

Лабораторная работа №5:

Спрайты

Цель:

Целью данной лабораторной работы является изучение способов отображения двумерных изображений в экранных и мировых координатах.

Справка: примеры использования спрайтов:

http://threejs.org/examples/webgl_sprites.html

Краткая справка:

Спрайтом, называется объект, способный выводить двумерные изображения в трёхмерной системе координат.

В используемой нами библиотеке, спрайты создаются следующим образом:

```
var texture = loader.load(name);
var material = new THREE.SpriteMaterial( { map: texture } );

var sprite = new THREE.Sprite( material);
sprite.center.set( 0.0, 1.0 );
sprite.scale.set( 80, 64, 1 );
sprite.position.set( 0, 0, 1 );
```

объект типа `sprite` является наследником объекта `Object3D`.

При добавлении в сцену, спрайт размещается в мировой системе координат.

Для того, чтобы разместить спрайт в экранной системе координат, нужно использовать ортогональную камеру и специфические настройки рендера:

```
//создание ортогональной камеры
cameraOrtho = new THREE.OrthographicCamera( - width / 2, width / 2, height / 2, -
height / 2, 1, 10 );
cameraOrtho.position.z = 10;

//сцена для хранения списка объектов размещаемых в экранных координатах
sceneOrtho = new THREE.Scene();

//отключение авто очистки рендера
renderer.autoClear = false;

//функция render
//процесс отрисовки сцены и объектов в экранных координатах
renderer.clear();
renderer.render( scene, camera );
renderer.clearDepth();
renderer.render( sceneOrtho, cameraOrtho );
```

Создание, добавление и обновление спрайтов будет выглядеть следующим образом:

```
//функция для создания спрайта
function addSprite(name)
{
    //загрузка текстуры спрайта
    var texture = loader.load(name);
    var material = new THREE.SpriteMaterial( { map: texture } );

    //создание спрайта
    sprite = new THREE.Sprite( material);
    //центр и размер спрайта
    sprite.center.set( 0.0, 1.0 );
    sprite.scale.set( 80, 64, 1 );
    //позиция спрайта (центр экрана)
    sprite.position.set( 0, 0, 1 );

    sceneOrtho.add(sprite);

    return sprite;
}

//функция для обновления позиции спрайта
function updateHUDSprite(sprite)
{
    var width = window.innerWidth / 2;
    var height = window.innerHeight / 2;

    sprite.position.set( -width, height, 1 ); // левый верхний угол экрана
}
```

Реализация кнопок при помощи спрайтов

Описать кнопку можно структурой следующего вида:

```
var button = {}; //структура для хранения кнопки
button.sprite = sprite; //спрайт для отображения кнопки
button.default = default_material; //материал кнопки по умолчанию
button.highlight = highlight_material; //материал кнопки с подсветкой
button.position = sprite.position; //позиция кнопки
button.width = sprite.scale.x; //ширина кнопки
button.height = sprite.scale.y; //высота кнопки
```

Позицию и размеры кнопки не обязательно хранить в виде отдельных значений и можно каждый раз запрашивать из спрайта.

Обнаружить наведение курсора мыши на кнопку, можно проверив координаты курсора в соответствующем событии:

```
function onDocumentMouseMove( event ) {
    var mouseX = event.clientX;
    var mouseY = event.clientY;
    //...
}
```

Изменить материал кнопки, при попадании курсора в её границы, можно присвоив материалу спрайта необходимое значение:

```
button.sprite.material = button.highlight;
```

Примечание: обратите внимание, что координаты, в которых перемещается курсор мыши, отличаются от координаты, в которых размещаются спрайты.



Соответственно, для того, чтобы определить, попадает курсор мыши в границы кнопки или нет, необходимо преобразовать координаты курсора мыши из одной системы координат в другую.

Задание: модифицировать приложения, разработанные в лабораторных работах 2 и 4:

- лаб. №4: добавить объекты типа “кнопка”, сделанные на основе спрайтов. При наведении на объект курсора мыши, его визуальное отображение должно изменяться. При нажатии на объект, должно выполняться добавление в сцену соответствующей модели.
- лаб. №2: добавить вывод в правый верхний угол экран спрайта, с краткой информацией о выбранной планете. Спрайты с информацией должны быть видны только в режиме слежения за планетой.
- лаб. №4: добавить систему частиц реализующую эффект дождя. В меню управления добавить возможность включать и выключать “дождь”, а также, регулировать силу ветра.
- лаб. №2: добавить над планетами спрайты с их названиями. Спрайты должны перемещаться вместе с планетами и отображаться только в режиме общего вида.

Справка: пример функции анимации на основе текстуры:

<http://stemkoski.github.io/Three.js/Texture-Animation.html>

Пример функции рендеринга текста в текстуру:

<http://stemkoski.github.io/Three.js/Sprite-Text-Labels.html>