Оглавление

[Импорт сцены из 3D Max в Ogre 2](#_Toc7164650)

[Материалы 3](#_Toc7164651)

[Простой текстурный переход 8](#_Toc7164652)

[Импорт 3d объектов: 12](#_Toc7164653)

# Импорт сцены из 3D Max в Ogre

1. Строим сцену в 3D Max
2. Экспортируем сцену с помощью плагина, при этом для сцены использовать измерение в пикселях (иначе объекты будут мелкими).
3. Переносим всю структуру папок в ресурсы проекта.
4. Используя функцию Scene\_Load(~Scene, ~SceneName) или имидж SceneLoader подключить сцену к проекту.
5. Для использования объектов сцены используется SceneObject, Player или PhysicObject. При этом свойство AddManually необходимо установить в 0, чтобы объект брался из сцены, а не создавался новый.
6. Обязательно необходимо использовать источник света, иначе объекты будут черными.



# Материалы

Шаблон файла материалов:

// <-Комментарий

//Название материала

material MaterialName

{

//техника

technique

{

//проход

pass

{

//текстурный блок

texture\_unit

{

//текстура

texture image.png

//масштаб

scale 0.1 0.1

}

}

}

}

**Материал** может состоять из множества техник

**Техника** - это один из способов достижения желаемого эффекта. Вы можете предоставить более одного метода, чтобы обеспечить запасные подходы, когда карта не имеет возможности визуализировать предпочтительный метод, или если вы хотите определить версии материала с более низким уровнем детализации, чтобы сохранить силу рендеринга, когда объекты более отдалены.

Каждая техника может состоять из множества **проходов**, то есть полный рендеринг объекта может быть выполнен несколько раз с различными настройками для получения составных эффектов. **[Ogre](https://ogrecave.github.io/ogre/api/latest/namespace_ogre.html" \o " Этот исходный файл является частью OGRE (движка объектно-ориентированной графики).)** может также разделить определенные вами проходы на множество проходов во время выполнения, если вы определите проход, который использует слишком много текстурных блоков для карты, на которой вы в данный момент работаете (обратите внимание, что он может делать это только в том случае, если вы не используете фрагментную программу ).

Рассмотрим различные варианты настроек материала:

**Сплошной цвет**

material Template/Red

{

technique

{

pass

{

texture\_unit

{

colour\_op\_ex source1 src\_manual src\_current 1 0 0

}

}

}

}

'source1' использует первый источник без изменений.

В этом случае первым источником является следующий аргумент.

«src\_manual» означает - использовать цвет, обозначенный тремя числами в конце.

Числа представляют значения RGB в диапазоне 0 и 1.

'src\_current' - это texture\_unit, который смешивается с 'src\_manual', но в этом случае, поскольку операция должна использовать первый источник без изменений, результирующая текстура - это просто сплошной цвет, обозначаемый «1 0 0».

**Простой затененный цвет**

material Template/Red

{

technique

{

pass

{

lighting on

ambient 0.3 0.1 0.1 1

diffuse 0.8 0.05 0.05 1

emissive 0 0 0 1

}

}

}

material Template/Blue

{

technique

{

pass

{

lighting on

ambient 0.3 0.3 0.3 1

diffuse 0.1 0.2 0.7 1

emissive 0 0 0 1

}

}

}

material Templates/RadioactiveGreen

{

technique

{

pass

{

lighting on

ambient 0.1 0.3 0.1 1

diffuse 0.2 0.2 0.2 1

emissive 0.05 0.8 0.05 1

}

}

}

Все цвета формируются с использованием трех значений от 0 до 1, которые представляют компоненты RGB. Некоторые параметры принимают 4 значения, четвертый - альфа-значение (также между 0 и 1). Значение 1 означает белый или максимальную яркость, 0 означает черный или низкую яркость.   
Ambient (Окружающая среда) определяет основной цвет объекта, без какого-либо освещения к нему. Т.к. цветовая модель является аддитивной, то это означает минимальную яркость объекта (когда он находится в абсолютной тени или не светит на него).

Слишком высокое значение (например, полностью белое (1 1 1)) приведет к тому, что все ваши объекты будут полностью белыми. Установка черного цвета (0 0 0) даст вам полностью черные тени.

Diffuse (Рассеянный) цвет определяет основной цвет вашего материала при наличии источников света. Он определяет, какая часть каждого цветового компонента отражается, определяя, какой цвет будет иметь материал, когда на него светит полный белый свет. Обратите внимание, что окружающий цвет добавляется к результату рассеивания, поэтому, если вы установите значение окружения, вам может потребоваться уменьшить значение рассеивания, чтобы компенсировать это.

Emissive - это количество света или цвет, который испускается от объекта, не нуждаясь в каком-либо свете, и дает эффект, как будто объект освещается своим собственным окружающим светом. Обратите внимание, что излучающий свет не освещает другие объекты в сцене, поэтому, если вам нужен светящийся объект, который освещает остальную часть сцены, вам придется добавить дополнительный свет в сцену. Если вы хотите, чтобы цвет окружающей среды отсутствовал, то вам нужен именно этот параметр, а не окружающий.   
Кроме того, есть «зеркальная» настройка, которая позволяет настроить зеркальную составляющую вашего материала. По умолчанию он отключен (0 0 0 0).   
Чтобы эти параметры (окружающий, рассеянный и излучающий) работали, вам нужно включить «освещение». Также использование 'colour\_op replace' в текстурном блоке вашего материала отключит эти настройки.

**Видимость с обеих сторон**

По умолчанию материал виден только с одной стороны.

Для плоских поверхностей (например, водной поверхности) часто нужна видимость с обеих сторон.

material TransparencyExample

{

technique

{

pass

{

//...

cull\_hardware none

cull\_software none

}

}

}

**Прозрачный цвет**

material Template/Red50

{

technique

{

pass

{

scene\_blend alpha\_blend

depth\_write off

texture\_unit

{

colour\_op\_ex source1 src\_manual src\_current 1 0 0

alpha\_op\_ex source1 src\_manual src\_current 0.5

}

}

}

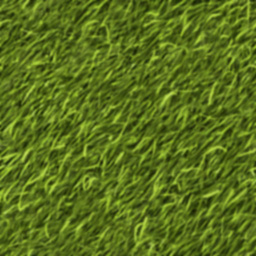
}

Строка 'colour\_op\_ex' объяснена выше. 'Alpha\_op\_ex' имеет ту же структуру аргументов, единственное отличие состоит в том, что значение в конце указывает альфа-значение для использования. Здесь цвет устанавливается на 50% прозрачным. Вы можете поэкспериментировать с различными аргументами для scene\_blend, чтобы придать материалу разные виды прозрачности.

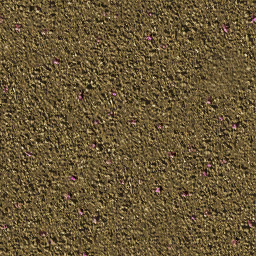
## Простой текстурный переход

**Требования к материалам**

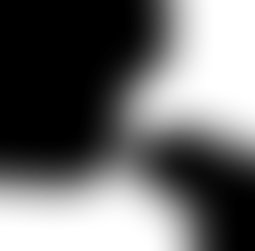
* смешанный файл должен быть в **8-битном** формате (24-битный не работает)



* *Material\_dirt.jpg* - Текстура назначения



* *Material\_alpha\_blend.png* - файл PNG с альфа-маской, которую мы хотим заимствовать для перехода.



Правило смешивания изображения: черная (или цветная) область будет исходной текстурой, а прозрачная область будет целевой текстурой

material Template/texture\_blend

{

technique

{

pass

{

texture\_unit

{

texture Material\_grass.png

}

texture\_unit

{

texture Material\_alpha\_blend.png

colour\_op alpha\_blend

}

texture\_unit

{

texture Material\_dirt.jpg

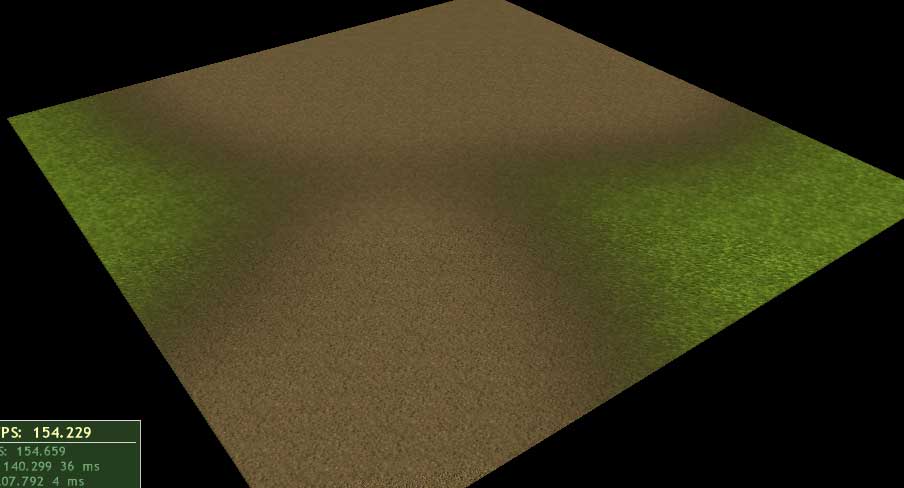
colour\_op\_ex blend\_current\_alpha src\_texture src\_current

}

}

}

}



**Витраж**

Предположим, у вас есть четырехугольник, и вы хотите, чтобы он был частично прозрачным, как витраж.

* image.png - Текстура, которую вы хотите быть прозрачной

material Template/TransparentTexture

{

technique

{

pass

{

lighting off

scene\_blend alpha\_blend

depth\_write off

texture\_unit

{

texture image.png

alpha\_op\_ex source1 src\_manual src\_texture 0.5

}

}

}

}

Подробнее:

<https://ogrecave.github.io/ogre/api/latest/_material-_scripts.html#SEC23>

# Импорт 3d объектов:

Ogre3d читает только свой формат .mesh, .sceleton

Можно конвертировать любой формат с помощью плагина 3d редактора

* 3dsMax – платный редактор
* Blender
* Assimp: <https://github.com/assimp/assimp>.
  + Open Asset Import Library (Assimp) — кроссплатформенная библиотека импорта 3D-моделей, обеспечивающая общий интерфейс программирования приложений (API) для различных форматов. Написана на C++, предоставляет интерфейс на языках C и C++.
  + <http://assimp.org/index.php> - сайт рзрабочиков

Рассмотрим 1вариант:

От 3ds MAX 9 до 3ds MAX 2018:

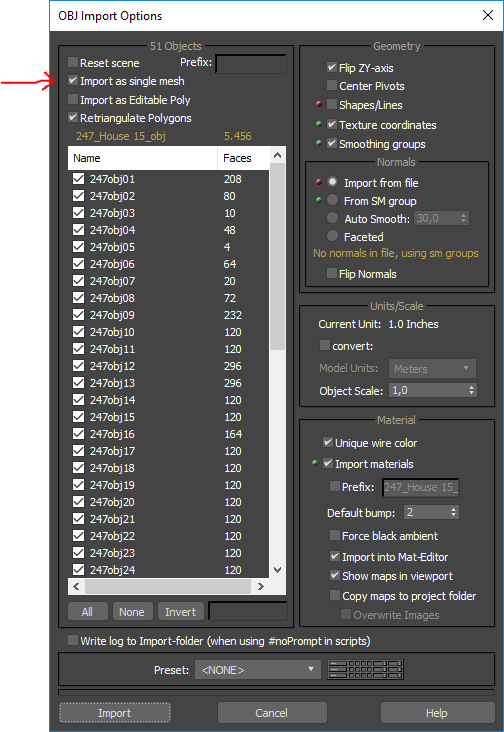
Зайти на сайт для ознакомления с информацией:

<http://wiki.ogre3d.org/Easy+Ogre+Exporter>

Перейти по ссылке [Install for Max 2011 to 2018](http://www.openspace3d.com/downloads/EasyOgreExporter) и скачать файл установки.

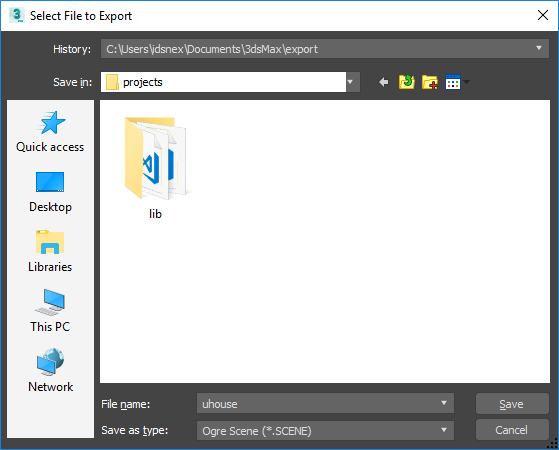
Во время установки версия 3ds Max должна определится сама.

Открываем 3ds Max редактор и импортируем модель (во время импорта есть возможно импортировать как 1 меш)

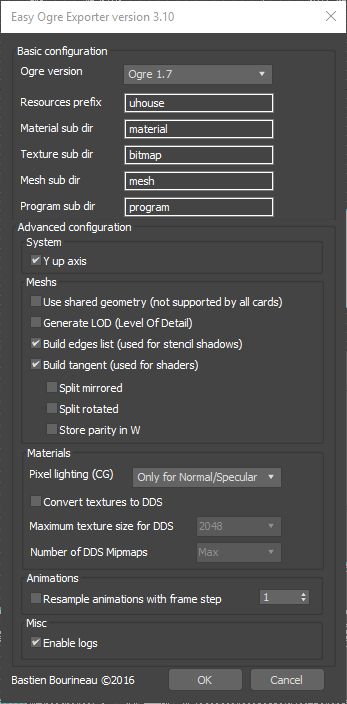


Далее заходим в File -> Export -> Export…

Выбираем тип Ogre Scene

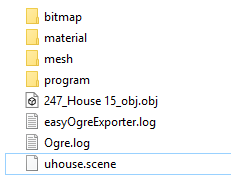


Затем выбираем версию Ogre (В stratum скорее всего Ogre 1.7) и настраиваем Export:



И нажимаем ок. Модель экспортирована в scene, откуда можно получить mesh.

Директория выглядит таким образом:



Для загрузки объекта нам нужен mesh и материал. Они расположены в соответствующих папках. Кладём данные файлы в rsrc проекта (папка ресурсов).

Получить модели можно в данных ресурсах:

<https://archive3d.net/>

<https://free3d.com/3d-models/>