

Компьютерная графика

Загрузка моделей. Построение локаций

Лекция 6

Демяненко Я.М. ЮФУ 2024 MAGnUS

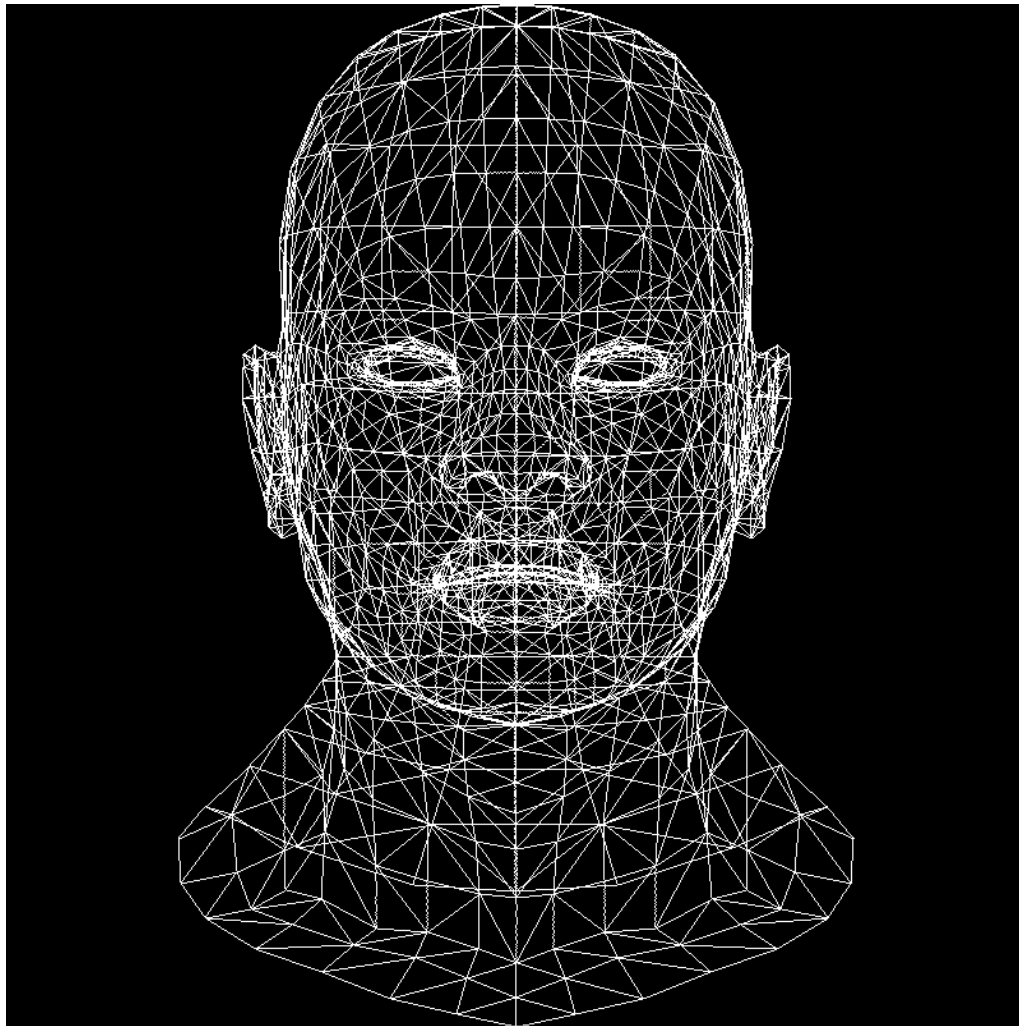
Реалистичные модели. Что бы хотелось?

Меши

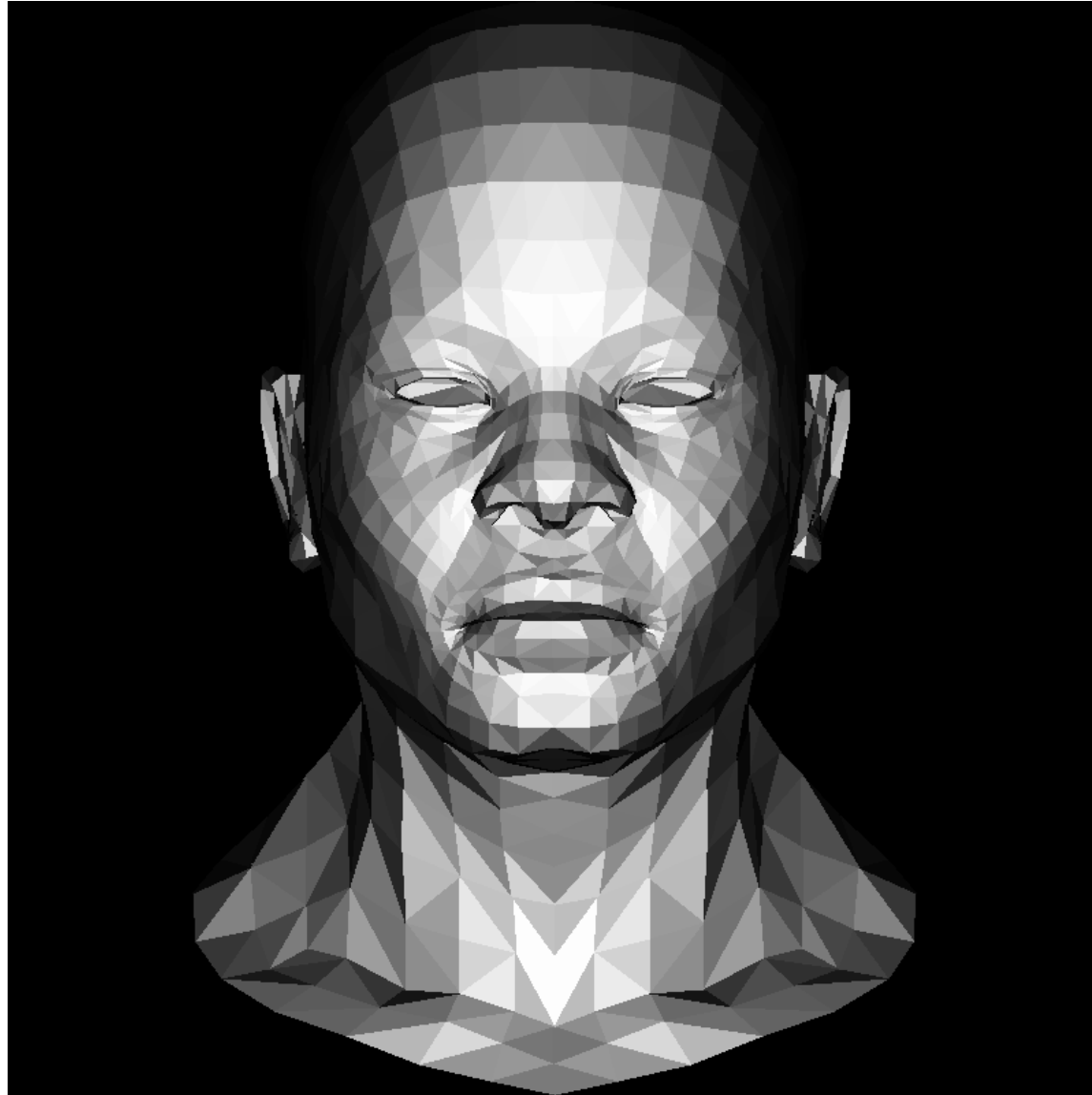


Что нужно хранить?

Первое приближение



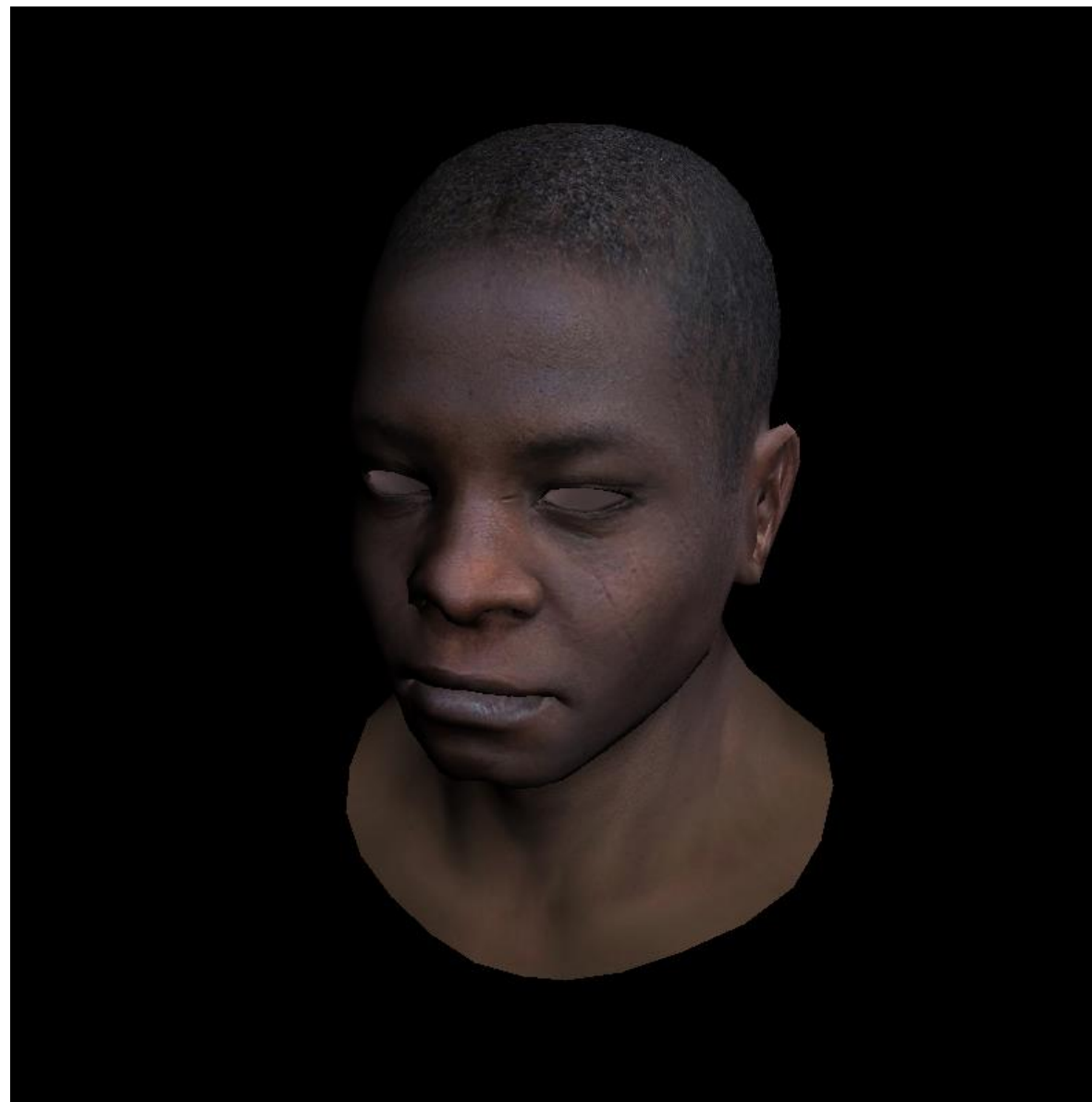
Что ещё?



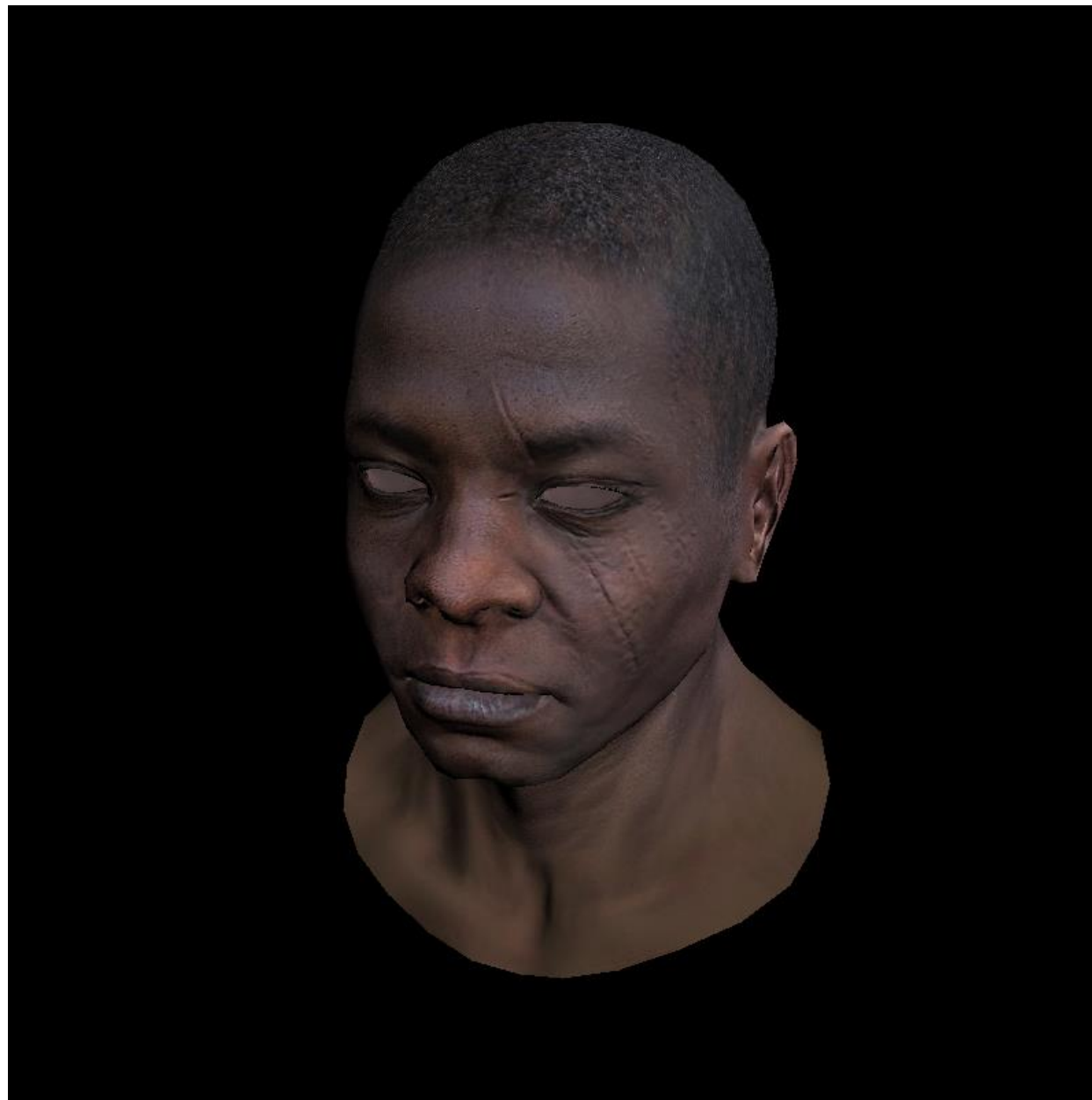
И...



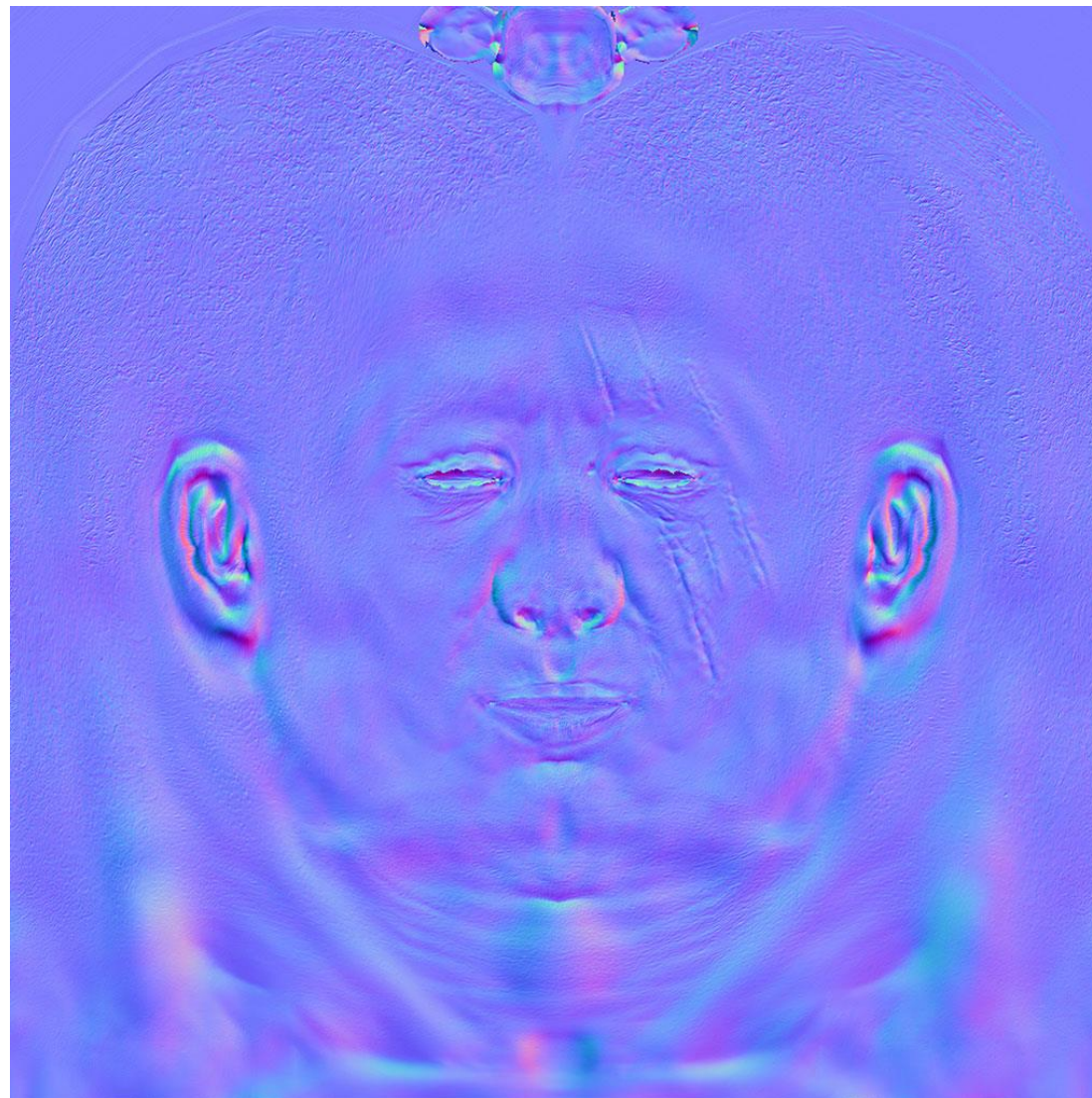
Нужны ли
новые данные?



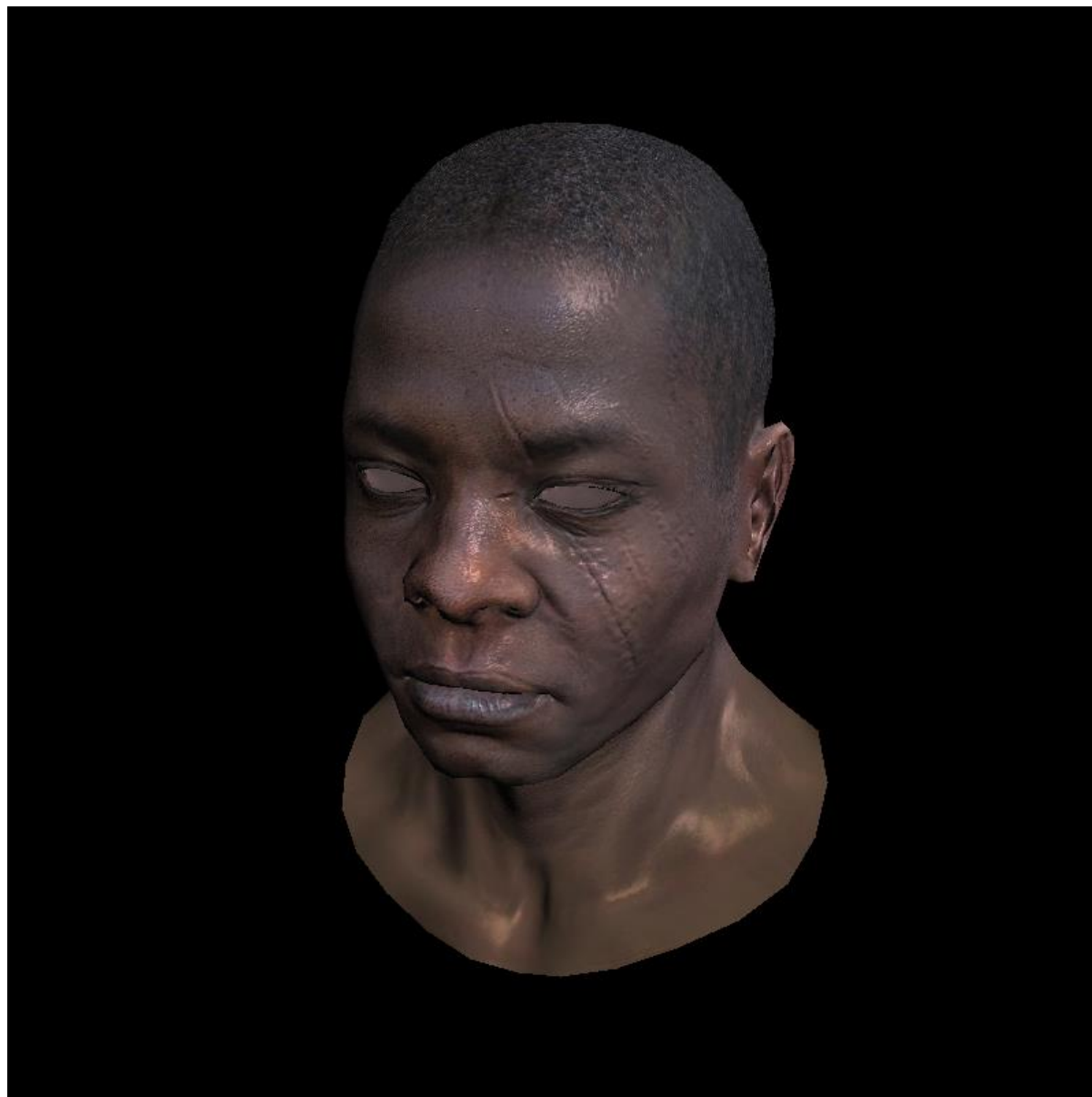
Что нужно для
этих изменений?



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 загрузке 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

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

Формат OBJ

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

Модели в формате obj



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
```



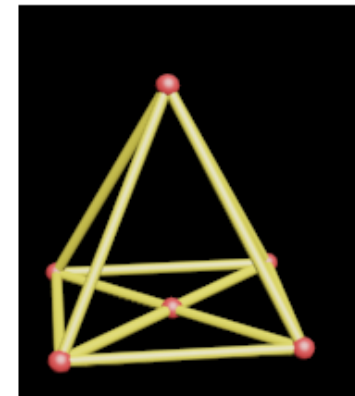
ПОЛИГОНЫ

```
cube_tringle.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 Expor
# File Created: 07.12.2013 14

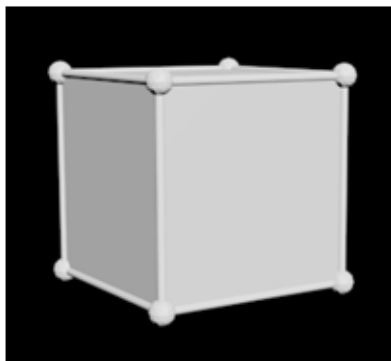
#
# object Box001
#

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

ПОЛИГОНЫ



```
cube_tringle_normals.obj
# 3ds Max Wavefront OBJ Expor
# File Created: 07.12.2013 14

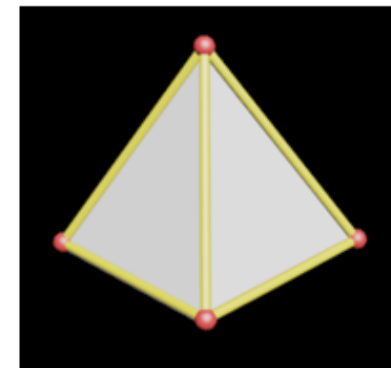
#
# object Box001
#

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

Треугольники



Координаты точек и текстуры

```
cube_polygon_texture.obj
# 3ds Max Wavefront OBJ Exporter v0.97
# File Created: 07.12.2013 14:17:03

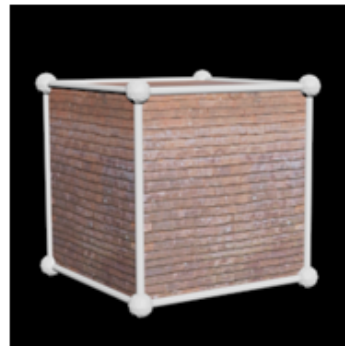
#
# object Box001
#

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_tringle_texture.obj
# 3ds Max Wavefront OBJ Exporter
# 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
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 - (
# 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 guruware
# 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 6/2/3 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

Пример с MTL

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

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

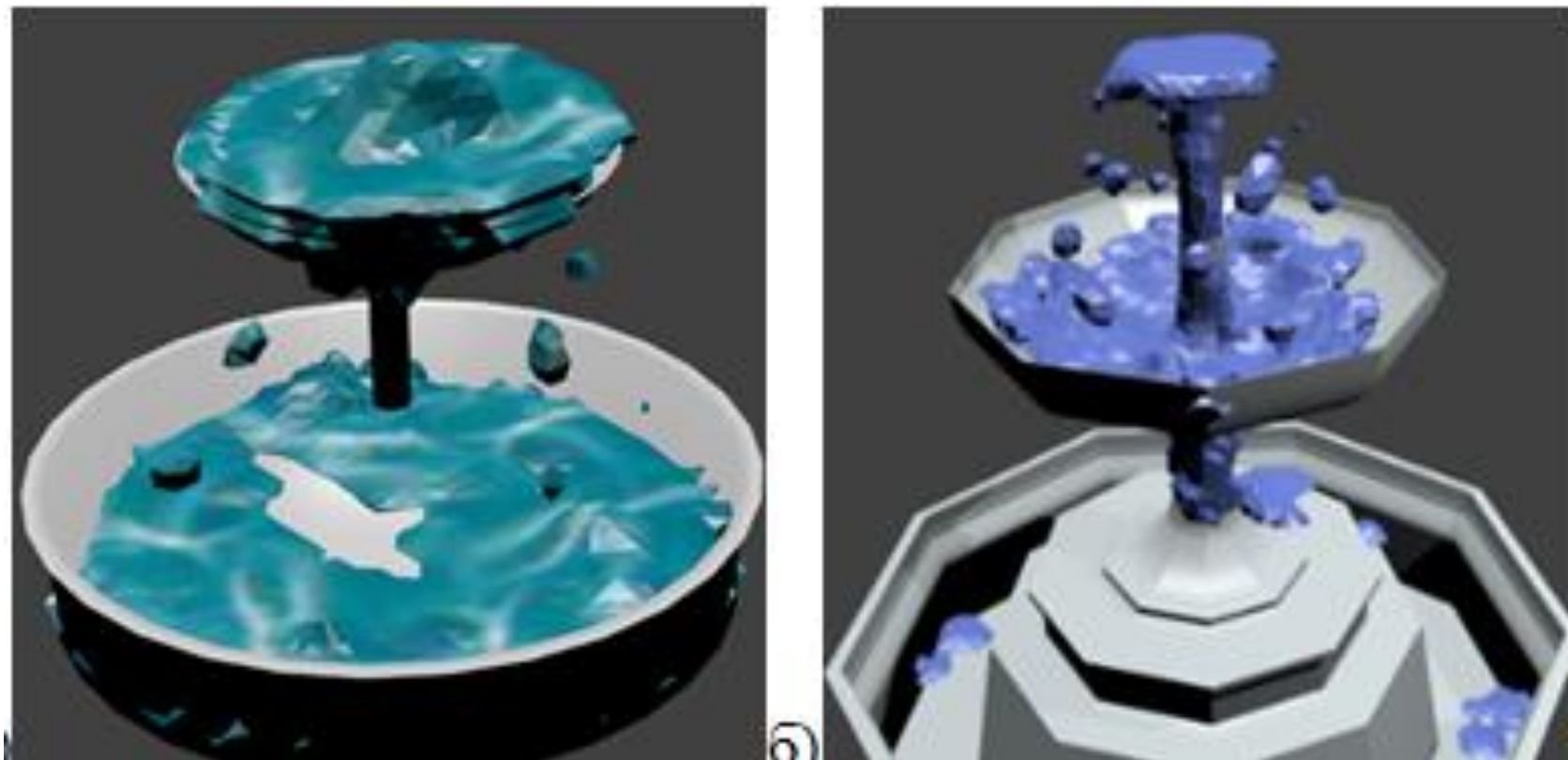
текстуры

деформацию

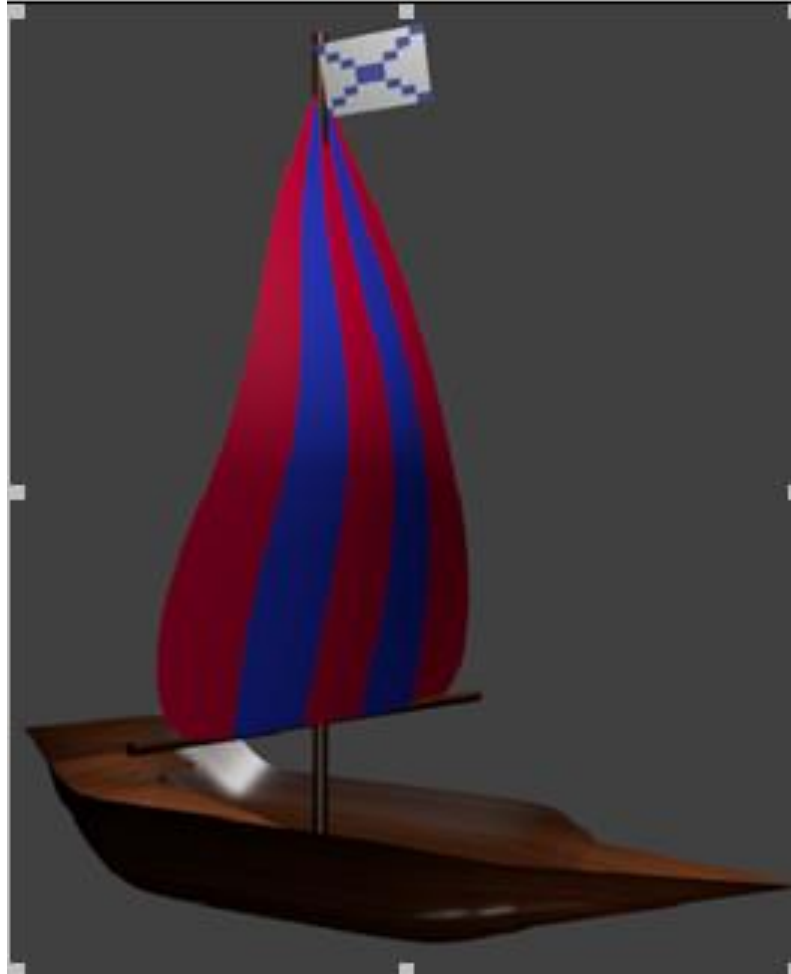
анимацию

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 Bulle, 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 exporter</authoring_tool>
  </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> </transparent>
        <transparency> <float>1</float> </transparency>
        <index_of_refraction> <float>1</float> </index_of_refraction>
      </lambert>
      <extra> <technique profile="FCOLLADA"></technique> </extra>
    </technique>
  </profile_COMMON>
</effect>
```

Геометрия

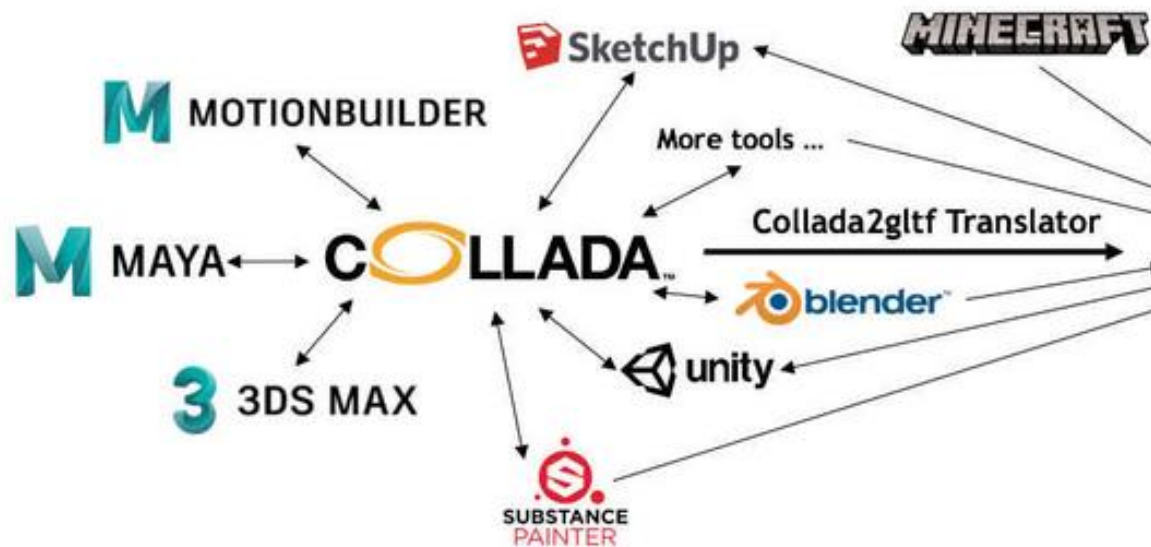
```
<geometry id="pCylinderShape1" name="pCylinderShape1">  
  <mesh>
```

```
    <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>
```

Collada\Gltf

COLLADA FOR AUTHORING INTERCHANGE

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



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

glTF FOR RUN-TIME TRANSMISSION

Compact file size and efficient processing/import



Apps and Engines Based on any 3D API



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

Библиотеки для загрузки моделей

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++ и др. языков

Данные импортируются в виде простой иерархической структуры данных. Настраиваемые эффекты пост-обработки (например, генерация касательного пространства, различные оптимизации) расширяют список возможностей[3].

«Assimp» в настоящее время поддерживает 57 формат для импортирования, включая форматы COLLADA (.dae), 3DS, DirectX (.x), Wavefront OBJ и Blender 3D (.blend)[4]. Версия 3.0 также обеспечивает экспорт некоторых форматов файлов[5].

http://assimp.sourceforge.net/main_features_formats.html

<https://github.com/assimp/assimp>

Three.js - Загрузка 3D-моделей

Вы можете найти все различные загрузчики в репозитории Three.js в каталоге `three/examples/jsm/loaders`.

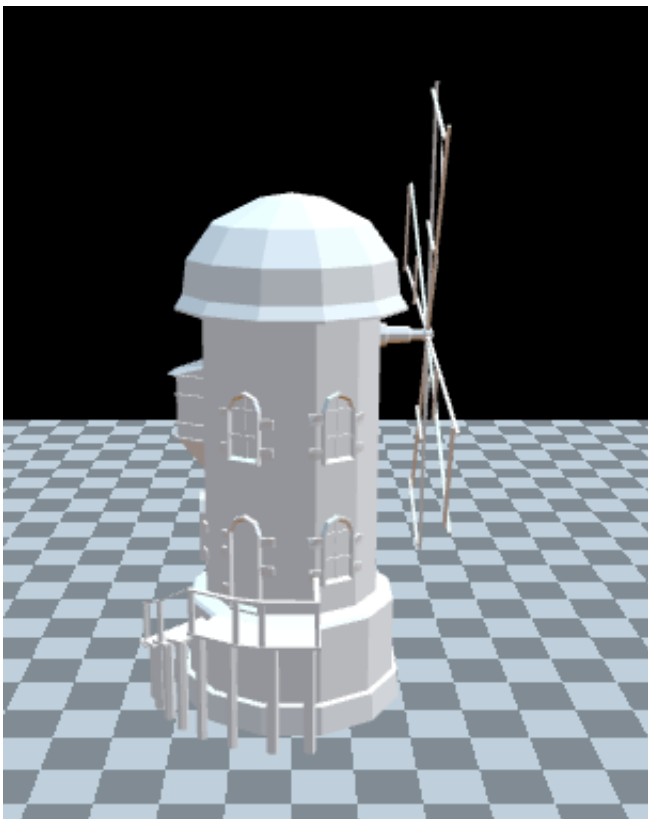
Первое, что нужно сделать, это включить OBJLoader загрузчик в скрипт.

```
import {OBJLoader} from 'three/addons/loaders/OBJLoader.js';
```

Затем, чтобы загрузить файл .OBJ, создаем экземпляр OBJLoader, передаем ему URL-адрес файла .OBJ и передаем обратный вызов, который добавляет загруженную модель в сцену.

```
{  
  const objLoader = new OBJLoader();  
  objLoader.load('resources/models/windmill/windmill.obj', (root) => {  
    scene.add(root);  
  });  
}
```

Результат



файлы .OBJ не имеют параметров материала.

Загрузчику .OBJ можно передать объект пары имя/материал. Когда он загружает файл .OBJ, любое имя материала, которое он находит, будет искать соответствующий материал в карте материалов, установленной в загрузчике. Если он найдет материал, соответствующий по названию, он будет использовать этот материал. В противном случае будет использоваться материал загрузчика по умолчанию.

Иногда файлы .OBJ поставляются с файлом .MTL, определяющим материалы.

Формат .MTL представляет собой простой ASCII, поэтому его легко просматривать.

Файл MTL блендера: 'windmill_001.blend'
Количество материалов: 2

newmtl Material

Ns 0.000000

Ka 1.000000 1.000000 1.000000

Kd 0.800000 0.800000 0.800000

Ks 0.000000 0.000000 0.000000

Ke 0.000000 0.000000 0.000000

Ni 1.000000

d 1.000000

illum 1

map_Kd windmill_001_lopatky_COL.jpg

map_Bump windmill_001_lopatky_NOR.jpg

newmtl windmill

Ns 0.000000

Ka 1.000000 1.000000 1.000000

Kd 0.800000 0.800000 0.800000

Ks 0.000000 0.000000 0.000000

Ke 0.000000 0.000000 0.000000

Ni 1.000000

d 1.000000

illum 1

map_Kd windmill_001_base_COL.jpg

map_Bump windmill_001_base_NOR.jpg

map_Ns windmill_001_base_SPEC.jpg

Теперь, когда у нас есть доступные текстуры, мы можем загрузить файл .MTL.

Сначала нам нужно включить MTLLoader;

```
import * as THREE from 'three';  
import {OrbitControls} from 'three/addons/controls/OrbitControls.js';  
import {OBJLoader} from 'three/addons/loaders/OBJLoader.js';  
import {MTLLoader} from 'three/addons/loaders/MTLLoader.js';
```

Затем мы сначала загружаем файл .MTL. Когда загрузка завершена, мы добавляем только что загруженные материалы к OBJLoader через setMaterials и затем загружаем файл .OBJ.

```
{  
  const mtlLoader = new MTLLoader();  
  mtlLoader.load('resources/models/windmill/windmill.mtl', (mtl) => {  
    mtl.preload();  
    objLoader.setMaterials(mtl);  
    objLoader.load('resources/models/windmill/windmill.obj', (root) => {  
      scene.add(root);  
    });  
  });  
}
```

Результат с материалами



Игровые локации в современных играх: как сюжет раскрывается через окружение

Что можете сказать?



Роль локаций в игре: пользователь не должен скучать



Возможные локации в одной игре

- пустыни,
- оазисы,
- города,
- пещеры,
- водные преграды,
- многое другое.

Каждая игровая локация имеет свои изюминки, а также уникальных врагов, если таковые предусмотрены сюжетом игры.

Развитие

Зоны могут постоянно развиваться,
добавляться новые сектора, истории,

при этом важно поддерживать сюжет,

а также учитывать современные требования к графике, технические возможности оборудования.

World of Warcraft (MMORPG), включает пять континентов с десятками локаций на каждом.



Казуальные разработки, стратегии, жанр hidden objects

в таких проектах дизайн игровых локаций максимально важен, ведь с их помощью создается уникальная атмосфера игры.





Основные виды локаций, которые сгруппированы по разным критериям

- Пустота и заселенность
- Опасность и спокойствие
- Природа
- Киберпанк
- Стимпанк
- Локации в играх жанра hidden object

Пустота и заселенность

Практически все многопользовательские, а особенно RPG-игры, имеют живые продуманные миры, где есть неигровые персонажи, иногда присутствуют животные.

NPC имеют реалистичное поведение, с ними можно взаимодействовать, причем не только по ходу сюжета.

Также есть незаселенные локации, например, в survival-играх, хоррорах, постапокалиптических шутерах, других подобных жанрах.

Пустые локации создают атмосферу тревожности, хаоса, беспорядка, которая передается геймеру.

Если в населенных зонах можно заинтересовать пользователя посредством действий NPC, пустые локации требуют отличной дизайнерской проработки деталей, поскольку внимание игрока будет сосредоточено на них.

Опасность и спокойствие

Обычно в играх комбинируют локации с высоким и низким уровнем опасности, позволяя пользователю отдохнуть между напряженными участками сюжета.

Спокойные зоны дают игроку возможность насладиться проработкой окружающего пространства, выполнить дополнительные действия (проверить экипировку, пополнить уровень здоровья, отправить письма).

Опасные локации обычно имеют гнетущую атмосферу, которая заставляет пользователя быть внимательнее.

Что можете сказать?



Природа

Это достаточно сложные по проработке локации, которые обычно являются переходами между основными зонами.

Для них нужно визуализировать естественный климат игрового мира, хорошо проработать погодные явления, включить детализированные растения, животных, вплоть до фотореалистичности.

Зачастую хорошо проработанные природные локации плотно исследуются игроками, поэтому для них разрабатывают дополнительные задания. На этом построены игры «Ведьмак 3», Uncharted 4, Assassin's Creed, где природа – важная часть сюжета.



Киберпанк

Еще одна интересная разновидность – киберпанк, где основной идеей мира является технический прогресс, вплоть до виртуальной реальности, который обостряет проблемы общества.

Для атмосферности левел-дизайнеры применяют много неона и подсветки, используют стиль, близкий к азиатскому, изоцряются на тему техники будущего, модификаций организмов.

Стимпанк

Он открывает простор для фантазии дизайнеров, позволяя им подумать о том, каким был бы мир, где энергия для жизни цивилизации генерируется паровыми двигателями.

Локации оформляются многочисленными облаками пара, шестеренками, трубами, прочими индастриал-объектами.

При этом исторический сеттинг стимпанка обычно разворачивается в альтернативных XIX-XX веках, что требует соответствующей проработки сеттинга и артов.



Локации в играх жанра hidden object



Ведьмак 3: Дикая охота



Death Stranding



Final Fantasy XV

