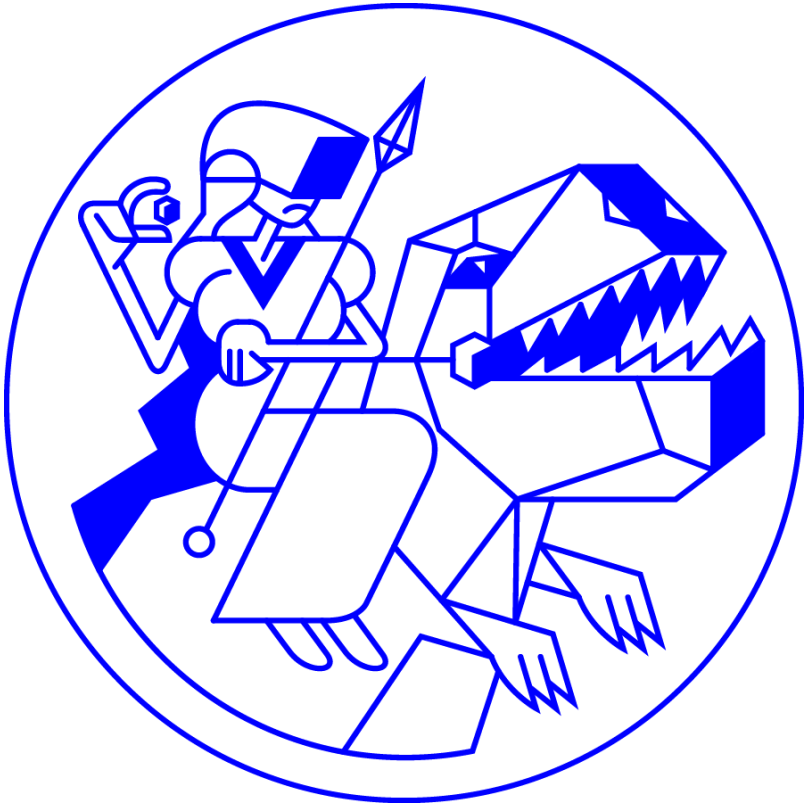
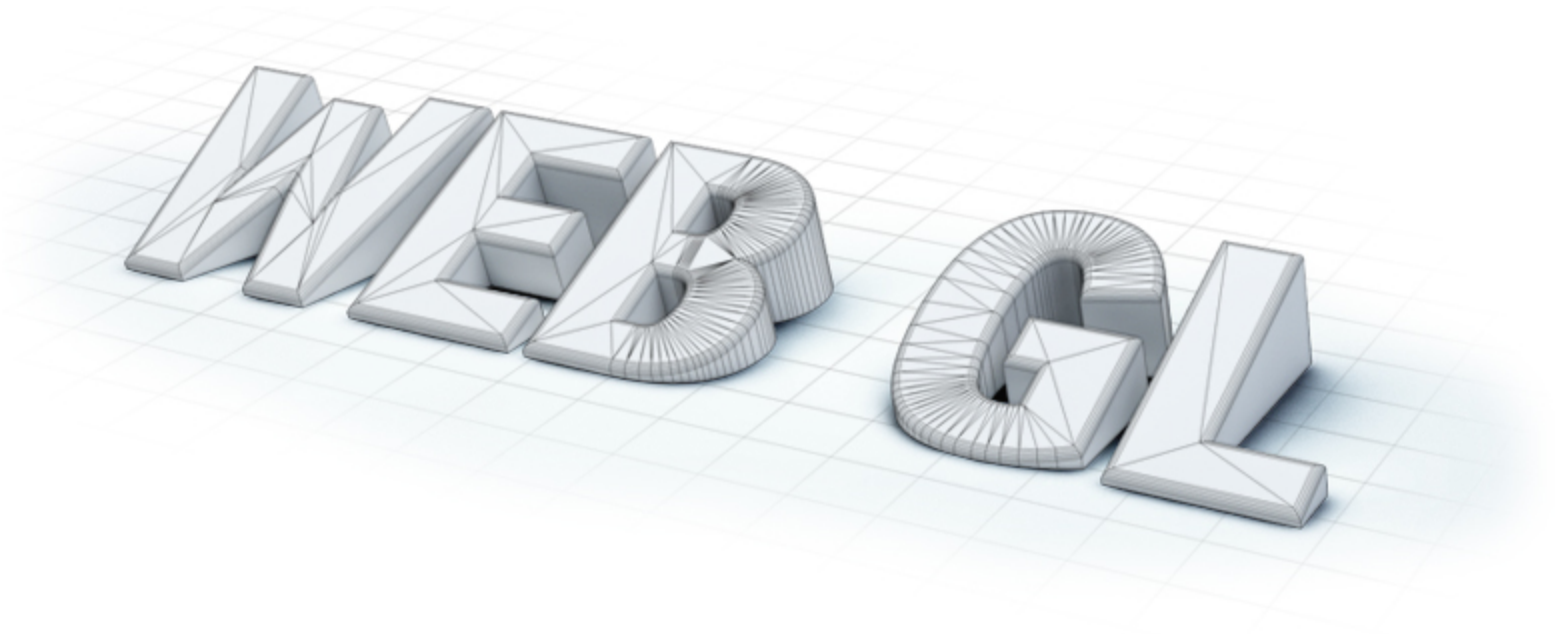


Основы веб-технологий

Василика Климова

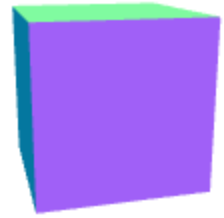
Разработчик интерфейсов

@lik04ka


WEBGL

WebGL

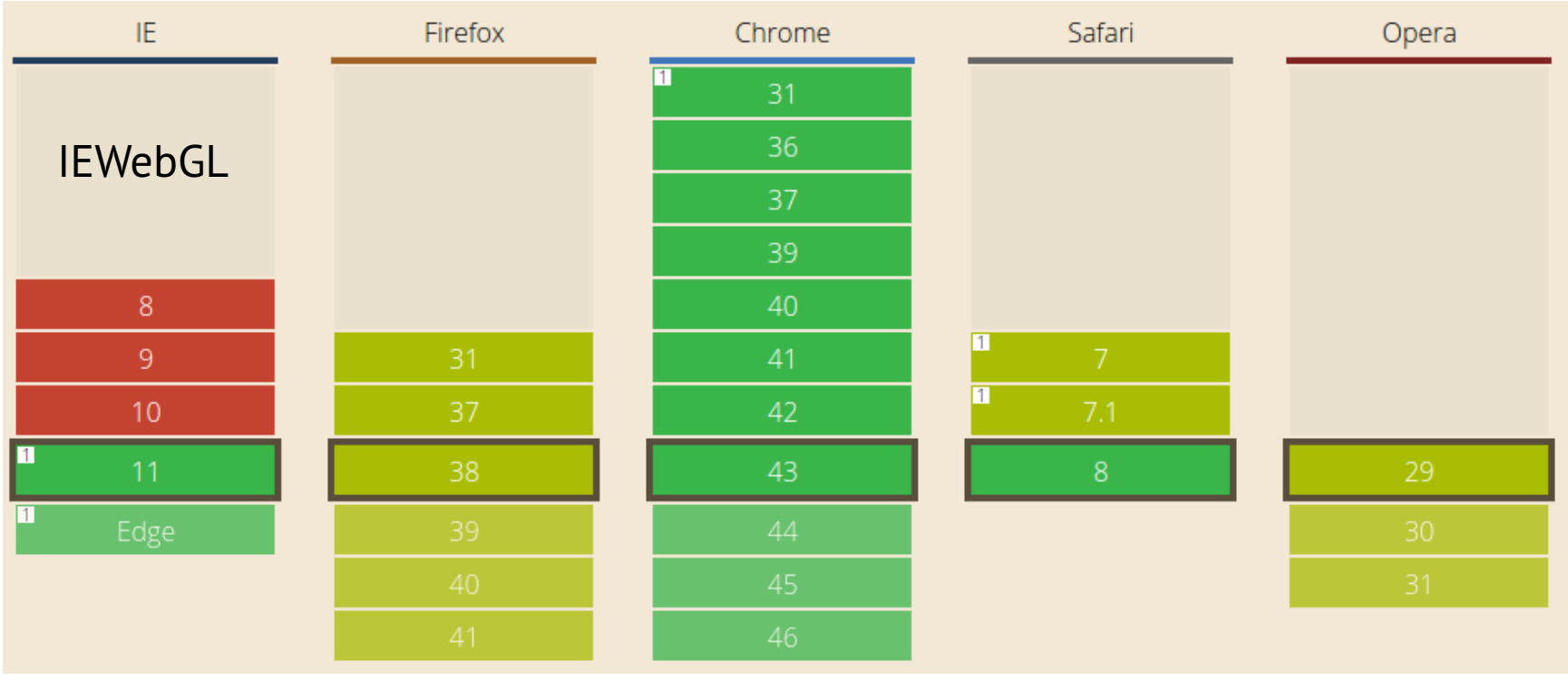
- HTML5 `<canvas>`
- OpenGL ES 2.0
- GLSL ES 1.1
- 2D/3D



KHRONOS GROUP

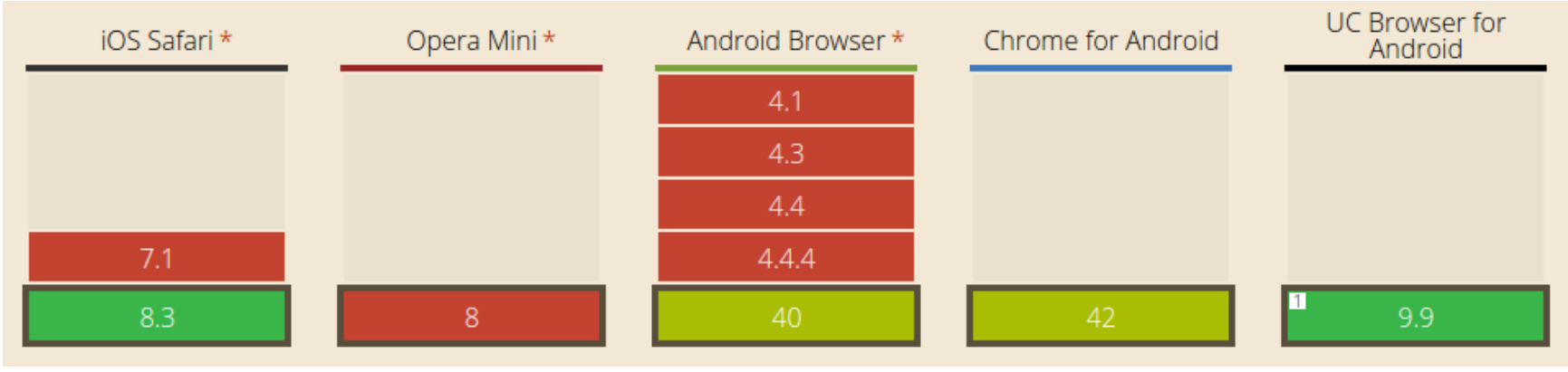
WebGL

Десктопные браузеры



caniuse.com/webgl

Мобильные браузеры

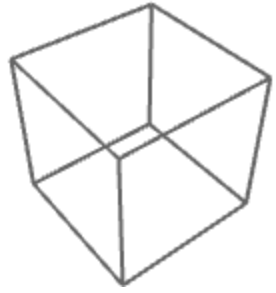


caniuse.com/webgl

Проверка поддержки WebGL

Your browser supports WebGL

You should see a spinning cube. If you do not, please
visit the support site for your browser.



get.webgl.org

Игры

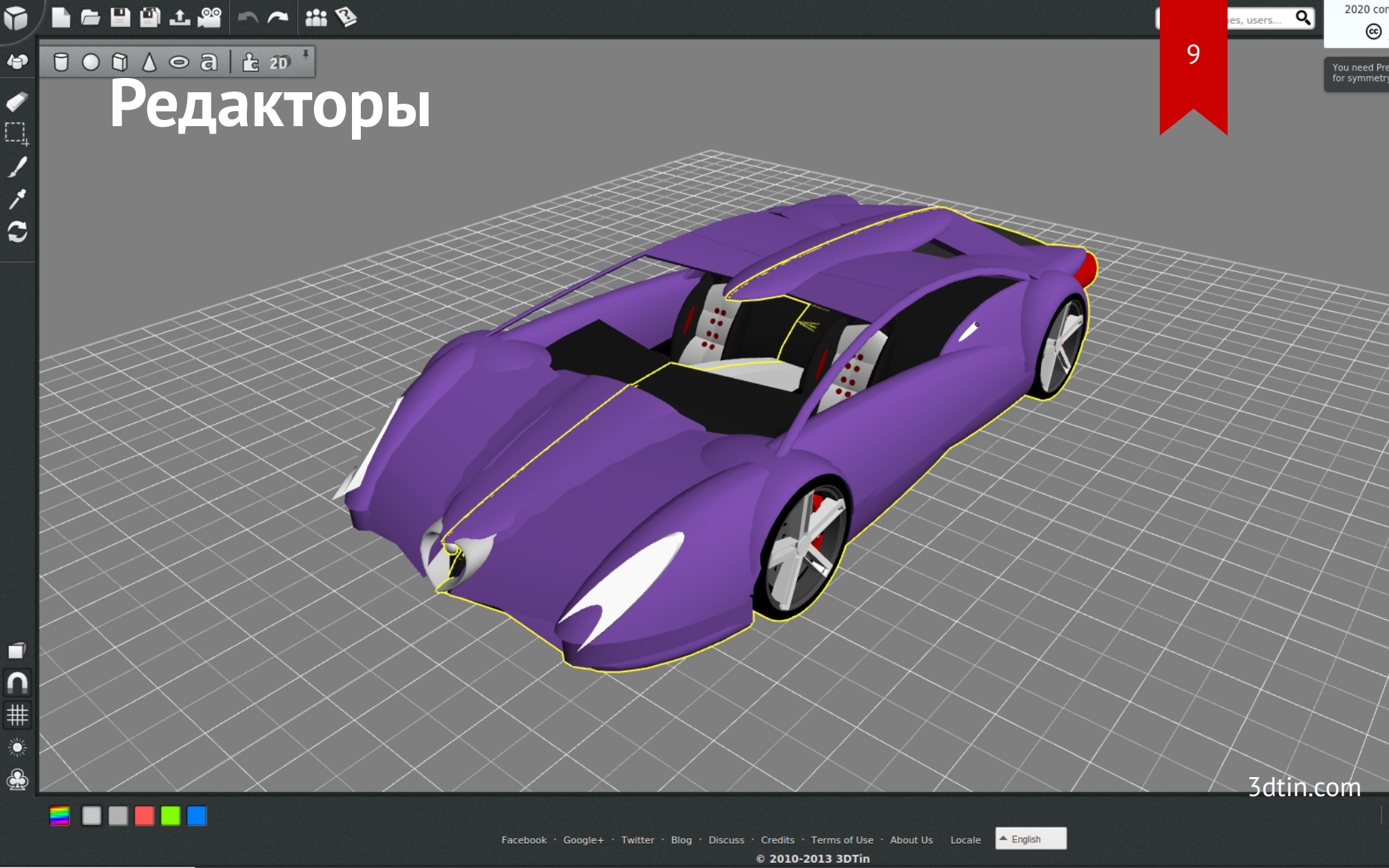
gooengine.com/pearl-boy

Навигация

bookcase.chromeexperiments.com

This is a Chrome Experiment

Редакторы



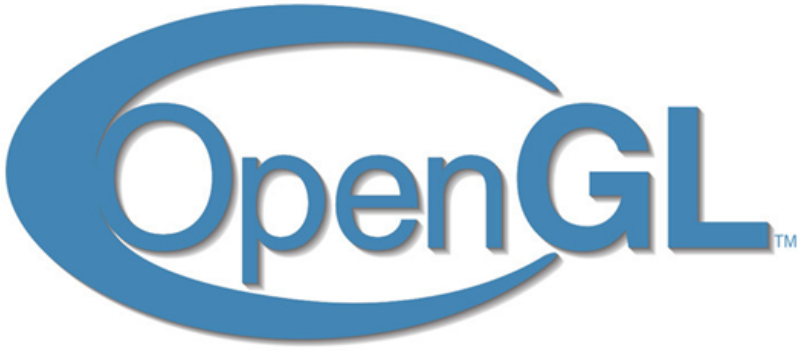
3dtin.com

Инфографика

Технологии 3D

O3D

Преимущества WebGL

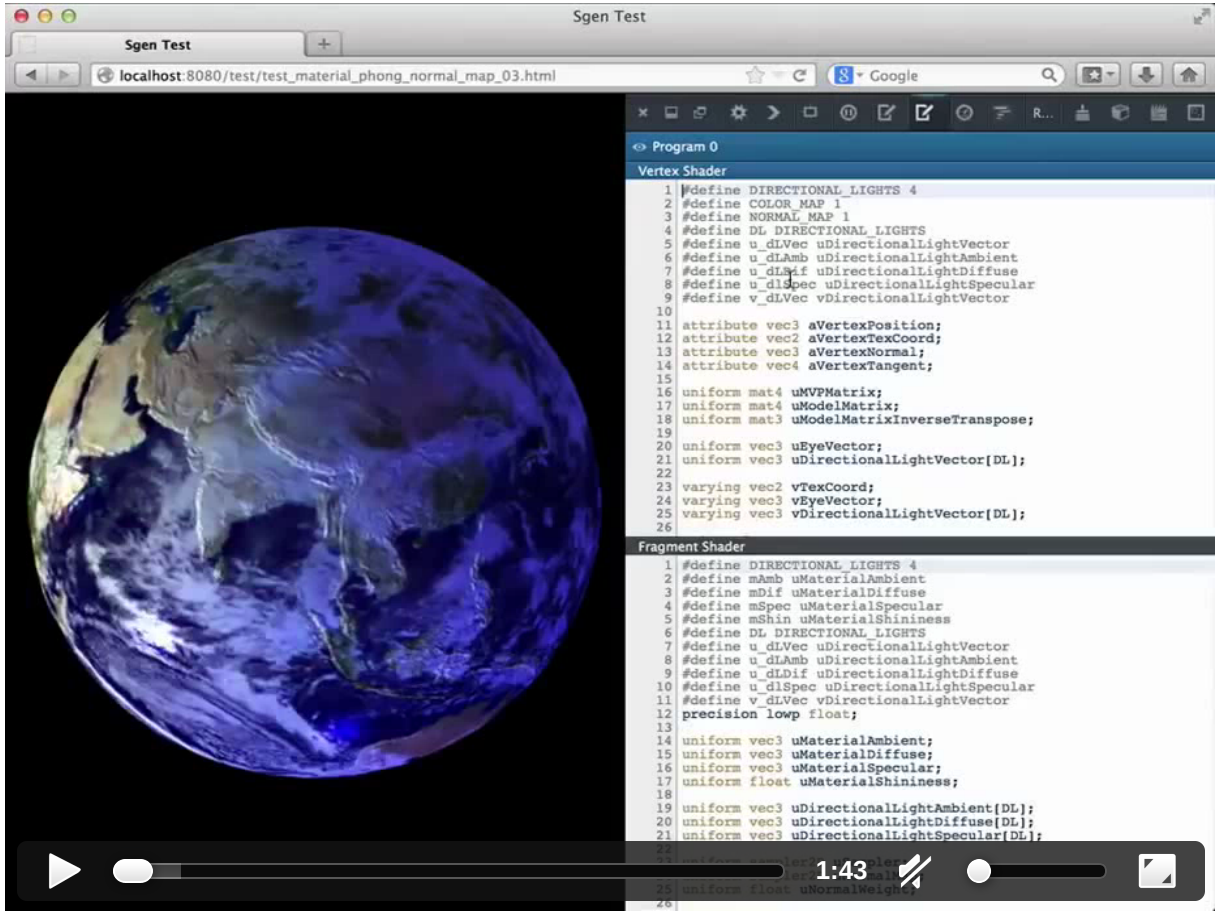


&

Преимущества WebGL

- Открытый стандарт
- Высокая производительность
- Автоматическое управление памятью
- Кроссплатформенность

Отсутствие компиляции



WebGL

+

+ GLSL ES

Программа на WebGL

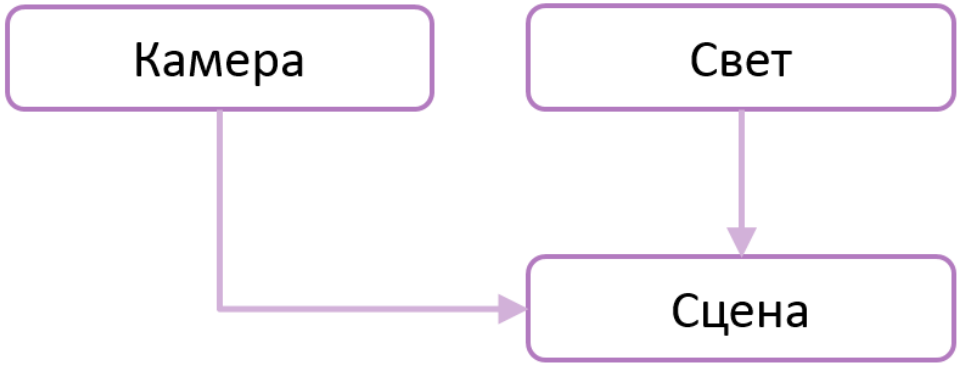
Точка


3D графика

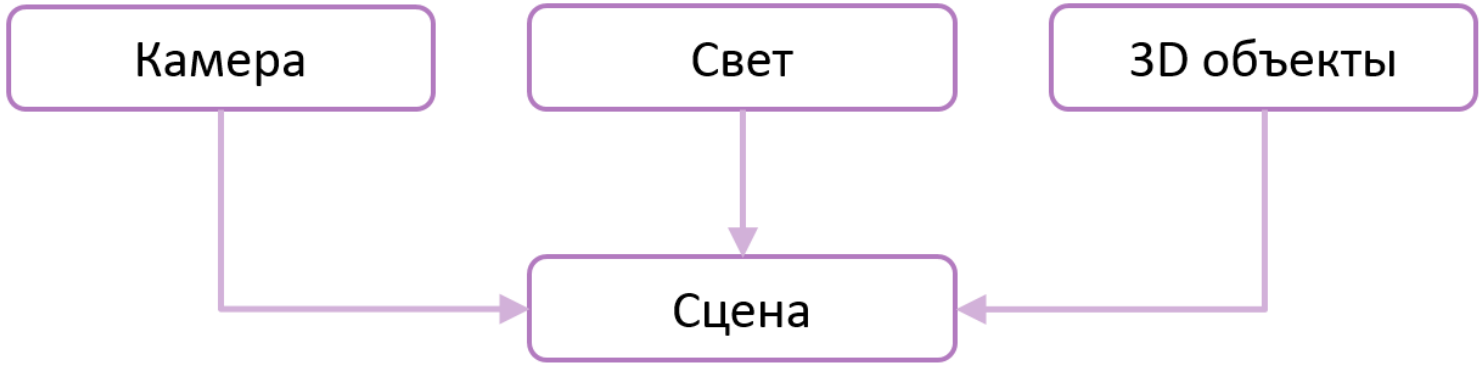
Основные понятия

- Сцена
- Свет
- Камера

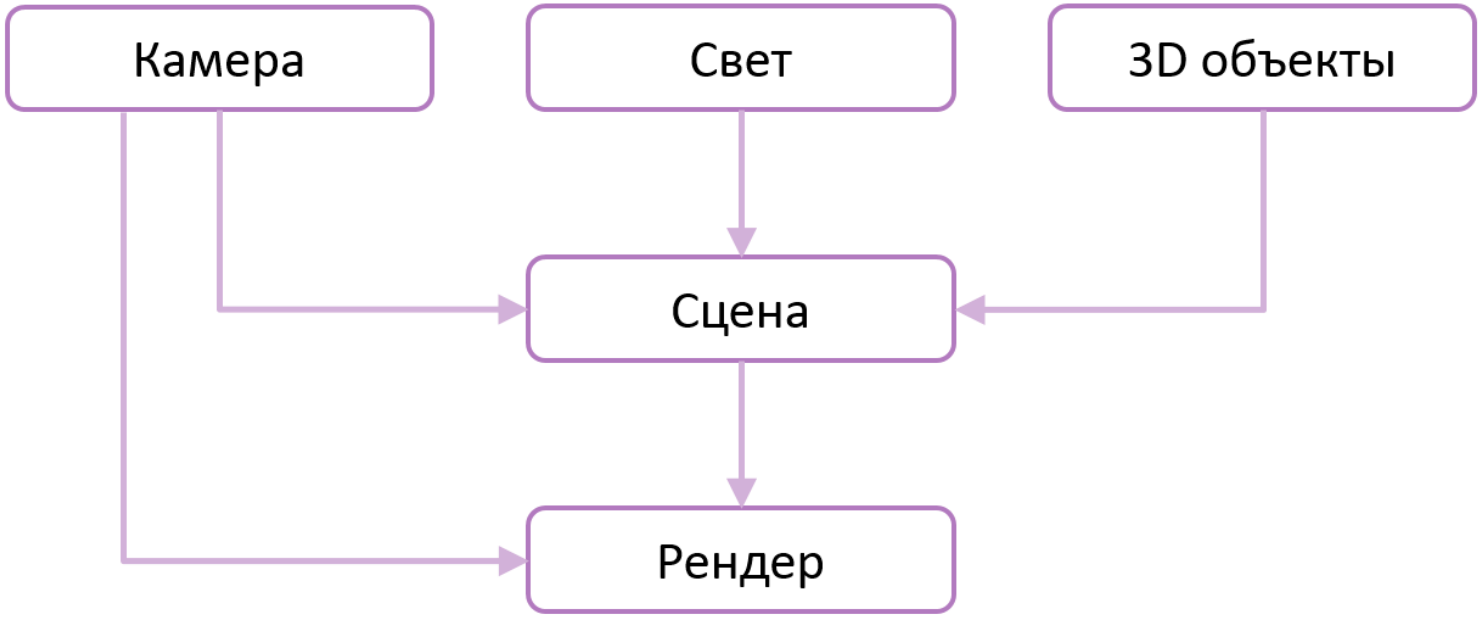
Взаимосвязь



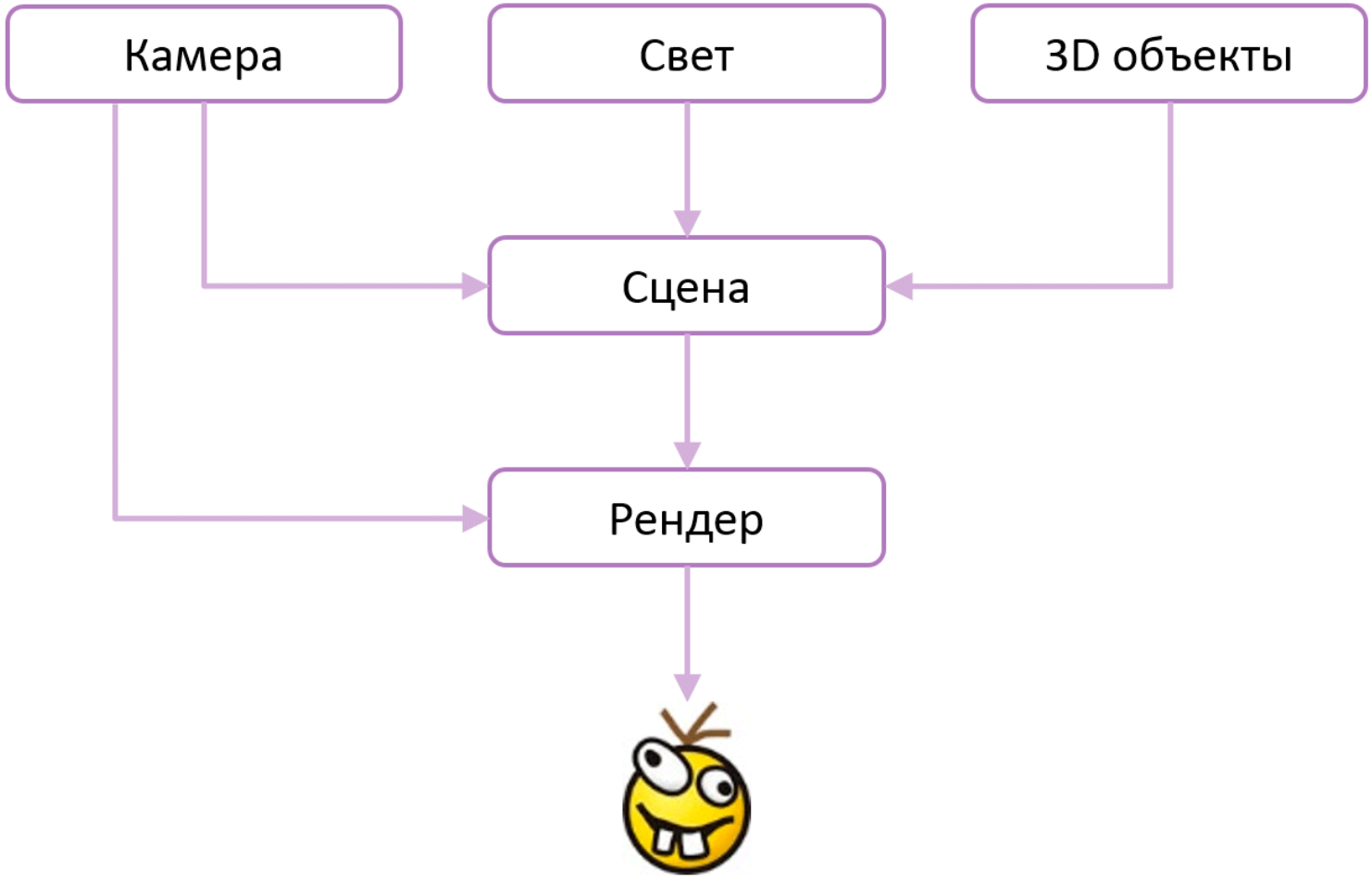
Взаимосвязь



Взаимосвязь



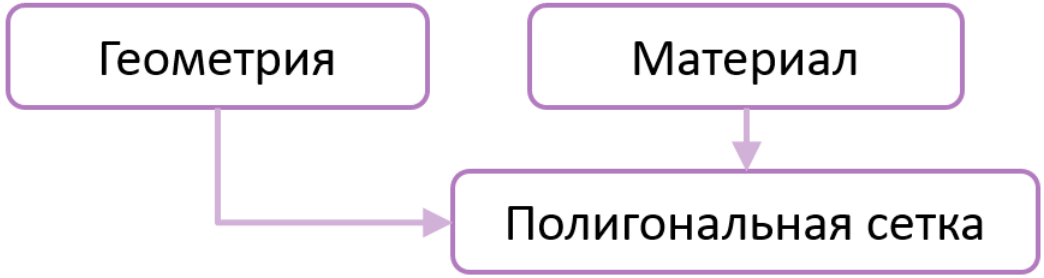
Взаимосвязь

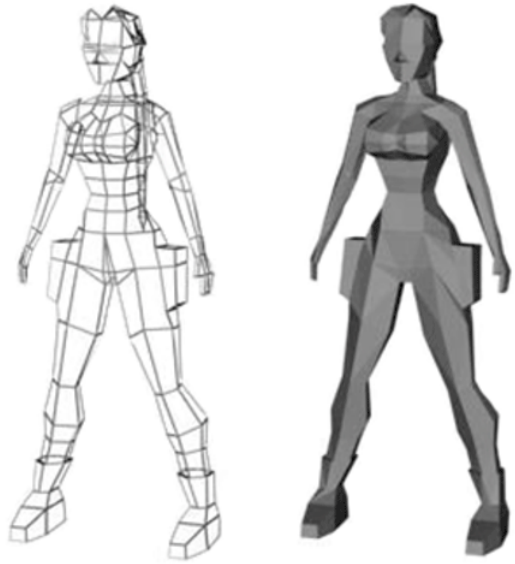


Меш

Геометрия

Меш





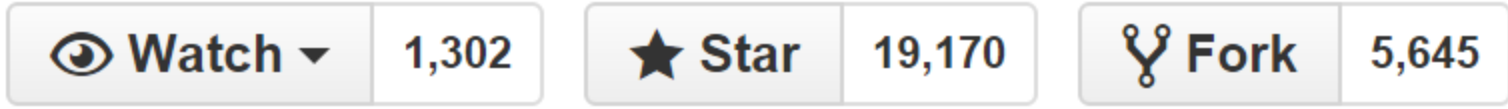
Меш

Визуализация

- Рендер
- Шейдер
- Анимация

threejs.org/examples/#webgl_particles_shapes

Three.js



three.min.js	420 kb
OBJLoader.js	8 kb
TrackballControls.js	14 kb

Экспортируемые форматы

obj	ply	binary
babylon	stl	wrl
vrm	dae (collada)	ctm
utf8	svg	pvr
vtk	awd	pdb

Входные данные

object.obj

Список вершин, с координатами (x,y,z)

```
v 13.825026512145996 -96.140419006347656 3.6714630126953125
v 15.503727912902832 -96.428817749023438 2.1252975463867187
v 14.864977836608887 -95.269874572753906 1.5220794677734375
v 12.35379695892334 -91.997489929199219 1.42974853515625
v 13.748141288757324 -93.280204772949219 1.0182418823242187
v 7.9108209609985352 -85.125518798828125 3.9087066650390625
v 11.508156776428223 -94.102958679199219 4.5504951477050781
v 13.997708320617676 -95.774742126464844 2.4739608764648437
v 13.337004661560059 -94.254402160644531 1.780059814453125
v 10.512875556945801 -90.527717590332031 2.5589218139648438
v 9.9423093795776367 -91.974296569824219 4.4449958801269531
v 12.121970176696777 -93.814460754394531 2.7707862854003906
v 8.8889608383178711 -89.88848876953125 4.7834281921386719
v 8.5577688217163086 -87.575393676757813 3.7520294189453125
v 65.255134582519531 50.347309112548828 32.405288696289063
v 66.766136169433594 50.430992126464844 33.477485656738281
v 63.850997924804687 51.162227630615234 31.309993743896484
v 61.49749755859375 53.767066955566406 30.037763595581055
```

Текстурные координаты (u,v)

```
vt 0.056318659335374832 0.62613308429718018
vt 0.061030514538288116 0.6244666576385498
vt 0.05973927304148674 0.62342357635498047
vt 0.060924597084522247 0.62328207492828369
vt 0.058572947978973389 0.62352561950683594
vt 0.056074466556310654 0.62256932258605957
vt 0.044292308390140533 0.48838406801223755
vt 0.04412538930773735 0.48736923933029175
vt 0.043935131281614304 0.48618751764297485
```

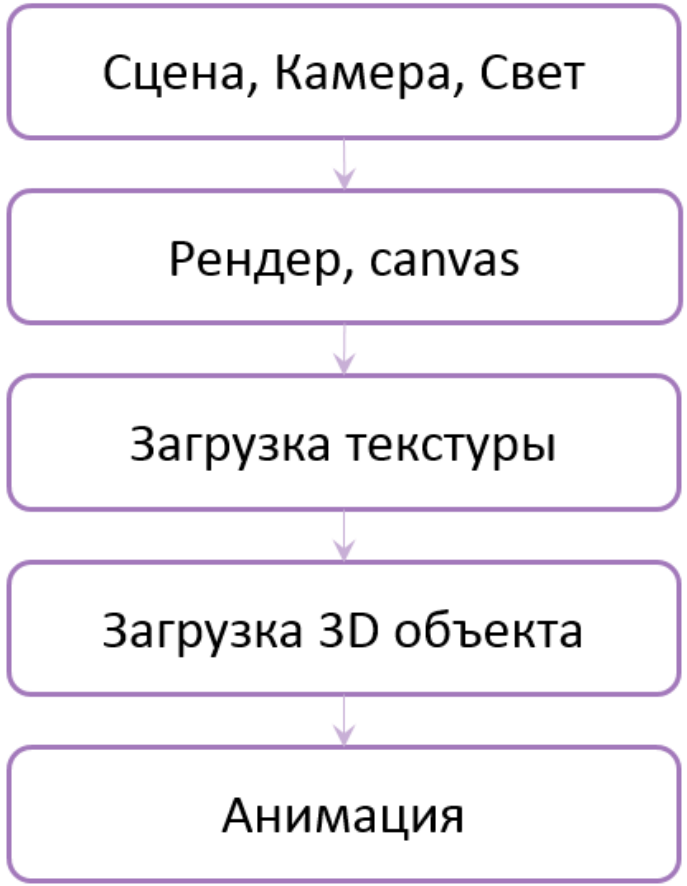
Определения поверхности(v/vt)

```
f 70220/29439 91099/29452 70267/29440
f 91099/29452 71051/29454 71061/29455
f 70265/29449 70272/29442 70288/29456
f 70262/29434 70272/29442 70265/29449
f 91099/29452 71061/29455 70299/29453
f 70265/29449 70288/29456 98620/29450
f 70267/29440 70299/29453 70363/29448
```

Входные данные

texture.jpg

Алгоритм


Сцена, Камера, Свет
Рендер, canvas
Загрузка текстуры
Загрузка 3D объекта
Анимация

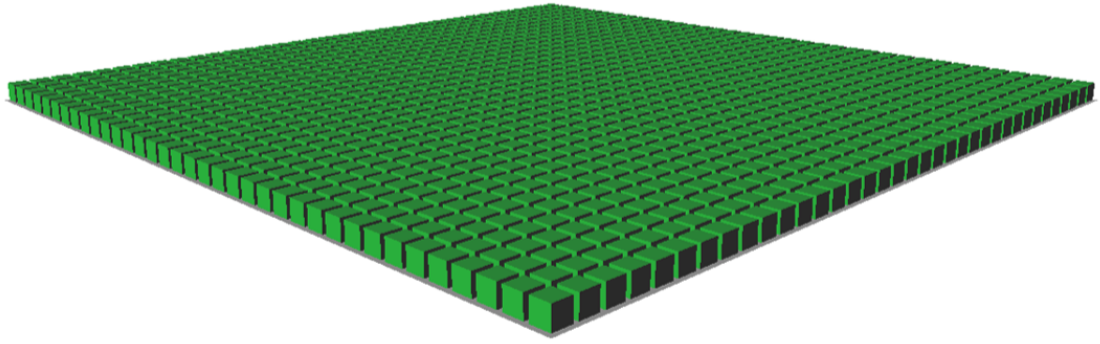
Выходной результат

Сцена

```
01. Player.container = document.getElementById("webgl-player");
02. Player.size = {
03.     width: Player.container.offsetWidth,
04.     height: Player.container.offsetHeight
05. };
06. Player.scene = new THREE.Scene();
```

Типы камер

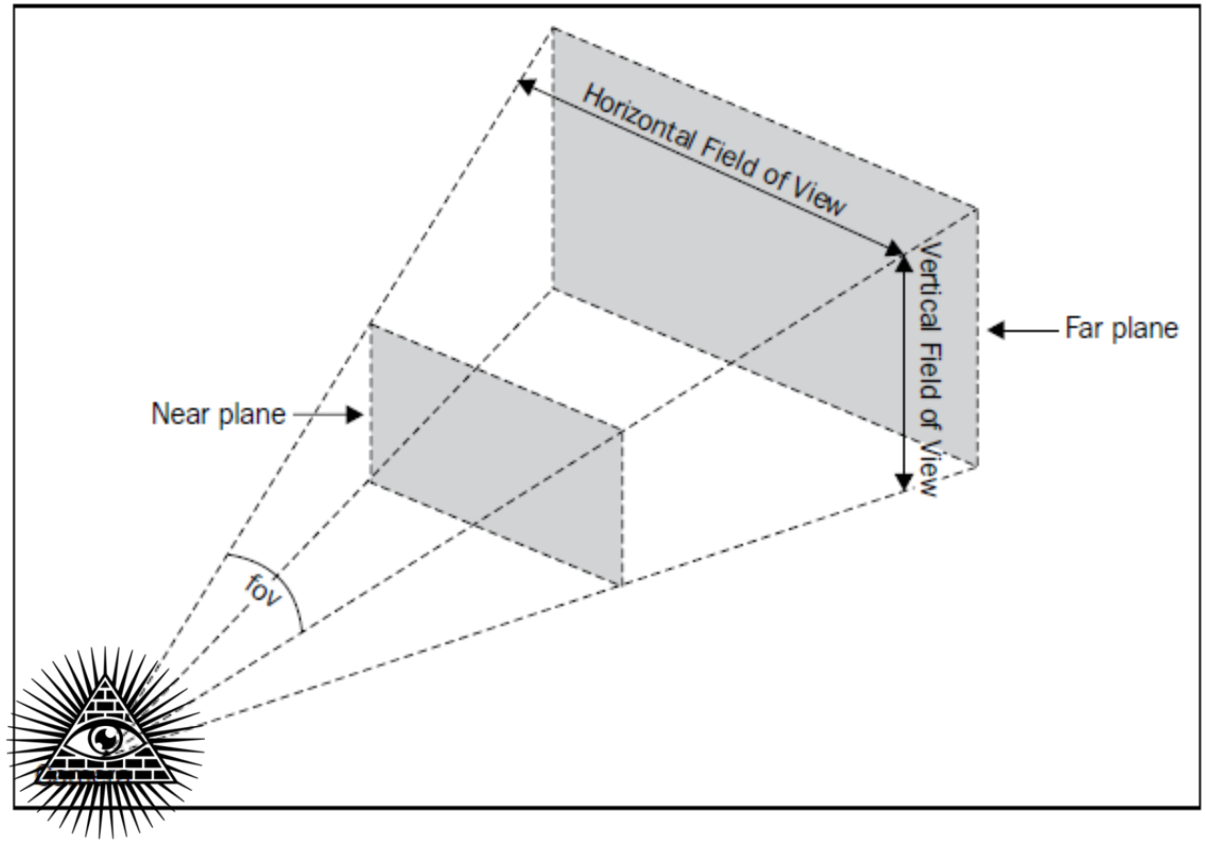
Перспективная проекция - PerspectiveCamera



Ортогональная проекция - OrthographicCamera

Объект Камера

```
PerspectiveCamera( fov,  aspect,  near,  far )
```



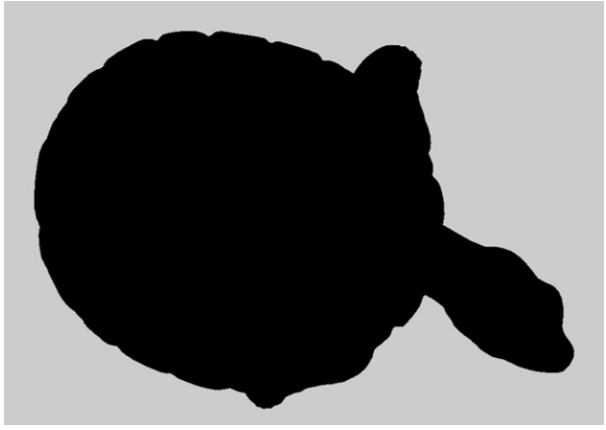
Добавление камеры

```
01. // PerspectiveCamera( fov,  aspect,  near,  far )
02. var aspect = Player.size.width / Player.size.height;
03. Player.camera = new THREE.PerspectiveCamera(45.0,
04. aspect, 2, 8000);
05. Player.camera.position.z = 300;
06. Player.scene.add(Player.camera);
```

Свет, рендер, canvas

```
01. var light = new THREE.AmbientLight();
02. Player.scene.add(Player.light);
03. Player.renderer = new THREE.WebGLRenderer({alpha: true});
04. Player.renderer.setSize(Player.size.width, Player.size.height);
05. // canvas
06. Player.container.appendChild(Player.renderer.domElement);
```

Свет



```
// Player.scene.add(light)
```

```
Player.scene.add(light)
```

WebGLRenderer

```
THREE.WebGLRenderer()
```

```
THREE.WebGLRenderer({alpha: true})
```

Текстура

```
01. var textureLoader = new THREE.TextureLoader();
02. textureLoader.load("texture.jpg", function(texture) {
03.     Player.texture = texture;
04.     Player.loadModel();
05. });
```

Загрузка 3D модели

```
01. loadModel: function() {
02.   var objectLoader = new THREE.OBJLoader();
03.   objectLoader.load("object.obj", function(object) {
04.     object.traverse(function(child) {
05.       if (child instanceof THREE.Mesh) {
06.         child.material.map = Player.texture; }
07.     });
08.     Player.scene.add(object);
09.   });
10. }
```

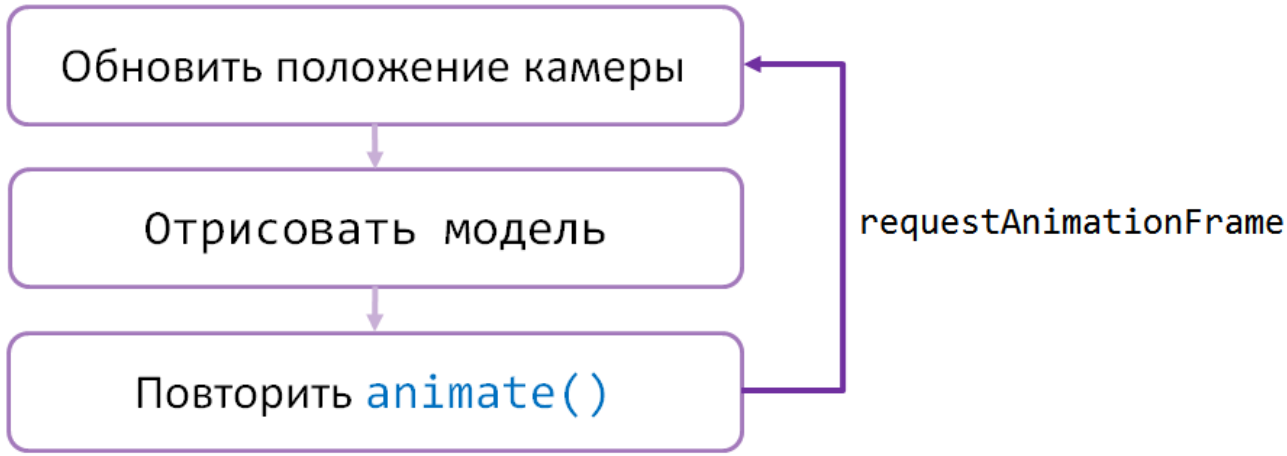
Отрисовка

```
01. Player.animate();
02. animate: function() {
03.     requestAnimationFrame(Player.animate);
04.     Player.renderer.render(Player.scene, Player.camera);
05. }
```

Вращение

```
01. Player.controls = new THREE.TrackballControls(
02.     Player.camera,
03.     Player.container
04. );
05. Player.animate();
06. animate: function() {
07.     requestAnimationFrame(Player.animate);
08.     Player.controls.update();
09.     Player.renderer.render(Player.scene, Player.camera);
10. }
```

requestAnimationFrame

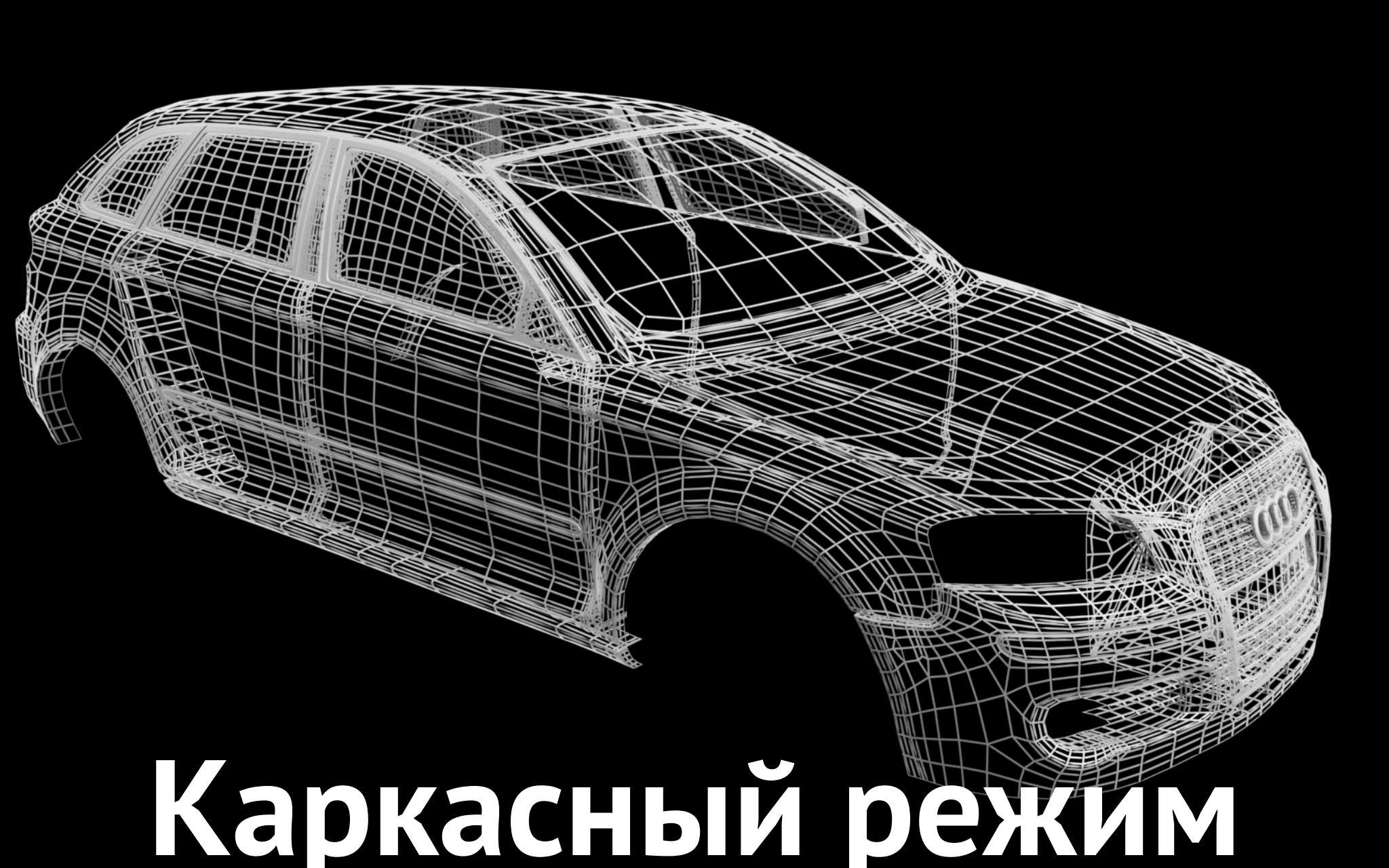



```
01. animate: function() {
02.     requestAnimationFrame(Player.animate);
03.     Player.controls.update();
04.     Player.renderer.render(Player.scene, Player.camera);
05. }
```

Отмена анимации

```
01. requestId = requestAnimationFrame(animate);
02. cancelAnimationFrame(requestId);
```

Модель с текстурой

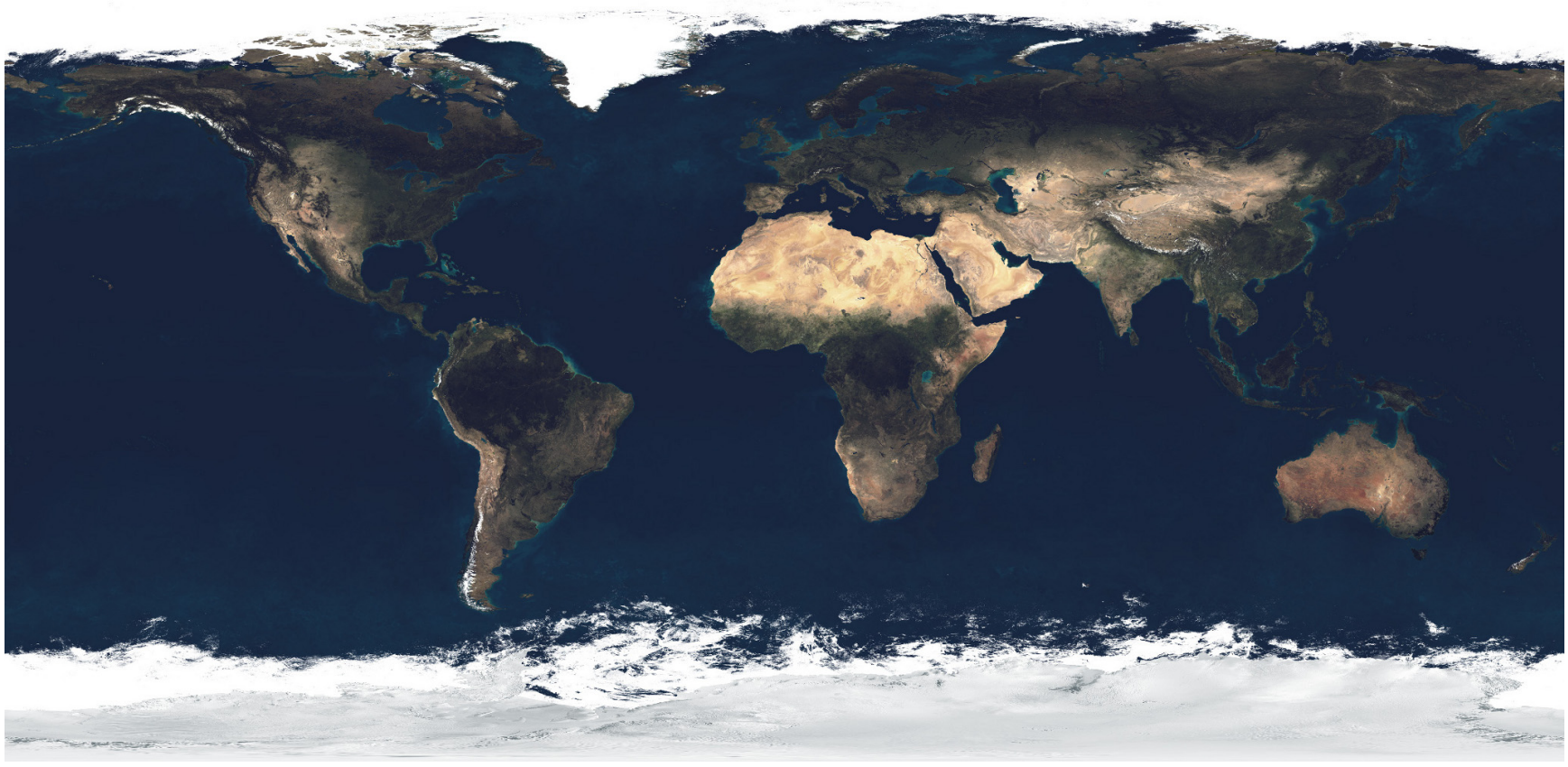


Каркасный режим

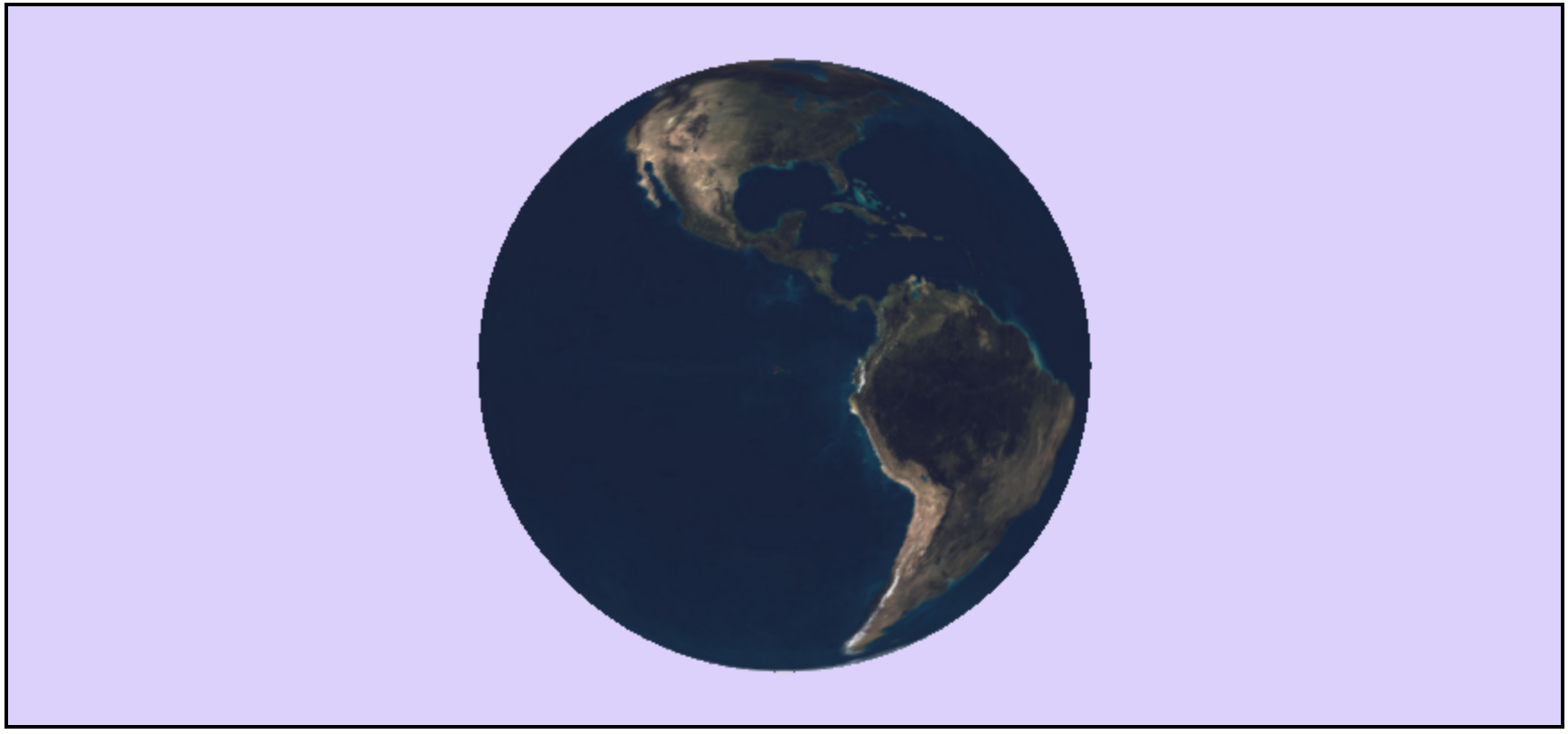
Создание сферы

```
01. geometry = new THREE.SphereGeometry(100, 50, 50);
02. material = new THREE.MeshPhongMaterial({map: texture});
03. mesh = new THREE.Mesh(geometry, material);
04. scene.add(mesh);
```

Текстура сферы



Земной шар



```
mesh.material.wireframe = false;
```

WebGL библиотеки

- Three.js
- Babylon.js
- Turbulenz
- PhiloGL

Полезные ссылки

- learningwebgl.com
- webglacademy.com
- davidscottlyons.com/threejs
- Книга WebGL. Программирование трехмерной графики
- Никита Северинов diductio.ru/course/2060/

Исходники

github.com/Likita

Демо

vasilika.ru

Интерактивность

Геймификация

Василика Климова


likita


vasilika.klimova


lik04ka