

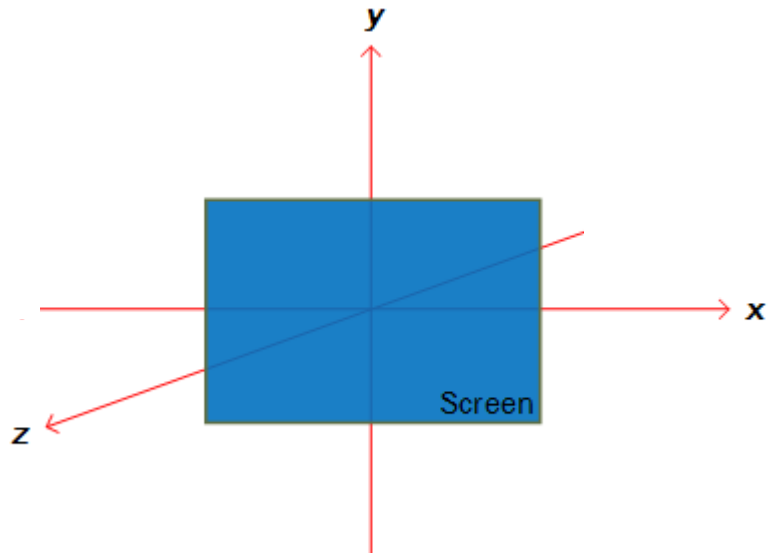
Grafika 3D w three.js

Kamera, światło, akcja!

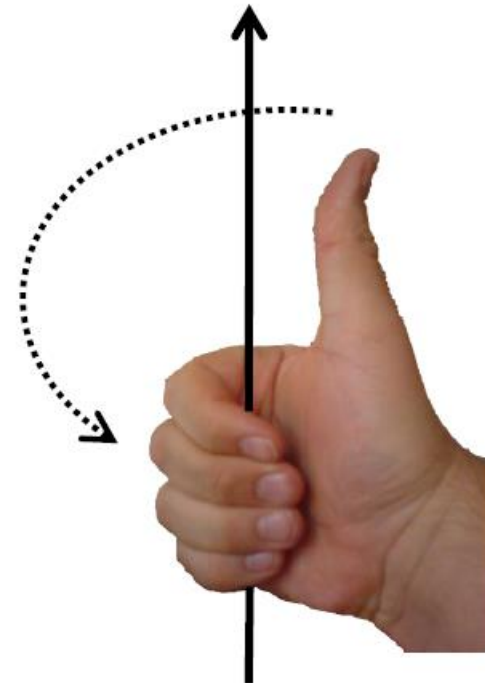
- Open source, licencja MIT
- Start w 2010
- Początkowo tylko CanvasRenderer i SVGRenderer
- Obecnie również WebGL
- 390 kontrybutorów

- Three.js Essentials - Jos Dirksen

Układ współrzędnych

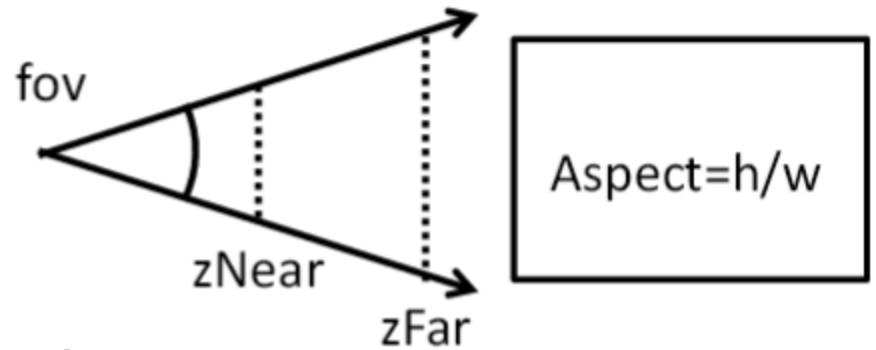


Prawoskrętny układ



Kamera z perspektywą

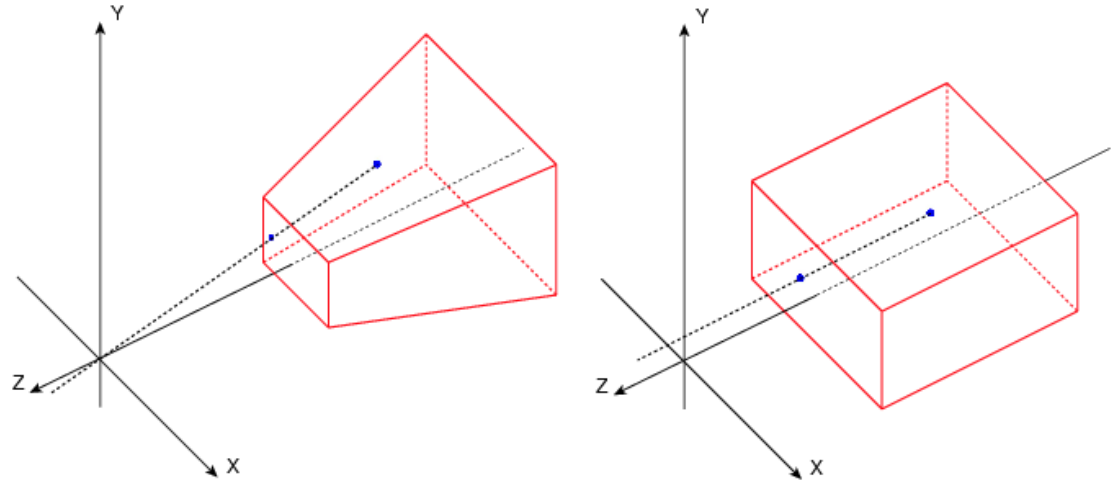
```
camera = new THREE.PerspectiveCamera(fov, aspect, zNear, zFar);
```



- fov - kąt rozwarcia "stożka" widzenia
- aspect - stosunek wysokości do szerokości płaszczyzn
- zNear - bliska płaszczyzna odcinania
- zFar – daleka płaszczyzna odcinania

Kamera ortogonalna

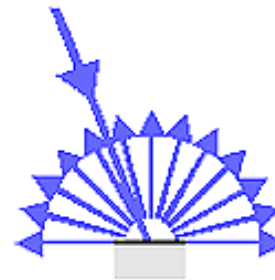
```
camera = new THREE. OrthographicCamera( left, right, top, bottom, near, far )
```



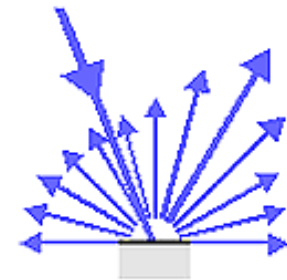
- left, right, top, bottom – wymiary płaszczyzn
- near – bliska płaszczyzna odcinania
- far – daleka płaszczyzna odcinania

Materiały

- MeshLambertMaterial – renderowany na lico, idealnie matowy
- MeshBasicMaterial – renderowany na lico, nie wspiera modelu oświetlenia
- MeshPhongMaterial – renderowany na pixel, odbicie z efektem Phong
- MeshNormalMaterial – kolor nie zależy od oświetlenia, tylko wektora normalnego, przydatny do testowania pozycji kamery



*Ideal diffuse reflection
(Lambertian surface)*



*Diffuse reflection with
directional component*

Oświetlenie

- `THREE.SpotLight` – światło z efektem stożka (tak jak w latarce)
- `THREE.AmbientLight` – proste światło, którego kolor jest dodawany do koloru materiału każdego obiektu na scenie
- `THREE.PointLight` – światło punktowe świecące równomiernie we wszystkich kierunkach
- `THREE.DirectionalLight` – światło kierunkowe, równoległe promienie, punkt w nieskończoności, np. słońce