

Лабораторна робота №4. Динамічне освітлення: налаштування джерел світла та Shadow Maps у Three.js

1. Мета роботи

- **Дослідити** типи джерел світла та їх фізичні характеристики (інтенсивність, спадання, колір).
- **Навчитися** налаштовувати систему генерування тіней (Shadow Mapping).
- **Опанувати** методи оптимізації тіней для забезпечення високої продуктивності веб-застосунку.

2. Теоретична частина

2.1. Типи світла в Three.js

1. **AmbientLight**: Рівномірно освітлює всі об'єкти в сцені. Не має напрямку і **не може відкидати тіні**. Використовується для пом'якшення надто темних областей.
2. **DirectionalLight**: Імітує сонце. Промені паралельні. **Підтримує тіні**. Використовує ортогональну камеру для розрахунку карти тіней.
3. **PointLight**: Світло від лампочки, розповсюджується в усіх напрямках. **Підтримує тіні**. Має параметри `distance` та `decay` (фізично коректне згасання).
4. **SpotLight**: Світло від прожектора або ліхтарика (конус). **Підтримує тіні**. Має кут розкриття (`angle`) та розмиття країв (`penumbra`).

2.2. Концепція Shadow Map (Карта тіней)

Three.js використовує техніку **Shadow Mapping**. Для кожного джерела світла, що відкидає тіні, сцена рендериться один раз з позиції цього джерела. Замість кольорів у спеціальну текстуру (Shadow Map) записується **глибина** (відстань від світла до найближчого об'єкта). Під час фінального рендеру відеокарта порівнює відстань поточної точки до світла зі значенням у карті тіней. Якщо точка далі — вона в тіні.

2.3. Shadow Camera Frustum

Оскільки Shadow Map — це текстура, ми маємо визначити область простору, яка в неї потрапить. Для `DirectionalLight` це прямокутний паралелепіпед (Frustum), що налаштовується через `light.shadow.camera` параметрами `left`, `right`, `top`, `bottom`, `near`, `far`. Об'єкти поза цими межами тіней не відкидатимуть.

2.4. Shadow Acne та Bias

Shadow Acne — це графічний артефакт у вигляді "зебри" або плям на поверхні, що виникає через обмежену точність Z-буфера.

- **Рішення:** Використання властивості `light.shadow.bias`. Невелике від'ємне значення (наприклад, `-0.0001`) допомагає "підняти" тінь над поверхнею.

3. Практичне завдання: Проєкт «Драматична сцена з тінями»

Алгоритм виконання:

1. **Ініціалізація:** Увімкніть підтримку тіней у рендерері:

```
renderer.shadowMap.enabled = true;  
renderer.shadowMap.type = THREE.PCFSoftShadowMap;
```
2. **Об'єкти:** Для мешів, що мають відкидати тінь, встановіть `castShadow = true`. Для поверхонь, на які падає тінь (підлога), — `receiveShadow = true`.
3. **Світло:** Створіть `DirectionalLight`, налаштуйте його позицію та активуйте `castShadow = true`.
4. **Налагодження:** Використовуйте `THREE.CameraHelper(light.shadow.camera)` для візуалізації робочої області тіней.

4. Завдання для самостійного виконання

1. **Рухоме світло:** Створіть анімацію для `PointLight`, щоб воно рухалося по колу або синусоїді над об'єктами. Проаналізуйте, як змінюється довжина та форма тіней.
2. **Фільтрація:** Спробуйте змінити `renderer.shadowMap.type` на `BasicShadowMap` (зубчасті краї) та порівняйте з `PCFSoftShadowMap`.
3. **Ефект ліхтарика:** Реалізуйте `SpotLight`, який слідує за курсором миші або прив'язаний до камери (імітація First-Person VR).

5. Контрольні запитання

1. Чому `AmbientLight` не може відкидати тіні з точки зору алгоритму Shadow Mapping?
2. Як роздільна здатність карти тіней (`mapSize`) впливає на споживання відеопам'яті (VRAM) та FPS?

3. Навіщо потрібен параметр `shadow.camera.updateProjectionMatrix()` при динамічній зміні меж `left/right/top/bottom`?
4. Як властивість `.target` у `SpotLight` впливає на напрямок освітлення та розрахунок тіней?