

Компьютерная графика

Загрузка моделей. Построение локаций

Лекция 5

Демяненко Я.М. ЮФУ 2025

Основные этапы создания и визуализации 3D моделей в кино и game-индустриях

- **Моделирование** – создание трехмерных объектов
- **Текстурирование** – наложение текстур и материалов на 3D-модели
- **Риггинг** – создание виртуального скелета, набора костей/суставов для последующей анимации персонажа
- **Анимация** – оживление, анимирование трёхмерного персонажа
- **Рендеринг** – 3D визуализация созданной графики
- **Композитинг** – объединение отдельных элементов в финальную сцену. К примеру, интегрирование 3D сцен в съёмочный материал, цветокоррекция и добавление эффектов

Моделирование

В фильмах	В играх
можно использовать криволинейные поверхности (NURBS-моделирование) и полигоны (полигональное моделирование)	обычно используют только полигональные модели, их проще всего визуализировать
обычно создают высокополигональные модели, рендеринг которых, проходит по несколько часов, а то и дней	используются низкополигональные модели, визуализация происходит прямо по ходу игры. Часто в компьютерных играх встречается LOD-технология (Level of Detail – уровень детализации)

3D-редакторы Autodesk Maya, Autodesk 3Ds Max, Cinema 4D, Modo и Blender (бесплатный) для цифрового скульптинга: ZBrush, Mudbox, 3D Coat, Sculptris

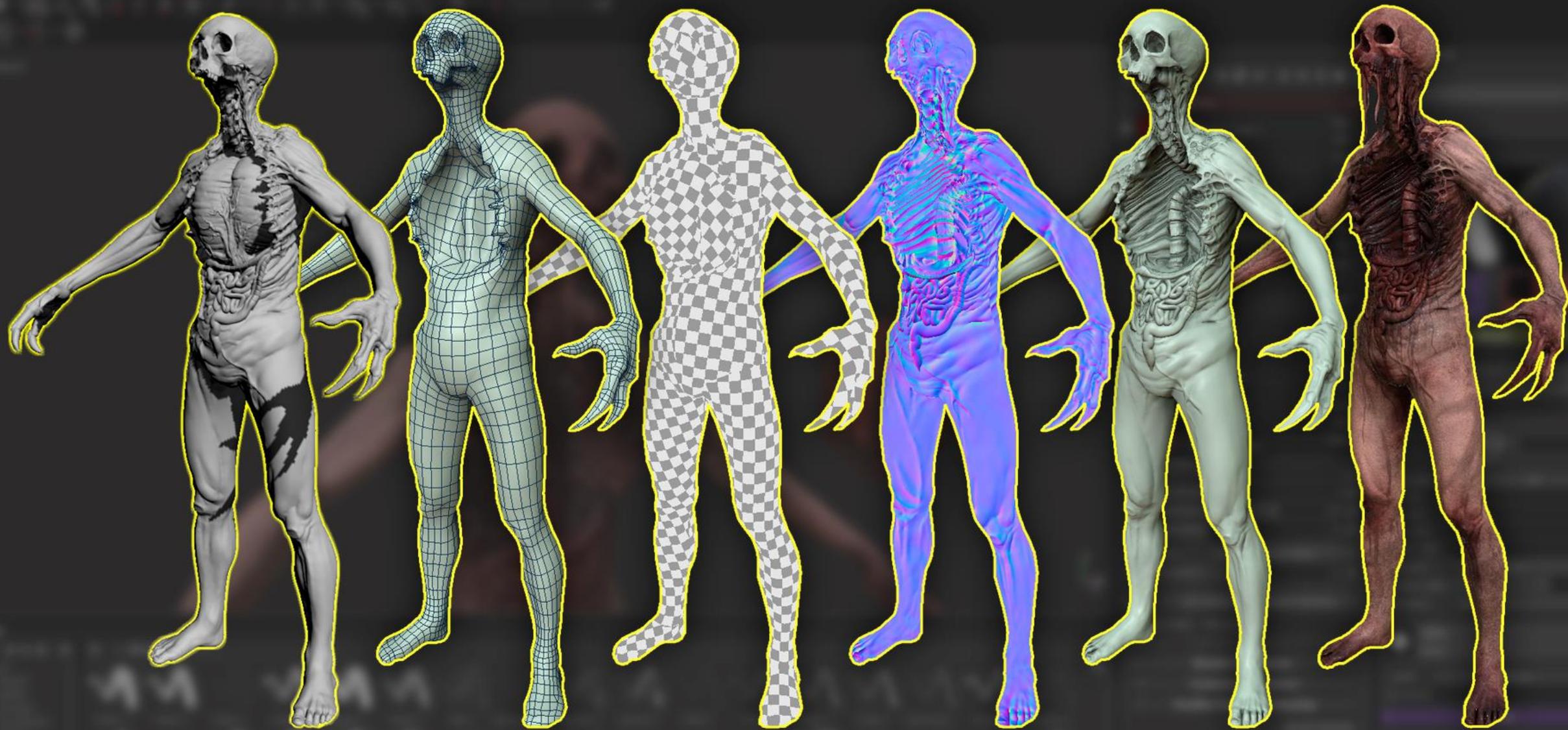
Основные этапы создания и визуализации 3D моделей в кино и game-индустриях

- Моделирование – создание трехмерных объектов
- **Текстурирование** – наложение текстур и материалов на 3D-модели
- Риггинг – создание виртуального скелета, набора костей/суставов для последующей анимации персонажа
- Анимация – оживление, анимирование трёхмерного персонажа
- Рендеринг – 3D визуализация созданной графики
- Композитинг – объединение отдельных элементов в финальную сцену. К примеру, интегрирование 3D сцен в съёмочный материал, цветокоррекция и добавление эффектов

Виды текстур

Создается целый набор текстур:

- цвет
- карта неровностей (bump)
- карта нормалей (normal map – создает видимость рельефа)
- карта рельефа (displacement – создает реальный рельеф)
- карта бликов (specular)
- карта прозрачности (alpha)
- и т д.



high poly → retopology → UV → normal map → AO → pbr textures

Зачем нужна ретопология в 3D моделировании?

Оптимизация производительности

Высокополигональные модели могут быть слишком тяжелыми для рендеринга в реальном времени, особенно в играх

Упрощение анимации

Модели с правильной топологией легче анимировать. Хорошо организованная сетка позволяет избежать проблем с деформацией при движении

Текстурирование и UV-развертка

Правильная топология облегчает процесс создания UV-разверток и текстурирования модели. Это особенно важно для создания реалистичных текстур, которые точно соответствуют форме модели

Сохранение деталей

Ретопология позволяет сохранить важные детали модели, при этом уменьшая количество полигонов

Текстурирование

Часто в **игровой индустрии моделер** ответственен и за моделирование, и текстурирование

Текстурированием в **кино** занимается отдельный специалист - **художник по текстурам**.
Перед его работой **моделер** создаёт текстурную развёртку (например, UV-развертка). Далее рисуются текстуры и привязываются к модели

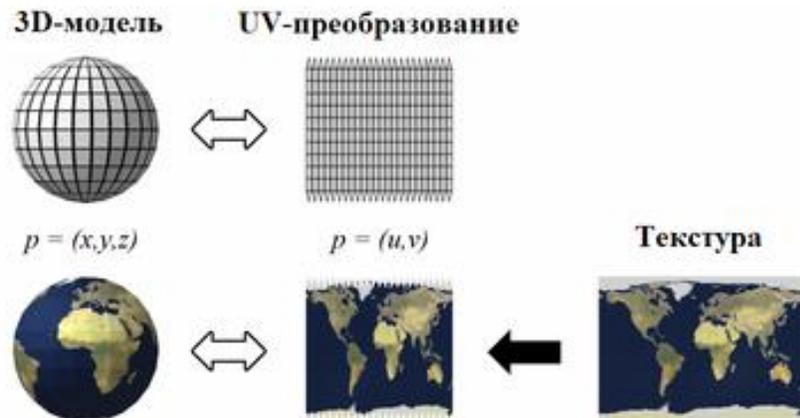
Создавать текстурные развёртки и текстуры можно в тех же программах, что и модели.
Но часто удобнее делать это в UVLayout (приложение для создания и редактирования UV-координат текстуры для 3D-поверхностей)

UV-преобразование или развёртка

UV-преобразование или развёртка в трёхмерной графике (англ. UV map) — соответствие между координатами на поверхности трёхмерного объекта (X, Y, Z) и координатами на текстуре (U, V)

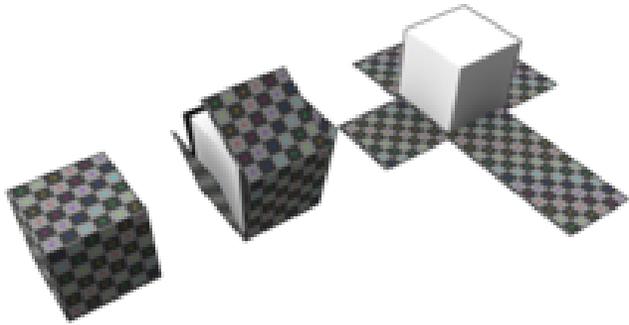
Значения U и V обычно изменяются от 0 до 1

Развёртка может строиться как вручную, так и автоматически — например, в 3Ds Studio MAX есть несколько алгоритмов автоматического развёртывания модели



Равнопромежуточная проекция — пример наложения двухмерной текстуры (координаты U, V) на трёхмерный глобус (координаты X, Y, Z)

Примеры развёрток



Пример наложения текстуры на куб.
Текстура имеет вид плоской развёртки



Шахматная текстура на сфере:
слева — общая матрица на всю сферу,
справа — развёртка с равнопромежуточной проекцией

Существует несколько противоречащих друг другу показателей качества развёртки

- Максимально полное использование площади текстуры
- Отсутствие областей с недостаточной или избыточной детализацией текстуры
- Отсутствие областей с излишними геометрическими искажениями
- Сходство со стандартными ракурсами, с которых обычно рисуется или фотографируется объект
- Удачно расположенные «швы». Они желательны, если есть естественный «разрыв» поверхности (швы одежды, кромки, сочленения и т. д.), и нежелательны, если таковых нет.
- Для частично симметричных объектов: удачное сочетание симметричных и асимметричных участков развёртки. Симметрия повышает детализацию текстуры; асимметричные детали «оживляют» объект.

PBR (Physically Based Rendering)

Метод рендеринга, основывающийся на физических свойствах материалов и их взаимодействии со светом

Основные компоненты PBR-текстур:

Albedo (или Diffuse) Основной цвет материала при нейтральном освещении

Metallic Карта показывает, является ли материал металлическим

Roughness Карта шероховатостей

Normal Map Эта карта добавляет детали на поверхность, создавая иллюзию неровностей и различных текстур

Normal Map не изменяет геометрию объекта, но позволяет добавлять мелкие детали. Например, царапины, небольшие углубления и т. д., что делает поверхность более реалистичной

PBR-текстуры используются в различных проектах, от игр до архитектурной визуализации

Основные этапы создания и визуализации 3D моделей в кино и game-индустриях

- Моделирование – создание трехмерных объектов
- Текстурирование – наложение текстур и материалов на 3D-модели
- **Риггинг** – создание виртуального скелета, набора костей/суставов для последующей анимации персонажа
- Анимация – оживление, анимирование трёхмерного персонажа
- Рендеринг – 3D визуализация созданной графики
- Композитинг – объединение отдельных элементов в финальную сцену. К примеру, интегрирование 3D сцен в съёмочный материал, цветокоррекция и добавление эффектов

Риггинг

Риггинг – создание «скелета», костей модели

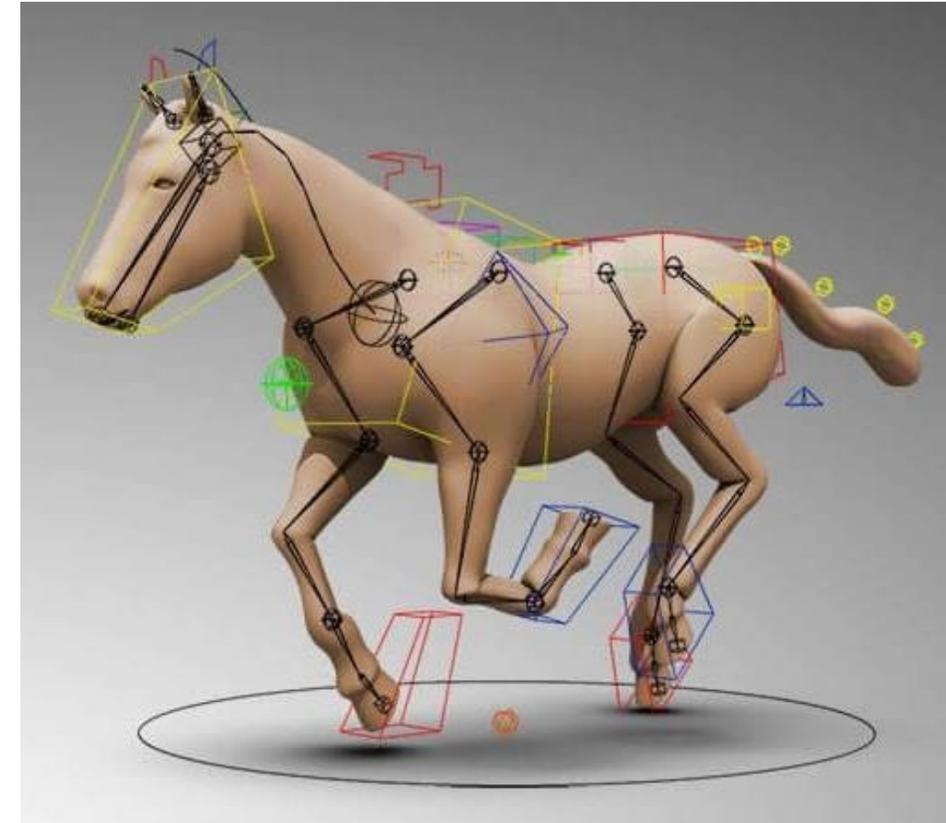
Занимаются этим в кино и game-индустрии художники по оснастке модели, **сетаперы** (Setup artist). Еще их называют **skinning, rigging artist**

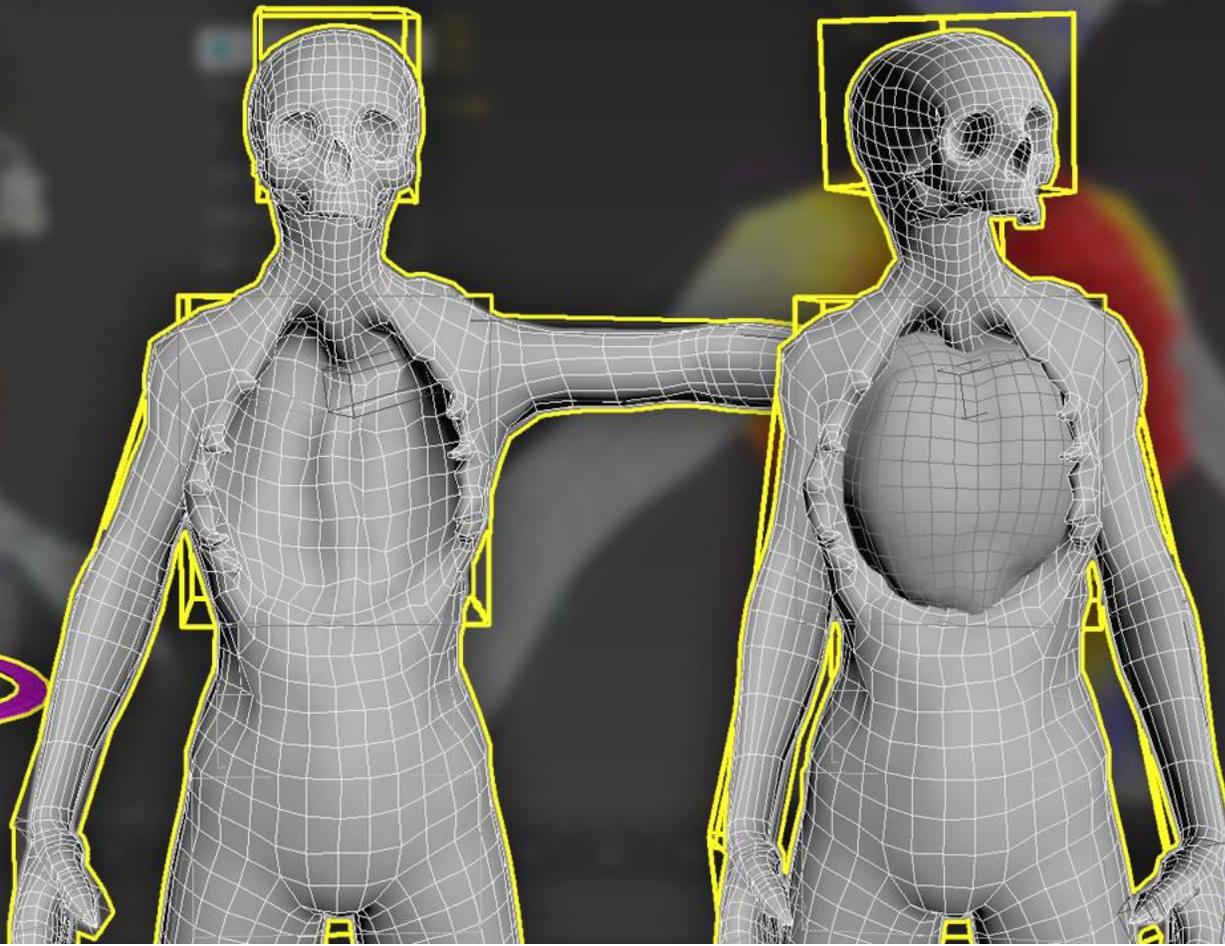
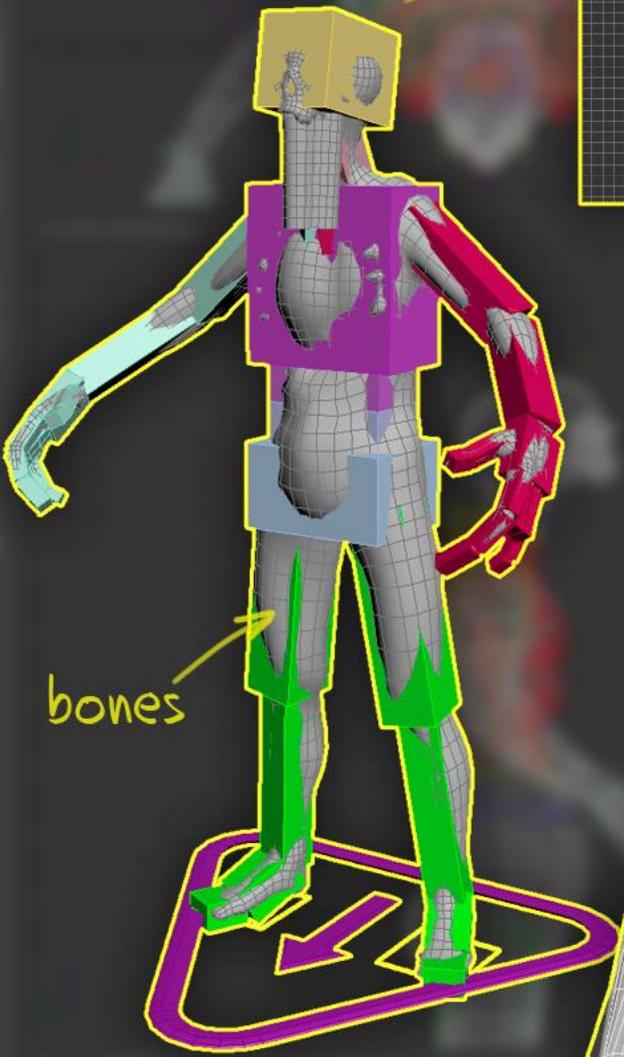
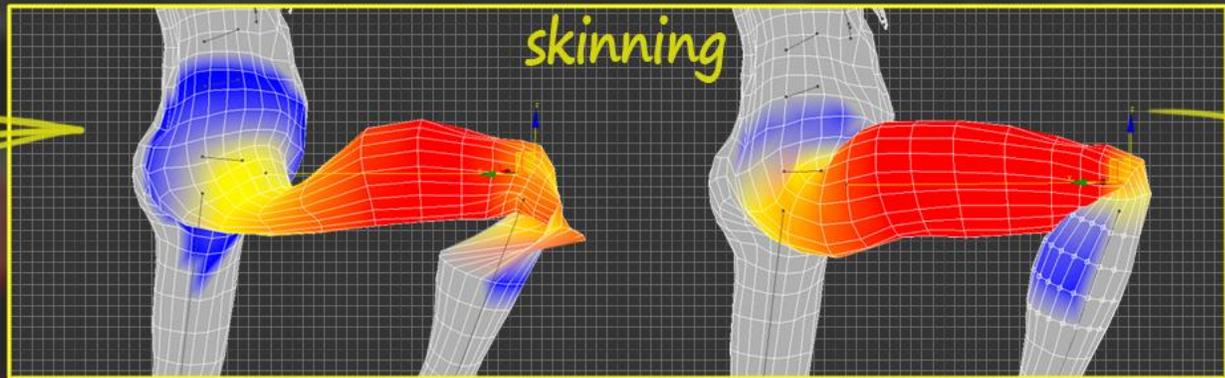
Сетаперы создают кости и контроллеры для управления этими костями, с помощью которых аниматоры могут оживить модель

В **кино** обычно создаётся множество сложных контроллеров для аниматоров. Например, для **лицевой анимации** (facial control rig) и мимики модели

В **играх** можно **обойтись** и без них, если персонаж не разговаривает в игре

Для риггинга модели подойдут те же 3D-редакторы, большинство из которых – комплексные 3D-пакеты.





Основные этапы создания и визуализации 3D моделей в кино и game-индустриях

- Моделирование – создание трехмерных объектов
- Текстурирование – наложение текстур и материалов на 3D-модели
- Риггинг – создание виртуального скелета, набора костей/суставов для последующей анимации персонажа
- **Анимация** – оживление, анимирование трёхмерного персонажа
- Рендеринг – 3D визуализация созданной графики
- Композитинг – объединение отдельных элементов в финальную сцену. К примеру, интегрирование 3D сцен в съёмочный материал, цветокоррекция и добавление эффектов

Анимация

Главная задача аниматора – сделать движения модели максимально реалистичными



Негласным лидером в создании трёхмерной анимации является Autodesk Maya
Помимо Maya отличные инструменты для анимации – 3Ds Max и Cinema 4D

Анимация

Простейшим методом анимации персонажей является **анимация по ключевым кадрам** (Keyframes)

Аниматор указывает положение персонажа в начальном и конечном кадрах движения, а положение в промежуточных кадрах вычисляется программой

Существует ещё **процедурная анимация**, при которой используется специальная программа для управления персонажем

Технология **Motion Capture** (система захвата движений)

Она подразумевает наложение движений реальных актеров на трёхмерных персонажей. Эта технология максимально упрощает анимацию, позволяя использовать уже готовые движения актеров

Основные этапы создания и визуализации 3D моделей в кино и game-индустриях

- Моделирование – создание трехмерных объектов
- Текстурирование – наложение текстур и материалов на 3D-модели
- Риггинг – создание виртуального скелета, набора костей/суставов для последующей анимации персонажа
- Анимация – оживление, анимирование трёхмерного персонажа
- **Рендеринг** – 3D визуализация созданной графики
- Композитинг – объединение отдельных элементов в финальную сцену. К примеру, интегрирование 3D сцен в съёмочный материал, цветокоррекция и добавление эффектов

Рендеринг

Существует два вида рендеринга:

- рендеринг в реальном времени
 - OpenGL
 - Vulkan
 - DirectX
- рендеринг не в реальном времени или пре-рендеринг
 - трассировка лучей (raytracing)
 - метод излучательности (radiosity)
 - трассировка путей

Стандартные программы трёхмерного моделирования включают и функцию рендеринга. Существуют и отдельные рендер-движки. Одни из самых мощных визуализаторов на сегодняшний день – Mental Ray, V-Ray, Renderman



Основные этапы создания и визуализации 3D моделей в кино и game-индустриях

- Моделирование – создание трехмерных объектов
- Текстурирование – наложение текстур и материалов на 3D-модели
- Риггинг – создание виртуального скелета, набора костей/суставов для последующей анимации персонажа
- Анимация – оживление, анимирование трёхмерного персонажа
- Рендеринг – 3D визуализация созданной графики
- **Композитинг** – объединение отдельных элементов в финальную сцену. К примеру, интегрирование 3D сцен в съёмочный материал, цветокоррекция и добавление эффектов

КОМПОЗИТИНГ

Композитинг является важным завершающим этапом постпродакшена

Композер объединяет все части в единое целое, интегрирует в съёмочный материал трёхмерных персонажей и другие 3D элементы, устраняет недочёты и убирает лишнее, работает над различными эффектами. Одним словом, создаёт одну реалистичную сцену

Композер является ответственным за финальный продукт – фильм, игру

Профессиональные программы для композитинга – Nuke, Adobe After Effects, Eyeon Fusion

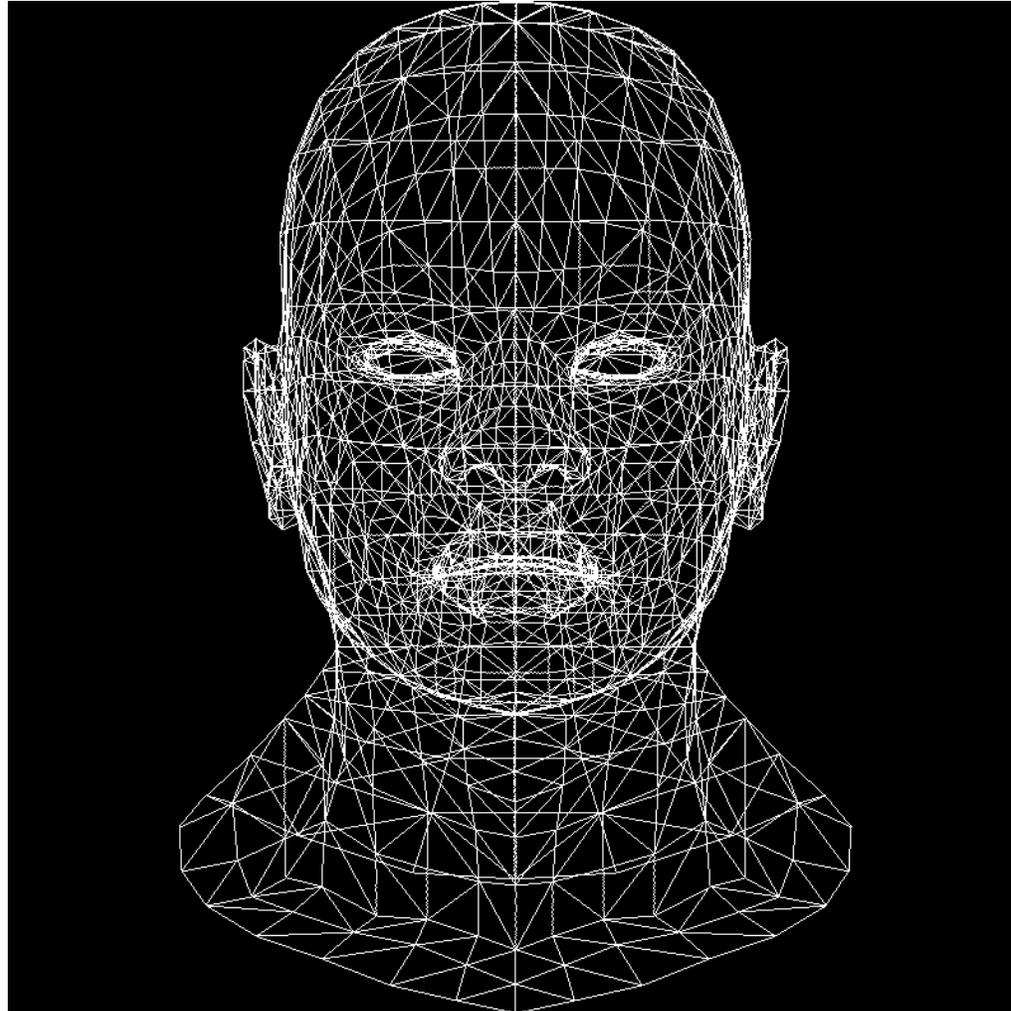
Реалистичные модели. Что бы хотелось?

Меши

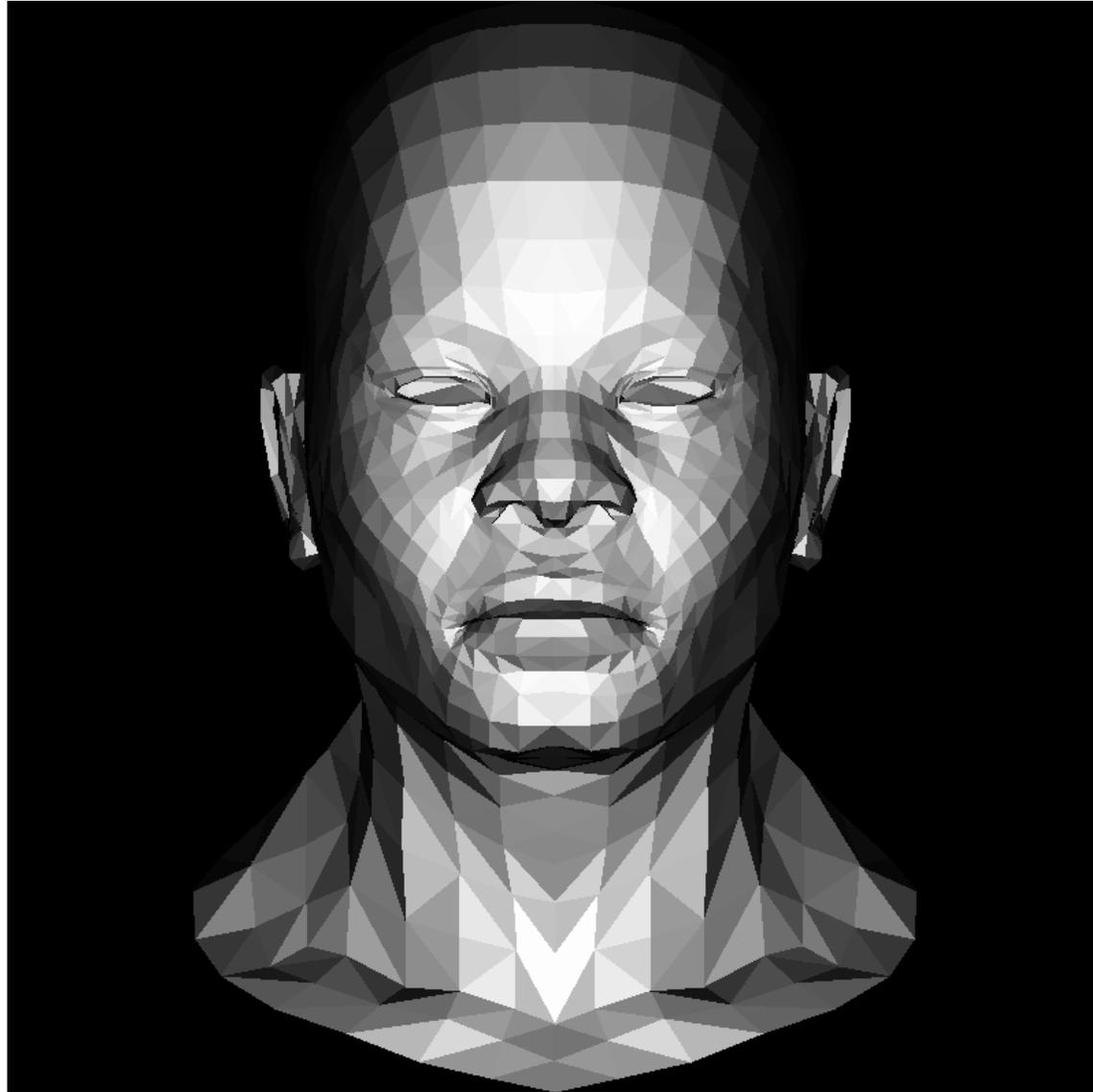


Что нужно хранить?

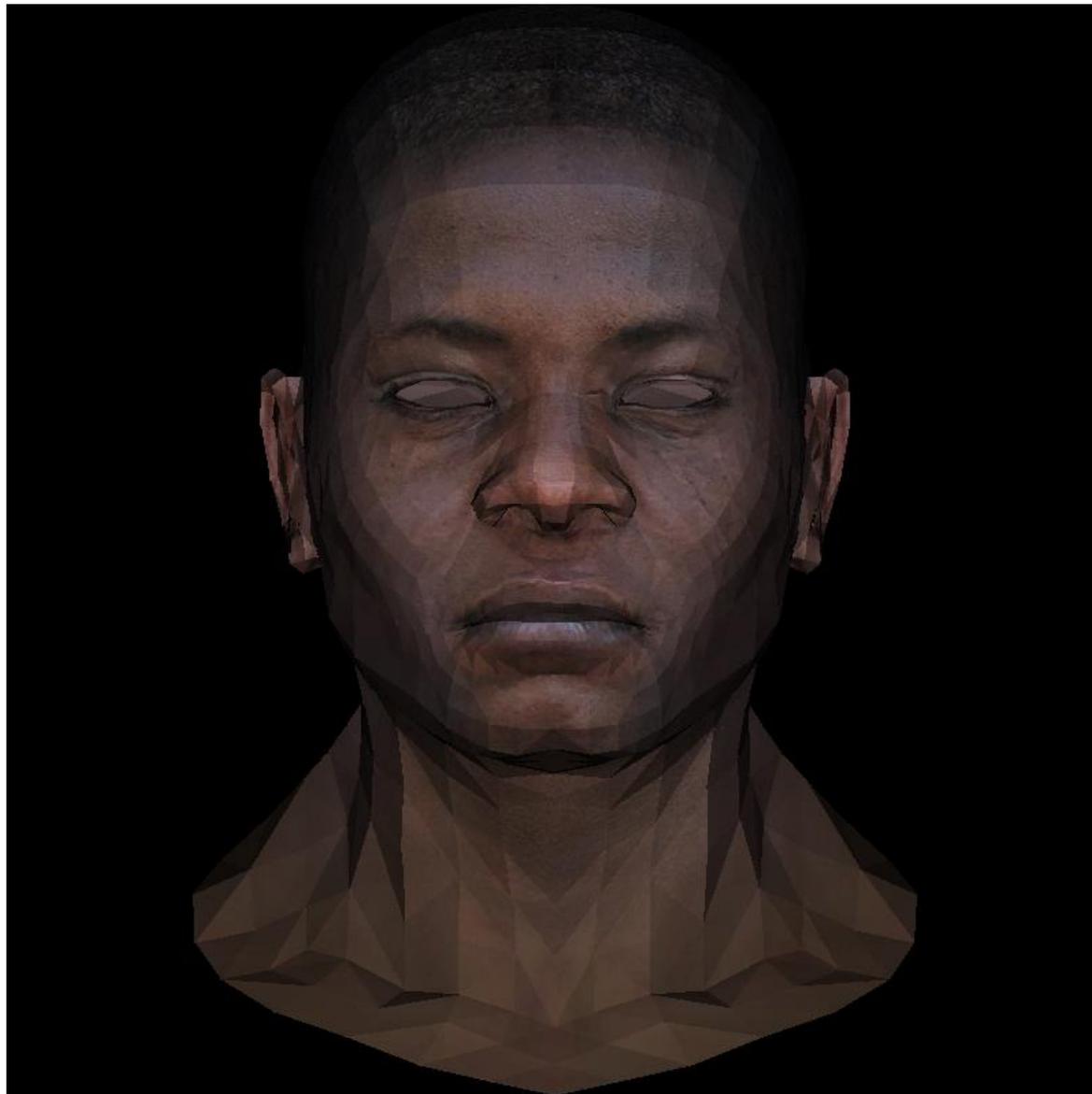
Первое приближение



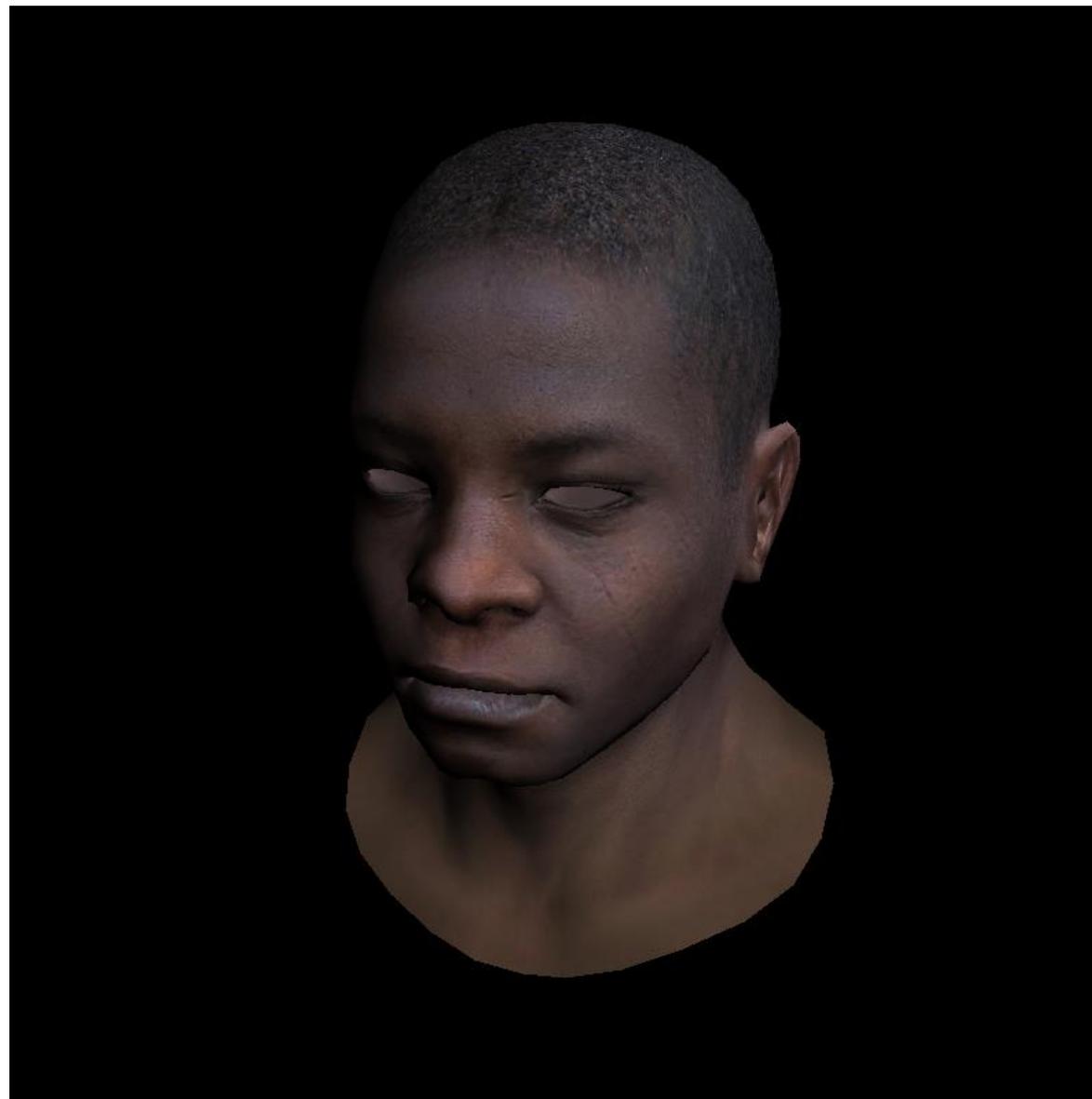
Что ещё?



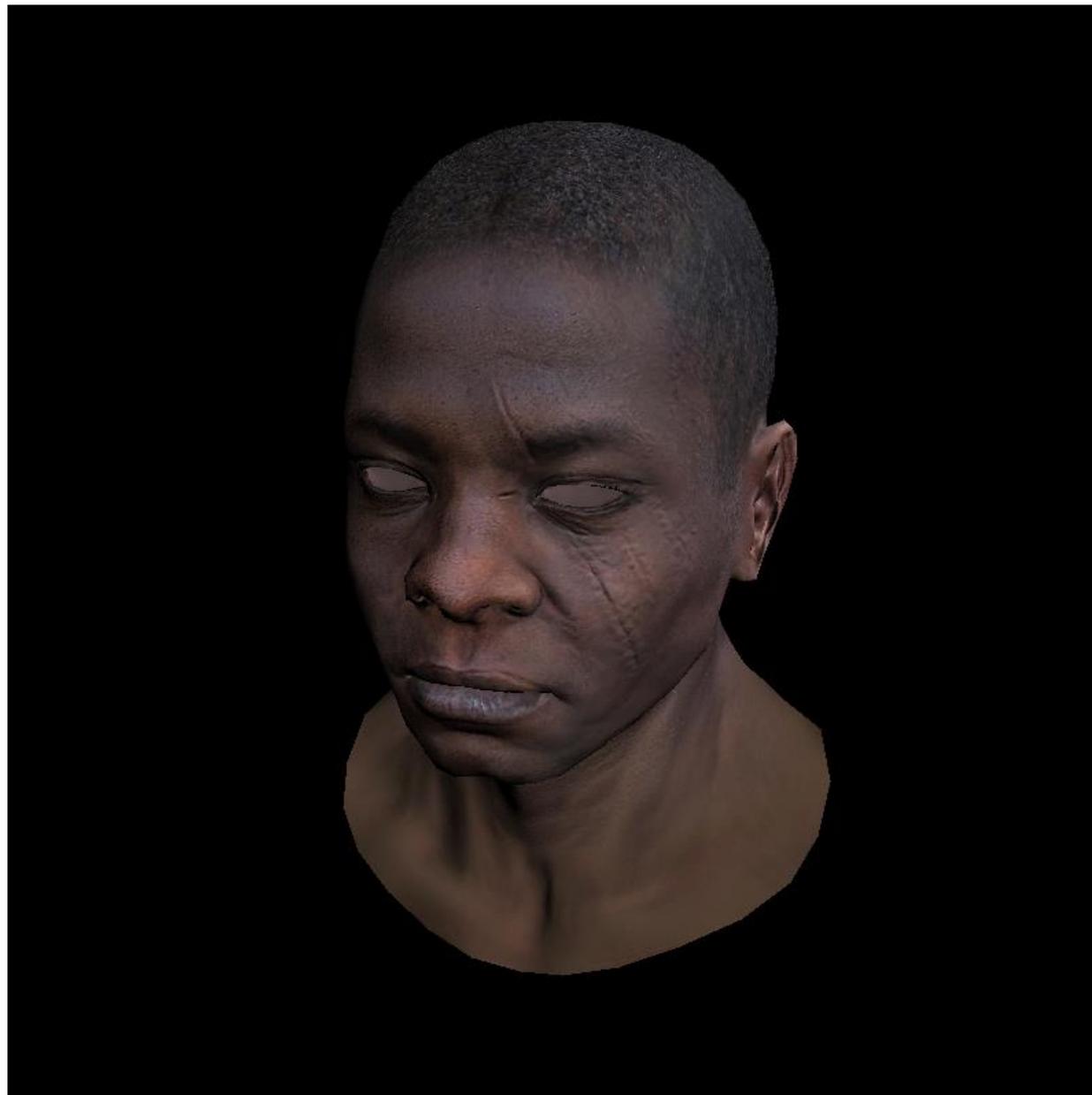
И...



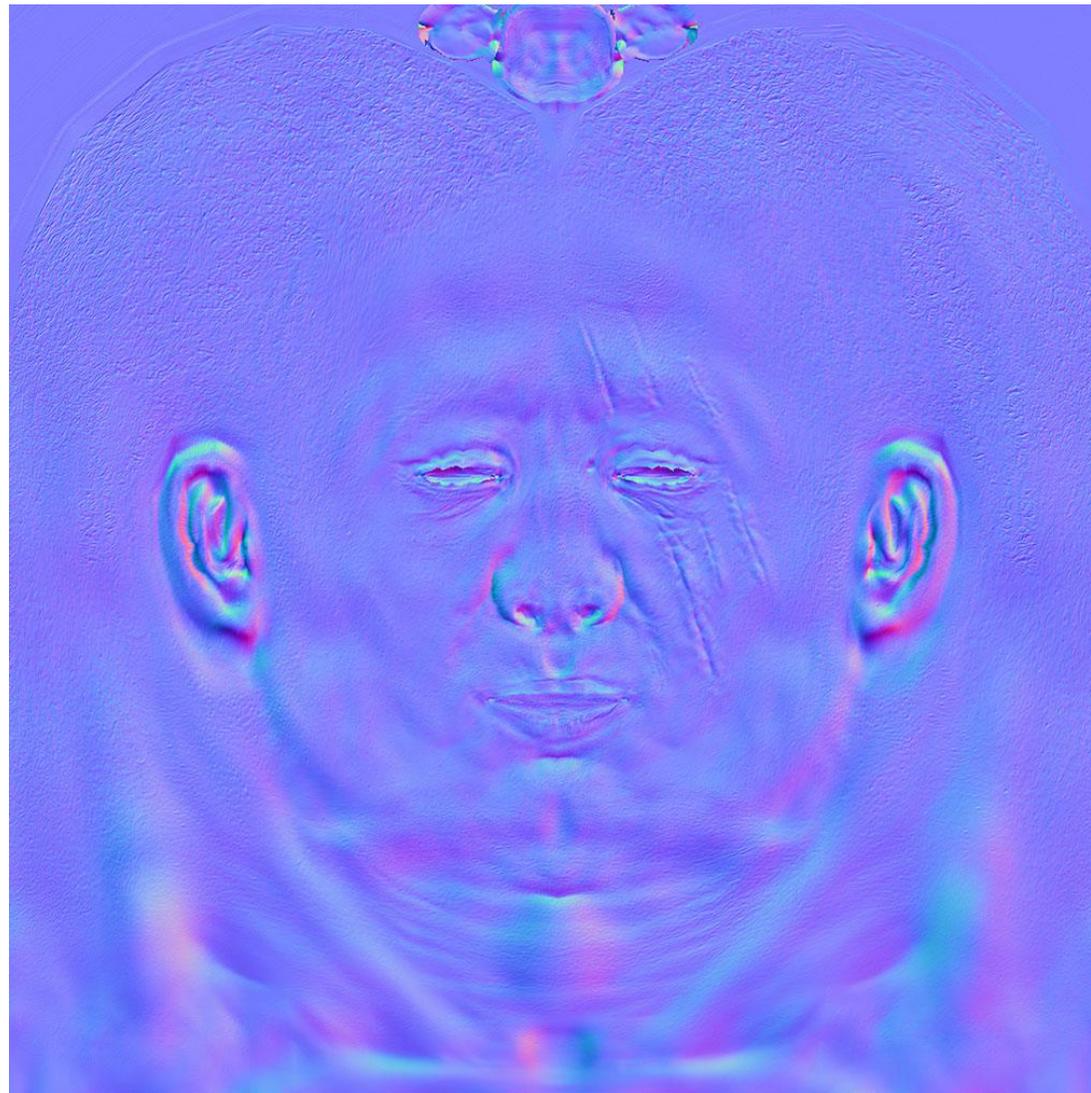
Нужны ли новые
данные?



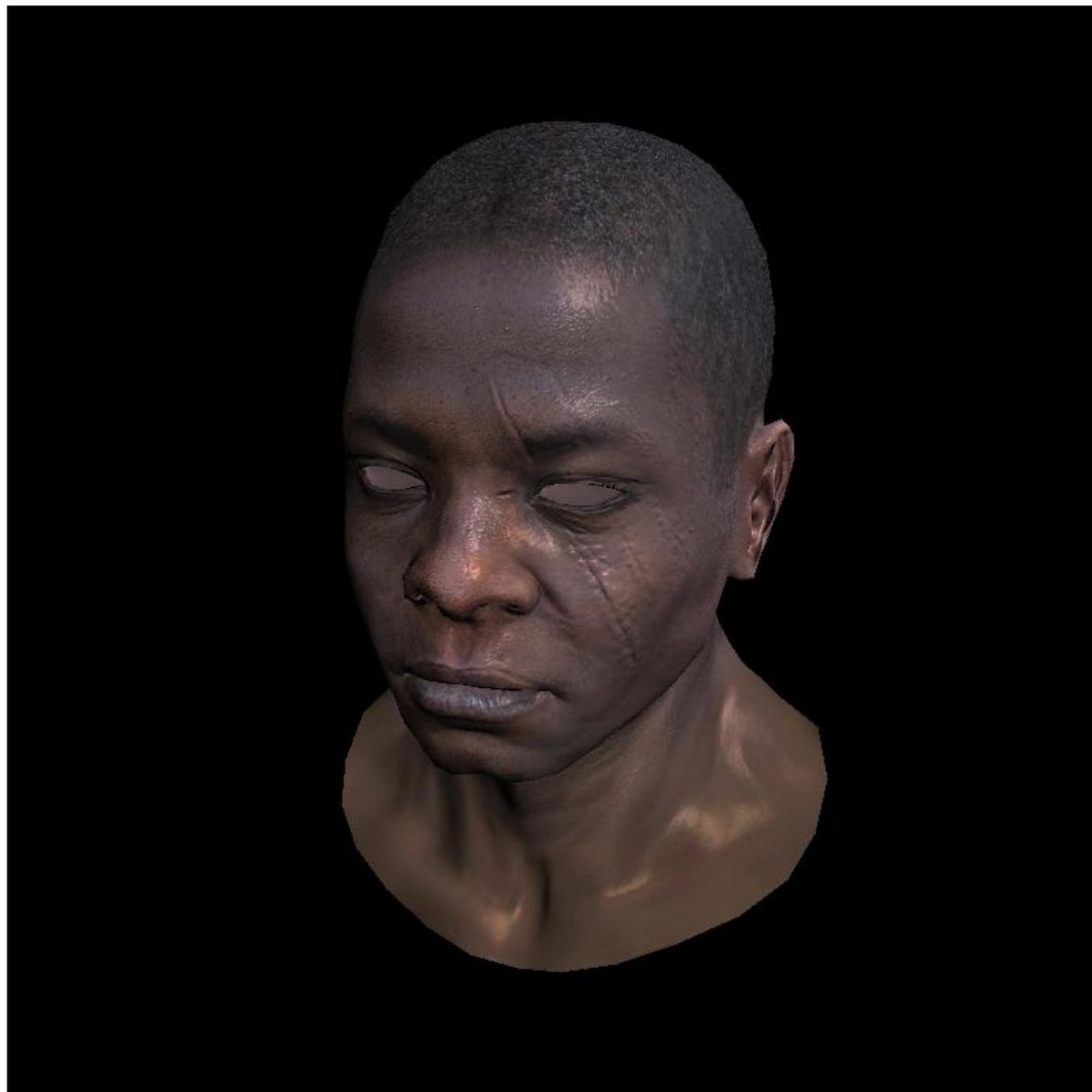
Что нужно для этих
изменений?



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

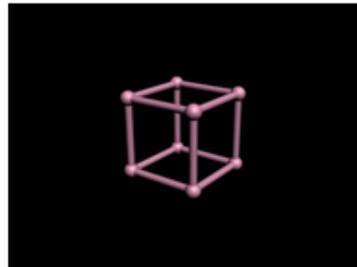
Координаты точек

```
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
```



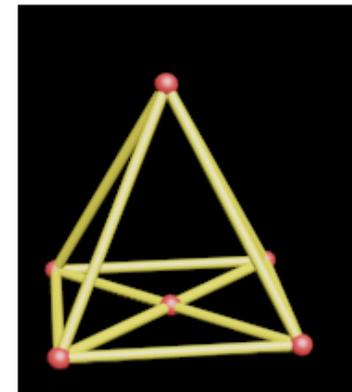
ПОЛИГОНЫ

```
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 Export
# File Created: 07.12.2013 14

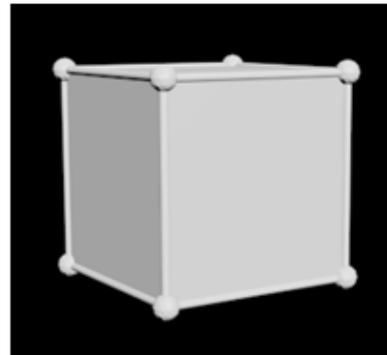
#
# object Box001
#

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

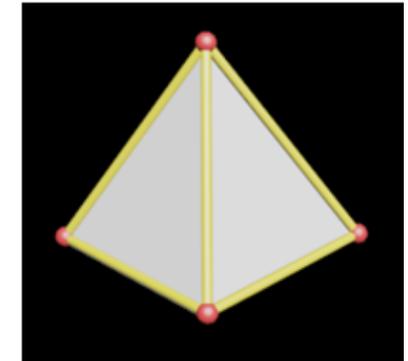
#
# object Box001
#

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 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

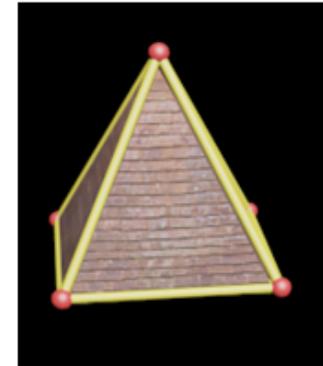
#
# object Box001
#

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

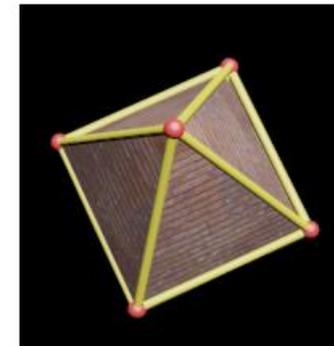
#
# object Box001
#
v -0.0005 0.0000 0.0005
v -0.0005 0.0000 -0.0005
v 0.0005 0.0000 -0.0005
v 0.0005 0.0000 0.0005
v -0.0005 0.0010 0.0005
v 0.0005 0.0010 0.0005
v 0.0005 0.0010 -0.0005
v -0.0005 0.0010 -0.0005
# 8 vertices

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



Информация о внешнем виде объектов(материалы)

- передаётся 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

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

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

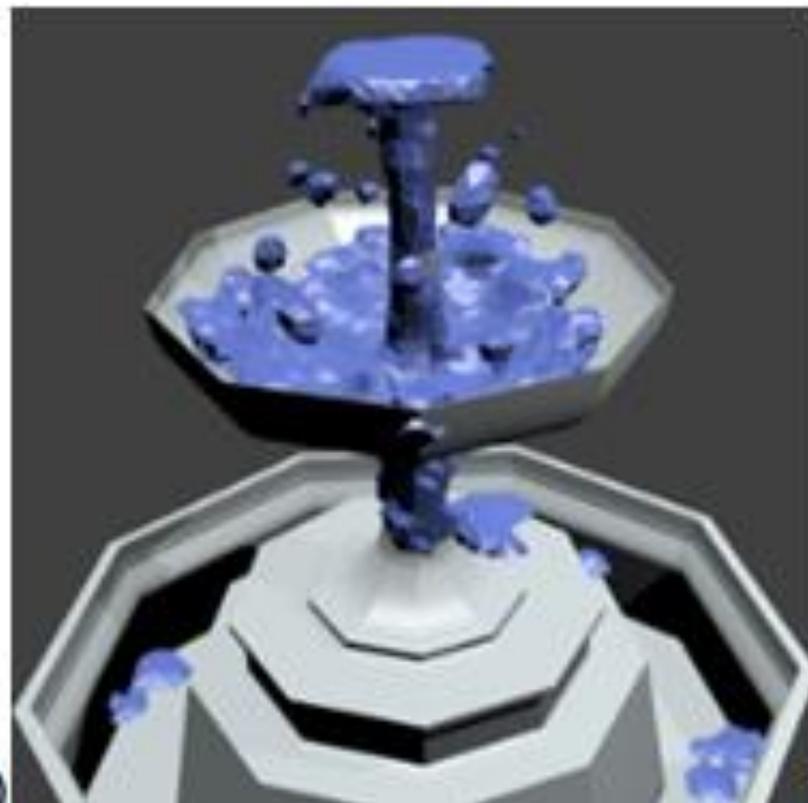
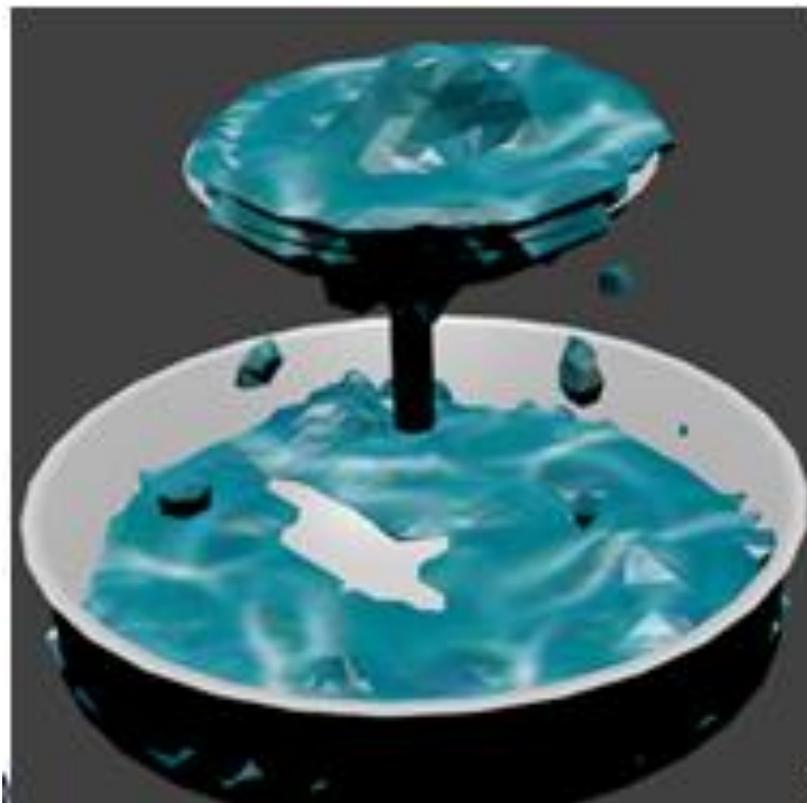
текстуры

деформацию

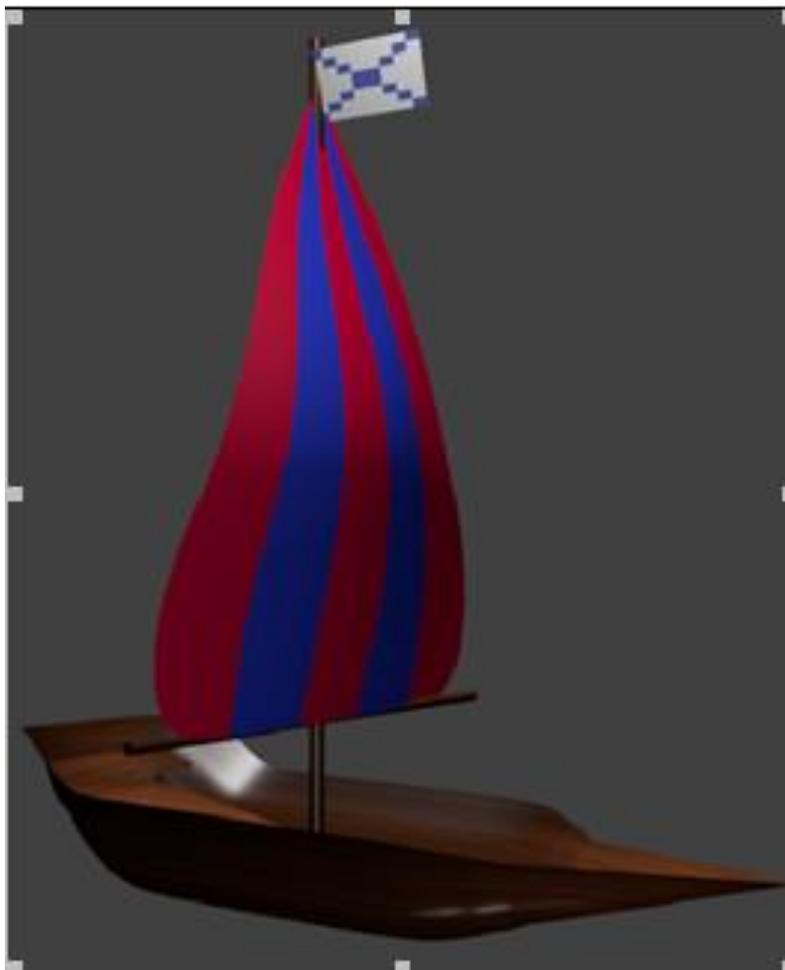
анимацию

ограничители (constraints) и ИК (цепочки инверсной кинематики)

Поддержка ограничителей



Поддержка физики



COLLADA

ОСНОВНАЯ ЧАСТЬ

- сцены
- камеры
- источники света
- модели
- анимации

ДОПОЛНИТЕЛЬНАЯ ЧАСТЬ

- 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.

Структура Collada *.dae

```
<?xml version="1.0" encoding="utf-8"?>
<COLLADA version="1.5.0" xmlns="http://www.collada.org/2008/03/COLLADASchema">
  <asset>
    <contributor>
      <author>tarakos GmbH</author>
      <author_email>ender.yemenicioglu@tarakos.com</author_email>
      <author_website>www.tarakos.com</author_website>
      <authoring_tool>tarakos VrmI To Collada Converter Version 1.1</authoring_tool>
      <copyright>tarakos GmbH</copyright>
    </contributor>
    <created>2015-10-08T13:26:43Z</created>
    <modified>2015-10-08T13:26:44Z</modified>
    <unit meter="1" name="meter" />
    <up_axis>Z_UP</up_axis>
  </asset>
  <library_geometries id="6eefceaa-ab1f-48a6-a540-130e478703fc" name="libGeo_World">
  <library_materials id="2ec95307-3a70-4111-b97d-7ef54c0c4110" name="libMaterial_World">
  <library_effects id="378ebafe-73d5-4708-b05b-f449ba598173" name="libEffect_World">
  <library_visual_scenes id="c3116fe0-8780-444a-a9e0-f07d459e69c3" name="libVisualScenes_World">
  <scene>
</COLLADA>
```

Геометрия

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

```
    <float_array id="pCylinderShapel-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="#pCylinderShapel-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>
```

Загрузка моделей

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 постобработки (например, генерация касательного пространства, различные оптимизации) расширяют список возможностей

«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

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

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

Three.js - Загрузка 3D-моделей

Загрузчики в репозитории Three.js в каталоге `three/examples/jsm/loaders`

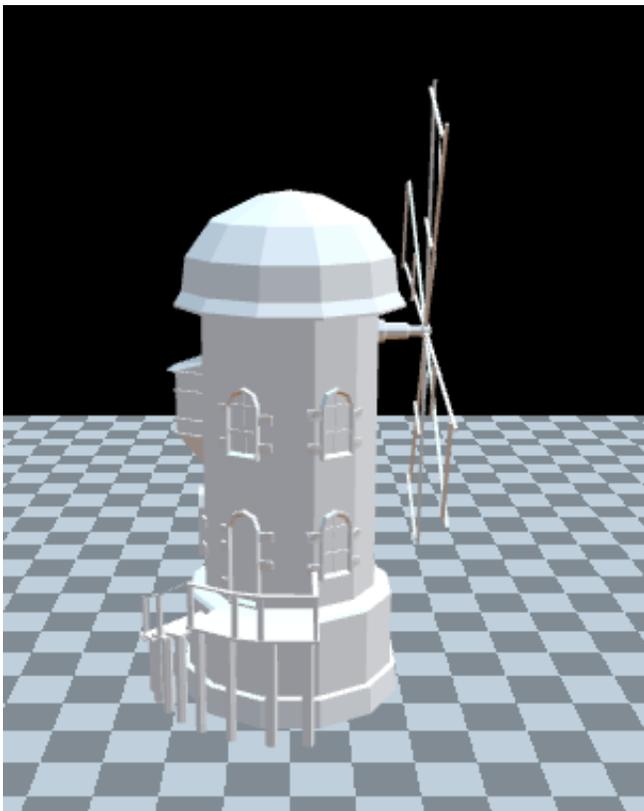
Первое, что нужно сделать, это включить OBJLoader загрузчик в скрипт

```
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 заселённость
- Опасность и спокойствие
- Природа
- Киберпанк
- Стимпанк
- Локации в играх жанра hidden object

Пустота и заселённость

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 незаселённые локации, например, в survival-играх, хоррорах, постапокалиптических шутерах, других подобных жанрах

Пустые локации создают атмосферу тревожности, хаоса, беспорядка, которая передаётся геймеру

Если в населённых 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 напряжёнными 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

Ещё 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

