

# Lighting Demos

## Ambient lighting Demo

20 minute demo

### II. Ambient lighting

- A. What is ambient lighting?
- B. Adding ambient lighting to a scene
- C. Configuring ambient lighting properties

Sure, here is a 20-minute demo for the section on ambient lighting:

[Start the demo by opening up the Three.js editor.]

### I. Introduction

Welcome everyone and explain that we will be covering the topic of ambient lighting in Three.js today.

### II. Ambient Lighting

#### A. What is ambient lighting?

Explain that ambient lighting is a type of lighting that provides a base level of illumination to a scene.

Unlike directional or point lights, ambient lighting does not cast shadows or create highlights.

Ambient lighting helps to give a scene a more natural and balanced look.

#### B. Adding ambient lighting to a scene

Open up the Three.js editor and create a new scene.

Show how to add an ambient light to the scene using the following code:

```
const ambientLight = new THREE.AmbientLight(0xffffff, 0.5);
scene.add(ambientLight);
```

Explain that the first parameter is the color of the light, and the second parameter is the intensity of the light.

### C. Configuring ambient lighting properties

Show how to adjust the intensity of the ambient light by changing the second parameter.

Explain that the intensity value is a decimal between 0 and 1, with 1 being the brightest.

Show how to change the color of the ambient light by changing the first parameter to a different hexadecimal value.

Explain that the ambient light color can be used to set the overall tone of the scene.

[Demo conclusion]

## Outline

### I. Introduction to Three.js lighting

- A. Importance of lighting in 3D graphics
- B. Overview of light types in Three.js
- C. Basic properties of lights (color, intensity, etc.)

### II. Ambient lighting

- A. What is ambient lighting?
- B. Adding ambient lighting to a scene
- C. Configuring ambient lighting properties

### III. Point lights

- A. What is a point light?
- B. Adding a point light to a scene
- C. Configuring point light properties (color, intensity, distance, etc.)
- D. Enabling shadows for point lights

### IV. Directional lights

- A. What is a directional light?
- B. Adding a directional light to a scene

- C. Configuring directional light properties (color, intensity, direction, etc.)
- D. Enabling shadows for directional lights

## V. Spot lights

- A. What is a spot light?
- B. Adding a spot light to a scene
- C. Configuring spot light properties (color, intensity, angle, distance, etc.)
- D. Enabling shadows for spot lights

## VI. Hemisphere lights

- A. What is a hemisphere light?
- B. Adding a hemisphere light to a scene
- C. Configuring hemisphere light properties (color, intensity, ground color, etc.)

## VII. Environment maps

- A. What is an environment map?
- B. Creating an environment map using a CubeTexture
- C. Mapping an environment map to a scene
- D. Configuring environment map properties (reflectivity, roughness, etc.)

## VIII. Light helpers

- A. What are light helpers?
- B. Using PointLightHelper and DirectionalLightHelper objects
- C. Configuring light helper properties

## IX. Practical examples

- A. Demonstrating how to use multiple light types in a scene
- B. Creating a night-time scene with directional and point lights
- C. Simulating sunlight with a directional light and environment map

## X. Conclusion and next steps

- A. Recap of key concepts and techniques
- B. Suggestions for further learning and experimentation
- C. Q&A session

This outline could be adjusted or expanded based on the level and interests of the audience, as well as the available time and resources.

# Env Map Implementation Demo

## Implementation Demo

### I. Introduction (5 minutes)

- A. Brief overview of the topic
- B. Importance of environment maps in 3D graphics
- C. Overview of the three types of environment maps: cube maps, spherical maps, and HDR maps

### II. Using Cube Maps (10 minutes)

- A. What is a cube map?
- B. Creating a cube texture using six individual images
- C. Loading a cube texture in Three.js
- D. Applying a cube texture to a mesh
- E. Live coding demo of creating and using a cube map in an Angular Three.js project

### III. Using Spherical Maps (10 minutes)

- A. What is a spherical map?
- B. Creating a spherical texture from a single image
- C. Loading a spherical texture in Three.js
- D. Applying a spherical texture to a mesh
- E. Live coding demo of creating and using a spherical map in an Angular Three.js project

### IV. Using HDR Maps (10 minutes)

- A. What is an HDR map?
- B. Loading an HDR texture in Three.js
- C. Creating a cube texture from an HDR map using PMREMGGenerator
- D. Applying a cube texture to a mesh
- E. Live coding demo of creating and using an HDR map in an Angular Three.js project

## V. Conclusion (5 minutes)

- A. Recap of the main points
- B. Resources for further learning
- C. Q&A

Note: The live coding demos can involve creating a basic scene with a mesh, applying the environment map, and adjusting some of the mesh properties to show the effect of the environment map.

## Cube, Spherical, HDR Map Demo

### 30 minute Code Demo

#### I. Introduction (5 minutes)

- A. Brief overview of the topic
- B. Importance of environment maps in 3D graphics
- C. Overview of the three types of environment maps: cube maps, spherical maps, and HDR maps

#### II. Using Cube Maps (10 minutes)

- A. What is a cube map?
- B. Creating a cube texture using six individual images
- C. Loading a cube texture in Three.js
- D. Applying a cube texture to a mesh
- E. Live coding demo of creating and using a cube map in an Angular Three.js project

#### III. Using Spherical Maps (10 minutes)

- A. What is a spherical map?
- B. Creating a spherical texture from a single image
- C. Loading a spherical texture in Three.js
- D. Applying a spherical texture to a mesh
- E. Live coding demo of creating and using a spherical map in an Angular Three.js project

#### IV. Using HDR Maps (10 minutes)

- A. What is an HDR map?
- B. Loading an HDR texture in Three.js
- C. Creating a cube texture from an HDR map using PMREMGGenerator
- D. Applying a cube texture to a mesh
- E. Live coding demo of creating and using an HDR map in an Angular Three.js project

#### V. Conclusion (5 minutes)

- A. Recap of the main points
- B. Resources for further learning
- C. Q&A

Note: The live coding demos can involve creating a basic scene with a mesh, applying the environment map, and adjusting some of the mesh properties to show the effect of the environment map.

## Procedural Maps

60 minute demo for Procedural maps in Angular Three.js:

### I. Introduction to Procedural maps

- Definition and explanation of procedural maps
- Advantages of using procedural maps
- Examples of procedural maps in popular games and movies

5 minute lecture exploring:

- Definition and explanation of procedural maps
- Advantages of using procedural maps
- Examples of procedural maps in popular games and movies

Procedural maps are textures that are generated algorithmically instead of being created manually. This means that they can be created on the fly, allowing for endless variations and unique textures.

One advantage of using procedural maps is that they can save a lot of memory and storage space, as only the algorithm needs to be stored rather than a large texture file. They can also be generated at different resolutions, making them ideal for use in games and other real-time applications.

Procedural maps have been used in many popular games and movies, such as Minecraft, No Man's Sky, and Avatar. In Minecraft, procedural maps are used to generate the blocky terrain of the game world. In No Man's Sky, they are used to create the vast and diverse universe of the game. And in Avatar, they were used to create the unique plant life and landscapes of Pandora.

Overall, procedural maps are a powerful tool in 3D graphics that offer many benefits over traditional texture mapping techniques.

## II. Generating procedural textures in Three.js

In Three.js, we can generate procedural textures using built-in generators or create custom generators for more specific use cases.

Built-in generators are pre-made functions that produce procedural textures with different properties. They are easy to use and a good starting point for beginners. Examples of built-in generators in Three.js include Checkerboard, Stripe, and Noise.

Creating custom generators allows us to have more control over the properties of the texture we want to generate. For example, we can create a custom generator to produce a specific pattern or to add custom noise to a texture.

To create a custom generator, we can define a function that takes in parameters such as frequency, amplitude, and offset, and use these values to generate the

texture. We can then use this custom generator in our Three.js application to generate the desired procedural texture.

Using built-in generators and creating custom generators provide us with a lot of flexibility when it comes to generating procedural textures in Three.js.

#### A. Using built-in generators

- Explanation of built-in generators available in Three.js
- Code demo to generate procedural textures using built-in generators

#### B. Creating custom generators

- Explanation of how to create custom procedural texture generators
- Code demo to create and use custom generators

### III. Applying procedural textures to a mesh

#### A. Creating a mesh

- Explanation of how to create a mesh in Three.js
- Code demo to create a mesh and add it to the scene

#### B. Creating materials with procedural textures

- Explanation of how to create a material with a procedural texture
- Code demo to create a material with a procedural texture and apply it to the mesh

#### C. Tweaking procedural textures



- Explanation of how to tweak procedural textures to achieve desired effects
- Code demo to show different ways to tweak procedural textures

#### IV. Advanced procedural maps techniques

##### A. Using noise functions to generate more complex textures

- Explanation of how noise functions can be used to create more complex textures
- Code demo to show how to use noise functions in Three.js

##### B. Combining procedural maps with other textures

- Explanation of how procedural maps can be combined with other textures to create more interesting effects
- Code demo to show how to combine procedural maps with other textures

#### V. Conclusion

- Recap of the concepts covered in the demo
- Suggestions for further learning and exploration of procedural maps in Three.js.