Оглавление

[1. Инструментарий 2](#_heading=h.30j0zll)

[2. Входные данные 2](#_heading=h.1fob9te)

[3. Требования к сцене 2](#_heading=h.3znysh7)

[4. Требования к геометрии 2](#_heading=h.2et92p0)

[4.1 Требования к геометрии персонажа 2](#_heading=h.tyjcwt)

[5. Требование к UV - развертке 3](#_heading=h.3dy6vkm)

[6. Требования к текстурам и материалам 3](#_heading=h.1t3h5sf)

[7. Требования к скелету 3](#_heading=h.4d34og8)

[8. Требования к анимации 4](#_heading=h.2s8eyo1)

[8.1 Требования к анимации персонажа 4](#_heading=h.17dp8vu)

[9. Требования к эффектам 4](#_heading=h.3rdcrjn)

# Инструментарий

* **Autodesk Maya 2011,** **32bit**.
* **Настройки:**
* Ось **Z** направлена вверх
* **Пространственные единицы: сантиметр**. 1 см в Maya соответствует 1 метр в игре
* **Углы** измеряются в **градусах**
* Количество **кадров** в секунду: **30**

# Входные данные

* **Все файлы** для ассета (геометрия, текстуры, концепты) должны находиться **в одной папке**
* В **наименовании файлов и объектов** (мешей, костей, контролов и т.д.) используйте только заглавные и строчные английскиебуквы, цифры и символ «\_» (подчеркивание). **Имена запрещено начинать с номера**
* Перед выполнением работы ознакомьтесь со структурой и иерархией на примере аналогично выполненного задания во избежание использования не приемлемых редактором функций, атрибутов и инструментов

# Требования к сцене

* Все статичные объекты должны сохраняться в формате **Maya Binary «\_.mb»**
* **Меш в ассете должен быть один** и с называнием **«\_»**, но могут быть и оговоренные исключения, например, в эффектах единый меш не обязателен.
* **История должна быть удалена** на всех объектах сцены (за исключением прискиненных мешей)
* Все **неиспользуемые связи должны быть удалены** из сцены. Майская сцена не должна содержать лишние объекты: модели героев, зданий, неиспользуемые текстуры и материалы и т.д. Все вспомогательные объекты должны быть референсом в сцене (File → Create Reference)
* Выполните ***Optimize Scene Size***, чтобы дополнительно очистить сцену

# Требования к геометрии

* Все **нормали** **граней** должны быть направлены к камере
* **Вершинные нормали** не должны быть заблокированы
* Сглаживание ребер (***softness/hardness***) настраивается для лучшего вида объекта
* **Двухсторонние полигоны** должны быть замоделированы сразу на две стороны. Точки между ними сшивать нельзя, чтобы не делать «плохой» геометрии
* Все **невидимые грани** должны быть удалены
* В геометрии **не должно быть** T-соединений, двойных полигонов, несшитых вершин и т.д.

## Требования к геометрии персонажа

* Максимальное **количество полигонов** на сложную модель: **2500**
* **Персонаж** должен быть **смоделирован в Т-позе**, удобной для дальнейшего скининга и анимации
* Не забывайте, что **сгибаемые части тела** (плечи, локти) должны сохранить пропорции после изгиба
* **Все части объекта должны быть собраны в единый меш**. Но не забывайте, что некоторые куски геометрии должны быть смоделированы отдельными частями, чтобы их можно было правильно анимировать
* **Необходимо сшивать вершины близкие в пространстве** (в случае пересечения частей геометрии), это улучшает рендер и освещение объекта, а также сохраняет пространство на текстуре
* Не забывайте, что **грани**, которые формируют силуэт объекта, особенно важны, потому что они всегда на виду
* Всегда продумывайте, что необходимо сделать геометрией, а что может быть передано **текстурами**
* **Меш** должен состоять из четких форм и линий

# Требование к UV - развертке

* Все **замкнутые области в UV-пространстве** **(shells)** должны быть скомпонованы с минимальными потерями пространства
* **Зеркальное отражение** можно использовать, если оно не слишком очевидно
* **UV- координаты** должны быть с минимальным количеством швов и искажений
* **Граничные области** должны быть выровнены по горизонтали и вертикали, чтобы избежать проблем с бейканными картами нормалей
* **Очевидные швы не допустимы**
* **Плотность** 1:1 тексель не требуется, если об этом не оговорено в задании
* При создании **UV - развертки** наиболее видимым частям должно отводиться больше места на текстуре

# Требования к текстурам и материалам

* **Текстуры** должны сохраняться в формате **TARGA «\_.tga».** **32bit** для текстуры с альфа-каналом и **24bit** без альфа-канала. (После финализации ассета исполнитель должен предоставить исходники для текстур в формате **Photoshop «\_.psd»** со слоями)
  + диффузная текстура ***(diffuse map)*** **«\_.tga»**
  + текстура бликов ***(secular map)*** **«\_SPC.tga»**
  + текстура нормалей ***(normal map)*** **«\_NRM.tga»**
* Текстура не обязательно, но желательно, должна быть квадратной. Главное, чтобы ее размеры в пикселях должны быть степенью двойки
* **Размеры текстур:**
  + **16х16, 32х32** – для мелких частиц (искорок, звездочек)
  + **64х64, 128х128** – для больших частиц с мелкими деталями (узорами, рунами)
  + **512х512, 1024х1024** – для текстур персонажей, башен, деревьев, домиков
  + **2048х2048** – для крупных объектов (большие куски локаций: горы, замки)
* **Альфа-канал** сохраняется как четвертый канал в карте цветов. Его не надо сохранять как отдельную текстуру
* **Карты нормалей** выполняются в *tangent space*
* **Материалы** используются стандарта **Lambert** и **Phong**
* **Не используйте** многослойные материалы или какие-либо ноды для процедурного текстурирования

# Требования к скелету

* Максимальное **количество костей** в деформирующем скелете: **64**
* **Число костей**, одновременно **влияющих** **на** каждую **вершину**: **4**
* Все **имена** костей должны быть уникальными
* **Первая кость** в иерархии должна называться **«root»**
* **Иерархия костей** не должна содержать пустые группы, только кости и констрейны
* Все **родители кости**, связанной со *skin cluster*, должны быть так же связанны со *skin cluster*. Это автоматически означает, что только *end-joints* могут не иметь связи со *skin cluster*
* Все **управляющие ноды** желательно помещать в отдельную группу
* **Управляющие контролы** анимационного рига должны быть выполнены в виде ***curves*** очевидной формы. Использование локаторов крайне нежелательно
* **Анимационный риг** должен содержать единый контрол, позволяющий перемещать, вращать и масштабировать персонажа целиком
* **Примером** хорошего анимационного рига может служить [AdvancedSkeleton](http://www.highend3d.com/maya/downloads/mel_scripts/animation/AdvancedSkeleton-194.html). Рекомендуется его непосредственное использование в игровых ассетах

# Требования к анимации

* Перед созданием любой анимации нужно убедится, что **объект правильно отмасштабирован**, и соответствует своему размеру
* **Геометрия** может быть заанимированна двумя способами:
  + В случае **простой анимации** геометрия импортится в редактор как статик меш и анимируется с помощью параметров трансформ ноды
  + В случае **сложной анимации** геометрия анимируются с помощью костей и/или приконстрейненных контролов
* В случае множественных вариаций, **анимационные клипы** должны сохраняться в формате **Maya ASCII format «\_.ma»**,в противном случае при импорте ассета в редактор создается один **«\_AutoClip.ma»**
* **Не допускается** использование в анимации деформеров (deformers) и прочие сложные элементы управления анимацией по мимо описанных

## Требования к анимации персонажа

* В сцене с персонажем должен быть ***Character set***, который содержит в себе каналы управляющих объектов (контролов). *Character Set* и клипы создаются для управляющего скелета
* **Каждой заказанной анимации** должен соответствовать один клип для Character Set в формате Maya Ascii **«\*.ma»**. Имя клипа совпадает с именем заказанной анимации. Названия и длительность клипов оговариваются предварительно
* **Move** – перемещение. Финальный вариант анимации должен представлять один зацикленный шаг персонажа «на месте». При создании анимации надо учитывать скорость перемещения данного персонажа (задается в ТЗ). Для соответствия анимации скорости перемещения, нужно чтобы при перемещении персонажа, его ноги не проскальзывали относительно земли. Аниматор может выбрать **один из двух способов производства**:
  + **Сделать линейную анимацию перемещения** главного контрола с обозначенной скоростью, и делать анимацию уже на перемещающемся персонаже таким образом, чтобы ноги не проскальзывали относительно земли.
  + **Сделать «честную» анимацию движения** путем перемещения всех контролов , проследив чтобы весь персонаж двигался с обозначенной скоростью, а затем сделать обратную компенсирующую анимацию на главном контроле.
* **attack** - атака, анимация с очень жесткими техническими ограничениями:
  + **Длительность анимации** attack **45 кадров (1,5 сек.)**
  + сам **момент удара** по противнику должен приходиться на **4ый кадр**
  + **Анимация зацикленная**, начальная и конечная позы должны соответствовать анимации **battleIdle**
  + **BattleIdle** – поза готовности к атаке. Короткая зацикленная анимация, которая проигрывается между ударами, и нужна для того, чтобы персонаж в любой момент мог «легко и быстро» нанести удар. Эта анимация максимально близко приближенна к начальной позе нанесения удара.
* **Idle** – безделье. Проигрывается, когда у героя нет задач или команд. Зацикленная анимация.
* **Death** – смерть. Максимальная длина этой анимации 3сек = 90frames.

# Требования к эффектам

* **Структура, расположение и наименование эффектов:**
* **Текстуры** **частиц** лежат в папке PFX\_Textures
* **Текстуры геометрии** лежат в папке GFX\_Textures
* **Имена текстур** должны быть коротки и понятны. Основные группы наименования текстур: Glow, Noise, Rays, Fire, Smoke, Spark, Sparks, Wave, Flash
* **Уникальные эффекты** для зданий, героев и крипов хранятся в папке с соответствующим ассетом. Имя файла эффекта соответствует имени анимации с приставкой FX. Если в названии эффекта название анимации не указано – эффект играется на объекте всегда. Например: **«FX.mb»** – эффект отображается постоянно с ассетом;  
  **«FXattack.mb»**, **«FXdeath.mb»** – эффекты проигрываются во время анимации **cast** и **death**. Клипы к эффектам называются аналогично названию главного файла эффекта: **«FX.ma»**, **«FXattack.ma»**, **«FXdeath.ma»**
* **Остальные эффекты** (сюда же относятся неуникальные эффекты для крипов и героев типа селекшена и аур) хранятся в папке **Effects**. В папке Effects есть папки – категории эффектов (например, Environment, Spells и т.д.). Каждый эффект лежит в своей папке, которая называется по имени эффекта.
* Используются **только спрайтовые частицы** ***(particleRenderType: sprites)***. Любые поля **(Fields)** и типы эммитеров ***(Emmiter)***
* **Не используются** элементы ***(nDynamics)*** и прочие сложные элементы генерации частиц по мимо описанных
* **Множество одинаковых частиц и эммитеров** объединяются в сет
* **Ориентация направленности отображения спрайта** частиц меняться в атрибутах **Scale**
* По мимо стандартных **каналов эммитеров, частиц, кривых и трансформ нод для анимации** используется устоявшийся набор атрибутов ***(ParticleShape: Per Particle (Array) Attributes)*** посредством градиентов ***(Ramp)*** или выражений ***(Expression)*** 
  + **Sprite Twist PP** – вращение
  + **Sprite Scale Y PP** – размер частиц по оси Y
  + **Sprite Scale X PP** – размер частиц по оси X
  + **Sprite Num PP** – с помощью этого параметра на систему частиц можно назначать материал с одной или с несколькими текстурами, порядок которых будет изменяться по закону, описанному в майке
  + **Opacity PP** – прозрачность
  + **RGB PP** – цвет
* [**Декали**](https://confluence.nivalnetwork.com/display/dot/Decals+Setup) ***(Decal)*** создается через **mel:**

global proc natDecal()

{

string $plane = one(`polyPlane -name "decal\_01" -w 1 -h 1 -sx 1 -sy 1 -ax 0 0 1`);

addAttr -ln "decal" -at "float" -dv 12 $plane;

setAttr -e -channelBox 1 ($plane + ".decal");

addAttr -ln "radius" -at "float" -dv 0.5 $plane;

setAttr -l 1 -k 0 ($plane + ".tz");

setAttr -l 1 -k 0 ($plane + ".rx");

setAttr -l 1 -k 0 ($plane + ".ry");

setAttr -l 1 -k 0 ($plane + ".sz");

setAttr -l 1 ($plane + ".rpx");

setAttr -l 1 ($plane + ".rpy");

setAttr -l 1 ($plane + ".rpz");

setAttr -l 1 ($plane + ".spx");

setAttr -l 1 ($plane + ".spy");

setAttr -l 1 ($plane + ".spz");

}

global proc string one(string $array[])

{

if (`size $array` > 0)

return $array[0];

else

return "";

}

natDecal();

* [**Blending modes**](http://www.simplefilter.de/en/basics/mixmods.html)– использование наложение цветов и обработки альфа каналов происходит рендером непосредственно в редакторе (игровом движке) в настройках **материала** объекта. После импорта эффекта или объекта в редактор они настраиваются набором параметров **Render state:**
  + **BlendMode** - наложение цветов и учет прозрачности
    - **off** - альфа канал не учитывается
    - **LerpByAlfa** - Линейная интерполяция по альфе (обычное наложение с учетом альфа канала как прозрачности)
    - **PremultiplietedLerp** - Смешенная линейная интерполяция через свечение альфа канала. (обычное наложение с учетом альфа канала как свечения) Только для объектов (геометрии), где не используется альфа канал как прозрачность. Пример: светящийся пузырек, кристалл
    - **AddColor** - Сложение цетов. Только для объектов, где не используется альфа канал как прозрачность
    - **AddColorMultyAlfa** - Сложение цветов с учетом альфа канала как прозрачности. Пример: Светящиеся искры, блики
    - **MulColor** - Умножение цветов без учета альфа канала как прозрачности
    - **MulInvColor** - Умножение на инвертированный цвет без учета альфа канала как прозрачности
  + **AlfaTest** – включает/выключает управление параметром AlfaTestRef
  + **AlfaTestRef** - (0-255) Учитывает границу пикселя альфа канала с которого начинает делать прозрачным изображение. При выключенном параметре = 0
  + **Culling** - включает/выключает двустороннее отображение нормалей
* **Искажение пространства (DistortionPinValue)**, такие как вода, жар от огня и прочее настраиваются посредством редактора в **материале** атрибутами:
  + **DistortionPinValue** – включает/выключает рендер эффекта в режим искажения
  + **Opacity** – прозрачность
  + **Distortion** – искажение
* **Трейcы** ***(Trace)*** – следы за оружием, и **связующие молнии, цепи** делаются в редакторе без использования Maya