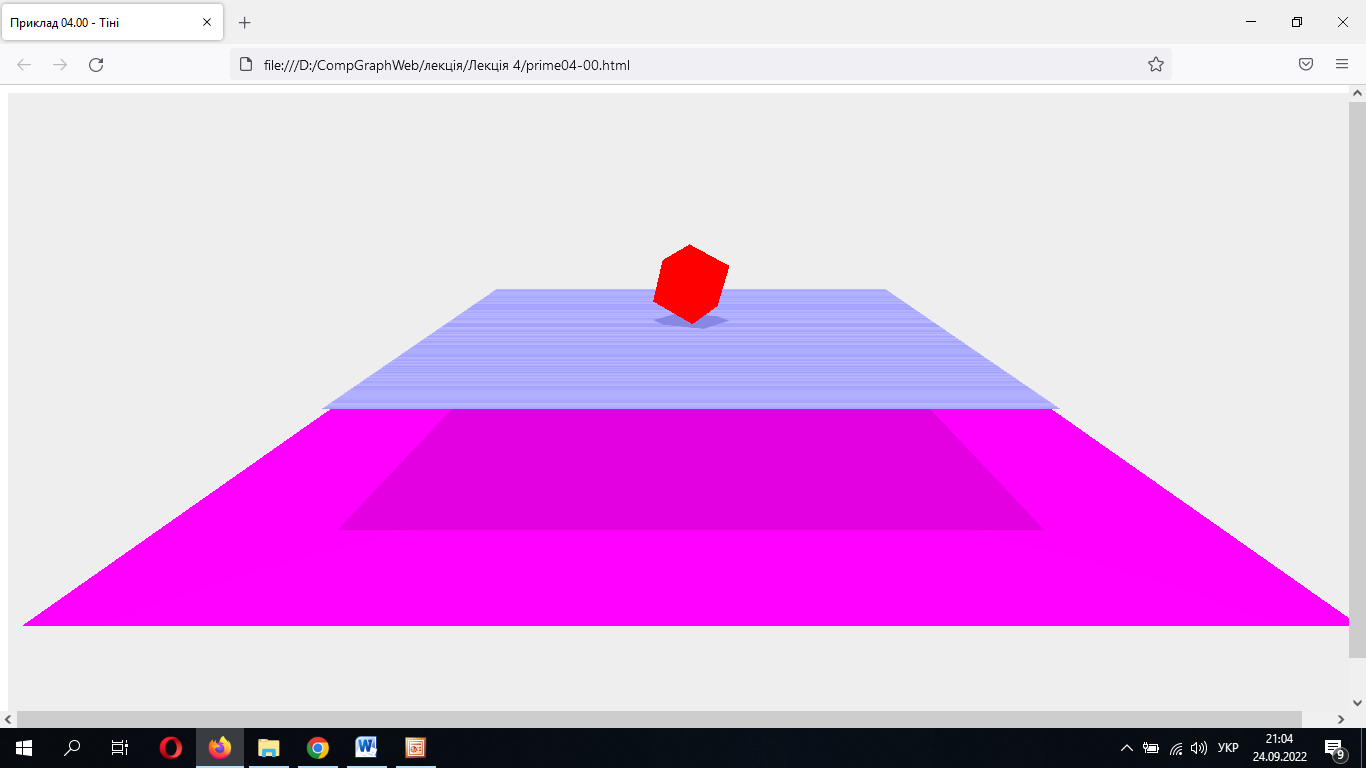
plane2.position.y = 0;

plane2.position.z = 0;

plane2.receiveShadow = true;

scene.add(plane2);

***Слайд 3***



Далі в лекції розглянемо, як використовувати різні види освітлення у three.js.

***Слайд 4***

***AmbientLight***

*AmbientLight*– це освітлення, яке освітлює всю сцену. Воно немає напряму і освітлює кожен об'єкт сцени однаково, незалежно від розташування об'єкта. Відповідно, цього світла немає позиції на осі координат. Це світло не може використовуватися для створення тіней.

**Конструктор**

AmbientLight (color, intensity)

***color***- Число RGB компонентів кольору. Значення за замовчуванням дорівнює 0xffffff (білий).

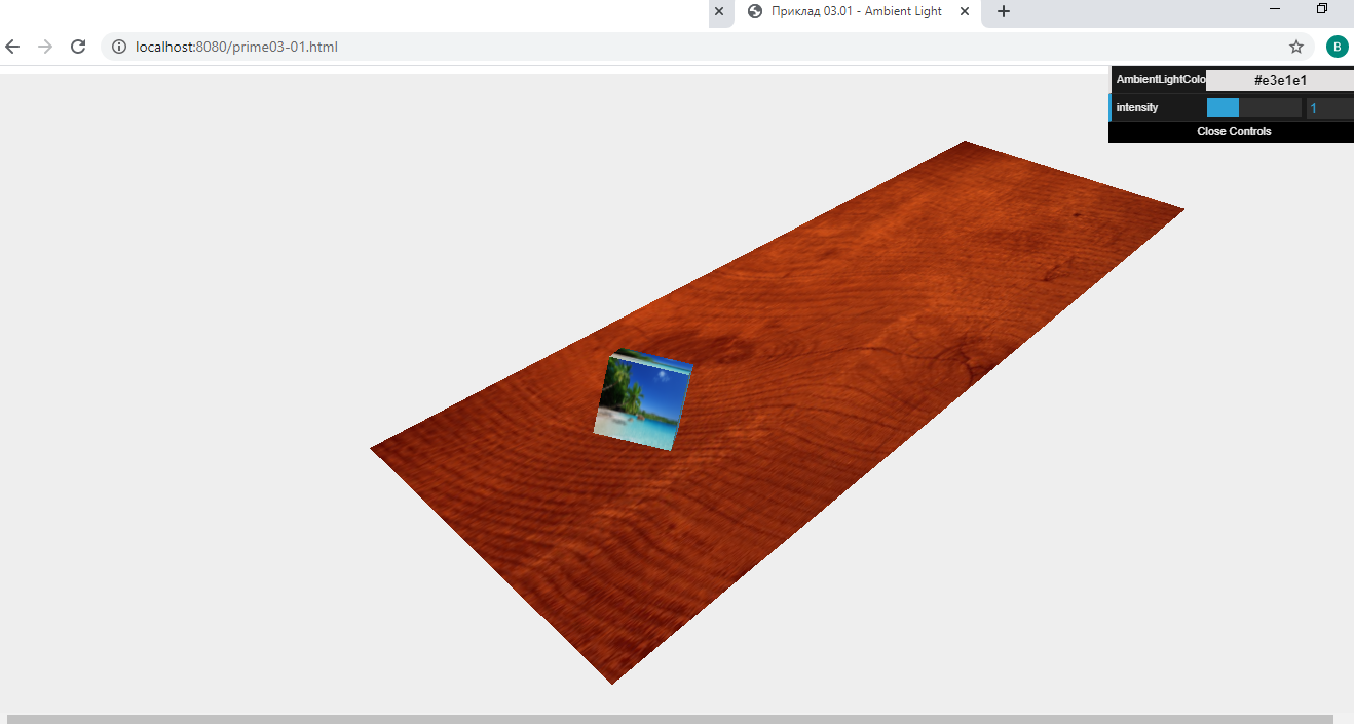
Додатковий, необов'язковий аргумент

***intensity***- Чисельне значення сили/інтенсивності світла. Значення за замовчуванням дорівнює 1.

Конструктор створює навколишнє (загальне) освітлення із заданими кольором та інтенсивністю.

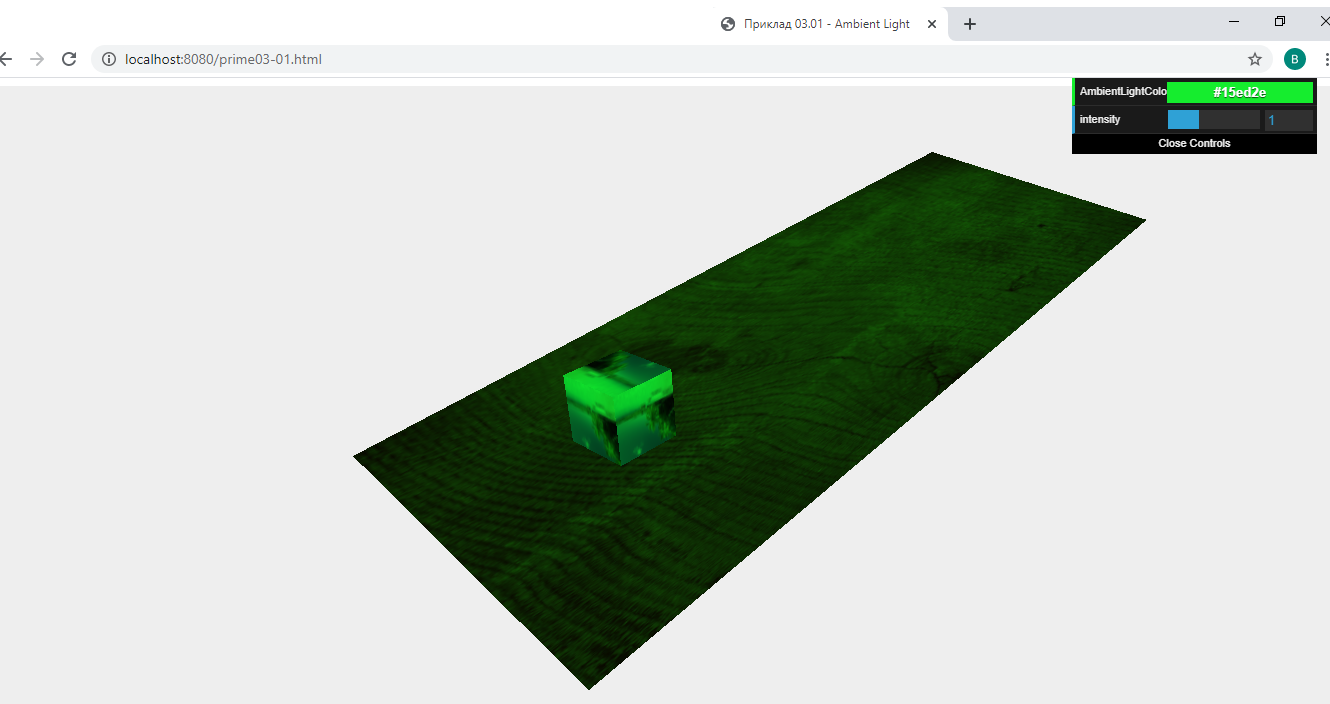
Ми не використовуємо AmbientLight як єдине джерело світла в сцені. Ми будемо використовувати його з іншими джерелами світла, такими як SpotLight або DirectionalLight, щоб пом'якшити тіні або додати колір.

У цьому прикладі (***prime04-01.html***) ми отримали простий інтерфейс користувача за допомогою бібліотеки ***dat.GUI***, який можна використовувати для зміни джерела світла AmbientLight, яке додається до сцени.



Якщо ми хочемо змінити цей колір на помітніший зелений колір, ми можемо зробити це в меню у правому верхньому кутку; об'єкти тоді будуть мати світло-зелене світло над ними. Це показано на наступному слайді:

***Слайд 5***



Як показано на скріншоті, зелений колір застосовується до всіх об'єктів та висвітлює всю сцену зеленим свіченням.

Розглянемо як ми можемо створити і використовувати джерело AmbientLight. Наступна пара рядків коду покаже нам, як створити джерело AmbientLight, а також як підключити його до меню керування dat.GUI:

***Слайд 6***

var ambiColor = "#e3e1e1";

var ambiIntensity = 1;

var ambientLight = New THREE.AmbientLight(ambiColor,ambiIntensity);

scene.add(ambientLight);

Додамо наступний JavaScript об'єкт:

var controls = new function () {

this.rotationSpeed = 0.02;

this.AmbientLightColor = ambiColor;

this.intensity = 1;

};

Далі ми передаємо цей об'єкт у новий об'єкт dat.GUI і визначаємо діапазон цих трьох властивостей, як показано нижче:

var gui = new dat.GUI();

gui.addColor(controls, 'AmbientLightColor').onChange(function (e) {

ambientLight.color = New THREE.Color(e);

});

gui.add(controls, 'intensity', 0, 3).onChange(function (e) {

ambientLight.intensity = e;

});

Створення джерела AmbientLight дуже очевидно. Оскільки джерело AmbientLight не має позиції, нам потрібно лише вказати колір (у шістнадцятковому форматі), використовуючи новий new THREE.AmbientLight(ambiColor,ambiIntensity).

У цьому прикладі зв'язали колір джерела AmbientLight з елементом управління.

У коді ми використовували функцію onChange Data файлу.

***Слайд 7***

***PointLight***

*PointLight*- освітлення, яке виходить з однієї точки у всіх напрямках (це джерело освітлення можна порівняти зі звичайною лампочкою).

#### Конструктор

PointLight( color, intensity, distance, decay )

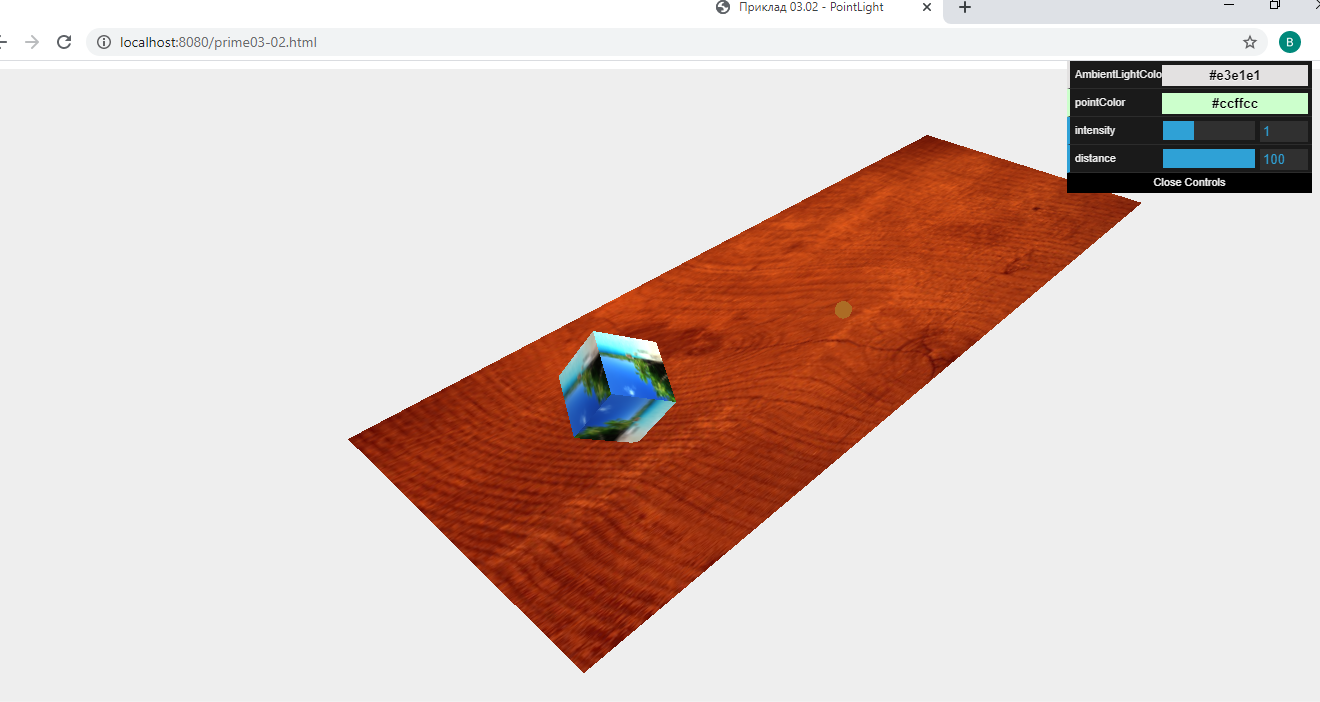
***color***- шістнадцяткове значення кольору освітлення за замовчуванням 0xffffff (білий). Додатковий, необов'язковий аргумент.

***intensity***- Чисельне значення сили/інтенсивності світла. Значення за замовчуванням дорівнює 1. Додатковий, необов'язковий аргумент.

***distance*** - Відстань від джерела світла, при якому інтенсивність світла дорівнює 0. Значення за замовчуванням дорівнює 0.

***decay*** - Значення, на яке зменшується інтенсивність світла зі збільшенням відстані від джерела.

Як і у випадку з усіма джерелами світла, ми маємо конкретний приклад, який ми можемо використовувати, щоб розібратися з PointLight. У прикладі розглядається переміщення джерела PointLight навколо простої сцени Three.js (***prime04-02.html***).



У цьому прикладі джерело PointLight рухається сценою. Щоб було зрозуміліше, де джерело PointLight: ми переміщаємо маленьку сферу тим самим шляхом.

Коли джерело світла рухатиметься, ми побачимо куб, освітлений цим світлом.

Розглянемо, як створити джерело PointLight:

***Слайд 8***

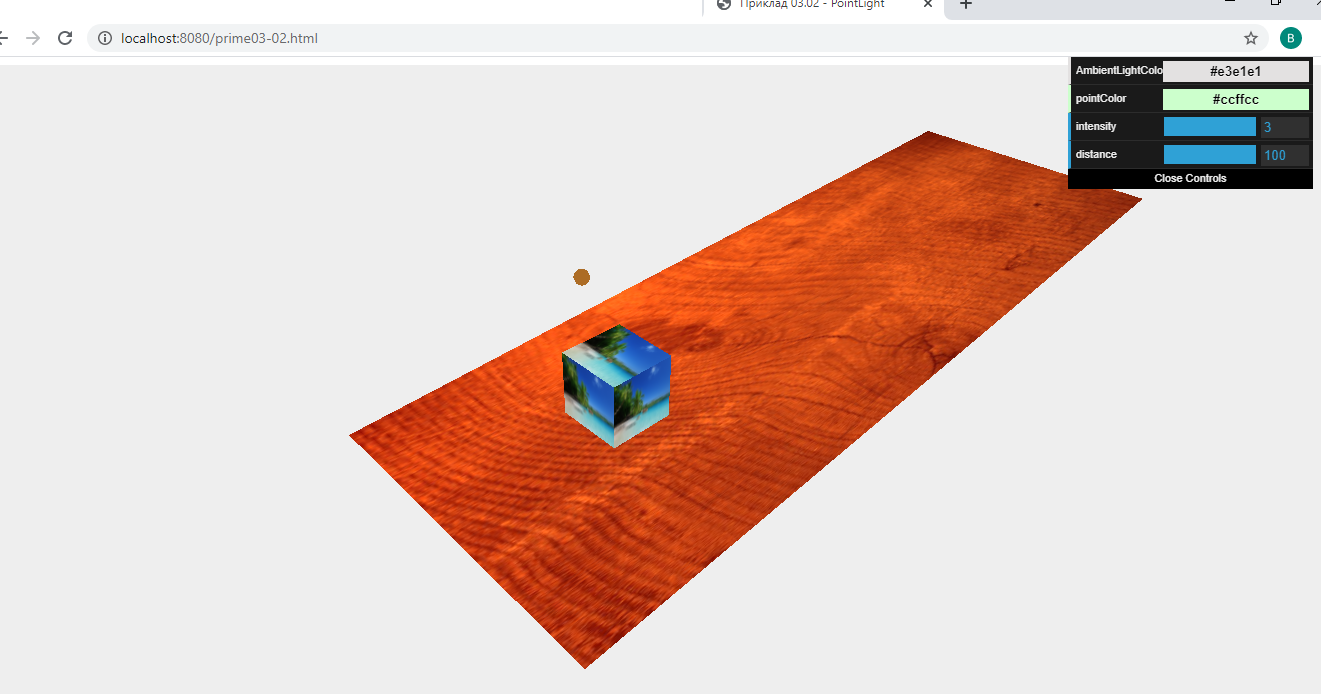
var pointColor = "#ccffcc"; //блакитний

var pointLight = New THREE.PointLight(pointColor);

pointLight.distance = 100;

scene.add(pointLight);

Тут нічого нового. Ми створили джерело світла певного кольору, встановили його положення і додали його до сцени. Перша властивість, яку ми розглянемо, – це інтенсивність. З цим ми можете встановити, наскільки яскраво світитиме світло. Якщо ми встановимо його на 0, ми не побачимо нічого, встановимо його на 1, і ми маємо яскравість за замовчуванням, а встановимо його на 3, і ми отримаємо світло, яке світитиме втричі яскравіше. На наступному знімку екрана, наприклад, ми встановили властивість інтенсивності світла 3:



Все, що вам потрібно зробити, щоб змінити властивість інтенсивності світла, це таке:

pointLight.intensity = 3;

***Слайд 9***

Ми також можемо використовувати DAT.GUI, як показано в наступному фрагменті коду:

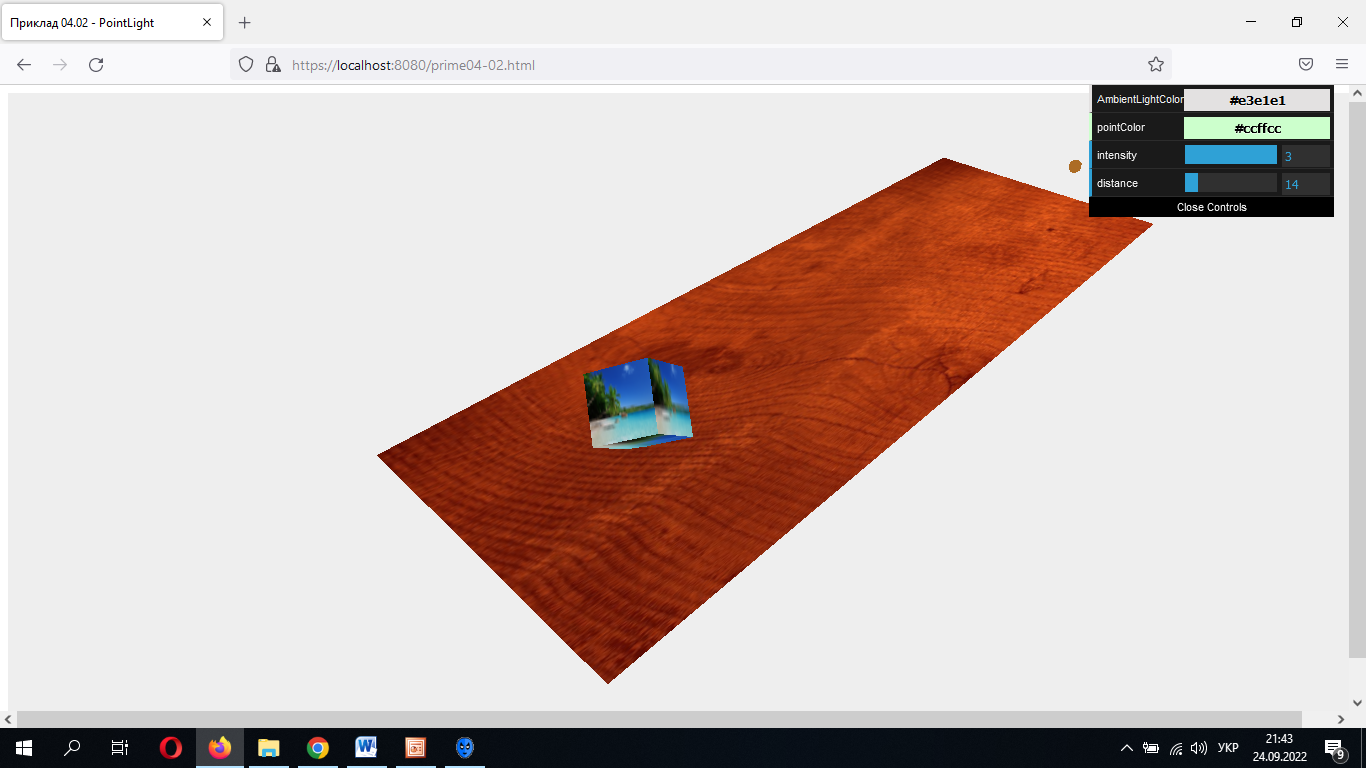
var controls = new function() {

this.intensity = 1; }

var gui = new dat.GUI();

gui.add(controls, 'intensity', 0, 3).onChange(function(e) { pointLight.intensity = e; });

Остання властивість джерела PointLight - дуже цікава і найкраща, пояснимо на прикладі. На наступному скріншоті ми побачимо ту саму сцену. знову, але цього разу з дуже високою інтенсивністю (оскільки у нас дуже яскраве світло), але з невеликою відстанню:



Властивість distance джерела SpotLight визначає, як далеко буде світло від джерела. Ми можемо встановити властивість так:

pointLight.distance = 14;.

У нашому прикладі яскравість (інтенсивність) світла повільно зменшуватиметься до 0 при відстань 14. Значення за замовчуванням для властивості distance дорівнює 0, яке означає, що світло не гасне на відстані.

***Слайд 10***

***SpotLight***

SpotLight - світло з ефектом конуса, є одним із джерел світла, яке ми використовуватимемо найчастіше (особливо якщо будемо використовувати тінь). Це світло має напрямок і кут, під яким воно випромінює світло. Світло PointLight також застосовується до SpotLight. Світло SpotLight також має низку додаткових властивостей:

|  |  |
| --- | --- |
| **Властивість** | **Опис** |
| castShadow | Якщо встановлено значення true, це світло відкидає тіні. |
| shadowCameraNear | Тінь камери перспективи біля параметра. Типово - 50 . |
| shadowCameraFar | Далекий параметр зрізаної піраміди камери перспективи. За замовчуванням – 5000. |
| shadowCameraFov | Параметр поля зору зрізаної камери перспективи. Типово - 50 . |
| target | Визначає, куди спрямоване світло. |
| shadowBias | Може використовуватися для усунення позиції візуалізованої тіні. |
| angle | Наскільки великий промінь від джерела світла. Це вимірюється у радіанах. За замовчуванням Math.PI/3. |
| exponent | Це значення визначає, наскільки швидко інтенсивність світла зменшується. |
| тільки Shadow | Якщо встановлено значення true, це світло лише відкидатиме тінь і не буде додано до сцени світло. |
| shadowCameraVisible | Якщо встановлено значення true, ми можемо побачити, як і де джерело світла відкидає тінь |
| shadowDarkness | За замовчуванням 0,5. Визначає, наскільки темним буде рендерінг тіні. Не може бути змінено після рендерингу сцени |
| shadowMapWidth | Ширина текстури карти тіні в пікселях. За замовчуванням – 512. |
| shadowMapHeight | Висота текстури карти тіні в пікселях. За замовчуванням - 512 . |

***Слайд 11***

Приклад створення світла SpotLight:

var pointColor = "#ffffff"; //білий

var spotLight = New THREE.SpotLight(pointColor);

spotLight.position.set(-40, 60, -10);

spotLight.castShadow = true;

spotLight.shadowCameraNear = 2;

spotLight.shadowCameraFar = 200;

spotLight.shadowCameraFov = 30; // поле зору

spotLight.target = plane;

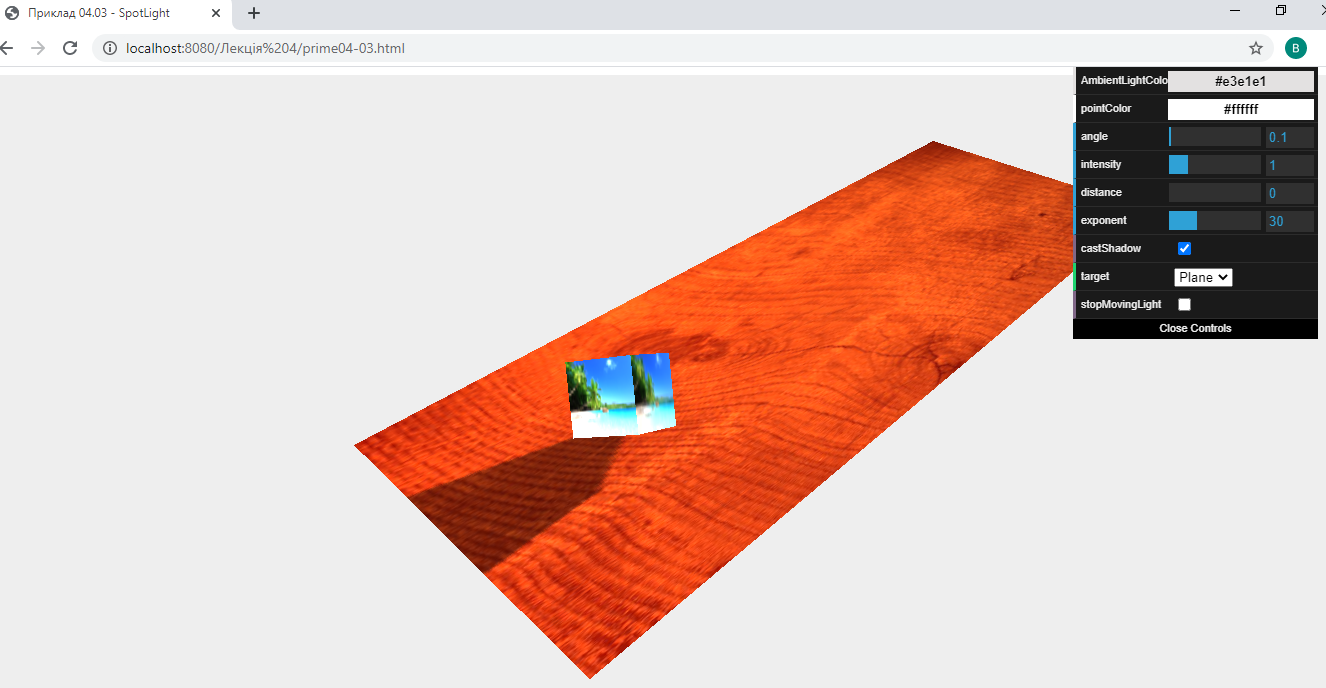
spotLight.distance = 0;

spotLight.angle = 0.4;

scene.add(spotLight);

***Слайд 12***

Не дуже відрізняється від джерела PointLight. Єдина різниця в тому, що ми встановимо для якості castShadow значення true, тому що нам потрібні тіні.



У цьому прикладі ми можемо встановити низку властивостей для SpotLight. Одне з них target – визначає, куди спрямоване світло.

У прикладі, ми створили джерело світла та направили його на площину. Ми також можемо направити його на другий об'єкт – куб. Але якщо ми не хочемо направляти світло на конкретний об'єкт, а довільну точку простору, то ми можемо зробити це, створивши порожній екземпляр THREE.Object3D () наступним чином:

var target = новий THREE.Object3D();

target.position = новий THREE.Vector3(5, 0, 0);

І встановимо якість джерела SpotLight, як показано:

spotlight.target = target;

У таблиці раніше в цій лекції ми розглянули кілька властивостей, які можна використовувати, щоб контролювати, як світло походить від джерела SpotLight. Властивості відстані та кута визначають форму конуса. Властивість angle визначає ширину конуса та за допомогою властивості відстані ми можемо встановити довжину конуса.

var coneLength = light.distance? light.distance: 10000;

var coneWidth = coneLength \* Math.tan (light.angle \* 0.5) \* 2;

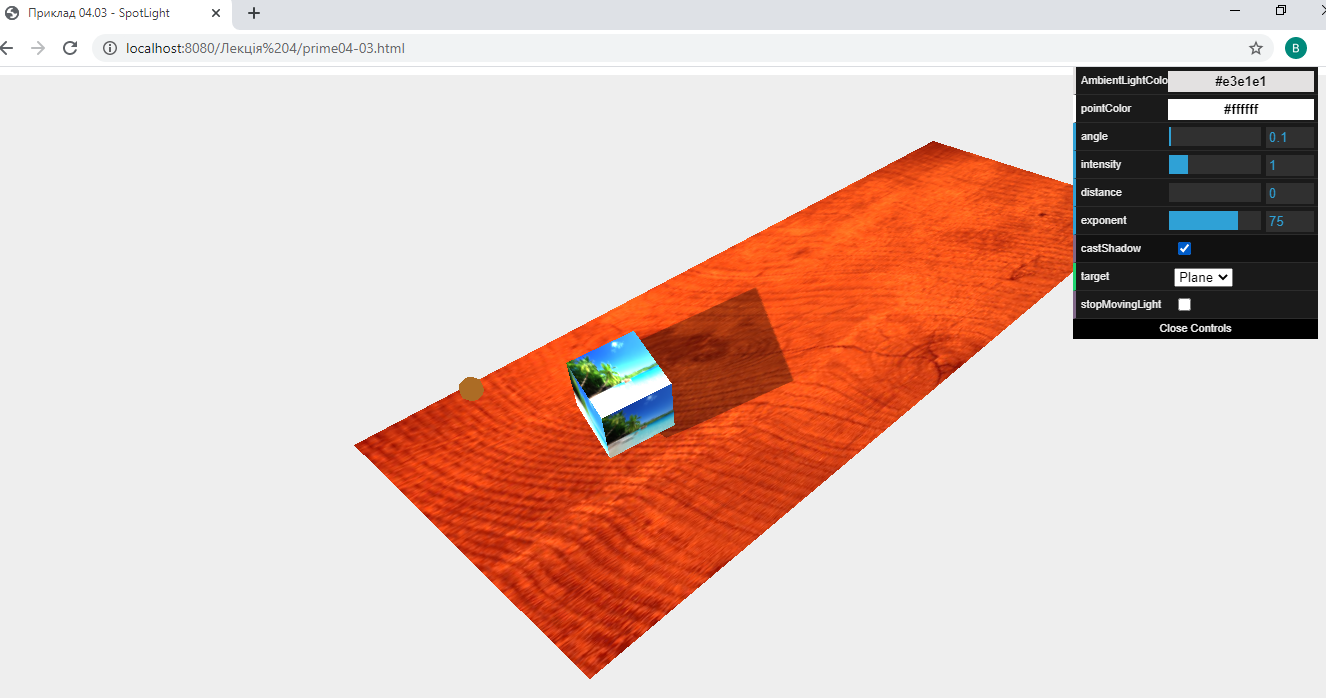
На наступному рисунку показано, як кут та довжина конуса визначають ширину конуса:

angle

conelength

cone width

Зазвичай нам не потрібно встановлювати ці значення, оскільки вони мають розумні значення за замовчуванням, але ми можемо використовувати ці властивості, наприклад для створення джерел SpotLight, які мають дуже вузький промінь або швидке зменшення інтенсивності світла. Остання властивість, яку ми розглянемо - це, щоб змінити сприйняття світла SpotLight, можна використовувати властивість експоненти. За допомогою цієї властивості ми можемо встановити, наскільки швидко зменшуватиметься інтенсивність світла від центру світлового конуса, як показано на наступному рисунку:



У нас дуже яскраве світло (висока інтенсивність), інтенсивність, яка швидко зменшується (високий показник) у міру віддалення від центру. Ми також могли б створити той самий ефект сфокусованого променя за рахунок використання малого значення показника ступеня та малого кута.

Перш ніж перейти до такого джерела світла, ми розглянемо тіні, тобто. властивості тіней доступні джерелу SpotLight. Ми вже розглянули раніше, що ми можемо отримати тіні, встановивши властивість castShadow джерела SpotLight як істина.

Бібліотека Three.js також дозволяє дуже детально контролювати, як рендерує тінь. Це робиться за допомогою декількох властивостей, які ми розглянули в таблиці раніше в цій лекції. За допомогою shadowCameraNear, shadowCameraFar та shadowCameraFov, ми можемо контролювати, як і де це світло буде відкидати тінь.

Це працює так само, як поле зору перспективної камери. Найпростіший спосіб побачити це в дії - встановити для властивості shadowCameraVisible значення true

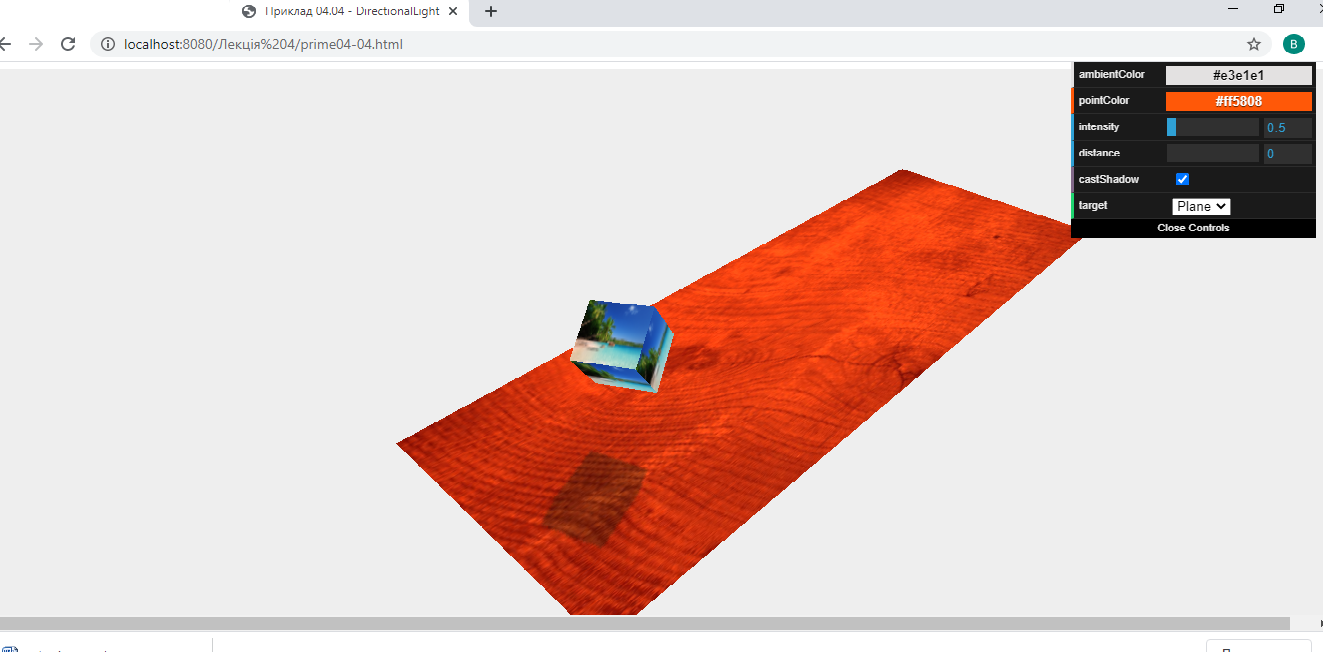
***DirectionalLight.***

Джерело DirectionalLight можна використовувати як джерело світла, яке є дуже далеко. Все світло промені, які він випромінює, є паралельними один одному. Хорошим прикладом є сонце.

Основна відмінність між джерелом DirectionalLight і SpotLight полягає в тому, що це світло не зменшуватиметься в міру віддалення від нього.

Вся область, освітлена джерелом DirectionalLight, отримує однакову інтенсивність світла.

Щоб побачити це у дії, розглянемо приклад (***prime04-04.html***), який показано на наступному знімку екрана:



Як бачите, на сцену не поширюється світловий конус. Все одержує таку ж інтенсивність світла. Як і у випадку з джерелом SpotLight, є кілька властивостей, які ми можемо встановити, контролювати інтенсивність світла та те, як він відкидає тіні. DirectionalLight має безліч властивостей, аналогічних властивостям SpotLight: target, intensity, distance, castShadow, onlyShadow, shadowCameraNear, shadowCameraFar, shadowDarkness, shadowCameraVisible, shadowMapWidth, shadowMapHeight і shadow.

Для джерела DirectionalLight:

оскільки всі промені паралельні один одному, у нас замість конуса світла є куб.

Все, що потрапляє в куб, може відкидати та отримувати тіні від світла. Як і у випадку з джерелом SpotLight, чим точніше ми визначимо область навколо об'єктів, тим краще виглядатимуть наші тіні. Ми можемо задати куб, використовуючи такі властивості:

directionalLight.shadowCameraNear = 2;

directionalLight.shadowCameraFar = 200;

directionalLight.shadowCameraLeft = -50;

directionalLight.shadowCameraRight = 50;

directionalLight.shadowCameraTop = 50;

directionalLight.shadowCameraBottom = -50;