

# Компьютерная графика

## Лекция 20

### Загрузка моделей

Я.М.Демяненко  
dem@math.sfedu.ru

Южный федеральный университет  
Институт математики, механики и компьютерных наук

2019

# Содержание

- 1 Форматы для хранения моделей
  - Реалистичные модели
  - Форматы
- 2 Меши
  - Что нужно хранить?
  - Требования к мешам
- 3 Формат OBJ
  - Собственно формат OBJ
  - Формат MTL
  - Пример
- 4 Формат COLLADA

# Содержание

## 1 Форматы для хранения моделей

- Реалистичные модели
- Форматы

## 2 Меши

- Что нужно хранить?
- Требования к мешам

## 3 Формат OBJ

- Собственно формат OBJ
- Формат MTL
- Пример

## 4 Формат COLLADA

## Что бы хотелось?



## Форматы для хранения моделей

- OBJ
  - очень простой
  - является текстовым
  - не поддерживает анимацию
- 3DS, MAX и BLEND
- бесчисленное количество форматов, используемых в играх и других приложениях
  - Quake I ( .mdl ), Quake II ( .md2 ), Quake III Mesh ( .md3 ), Quake III Map/BSP ( .pk3 )
  - Doom 3 ( .md5\* ), Unreal ( .3d )
  - PovRAY Raw ( .raw ), Terragen Terrain ( .ter ), 3D GameStudio/3DGS ( .mdl )
- COLLADA
  - COLLABorative Design Activity
  - основан на XML
  - поддерживает анимацию

## Библиотеки для загрузки моделей

- Open Asset Import Library (short name: Assimp) is a portable Open Source library
- [http://assimp.sourceforge.net/main\\_features\\_formats.html](http://assimp.sourceforge.net/main_features_formats.html)
- <https://github.com/assimp/assimp>
- Assimp в настоящее время поддерживает 41 формат для импортирования, включая форматы COLLADA (.dae), 3DS, DirectX X, Wavefront OBJ и Blender 3D

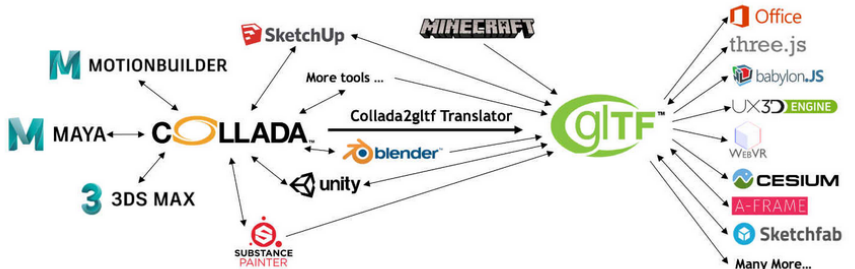
# Collada\_Gltf

## COLLADA FOR AUTHORING INTERCHANGE

Retains extensive data to enable editable assets to be passed between authoring tools

## glTF FOR RUN-TIME TRANSMISSION

Compact file size and efficient processing/import



COLLADA is intended to be used for intermediate interchange, glTF is designed for run-time delivery

glTF can be generated from intermediate formats such as COLLADA, or exported directly from authoring tools to fit individual use cases

Apps and Engines Based on any 3D API

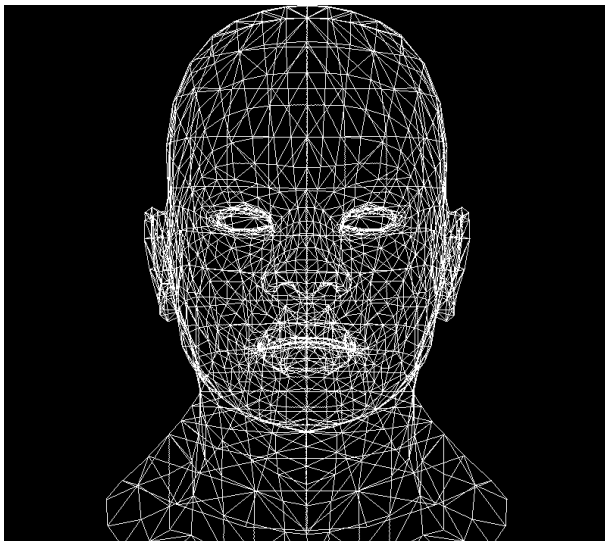


# Содержание

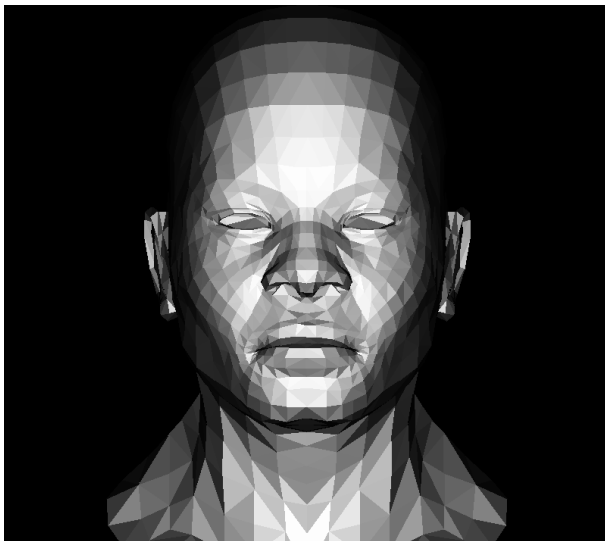
- 1 Форматы для хранения моделей
  - Реалистичные модели
  - Форматы
- 2 Меш
  - Что нужно хранить?
  - Требования к мешам
- 3 Формат OBJ
  - Собственно формат OBJ
  - Формат MTL
  - Пример
- 4 Формат COLLADA



## Первое приближение



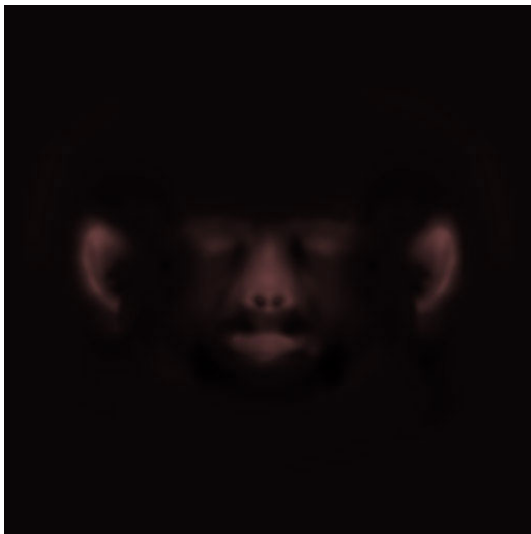
## Что ещё?



И...



## Развертка текстуры



## Нужны ли новые данные?



## Что нужно для этих изменений?



## Карта нормалей



А теперь...?





## Данные, обязательные для экспорта

- координаты вершин
- индексы полигонов
- нормали
- материалы

## Правильный экспорт мешей

- Используйте для всего меша не больше одной текстуры и не больше одного материала
- Если после загрузки меш отображается очень темным, то попробуйте осуществить либо в редакторе либо в функции загрузки меша операцию обращение нормалей
- Если после загрузки меш отображается в виде маленькой точки или не виден совсем, подберите его масштаб
- Указывайте отдельно имя файла меша и отдельно текстуру к нему для легкой замены текстуры

# Содержание

- 1 Форматы для хранения моделей
  - Реалистичные модели
  - Форматы
- 2 Меши
  - Что нужно хранить?
  - Требования к мешам
- 3 Формат OBJ
  - Собственно формат OBJ
  - Формат MTL
  - Пример
- 4 Формат COLLADA

## Формат OBJ

- разработан в Wavefront Technologies для их анимационного пакета Advanced Visualizer
- формат файла является открытым
- может быть экспортирован/импортирован в e-Frontier's Poser, Maya, XSI, Blender, MeshLab, Misfit Model 3D, 3D Studio Max и Rhinoceros 3D, Hexagon, CATIA, Newtek Lightwave, Art of Illusion, milkshape 3d, Modo, Cinema 4D, Zanoza Modeller

## OBJ — это формат файлов описания геометрии

- содержит только 3D геометрию
  - позицию каждой вершины
  - нормаль для каждой вершины
  - связь координат текстуры с вершиной
  - параметры для создания полигонов

## Типы определений

# Список вершин, с координатами  $(x, y, z [, w])$ ,

#  $w$  по умолчанию 1.0

v 0.123 0.234 0.345 1.0

v ...

# Текстурные координаты  $(u, v [, w])$ ,

#  $w$  по умолчанию 0

vt 0.500 1 [0]

vt ...

# Нормали  $(x, y, z)$ ; нормали могут быть не нормированными

vn 0.707 0.000 0.707

vn ...

## Типы определений

```
# Определения поверхности сторон()  
# f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3 v4/vt4/vn4 ...  
f 1 2 3  
f 3/1 4/2 5/3  
f 6/4/1 3/5/3 7/6/5  
f 6//1 3//3 7//5  
f ...  
# Группа  
g Group1  
# Объект  
o Object1
```

# Координаты точек

```
cube_polygon.obj
# 3ds Max Wavefront OBJ Export
# File Created: 07.12.2013 14:

#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

g Box001
f 1 2 3 4
f 5 6 7 8
f 1 4 6 5
f 4 3 7 6
f 3 2 8 7
f 2 1 5 8
# 6 polygons
```

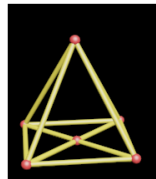


ПОЛИГОНЫ

```
cube_triangle.obj
# 3ds Max Wavefront OBJ Export
# File Created: 07.12.2013 14:

#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

g Box001
f 1 2 3
f 3 4 1
f 5 6 7
f 7 8 5
f 1 4 6
f 6 5 1
f 4 3 7
f 7 6 4
f 3 2 8
f 8 7 3
f 2 1 5
f 5 8 2
# 12 faces
```



Треугольники



# Координаты точек и нормали

```
cube_polygon_normal.obj
# 3ds Max WaveFront OBJ Export
# File Created: 07.12.2013 14

#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

vn 0.0000 -1.0000 -0.0000
vn 0.0000 1.0000 -0.0000
vn 0.0000 0.0000 1.0000
vn 1.0000 0.0000 -0.0000
vn 0.0000 0.0000 -1.0000
vn -1.0000 0.0000 -0.0000
# 6 vertex normals

g Box001
f 1/1/1 2/1/1 3/1/1 4/1/1
f 5/1/2 6/1/2 7/1/2 8/1/2
f 1/1/3 4/1/3 6/1/3 5/1/3
f 4/1/4 3/1/4 7/1/4 6/1/4
f 3/1/5 2/1/5 8/1/5 7/1/5
f 2/1/6 1/1/6 5/1/6 8/1/6
# 6 polygons
```

ПОЛИГОНЫ



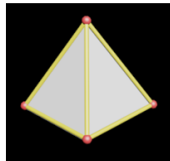
```
cube_triangle_normals.obj
# 3ds Max WaveFront OBJ Export
# File Created: 07.12.2013 14

#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

vn 0.0000 -1.0000 -0.0000
vn 0.0000 1.0000 -0.0000
vn 0.0000 0.0000 1.0000
vn 1.0000 0.0000 -0.0000
vn 0.0000 0.0000 -1.0000
vn -1.0000 0.0000 -0.0000
# 6 vertex normals

g Box001
f 1/1/1 2/1/1 3/1/1
f 3/1/1 4/1/1 1/1/1
f 5/1/2 6/1/2 7/1/2
f 7/1/2 8/1/2 5/1/2
f 1/1/3 4/1/3 6/1/3
f 6/1/3 5/1/3 1/1/3
f 4/1/4 3/1/4 7/1/4
f 7/1/4 6/1/4 4/1/4
f 3/1/5 2/1/5 8/1/5
f 8/1/5 7/1/5 3/1/5
f 2/1/6 1/1/6 5/1/6
f 5/1/6 8/1/6 2/1/6
# 12 faces
```

Треугольники



# Координаты точек и текстур

```
cube_polygon_texture.obj
# 3ds Max Wavefront OBJ Exporter v0.9;
# File Created: 07.12.2013 14:17:03

#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

vt 1.0000 0.0000 0.0000
vt 1.0000 1.0000 0.0000
vt 0.0000 1.0000 0.0000
vt 0.0000 0.0000 0.0000
# 4 texture coords

g Box001
f 1/1 2/2 3/3 4/4
f 5/4 6/1 7/2 8/3
f 1/4 4/1 6/2 5/3
f 4/4 3/1 7/2 6/3
f 3/4 2/1 8/2 7/3
f 2/4 1/1 5/2 8/3
# 6 polygons
```

ПОЛИГОНЫ

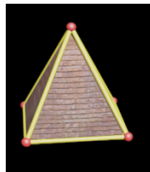


```
cube_triangle_texture.obj
# 3ds Max Wavefront OBJ Exporter
# File Created: 07.12.2013 14
#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

vt 1.0000 0.0000 0.0000
vt 1.0000 1.0000 0.0000
vt 0.0000 1.0000 0.0000
vt 0.0000 0.0000 0.0000
# 4 texture coords

g Box001
f 1/1 2/2 3/3
f 3/3 4/4 1/1
f 5/4 6/1 7/2
f 7/2 8/3 5/4
f 1/4 4/1 6/2
f 6/2 5/3 1/4
f 4/4 3/1 7/2
f 7/2 6/3 4/4
f 3/4 2/1 8/2
f 8/2 7/3 3/4
f 2/4 1/1 5/2
f 5/2 8/3 2/4
# 12 faces
```

Треугольники



Всё

```
cube_polygon_texture_normal.obj
# 3ds Max Wavefront OBJ Exporter v0.97b - (c)
# File Created: 07.12.2013 14:17:57

#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

vn 0.0000 -1.0000 -0.0000
vn 0.0000 1.0000 -0.0000
vn 0.0000 0.0000 1.0000
vn 1.0000 0.0000 -0.0000
vn 0.0000 0.0000 -1.0000
vn -1.0000 0.0000 -0.0000
# 6 vertex normals

vt 1.0000 0.0000 0.0000
vt 1.0000 1.0000 0.0000
vt 0.0000 1.0000 0.0000
vt 0.0000 0.0000 0.0000
# 4 texture coords

g Box001
f 1/1/1 2/2/1 3/3/1 4/4/1
f 5/4/2 6/1/2 7/2/2 8/3/2
f 1/4/3 4/1/3 6/2/3 5/3/3
f 4/4/4 3/1/4 7/2/4 6/3/4
f 3/4/5 2/1/5 8/2/5 7/3/5
f 2/4/6 1/1/6 5/2/6 8/3/6
# 6 polygons
```

ПОЛИГОНЫ



```
cube_triangle_texture_normal.obj
# 3ds Max Wavefront OBJ Exporter v0.97b - (c)2007 gururaw
# File Created: 07.12.2013 14:15:28

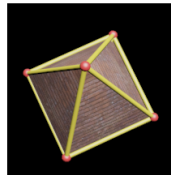
#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

vn 0.0000 -1.0000 -0.0000
vn 0.0000 1.0000 -0.0000
vn 0.0000 0.0000 1.0000
vn 1.0000 0.0000 -0.0000
vn 0.0000 0.0000 -1.0000
vn -1.0000 0.0000 -0.0000
# 6 vertex normals

vt 1.0000 0.0000 0.0000
vt 1.0000 1.0000 0.0000
vt 0.0000 1.0000 0.0000
vt 0.0000 0.0000 0.0000
# 4 texture coords

g Box001
f 1/1/1 2/2/1 3/3/1
f 3/3/1 4/4/1 1/1/1
f 5/4/2 6/1/2 7/2/2
f 7/2/2 8/3/2 5/4/2
f 1/4/3 4/1/3 6/2/3
f 4/2/3 5/3/3 1/4/3
f 4/4/4 3/1/4 7/2/4
f 7/2/4 6/3/4 4/4/4
f 3/4/5 2/1/5 8/2/5
f 8/2/5 7/3/5 3/4/5
f 2/4/6 1/1/6 5/2/6
f 5/2/6 8/3/6 2/4/6
# 12 faces
```

Треугольники



## Модель в формате obj



## Информация о внешнем виде объектов(материалы)

- передается в файлах-спутниках в формате MTL (Material Library)
- ссылка с помощью директивы: `mtllib [имя внешнего MTL файла]`
- информация представлена в текстовом виде

# MTL

# Объявление очередного материала  
newmtl названиеМатериала1

# Цвета

# Цвет окружающего освещения желтый()

Ka 1,000 1,000 0,000

# Диффузный цвет белый()

Kd 1,000 1,000 1,000

# Параметры отражения

# Цвет зеркального отражения (0;0;0 – выключен)

Ks 0,000 0,000 0,000

# Коэффициент зеркального отражения от( 0 до 1000)

Ns 10,000

# MTL

# Параметры прозрачности

# Прозрачность указывается с помощью директивы d  
d 0,9

# или в других реализациях формата с помощью Tr  
Tr 0,9

# Следующий материал  
newmtl названиеМатериала2

...

## Общая структура MTL

```
newmtl my_red
```

```
Material color & illumination statements
```

```
texture map statements
```

```
reflection map statement
```



## Texture maps 1

```
newmtl Textured
Ka 1.000 1.000 1.000
Kd 1.000 1.000 1.000
Ks 0.000 0.000 0.000
d 1.0
illum 2
```

## Texture maps 2

```
map_Ka lenna.tga
# the ambient texture map
map_Kd lenna.tga
# the diffuse texture map (most of the time,
# it will be the same as the ambient texture map)
map_Ks lenna.tga
# specular color texture map
map_Ns lenna_spec.tga
# specular highlight component
map_d lenna_alpha.tga
# the alpha texture map
```

## Вершины

```
#  
mtllib ./box2.mtl  
g  
# object (null) to come ...  
#  
v -1.968504 0.000000 1.968504 [1]  
v 1.968504 0.000000 1.968504 [2]  
v -1.968504 0.000000 -1.968504 [3]  
v 1.968504 0.000000 -1.968504 [4]  
v -1.968504 3.937008 1.968504 [5]  
v 1.968504 3.937008 1.968504 [6]  
v -1.968504 3.937008 -1.968504 [7]  
v 1.968504 3.937008 -1.968504 [8]  
# 8 vertices
```

## Текстурные координаты

```
vt 0.000000 0.000000 0.000000 [1]
vt 1.000000 0.000000 0.000000 [2]
vt 0.000000 1.000000 0.000000 [3]
vt 1.000000 1.000000 0.000000 [4]
vt 0.000000 0.000000 0.000000 [5]
vt 1.000000 0.000000 0.000000 [6]
vt 0.000000 1.000000 0.000000 [7]
vt 1.000000 1.000000 0.000000 [8]
vt 0.000000 0.000000 0.000000 [9]
vt 1.000000 0.000000 0.000000 [10]
vt 0.000000 1.000000 0.000000 [11]
vt 1.000000 1.000000 0.000000 [12]
# 12 texture vertices
```

## Нормали

```
vn 0.000000 -2.000000 0.000000 [1]
vn 0.000000 -1.000000 0.000000 [2]
vn 0.000000 -1.000000 0.000000 [3]
vn 0.000000 -2.000000 0.000000 [4]
vn 0.000000 2.000000 0.000000 [5]
vn 0.000000 1.000000 0.000000 [6]
vn 0.000000 1.000000 0.000000 [7]
vn 0.000000 2.000000 0.000000 [8]
# 8 vertex normals
```

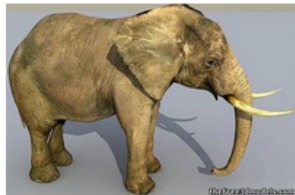
## Грани

```
g (null)
f 1/10/1 3/12/3 4/11/4
f 4/11/4 2/9/2 1/10/1
f 5/9/5 6/10/6 8/12/8
f 8/12/8 7/11/7 5/9/5
f 1/5/1 2/6/2 6/8/6
f 6/8/6 5/7/5 1/5/1
f 2/1/2 4/2/4 8/4/8
f 8/4/8 6/3/6 2/1/2
f 4/5/4 3/6/3 7/8/7
f 7/8/7 8/7/8 4/5/4
f 3/1/3 1/2/1 5/4/5
f 5/4/5 7/3/7 3/1/3
# 12 faces
```

## Получаем

```
indices [ count++]=0[new ] ;  
indices [ count++]=1[new ] ;  
indices [ count++]=2[new ] ;  
indices [ count++]=3[new ] ;  
indices [ count++]=2[copy ] ;  
indices [ count++]=4[new ] ;
```

## Модели в формате obj



<https://free3d.com/>



# Содержание

- 1 Форматы для хранения моделей
  - Реалистичные модели
  - Форматы
- 2 Меши
  - Что нужно хранить?
  - Требования к мешам
- 3 Формат OBJ
  - Собственно формат OBJ
  - Формат MTL
  - Пример
- 4 Формат COLLADA

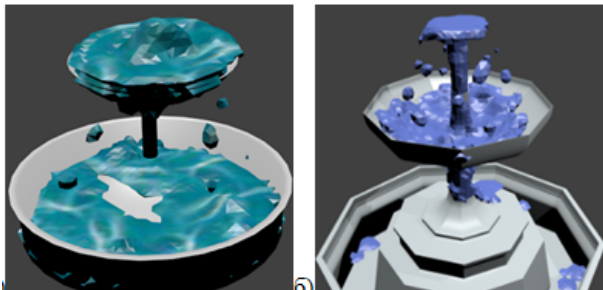
## Collada \*.dae

- Был задуман как промежуточный формат для переноса файлов
- Реализована поддержка таких программ, как
  - Maya (используя ColladaMaya); 3ds Max (при помощи ColladaMax); Poser (v.7.0); Lightwave 3D (version 9.5); Cinema 4D (MAXON); Softimage|XSI; Houdini; MeshLab; CityScape, CityEngine, SketchUp, Blender, modo и Strata 3D. Adobe Photoshop с версии CS3. Игровые движки, такие как Unreal engine, Unity и Torque 3D,
- Библиотека Assimp также поддерживает формат COLLADA
- 27 марта 2013 года Khronos Group анонсировали, что COLLADA 1.5.0 был опубликован в качестве официального стандарта ISO

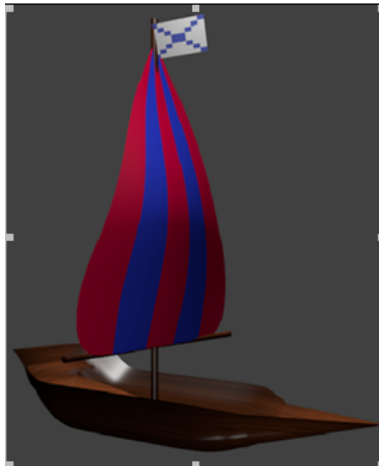
## COLLADA поддерживает

- геометрию (вершины, NURBS )
- материалы
- текстуры
- деформацию
- анимацию
- ограничители (constraints) и ИК (цепочки инверсной кинематики)

## Поддержка ограничителей



## Поддержка физики



# COLLADA - основная часть

- сцены
- камеры
- источники света
- модели
- анимации

## COLLADA - дополнительная часть

- состоит из, выделенных в отдельный модуль,
  - COLLADA FX - отвечает за эффекты, их параметры, код шейдеров.
  - COLLADA Physics - отвечает за физические объекты, их параметры. Сюда входят все основные физические объекты, настройки симуляции, данный модуль поддерживается многими физическими движками, такими как Bullet, Open Dynamics Engine, PAL и NVIDIA PhysX.

## Collada \*.dae

```
<?xml version="1.0" encoding="utf-8"?>  
<COLLADA xmlns="http://www.collada.org/  
2005/11/COLLADASchema" version="1.4.1">  
  <asset>  
    ...  
  </asset>  
<library_images>  
  ...  
</library_images>  
<library_materials>  
  ...  
</library_materials>  
  ...  
</COLLADA>
```



## блок <asset>

```
<asset>  
<contributor>  
<authoring_tool>Converted by the PolyTrans Collada exp  
</contributor>  
<created>20XX-04-09T14:24:50Z</created>  
<modified>20XX-04-09T14:24:50Z</modified>  
<unit meter="0.010000" name="centimeters"/>  
  <up_axis>Y_UP</up_axis>  
</asset>
```

## блок <library\_images>

```
<image id="ceramic_tif" name="ceramic_tif">  
  <init_from>file:///ceramic.tif</init_from>  
</image>
```

## блок <library\_materials>

```
<material id="/lambert1" name="/lambert1">  
  <instance_effect url="#/lambert1-fx">  
  </instance_effect>  
</material>
```

## блок <library\_effects>

```
<effect id="lambert1-fx">  
  <profile_COMMON>  
    <technique sid="common">  
      <lambert>  
        <emission> <color>0 0 0 1</color> </emission>  
        <ambient> <color>0 0 0 1</color> </ambient>  
        <diffuse> <color>0.5 0.5 0.5 1</color> </diffuse>  
        <transparent opaque="A_ONE"> <color>0 0 0 1</color>  
          <transparency> <float>1</float> </transparency>  
          <index_of_refraction> <float>1</float> </index_of_refraction>  
        </lambert>  
        <extra> <technique profile="FCOLLADA"></technique>  
      </technique>  
    </profile_COMMON>  
  </effect>
```

# Геометрия

```
<geometry id="pCylinderShape1" name="pCylinderShape1">  
  <mesh>
```

```
    <float_array id="pCylinderShape1-positions-array" count="126">20 -90  
-34.641  
-20 -90 -34.641 -40 -90 -0.000005 -20 -90 34.641 20 -90 34.641 40 -90 0 20 -60  
-34.641  
-20 -60 -34.641 -40 -60 -0.000005 -20 -60 34.641 20 -60 34.641 40 -60 0 20 -30  
-34.641  
-20 -30 -34.641 -40 -30 -0.000005 -20 -30 34.641 20 -30 34.641 40 -30 0 20 0  
-34.641  
-20 0 -34.641 -40 0 -0.000005 -20 0 34.641 20 0 34.641 40 0 0 20 30 -34.641 -20 30  
-34.641  
-40 30 -0.000005 -20 30 34.641 20 30 34.641 40 30 0 20 60 -34.641 -20 60 -34.641  
-40 60 -0.000005 -20 60 34.641 20 60 34.641 40 60 0 20 90 -34.641 -20 90 -34.641  
-40 90 -0.000005 -20 90 34.641 20 90 34.641 40 90 0</float_array>  
    <technique_common>  
      <accessor source="#pCylinderShape1-positions-array" count="42"  
stride="3">  
        <param name="X" type="float"></param>  
        <param name="Y" type="float"></param>  
        <param name="Z" type="float"></param>  
      </accessor>  
    </technique_common>
```

## Ссылки

- официальный сайт - <https://collada.org/>
- Twitter разработчиков COLLADA - <https://twitter.com/collada>
- спецификация COLLADA 1.5 от Khronos Group -  
[http://www.gamasutra.com/php-bin/news\\_index.php?story=19729](http://www.gamasutra.com/php-bin/news_index.php?story=19729)