

CAPH.WUI.ENGINE.RENDERING.DIRECTIONALLIGHT

A light source that shines from a specific direction not from a specific position. This light will behave as though it is infinitely far away and the rays produced from it are all parallel. The best analogy would be a light source that acts like the sun; the sun is so far away that all sunlight hitting objects comes from the same angle. Affects objects using LambertMaterial or PhongMaterial.

Contents

Constructor

[build](#)

[DirectionalLight](#)

Methods

[build](#)

[DirectionalLight](#)

Constructor

build		
Description		
DirectionalLight 'build' method is used to build DirectionalLight object		
Parameters		
hex	Number	Numeric value of the RGB component of the color.
intensity	Number	Numeric value of the light's strength/intensity.
Emulator Support	Y	
SDK Constraint	None	
Example		

```
var AccRendering = caph.wui.engine.rendering;

var basePage = new AccRendering.BasePage();

var geometry = new AccRendering.SphereGeometry().build(100, 100, 100);

var matrial = new AccRendering.BasicMaterial();
var object = new AccRendering.Mesh3d(geometry, matrial.build({ color: Math.random() * 0xfffff, opacity: 0.5 }));

basePage.add(object);

var light = new AccRendering.DirectionalLight().build(0xFFFF00);
light.position.set(500, 0, 0);
basePage.scene.add(light);

var renderer = new AccRendering.CanvasRenderer();
renderer.render(basePage);
```

DirectionalLight

Description

DirectionalLight 'build' method is used to build DirectionalLight object. Return a factory to create a DirectionalLight.

Emulator Support	Y
SDK Constraint	None

SDK Constraint	None
----------------	------

Example

```
var AccRendering = caph.wui.engine.rendering;

var basePage = new AccRendering.BasePage();

var geometry = new AccRendering.SphereGeometry().build(100, 100, 100);

var matrial = new AccRendering.BasicMaterial();
var object = new AccRendering.Mesh3d(geometry, matrial.build({ color: Math.random() * 0xfffff, opacity: 0.5 }));

basePage.add(object);

var light = new AccRendering.DirectionalLight().build(0xFFFF00);
light.position.set(500, 0, 0);
basePage.scene.add(light);

var renderer = new AccRendering.CanvasRenderer();
renderer.render(basePage);
```

Methods

build

Description

(Constructor) DirectionalLight 'build' method is used to build DirectionalLight object

Parameters	<ul style="list-style-type: none">■hex<ul style="list-style-type: none">- Number- Numeric value of the RGB component of the color.■intensity (Optional)<ul style="list-style-type: none">- Number- Numeric value of the light's strength/intensity.
Return	<ul style="list-style-type: none">■Object<ul style="list-style-type: none">- An instance of THREE.DirectionalLight
Emulator Support	Y
SDK Constraint	none

Example

```
var AccRendering = caph.wui.engine.rendering;

var basePage = new AccRendering.BasePage();

var geometry = new AccRendering.SphereGeometry().build(100, 100, 100);

var matrial = new AccRendering.BasicMaterial();
var object = new AccRendering.Mesh3d(geometry, matrial.build({ color: Math.random() * 0xfffff, opacity: 0.5 }));

basePage.add(object);

var light = new AccRendering.DirectionalLight().build(0xFFFFF00);
light.position.set(500, 0, 0);
basePage.scene.add(light);

var renderer = new AccRendering.CanvasRenderer();
renderer.render(basePage);
```

DirectionalLight

Description

(Constructor) DirectionalLight 'build' method is used to build DirectionalLight object. Return a factory to create a DirectionalLight.

Parameters	■Void
Return	■Object - An instance of DirectionalLight
Emulator Support	Y
SDK Constraint	none

Example

```
var AccRendering = caph.wui.engine.rendering;

var basePage = new AccRendering.BasePage();

var geometry = new AccRendering.SphereGeometry().build(100, 100, 100);

var matrial = new AccRendering.BasicMaterial();
var object = new AccRendering.Mesh3d(geometry, matrial.build({ color: Math.random() * 0xfffff, opacity: 0.5 }));

basePage.add(object);

var light = new AccRendering.DirectionalLight().build(0xFFFFF00);
light.position.set(500, 0, 0);
basePage.scene.add(light);

var renderer = new AccRendering.CanvasRenderer();
renderer.render(basePage);
```