

Виртуальная студия
Фокус

Создание 3D-сцен

Руководство пользователя

Версия от 15 мая 2009 г.

VS 1.63

Copyright © SoftLab-NSK Ltd.

Содержание

1	ВВЕДЕНИЕ	3
2	СОЗДАНИЕ 3D-СЦЕН ДЛЯ <i>HOTACTIONS</i> В <i>3D STUDIO MAX</i>.....	4
2.1	Ограничения на экспортируемые параметры.....	4
2.1.1	Экспорт общего трека сцены	4
2.1.2	Геометрия	4
2.1.3	Камеры	4
2.1.4	Источники света	4
2.1.5	Иерархия	5
2.1.6	Параметры среды Environment.....	5
2.1.7	Материалы	5
2.1.8	Текстуры.....	6
2.1.9	Текстурные координаты	7
2.1.10	Размеры текстур	7
2.1.11	Вспомогательные объекты.....	7
2.1.12	Анимация	8
3	РЕАЛИЗАЦИЯ МОРФИНГА В ВИРТУАЛЬНОЙ СТУДИИ	9
4	ИМИТАЦИЯ СВЕТОВЫХ ЭФФЕКТОВ И НЕРОВНОСТЕЙ ПОВЕРХНОСТЕЙ	10
5	ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ МАТЕРИАЛОВ С ОТРАЖЕНИЕМ	11
6	СОКРАЩЕНИЕ КОЛИЧЕСТВА МНОГОУГОЛЬНИКОВ В 3D-СЦЕНЕ	12
7	ЖИВОЕ ВИДЕО	13
7.1	Создание текстуры для видеоматериала	13
8	ИМЕНОВАНИЕ ОБЪЕКТОВ В 3D-СЦЕНАХ И ЗАРЕЗЕРВИРОВАННЫЕ ПРЕФИКСЫ ИМЁН.....	15
9	ПРИМЕР ЭКСПОРТА И ИСПОЛЬЗОВАНИЯ В <i>HOTACTIONS</i> ПРОСТЕЙШЕЙ СЦЕНЫ.....	16
9.1	Составляющие минимальной сцены виртуальной студии.....	16
9.2	Экспорт сцены и создание нового проекта в <i>HotActions</i>	17
9.3	Работа со сценой с помощью <i>Action'ов</i>	19
9.3.1	Создание <i>Action'а</i> , инициализирующего сцену	20
9.3.2	Использование именованных треков	22
9.3.3	Работа с виртуальными камерами	24
9.3.4	Работа с видеофайлами	24
9.3.5	Работа с графическими файлами.....	25
9.3.6	Использование одних <i>Action'ов</i> в других.....	25
9.3.7	Создание кнопок для выполнения определённых <i>Action'ов</i>	26
10	ЭКСПОРТ СЦЕН ИЗ <i>3D STUDIO MAX</i> В ФАЙЛ *.3D.....	28
10.1	Установка Plug-in'a для <i>3D Studio MAX (MAYA)</i>	28
10.2	Стандартная утилита экспорта	29
11	ИСПОЛЬЗОВАНИЕ <i>VS EXPORTER</i>	30
11.1	Загрузка <i>VS Exporter</i>	30
11.2	Просмотр сцены с помощью <i>VS Exporter</i>	31
11.3	Сохранение сцены в формате <i>HotActions</i> с помощью <i>VS Exporter</i>	32

1 Введение

Излагаемый материал предполагает знакомство читателя с приложением *3D Studio Max* (версии 4.2 и выше, если не оговорено иное) и умение с его помощью создавать объекты 3D-сцены, виртуальные камеры, задавать их анимацию и т.п. При работе в *3D Studio Max* необходимо учитывать рекомендации и ограничения, рассматриваемые в главе 2.

Для использования в проекте виртуальной студии сцена, приготовленная в *3D Studio Max*, должна быть экспортирована в файл с расширением **3d**. Использование компонента *VS Exporter* (см. главу 11) позволяет предварительно просматривать сцену в 3D-формате непосредственно из *3D Studio MAX*. Настоятельно рекомендуется пользоваться указанной возможностью, так как при этом используются механизмы рендеринга виртуальной студии, которые могут давать результаты, отличающиеся от рендеринга *3D Studio MAX*. Различия могут заметно усилиться при несоблюдении ограничений экспорта, перечисленных в разделах 2.1.4, 2.1.8.

2 Создание 3D-сцен для *HotActions* в *3D Studio MAX*


2.1 Ограничения на экспортируемые параметры

Экспорт и последующее использование в виртуальной студии созданной в *3D Studio Max* сцены предполагает выполнение некоторых требований к 3D-сцене. Эти ограничения похожи на те, с которыми сталкиваются разработчики компьютерных игр. В частности, в виртуальной студии используются возможности графического 3D-ускорителя. Кроме того, необходимо обеспечить успешное функционирование в режиме реального времени, когда все необходимые операции требуются произвести за время, меньшее длительности полукадра видео.

Приведенные ниже ограничения достаточно жёсткие. Однако, если следовать перечисленным ниже рекомендациям, можно получить действительно впечатляющие результаты.

2.1.1 Экспорт общего трека сцены

Объекты сцены могут иметь анимацию с помощью треков (см. также раздел 2.1.12). Для управления траекториями в *HotActions* необходимо присвоить им имена, поскольку в приложении *3D Studio Max* имена траекторий не задаются.

Кнопки  на панели окна сцены (Рис. 18) служат для воспроизведения и остановки общего (неименованного) трека сцены. С помощью этих кнопок можно просмотреть общий трек сцены до того, как инициализировали сцену в проекте. Именование треков конкретных объектов в процессе инициализации исключает их из проигрывания кнопками этой панели, и дальнейшее управление такими треками осуществляется с помощью команд. Подробное описание команд работы с треками содержится в главе 4 руководства пользователя по использованию команд сценария.

Для дальнейшего использования сцены необходимо добавить её в существующий проект приложения *HotActions* или создать новый проект (см. главу 3 Руководства пользователя *HotActions*).

2.1.2 Геометрия

Экспортируются параметры:

- **Standard primitives;**
- **Extended primitives;**
- **Compound objects.**

2.1.3 Камеры

Из камер экспортируется только:

- **Target camera** – направленная камера. Виртуальные камеры этого типа чаще всего управляются в сцене с помощью пустых (*Dummy*) объектов (см. раздел 9.1). **Target camera** может быть прикреплена к любому анимируемому объекту в сцене. Треки создаются именно для этих вспомогательных объектов. Это позволяет корректно вращать камеру.
- **Free camera** – свободная камера, не имеющая точки нацеливания.

2.1.4 Источники света

Можно использовать до 8 источников света. Однако по возможности необходимо минимизировать их количество, поскольку каждый новый источник света снижает скорость рендеринга в реальном времени. Возможно использование следующих типов источников:

- **Omni** – точечный всенаправленный источник света, находящийся в точке и излучающий во всех направлениях трёхмерного пространства сцены.
- **Target Spot** – целевой точечный направленный источник света, состоящий из собственно излучателя и Target (точки цели), задающий направление лучей. Лучи света расходятся конусом. Поверхность, освещаемая таким источником, должна быть тесселирована, степень тесселирования подбирается экспериментально).

- **Target Direct** – нацеленный прямой источник света, излучаемый плоскостью. Испускает пучок параллельных лучей света.
- **Free Spot** – свободный точечный источник без целевого объекта, лучи которого расходятся конусом. Направление светового луча меняется вращением осветителя.
- **Free Direct** – свободный направленный, излучающий из плоскости. Испускает пучок параллельных лучей света.

Поддерживаются следующие параметры источников света:

- **Multiplier** - интенсивность света.
- **Attenuation** - область затухания - поддерживается, если поверхность тесселирована.
- **Light Color** – цвет источника.
- **Spotlight/Directional Parameters** – диапазон углов, регулирующих параметры направленного луча – поддерживается только для направленных источников света.

НЕ поддерживается **Inverse Square Decay** и тени.

2.1.5 Иерархия

Поддерживаются:

- **Pivot;**
- **Linking;**
- **Unlink.**

2.1.6 Параметры среды Environment

Поддерживаются:

- **Background Color** – цвет фона;
- **Ambient Global Lighting** – цвет рассеянного света;
- **Fog Atmospheric Effect** – эффект дымки (тумана).

2.1.7 Материалы

Поддерживаются типы:

- **Standard;**
- **Multi/sub-object;**
- Для материала с типом **Standard** из **Shader Basic Parameters** экспортируется только **Blinn** (раскраска по Блинну)(Рис. 1), другие игнорируются. Из **Blinn Basic Parameters** экспортируются только параметры: **Ambient, Diffuse, Self-Illumination, Opacity, Specular, Specular Level, Glossiness**. Другие параметры закраски игнорируются.

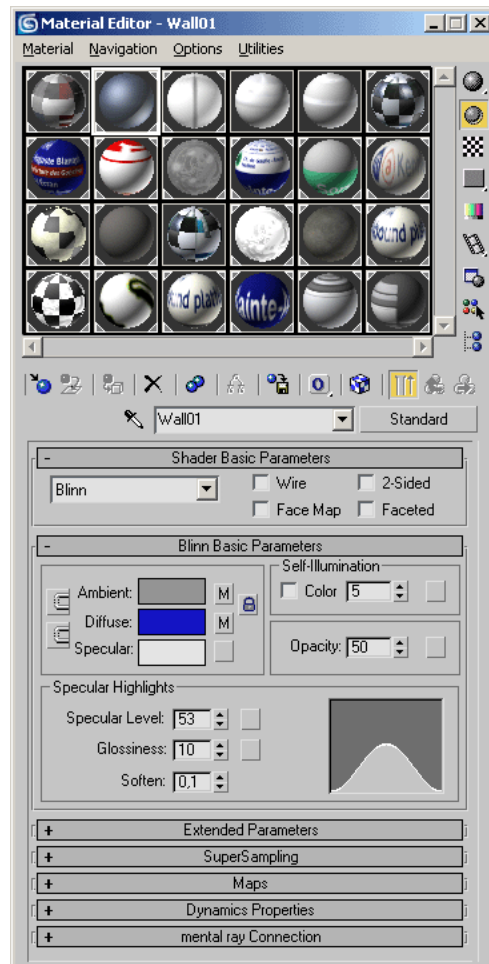




Рис. 1. Диалог выбора материала.

2.1.8 Текстуры


В качестве карт текстур используются только **Bitmap**, как статические изображения, так и видеофайлы:

- **Diffuse color;**
- **Ambient color;**

 *Внимание:* этот цвет используется не как **Ambient color**. **Ambient color map** экспортируется в 3d-формат как дополнительная **Diffuse color map**, а если установить для неё значение **Alpha Source** в **RGB Intensity**, то как **LightMap**. Использование **LightMap** полезно тем, что эта текстура обладает независимым набором координат, то есть можно изменять основную текстуру (**Diffuse Color**), не меняя при этом текстуру освещённости, тени и блики остаются прежними.

 *Использование видеофайлов в качестве текстур приводит к заметному увеличению размера результирующего файла экспорта. Это, в свою очередь приводит к увеличению времени загрузки, а также, если файл слишком велик, к проблемам с производительностью. Для каждой сцены допустимый общий объем экспортируемых видеофайлов следует подбирать экспериментально.*

- **Opacity;**
- **Reflection.**

 *Внимание:* при использовании этой карты текстуры необходимо предварительное ознакомление с особенностями, описанными в главе 5.

2.1.9 Текстурные координаты

Для текстур – экспортируются:


- **UV offset**;
- **UV tiling**;
- **W Angles**;
- из **Texture/environ** – только текстура;
- из **Mapping** – только **Explicit Map Channel**.

Максимальное число текстур для материала равно трём. Допустимы следующие комбинации:

- **Ambient color + Diffuse color + Opacity**:
 - если **Alpha source** в **Bitmap parameters** отмечены как **No (Opaque)**, карты будут просто смешаны пропорционально выставленным значениям (**Amount**). Размеры текстур (в текселях – аналог пикселей) для **Diffuse color** и **Opacity** должны быть идентичны. Значения **U** и **V** для **Diffuse color + Opacity** должны быть идентичны;
 - если **Alpha source** в **Bitmap parameters** отмечены как **RGB Intensity**, карты будут смешаны специальным способом (**multiply**). В этом случае **Amount**-значения параметров текстуры игнорируются.
- **Diffuse color + reflection**:
режимы смешивания такие же, как в **Ambient color + Diffuse color**.
- **Diffuse color + Opacity**.

2.1.10 Размеры текстур

Размеры текстур (в текселях) для **Diffuse color + Opacity** должны быть идентичны. Значения **UV** для **Diffuse color + Opacity** должны быть идентичны.

 *Внимание: Ограничения на размер карты определяются типом видеокарты, установленной в компьютере: 2048 текселов для GeForce 3 или до 8192 текселов для более поздних выпусков GeForce. Желательно контролировать размер сторон текстур. При экспорте в 3D-файл или загрузке 3D-файла сцены в HotActions(см. раздел 11.2) стороны текстур автоматически перемасштабируются так, чтобы они равнялись 2 текселям в степени N, где N – целое число. Таким образом, в текселях это будет: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. Причём масштабирование всегда происходит к ближайшему верхнему пределу 2^N для того, чтобы не было потеряно качество текстуры. Например, текстура с размерами 257x513 будет масштабирована к размерам 512x1024, а не к 256x512, хотя разница между 256x512 и 257x513 не превышает по 1 пикселу в ширину и высоту. При подобном масштабировании без необходимости будут задействованы дополнительные ресурсы компьютера. Поэтому процесс лучше контролировать и делать так: вначале создать текстуру (например, в Photoshop) в естественных пропорциях. После чего масштабированием привести стороны текстуры к ближайшим допустимым значениям, но так, чтобы качество картинки не пострадало. Может получиться нечто сплющенное или растянутое. В данном случае это не имеет значения, ведь в 3D-сцене пропорции зависят исключительно от того, как заданы текстурные координаты (**UVW Mapping**). С одной стороны, делать текстуры с предлагаемыми размерами не является жёстким требованием, но с другой стороны – крайне желательно (особенно для сцен, где этих текстур много).*

2.1.11 Вспомогательные объекты

Экспортируется только **Dummy**.


2.1.12 Анимация

Анимация поддерживается только для параметров узлов (камер, источников света и собственно объектов сцены). **Bones** не поддерживаются, не поддерживаются динамические изменения материала (прозрачность и так далее), не поддерживается **Visibility Track**.

3 Реализация морфинга в виртуальной студии

Процедура **Морфинга** выполняется в виртуальной студии командой сценария MORPH, которая относится к разделу команд RENDER (см. главу 5 Руководства пользователя по использованию команд сценария). Чтобы объекты можно было подвергать морфингу, в исходной сцене 3DS MAX нужно:

1. создать основной объект;
2. создать несколько копий объекта в соответствии с предполагаемым количеством стадий морфинга, по одной копии на каждую стадию;
3. отредактировать созданные копии объекта для создания стадий морфинга.

 *Внимание: объекты на всех стадиях морфинга должны содержать столько же вершин, сколько их у основного объекта. Чтобы стадии морфинга (но не основной объект) были невидимыми, используется редактор траекторий.*

4 Имитация световых эффектов и неровностей поверхностей

Тени и световые блики очень важны для визуального восприятия сцены. Простое сокращение количества света может привести к тому, что сцена будет выглядеть плоско и невыразительно. Необходимо иметь в виду и то, что экспорт теней источников света не поддерживается. Решением этой проблемы может быть имитация теней и световых бликов путем использования текстур. Это можно сделать различными способами. Ниже рекомендуется один из них.

- Создайте столько и такого типа источников света, сколько необходимо для впечатляющего освещения сцены и создания теней. При этом игнорируйте все ограничения, описанные выше.
- Используйте возможности *3D Studio Max* для создания (рендеринга) текстур с «впечатанными» тенями и бликами или светотеневых текстур (**Lighting Map**). Начиная с версии 5.0 в *3D Studio Max* предусмотрен рендеринг в текстуру (модуль **Render To Texture**). Рекомендуемые установки при использовании этого модуля представлены на Рис. 2.

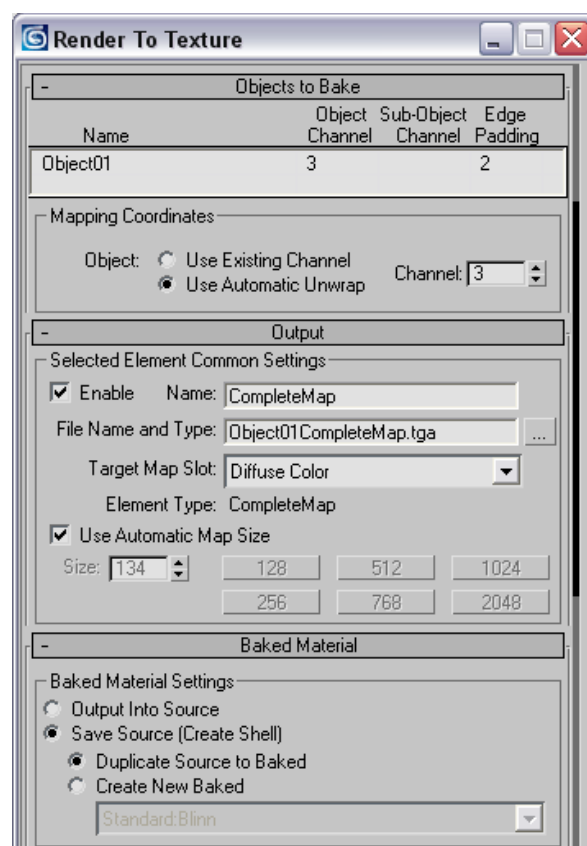


Рис. 2. Рекомендуемые настройки модуля *Render To Texture* при пререндеринге текстур.

При использовании автоматической установки размеров текстур (**Use Automatic Map Size**) нужно помнить, что не во всех версиях *3DS MAX* определение размеров происходит корректно. Результат установки следует проверить, просмотрев размеры текстур в приложении *HotActions* (глава 11).

- Удалите все вспомогательные источники света, оставив только необходимые, количеством не более 8 (раздел 2.1.4).

Использование модуля *3DS MAX Render To Texture* в сильной степени улучшает визуальные характеристики сцены в приложении *HotActions*. При этом необходимо помнить, что размер видеопамати графического ускорителя ограничен и чрезмерное её переполнение может приводить к пропуску кадров при показе. Поэтому предварительный рендеринг рекомендуется производить только для наиболее выигрышных в визуальном отношении текстур, не забывая при этом оптимизировать их размеры (раздел 2.1.10).

5 Особенности использования материалов с отражением

В связи с особенностями аппаратного отображения поверхностей с отражением, на стыках больших граней могут возникнуть изломы изображения. Эти искажения вызваны отсутствием коррекции перспективного преобразования.

Если искажения слишком заметны, то возможными решениями этой проблемы являются:

- разбиение крупных граней (тесселяция);
- сокрытие границ между гранями, то есть размещение объектов в сцене так, чтобы границы их крупных граней не отражались;
- в случае большого угла между гранями, особенно между крупными, имеет смысл его сгладить, добавив небольшие лишние грани.

6 Сокращение количества многоугольников в 3D-сцене

 *Внимание! 3D-сцена не должна содержать слишком много граней и не должна быть перегружена текстурами.*

Объекты сцены необходимо оптимизировать. Это можно сделать различными способами, ниже приводятся лишь основные положения.

- Желательно знать возможные позиции камеры в сцене. Чем дальше объект от камеры, тем менее детальным он может быть.
- Удалите или упростите все объекты, которые не важны для восприятия сцены и сценария.
- Удалите все объекты, которые могут моделироваться текстурой, например, лист бумаги на столе и так далее.
- Оптимизируйте плоские и слабо изогнутые поверхности объектов, редактируя параметры объектов и/или используя модификатор **Optimize**.
- Удалите все поверхности объектов, которые никогда не будут наблюдаться камерой.

7 Живое видео

Материал, используемый для видеоизображения, должен содержать **Diffuse Map**. После инициализации видеоматериала в сцене эта карта будет заменена видеоизображением. Подробности процесса описаны в главе 7 руководства по использованию команд сценария. При этом все остальные карты и параметры материала будут игнорироваться.

Название материала не имеет значения. В примерах, которые обычно находятся в директории **VS_Samples** на диске **D:**, материалы, чаще всего, называются **VIDEO1** и **VIDEO2**, если используются два потока входного видео одновременно. Если задействован только один поток входного видео, то материал часто называют VIDEO или VIDEO_m. Последнее может быть даже удобнее, поскольку такое название позволяет отличать имя материала от составляющего слова VIDEO, используемого в командах. Например, VIDEO в `RENDER.VIDEO.LIVE_1.CREATE = 1` – часть команды. А VIDEO_m в `RENDER.MATERIAL.VIDEO_m.SOURCE = LIVE_1` – имя материала.


7.1 Создание текстуры для видеоматериала

Изображение на текстуре, используемой для видеоматериала, не имеет значения, поскольку она будет замещена живым видео после инициализации сцены в *HotActions*. Однако, для того, чтобы разработчик сцены лучше представлял себе общую композицию, рекомендуется использовать в качестве видео текстуры изображение актёра, причем, в том ракурсе и в таком плане (поясной, во весь рост и тому подобное), в каком он предположительно будет выглядеть в видеискателе камеры. Для этого используются два типа текстур – **Diffuse** и **Opacity**.


Для **Diffuse Map** используйте фотографию (рисунок) условного актёра, соблюдая обычные рекомендации для текстур (см. раздел 2.1.10).

Для **Opacity Map** – маску. Она создаётся в графическом редакторе на основе той же фотографии или рисунка. Белым цветом заполните силуэт актёра – это будет непрозрачной частью материала. Остальное поле маски заполните чёрным цветом – это прозрачная часть.

Все операции для **UVW Mapping** сохранятся при замене на живое видео. Мы рекомендуем применить видеоматериал к прямоугольнику с отношением сторон 4:3 и для PAL, и для NTSC (16 : 9 для HD).

 *Геометрический аспект сторон объекта с видеотекстурой 4:3, предлагаемый для имитации экрана телевизора, не имеет отношения к разрешению ТВ формата (720x576 для PAL или 720x480 для NTSC). Аспект пикселя может быть различным для разных ТВ форматов, но ни для PAL, ни для NTSC он не является единичным.*


Текстура должна быть вписана в этот прямоугольник. Размеры текстуры по горизонтали и вертикали значения не имеют, поскольку после назначения на этот материал видео все параметры берутся из видео. Изменить пропорции сторон прямоугольника после экспорта сцены можно, используя в инициализирующем *Action*'е команды сценария **DATA.NODE.«Object».**`SCL = fa, fb, fc`. После изменения пропорций может потребоваться сдвинуть объект, что также возможно с помощью соответствующих команд (см. руководство «Использование команд сценария»). Тем не менее, допустимо использовать основой для видео любой материал сцены, содержащий **Diffuse Map**. Таким образом, видеоизображение может быть при необходимости показано на любом объекте сцены – стене, экране или даже чайнике.

 *Для избежания поворота плоскости с изображением актёра относительно виртуальной камеры так, что теряется объёмность изображения и оно видится как плоская грань, рекомендуется ставить перед названием объекта символ «!». Тогда при смещениях виртуальной камеры объект тоже всегда смещается таким образом, чтобы плоскость с его изображением оставалась перпендикулярной лучу зрения. Объект ведёт себя подобно флюгеру, поворачиваясь к камере вокруг своей вертикальной оси (Z) (см. главу 8) Для таких объектов настоятельно рекомендуется применить модификатор `Reset XForm`. Это связано с тем, что в 3D Studio Max направление осей координат создаваемого объекта не обязательно*

совпадает с «мировыми» координатами, то есть, «собственная» ось Z может быть направлена не вверх.

8 Именованние объектов в 3D-сценах и зарезервированные префиксы имён

Если имя объекта состоит более чем из 40 символов, при экспорте сцены в *HotActions* оно сокращается с присвоением номера таким образом, что количество символов не превышает 40.

 Желательно аккуратно называть материалы и объекты соответствующими их назначению именами, тем самым упрощая работу с ними в *HotActions*.

Специальные имена используются для объектов, к которым должно происходить неявное обращение, например при манипулировании мышью или джойстиком сразу после открытия сцены:

- Объект с именем **World** выбирается для манипулирования при открытии сцены.
- Нельзя называть объект в сцене именем **<Current>**. Это специальное имя используется в командах сценария для обозначения текущего выбранного объекта. При смене объекта манипуляций соответственно изменяется имя, на которое указывает **<Current>**.

Кроме того, особое значение имеют следующие символы в начале имени объекта:

<i>Символы, влияющие на положение объекта относительно виртуальной камеры</i>	
#	Объект всегда обращён к лучу зрения своей фронтальной частью. Все углы вращения объекта равны нулю в системе координат наблюдателя.
!	То же, что и #, но при этом объект ведёт себя подобно флюгеру, поворачиваясь к камере вокруг своей вертикальной оси (Z).
:	То же, что и #, с различием, что объект медленней уменьшается с расстоянием. Используется для источников света, которые на расстоянии имеют тенденцию казаться большими, чем в действительности.
<i>Символы, изменяющие приоритет для рендеринга. Система использует алгоритм Z-буфера, который не требует приоритетной сортировки, для устранения невидимых зрителю поверхностей. Но могут быть случаи, когда задание приоритета необходимо, например, в сцене с полупрозрачными объектами.</i>	
< >	Изменяет приоритет объекта относительно других объектов с тем же самым предком в дереве сцены. Имена объектов, которые ближе к зрителю, должны быть с префиксом '<', те, которые дальше – с '>'.
<< >>	То же, что и '<', и '>', но используется, чтобы изменить приоритет относительно всех объектов сцены.

9 Пример экспорта и использования в *HotActions* простейшей сцены

Для того, чтобы использовать сцену в качестве виртуальных декораций, необходимо содержание в сцене как минимум:

- объекта с присвоенным ему текстурным материалом (для видео);
- камеры;
- источника света.

В следующих главах мы рассмотрим простейшую сцену, созданную с помощью приложения *3D Studio MAX* и поставляемую в качестве примера. Это пример минимальной сцены, изображающей «виртуальную студию».

Отметим также, что конечный результат описанной далее работы по созданию проекта и его содержимого также имеется среди поставленных примеров – проект **simple.vsp** (папка **Simple**, которая находится в директории **VS_Samples\VS_Basic**, обычно на диске **D:**). Файл **readme.txt** этого проекта содержит краткое описание назначения отдельных *Action*'ов, сами команды которых также подробно прокомментированы.

9.1 Составляющие минимальной сцены виртуальной студии

Запустите *3D Studio MAX*. С помощью команды **Open** меню **File** откройте файл «**Simple.max**». Сцена представляет собой готовый пример декораций с наиболее типичными элементами, обычно используемыми в Виртуальной студии для создания проектов.

К обязательным объектам относятся:

- камера (типа **Target**);
- источник света (типа **Omni**);
- плоский прямоугольник с изображением актёра, материалу этого объекта в проекте приложения *HotActions* будет назначен поток видео с камеры.

К необязательным объектам, без которых, тем не менее, трудно представить себе какую-либо студию, относятся:

- пол, а также зелёный и синий фоны для актёра (в нашем примере – объекты **back_blue** и **back_green**, из которых, чаще всего, оставляют видимым только нужный, пряча другой);
- стол или любой объект, который может понадобиться, чтобы скрыть нижнюю часть туловища актёра, если предусматривается, что актёр в камере не в полный рост. Например, при съёмке актёра по пояс, что довольно часто встречается, когда надо посадить диктора за стол;
- экран для вывода видеофрагментов, слайдов и так далее. Текстуру, используемую для материала экрана обычно впоследствии, уже в проекте *HotActions*, заменяют видеопотоком из видеофайла или входным видео.

Также в сцене использована вторая камера с именем **Camera02**, управляемая посредством пустого (*Dummy*) объекта **CAMERA2**, для наглядной демонстрации возможности мгновенного переключения с одной камеры на другую при работе с приложением *HotActions*. Первая камера с именем **Camera01** управляемая с помощью *Dummy* объекта **CAMERA1**, содержит анимацию движения камеры.

Отметим также некоторые важные особенности создания составляющих сцены:

- Для вывода изображения «актёра» в проекте виртуальной студии, в сцене предварительно создаётся заготовка - плоский прямоугольник с соотношением сторон 4:3 (16:9) в соответствии с рекомендациями раздела 7.1. Для создания материала, который впоследствии будет заменён живым видео, карте **Diffuse** назначается фотография актёра, а в карте **Opacity** используется чёрно-белая маска этой фотографии. Для избежания поворота камеры под сильным углом к плоскости «актёра», при котором его объёмное изображение видится как плоская грань, рекомендуется ставить в названии объекта первым символом «!». При этом объект ведёт себя подобно флюгеру, всегда поворачиваясь к камере фронтальной

частью вокруг своей вертикальной оси (Z). Именно такой объект и присутствует в сцене (с именем **lactress**).

☞ При таком поведении объекта («флюгер») используются локальные координаты объекта, которые не обязательно совпадают с мировыми. Поэтому, во избежание путаницы рекомендуем в *3D Studio MAX* применить к объекту, в имени которого используется символ «!», операцию «Reset Xform», которая приводит локальные координаты объекта в соответствие с мировыми.

- В качестве фона можно использовать разнообразные декорации, используя анимацию, созданную в приложении *3D Studio MAX*. В данном примере фон – плоский простейший объект. Так как актёр снимается на синем или зелёном фоне, то для более естественного помещения его в виртуальные декорации желательно создать фон, близкий к зелёным или синим тонам в зависимости от того, на каком фоне планируется снимать актёра. Можно, как в данном примере, создать сразу два фона и, по необходимости, с помощью команд сценария в *HotActions* скрывать тот или иной.
- Стол для актёра. Этот объект наиболее близок к камере, поэтому ему нужно уделять наибольшее внимание при создании сцены, делать более детальным. Имеет смысл использовать на поверхности такого объекта карту отражения. В данном примере при приближении камеры к актёру хорошо видно, как с помощью карты **Reflection** имитируются блики материала. Так как при наличии угла между большими гранями могут возникать изломы изображения, поверхность цилиндров сделана высотой в одну грань (см. главу 5).
- В данном примере экран – плоский объект без толщины, с материалом, имеющим одну карту типа **Diffuse** (подобно плоскости актёра). Ниже показано, как на экран можно вывести изображение из AVI-файла или показать слайд непосредственно в *HotActions*.

То, как будет выглядеть сцена в *HotActions* (Рис. 3), можно посмотреть с помощью приложения *VS Exporter* (см. раздел 11.2).

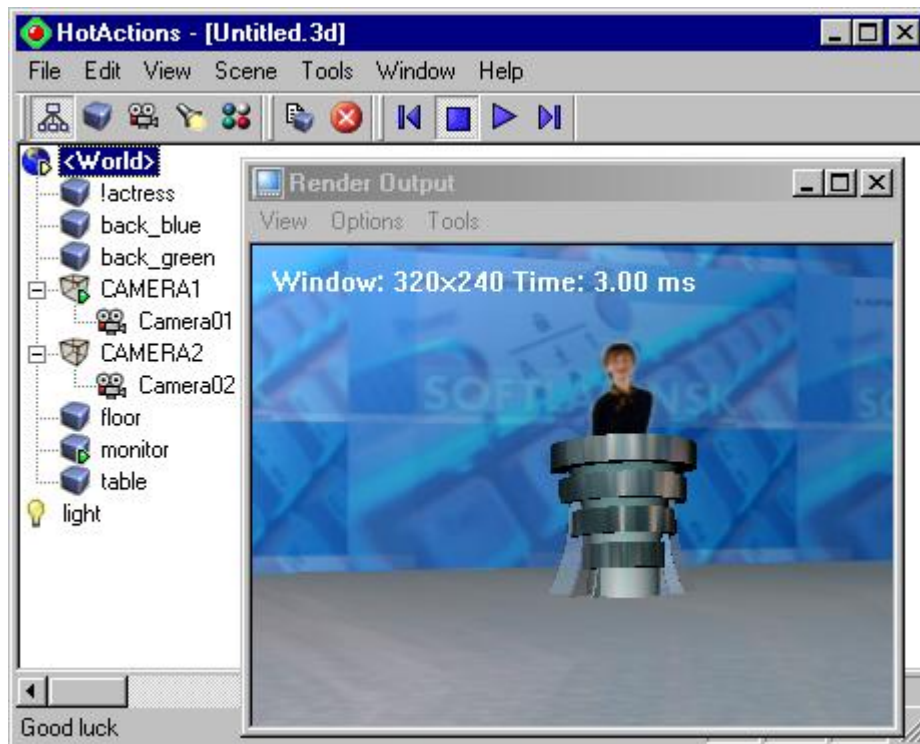


Рис. 3. Сцена примера, открытая с помощью *VS Exporter*.

9.2 Экспорт сцены и создание нового проекта в *HotActions*

Для экспорта сцены достаточно сохранить её с помощью команды **Save As...** меню **File** главного окна *HotActions* (Рис. 3), указав имя, например «Simple», и папку в появившемся диалоге. Подробнее экспорт описан в разделе 11.3.

В результате экспорта сцены получается файл формата *.3d – файл сцены. Для поставляемого примера сцены уже имеется экспортированный файл с тем же именем и в той же директории, где и «Simple.max», но с расширением *.3d.

Файлы сцены можно использовать в существующих проектах приложения *HotActions* или создать новый проект. Для создания нового проекта надо запустить приложение *HotActions* и выбрать команду **New** в меню **File** главного окна. Выберите в появившемся диалоге **VS Project** и введите имя проекта, например «Simple», в поле **File Name** и укажите папку для его сохранения (Рис. 4).

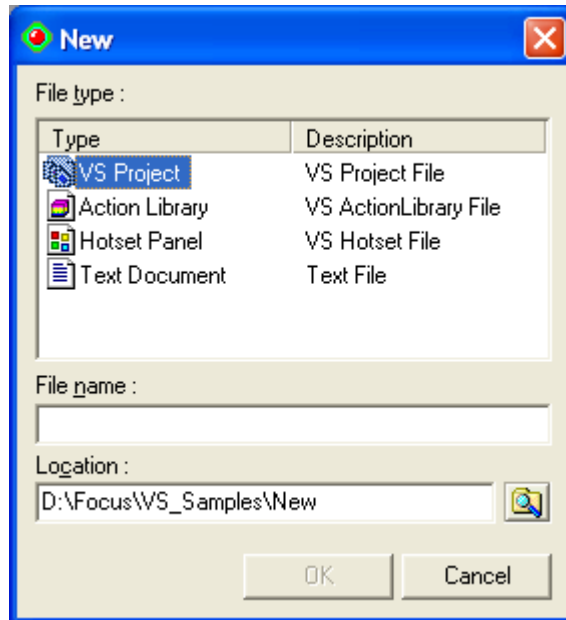


Рис. 4. Диалог выбора типа нового файла для создания.

Теперь с помощью команды **Add File to Group** контекстного меню проекта для группы **Scenes** (Рис. 5), добавьте файл сцены **Simple.3d** в проект.

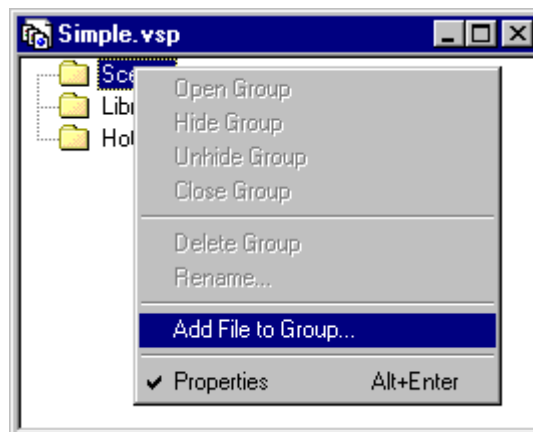


Рис. 5. Добавление файлов в проект.

В новом проекте необходимо создать библиотеку *Action*’ов для реализации основных действий со сценами. Также нам понадобится, как минимум, один *Hotset*, где мы поместим кнопки, нажатие которых и будет запускать *Action*’ы из библиотеки.

Библиотека *Action*’ов (*Actions Library*) создаётся с помощью выбора команды **Actions Library** в меню **New** (Рис. 4), которое открывается командой **New** в меню **File** основного окна приложения *HotActions*.

Hotset создаётся с помощью выбора **Hotset Panel** в диалоге **New** (Рис. 4). При первом сохранении созданных файлов запрашивается имя файла и директория для сохранения. Для удобства назовём их «Simple» и сохраним в одной директории с файлом проекта. Здесь необходимо отметить, что чаще всего все файлы, входящие в один проект, создаются с одинаковыми именами и различаются лишь расширениями.

Важным моментом является задание директорий для видео файлов и файлов изображения, работа с которыми описана в следующих разделах. Это можно сделать на панели **Directories** диалога *Properties* для любой из групп или файла в окне проекта.

9.3 Работа со сценой с помощью *Action*'ов

Написание *Action*'ов является легко изменяемым и переносимым способом обеспечить взаимодействие со сценой – её показ, запуск отдельных треков объектов и так далее. Следующая последовательность действий является стандартной для создания *Action*'а в открытой библиотеке.

- Щёлкните правой кнопки мыши на свободном пространстве в окне библиотеки.
- В открывшемся контекстном меню (Рис. 6) выберите команду **New Action**. Фокус устанавливается на имени нового *Action*'а, чтобы можно было сразу изменить его имя.

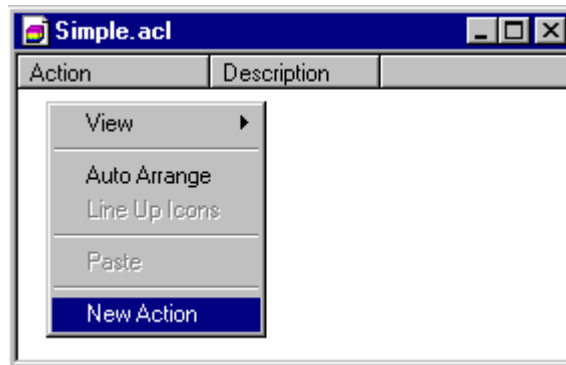


Рис. 6. Создание нового *Action*'а в библиотеке.

После создания нового *Action*'а можно вписать в специально отведённое в нём поле команды, которые он будет содержать. Для этого щёлкните по нему правой кнопкой мыши и в контекстном меню выберите команду **Edit** (Рис. 7, слева). Откроется диалог *Properties*. Текстовое поле **Body** (Рис. 7, справа) панели **Action** этого диалога предназначено для ввода и редактирования команд сценария. Если первым знаком команды ввести знак «:», то в появившемся локальном меню подсказок (Рис. 7, справа) можно выбрать одну из доступных команд. Дальнейший ввод слов команды в системе подсказок возможен по нажатию символа разделения (точки). Например, для набора команды DATA.CURRENT.CAMERA после выбора из списка слова DATA нужно напечатать (.) - точку, затем в появившемся списке команд выбрать слово CURRENT, снова напечатать (.) и так до полного набора команды. Такой механизм облегчает создание команд и уменьшает вероятность ошибки.

После набора или изменения команд *Action*'а, можно проверить их исполнение, если нажать на кнопку **Test** в диалоге *Properties* или два раза щёлкнуть по этому *Action*'у в библиотеке *.acf. Кнопка **Apply** просто сохраняет изменения без запуска *Action*'а.

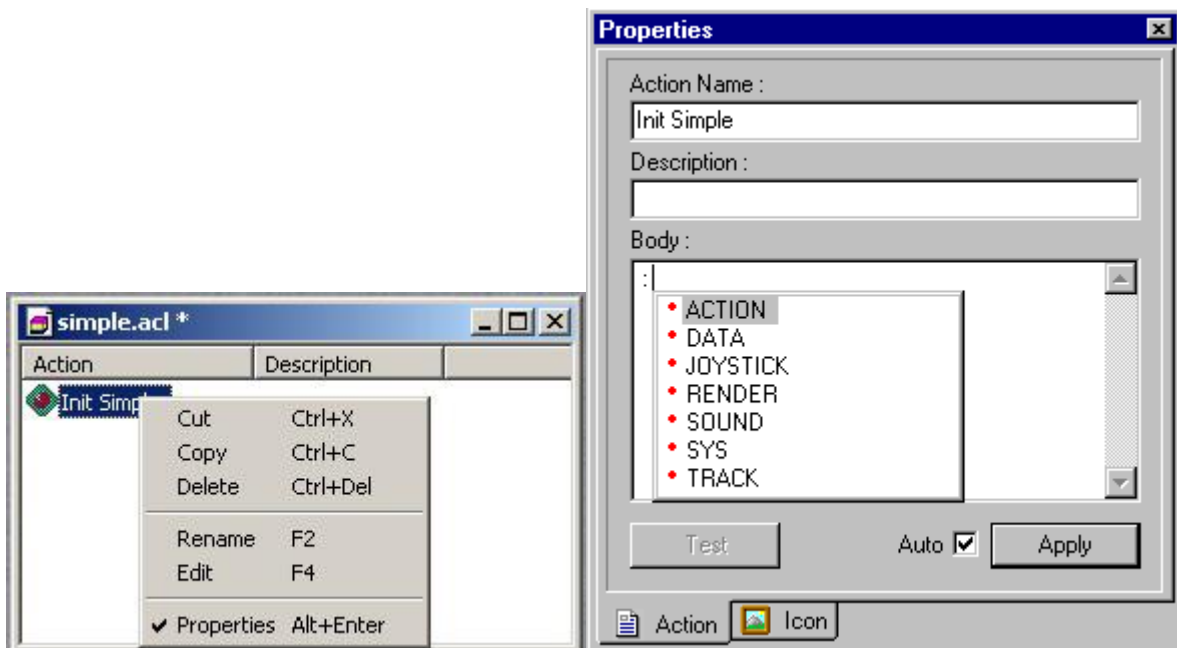


Рис. 7. Локальное меню для *Action*'а в библиотеке (слева) и панель *Action* диалога *Properties* для него.

Приведение сцены в начальное рабочее состояние является задачей инициализирующих *Action*'ов, с рассмотрения которых и начинается данный раздел.

9.3.1 Создание *Action*'а, инициализирующего сцену

Особую группу *Action*'ов представляют инициализирующие (*\$STARTUP\$*) *Action*'ы. Их задачей является приведение сцены в начальное рабочее состояние, например, при загрузке сцены.

В библиотеке следует создать новый *Action* и назвать его **Init Simple**. Этот *Action* должен содержать следующие команды, выполняющие необходимые действия.

- Прежде всего, необходимо загрузить файл сцены: `DATA.OPEN = «simple.3d»`.
- Делаем видимым нужный фон, пряча оставшийся ненужным. При работе с синим фоном, соответственно, прячем зелёный фон. В проекте эти команды помещены также в отдельные *Action*'ы – **Blue backdrop** и **Green backdrop**, для возможности отдельного вызова:

```
:DATA.NODE.back_blue.HIDE = 0
:DATA.NODE.back_green.HIDE = 1
:DATA.NODE.monitor.HIDE = 0.
```

- Для управления траекториями в *HotActions* им необходимо присвоить имена. В сцене имеется два объекта, имеющих анимацию. Первый – виртуальная камера, управление которой осуществляется с помощью пустого (*Dummy*) объекта **CAMERA1**. Второй – экран, объект с именем **monitor**. Следующие команды именуют треки, принадлежащие этим объектам:

//именуем треки, устанавливаем начальные позиции:

```
:TRACK.TV.NODE = «monitor»
:TRACK.TV.START = 30
:TRACK.CAMERA.NODE = «Camera01»
:TRACK.CAMERA.START = 0.
```

Предпоследняя команда присваивает траектории объекта **CAMERA1** имя «CAMERA», которое в дальнейшем используется для работы с траекториями этого объекта, например, для запуска.

- Необходимо создать видеопотоки для сцены и связать их с имеющимися материалами:

```
//создаётся видеопоток LIVE_1:
```



```

:RENDER.VIDEO.LIVE_1.CREATE = 1
//устанавливается формат видеопотока, включается альфа-смешивание, а также
//отсечение краёв кадра:
RENDER.VIDEO.LIVE_1.FORMAT = ALPHA, CROP
//видеопоток запускается:
RENDER.VIDEO.LIVE_1.START = 1
//ожидание, пока видеопоток запустится
SYS.WAIT = VIDEO.LIVE_1.START
//назначает материалу с именем VIDEO_m видео текстуру из видеопотока LIVE_1,
//которая будет наложена на все объекты в сцене с этим материалом:
RENDER.MATERIAL.VIDEO_m.SOURCE = LIVE_1
//создание файлового видеопотока MSDS_1:
:RENDER.VIDEO.MSDS_1.CREATE = 1
//определение режима воспроизведения видеофайла: непрерывный, зацикленный
(LOOP), со звуком (AUDIO), первым показывается нижнее поле (LFF), включена
трилинейная фильтрация для корректировки искажений показа удалённого
видеоизображения (MIPMAP)
:RENDER.VIDEO.MSDS_1.FORMAT=LOOP, AUDIO, LFF, MIPMAP
//определение источника данных для видеопотока MSDS_1:
:RENDER.VIDEO.MSDS_1.DATA = Biathlon_Bjordalen_LFF.avi
//старт видеопотока MSDS_1:
:RENDER.VIDEO.MSDS_1.START = 1
//ожидание, когда видеопоток MSDS_1 запустится
:SYS.WAIT = VIDEO.MSDS_1.START
//показ на материале телевизора видеопотока MSDS_1:
:RENDER.MATERIAL.monitor_m.SOURCE = MSDS_1.

```

Единственным объектом в нашей сцене, имеющим материал с именем **VIDEO_m**, является плоскость **lactress** – «актёр». Таким образом, мы обеспечили появление в сцене «живого видео».

Подробно о каждой команде можно узнать в руководстве пользователя по использованию команд сценария. Отметим только, что в инициализирующих *Action*'ах кроме команд загрузки сцены и именования треков могут, как и в обычном *Action*'е, находиться любые команды их модификации.

Теперь для дальнейшей работы, при каждом открытии сцены «**Simple.3d**» в составе проекта, необходимо исполнять написанный нами *Action* с именем **Init Simple**. Приложение *HotActions* автоматически исполняет содержимое папки **\$STARTUP\$** при нажатии кнопки  на панели инструментов основного окна приложения. Поэтому рекомендуется инициализирующие *Action*'ы помещать в папку **\$STARTUP\$**. При соблюдении этой рекомендации, чтобы привести даже незнакомый проект в рабочее состояние, достаточно будет нажать на кнопку .

Папка **\$STARTUP\$** всегда создаётся при создании *Hotset*'а. По умолчанию эта папка скрыта, но можно работать с ней, перейдя в режим конфигурирования для *Hotset*'а. Для этого нужно щёлкнуть два раза правой кнопкой мыши на свободном месте в *Hotset*'е или включить опцию **Configure** в контекстном меню *Hotset*'а (Рис. 8, слева).

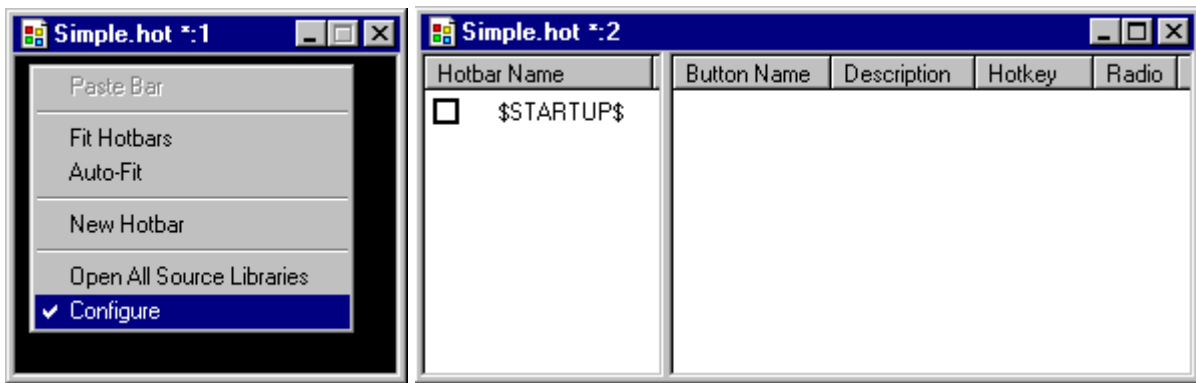


Рис. 8. Контекстное меню *Hotset*'а (слева) и его альтернативное представление (справа).

В этом режиме *Hotset* представляется альтернативным документом с папками (Рис. 8, справа). Все папки размещены в левой половине окна документа, а содержимое выбранной папки показывается в его правой половине, это ссылки на *Action*'ы, соответствующие кнопкам *Hotbar*'а. При пустом *Hotset*'е единственная папка, которая в нём имеется, это папка *\$STARTUP\$*.

Нужно переместить *Action* с именем **Init Simple** из библиотеки в папку *\$STARTUP\$* и закрыть окно альтернативного представления *Hotset*'а. Так как папка *\$STARTUP\$* скрыта по умолчанию (квадратик слева от имени очищен, Рис. 8, слева), то никаких кнопок, соответствующих **Init Simple**, в *Hotset*'е не появится.

9.3.2 Использование именованных треков

Треки объектов **Camera01** и **monitor** именованы (см. раздел 9.3.1) и доступны для управления ими. Панель **Track** диалога *Properties* этих объектов содержит информацию о треке объекта отдельно по его составляющим (Рис. 9).

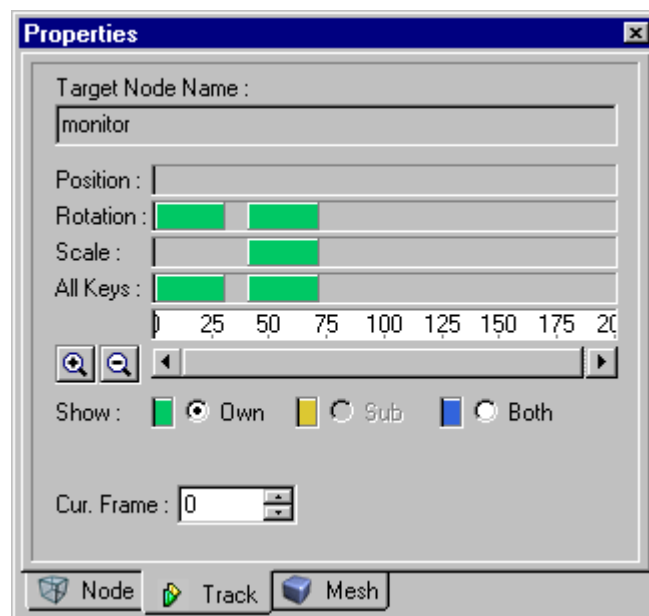


Рис. 9. Панель *Track* диалога *Properties*.

Имеющимся изменениям позиции, угла поворота и масштаба объекта на панели соответствуют цветные прямоугольники на соответствующей линии (Рис. 9). Цвет прямоугольников зависит от того, какая радио-кнопка выбрана в группе **Show**.

Группа радио-кнопок **Show** позволяет сгруппировать и показать следующие треки:

- собственные треки объекта (установленный переключатель **Own**);
- суммарную информацию о треках его потомков (установленный переключатель **Sub**);

- собственные треки объекта вместе с треками его потомков (установленный переключатель **Both**).

Представленную информацию можно эффективно использовать в командах сценария. Например, создать *Action* для появления экрана в сцене. Назовём его **Show TV** и поместим в него следующие команды:

```
:DATA.NODE.monitor.HIDE = 0
:TRACK.TV.GOTO = 30
:SYS.WAIT = "TRACK.TV".
```

Команда TRACK.TV.GOTO = 30 запускает трек **TV** с текущего по 30-ый кадр, обеспечивая появление экрана.

Результат аналогичной команды (следует поместить её в *Action* с именем, например, **Hide TV**) – плавное удаление экрана из сцены:

```
//запуск трека с 40-ого по 72-ой кадры:
:TRACK.TV.START = 40,72
:SYS.WAIT="TRACK.TV"
:TRACK.TV.START = 0
:DATA.NODE.monitor.HIDE = 1.
```

Таким же образом можно создать *Action*'ы, использующие трек **CAMERA** для управления виртуальной камерой:

Go to Close Up, служащий для приближения к актёру, с командами:

```
:DATA.CURRENT.CAMERA = «Camera01»
:TRACK.CAMERA.GOTO = 90
SYS.WAIT = "TRACK.CAMERA".
```

Go to Long shot с командами:

```
:DATA.CURRENT.CAMERA = «Camera01»
:TRACK.CAMERA.GOTO = 50
SYS.WAIT = "TRACK.CAMERA".
```

Нам также понадобится *Action* для приближения камеры к экрану и фокусирования на нём:

Go to TV с командами:

```
ACTION.START = "Show TV"
SYS.WAIT = "ACTION.Show TV"
:DATA.CURRENT.CAMERA = «Camera01»
:TRACK.CAMERA.GOTO = 120
SYS.WAIT = "TRACK.CAMERA".
```

Последний упоминаемый здесь *Action*, использующий именованные треки, служит для переключения на дистанционный план – *Action* с именем **Distant View** и телом из команды

```
:DATA.CURRENT.CAMERA = «Camera01»
:TRACK.CAMERA.START = 0.
```

Конечно, задание конкретных номеров в кадрах базируется на знании того, что происходит с объектом в эти моменты. Вместе с тем панель **Track** предоставляет достаточно информации, чтобы разобраться с треками объектов незнакомой сцены.

В самом деле, после именованя трека любого объекта с помощью команд вида TRACK.Name of Track.NODE = Name of Object в инициализирующем *Action*'е можно запустить его полностью. В качестве начального кадра задается нулевой, а в качестве последнего – кадр, который можно примерно определить с помощью линии **All Keys** панели **Track**. Например: TRACK.TV.START = 0, 180.

После этого, сужая диапазон кадров, становится возможным выделить отдельное нужное действие.

9.3.3 Работа с виртуальными камерами

В сцене имеется две камеры – **Camera01** и **Camera02**, управление которыми осуществляется с помощью пустых (*Dummy*) объектов **CAMERA1** и **CAMERA2**.

Camera01 имеет анимацию, которая используется для реализации некоторых действий, таких, как наезд камеры на актёра или появление экрана в предыдущем разделе. Виртуальная камера **Camera02** не имеет анимации и остаётся направленной на экран.

Для того, чтобы переключать виртуальные камеры в сцене, меняя область видимости, необходимо выполнить следующие команды (следует поместить их в отдельные *Action*'ы с именами **Camera1** и **Camera2**):

```
//устанавливает область видимости первой камеры
:DATA.CURRENT.CAMERA = «Camera01»;
//устанавливает область видимости второй камеры
:DATA.CURRENT.CAMERA = «Camera02».
```

9.3.4 Работа с видеофайлами

С помощью команд сценария можно написать *Action*, помещающий видеоизображение из видеофайла, например, на импровизированный экран. Это было сделано в инициализирующем *Action*'е (см. раздел 9.3.1), но могло быть вынесено в отдельный *Action*, с именем «Play DV»:

```
//создание файлового видеопотока MSDS_1:
:RENDER.VIDEO.MSDS_1.CREATE = 1
//определение режима воспроизведения видеофайла: со звуком (AUDIO), первым показывается
нижнее поле (LFF), включена трилинейная фильтрация для корректировки искажений показа
удалённого видеоизображения (MIPMAP)
:RENDER.VIDEO.MSDS_1.FORMAT= AUDIO, LFF, MIPMAP
//назначение источника данных для видеопотока MSDS_1:
:RENDER.VIDEO.MSDS_1.DATA = Biathlon_Bjordalen_LFF.avi
//старт видеопотока MSDS_1:
:RENDER.VIDEO.MSDS_1.START = 1
//ожидание, когда видеопоток MSDS_1 запустится
:SYS.WAIT = VIDEO.MSDS_1.START
//показ на материале телевизора видеопотока MSDS_1:
:RENDER.MATERIAL.monitor_m.SOURCE = MSDS_1.
//после того, как видеопоток (файловый или «живого» видео) создан и запущен, его можно
//назначить материалом любому объекту в сцене, например, видео с LIVE_1 поместить на
//экран:
:RENDER.MATERIAL.monitor_m.SOURCE = LIVE_1
//ожидание окончания проигрывания файла:
:SYS.WAIT = «VIDEO.MSDS_1.END»
//возвращение изначального материала экрану:
:RENDER.MATERIAL.monitor_m.SOURCE = 0.
```

Команды, выполняемые в данном *Action*'е, во многом аналогичны действиям по созданию видеопотока **LIVE_1** (см. раздел 9.3.1).

Чтобы материал объекта вернулся в своё исходное состояние после работы с видео, надо назначить ему источником несуществующий поток. Если поток с заданным именем не найден, то используется начальная карта. Например, вернём экрану его текстуру:

```
:RENDER.MATERIAL.monitor_m.SOURCE = 0.
```

Пример полноценной работы с видеофайлом, использующий вышеописанные *Action*'ы, приведён в разделе 9.3.6.

9.3.5 Работа с графическими файлами

Приложение *HotActions* обеспечивает с помощью команд сценария работу с графическими файлами.

Создадим в качестве примера *Action* (с именем «**Show Slide**»), помещающий изображение из JPG-файла на импровизированный экран. Подразумевается, что файл *Slide.jpg* существует в папке проекта:

```
:DATA.MATERIAL.monitor_m.MAP = «Slide.jpg».
```

Чтобы материал объекта вернулся в своё исходное состояние, достаточно выполнить следующую команду:

```
:DATA.MATERIAL.monitor_m.RESET = 1
```

При её исполнении файл *Slide.jpg* будет удалён также и из оперативной памяти видеокарты, что означает, что при следующем назначении этой текстуры на материал *monitor_m* будет задействован дополнительный системный ресурс для преобразования файла *Slide.jpg* во внутренний формат памяти видеокарты. Чтобы этого избежать, для возвращения материалу *monitor_m* первоначального изображения может быть использована команда

```
:DATA.MATERIAL.monitor_m.MAP = 0
```

При исполнении такой команды файл текстуры *Slide.jpg* не удаляется из оперативной памяти видеокарты и при следующем его назначении не потребуются его дополнительное преобразование.

В то же время не имеет особого смысла показывать слайд на экране, если экран отсутствует в сцене. Можно, конечно, выполнить ручную *Action* с именем «**Show TV**», а затем *Action* с именем «**Show Slide**», но есть более удобный способ, связанный с использованием возможности запуска одних *Action*'ов из других. Описанию работы со сценарными последовательностями посвящён следующий раздел.

9.3.6 Использование одних Action'ов в других

В предыдущем разделе описывалось создание *Action*'а («**Show Slide**»), помещающего изображение из JPG-файла на экран в сцене. Существует возможность создать более сложное действие, используя уже готовые *Action*'ы. Например, можно создать *Action* под именем «**TV Show Slide**», в котором экран появляется, показывает слайд в течение 5 секунд и удаляется обратно. В нужные моменты следует фокусировать камеру на происходящем. Поскольку все описанные действия уже имеются в виде готовых *Action*'ов, то можно использовать эти *Action*'ы.

Для того, чтобы все действия происходили последовательно, одно за другим, можно воспользоваться тем, что после завершения исполнения *Action*-а формируется сообщение о завершении «ACTION.Имя_Action'а». Эти сообщения появляются в отладочном окне приложения **Debug Output** (глава 6 Руководства пользователя *HotActions* и глава 18 Руководства пользователя по использованию команд сценария), правые части этих сообщений можно использовать в качестве параметров команды *SYS.WAIT* = «событие» для ожидания окончания исполнения *Action*'а:

```
//фокусируем камеру на том месте, где появится экран и выводим его:
```

```
:ACTION.START = «Go to TV»
```

```
//ожидание завершения исполнения Action'а, фокусирующего камеру:
```

```
:SYS.WAIT = ACTION.«Go to TV»
```

```
//помещаем изображение из JPG-файла на экран с помощью Action'а, созданного в предыдущем разделе:
```

```
:ACTION.START = «Show Slide»
```

```
//пятисекундная пауза:
```

```
:SYS.DELAY = 5.0
```

```
//запускаем Action, удаляющий экран из сцены:
```

```
:ACTION.START = «Hide TV»
```

//ожидание завершения *Action*'а, удаляющего экран из сцены:

:SYS.WAIT = ACTION.«Hide TV»

//снова фокусируем камеру на актёре:

:ACTION.START = «Go to Close Up»

:SYS.WAIT = ACTION.«Go to Close Up»

//возвращаем изначальный материал экрану:

:DATA.MATERIAL.monitor_m.MAP = 0

Для того, чтобы наш *Action* вместо слайда показывал содержимое определённого файла видео, достаточно создать *Action* с именем «**TV Show DV**», незначительно отличающийся от приведённого выше:

//фокусируем камеру на том месте, где появится экран и выводим его:

:ACTION.START = «Go to TV»

//ожидание завершения *Action*'а, фокусирующего камеру:

:SYS.WAIT = ACTION.«Go to TV»

//показываем видео из файла на экране с помощью *Action*'а, созданного в разделе 9.3.4:

:ACTION.START = «Play DV»

//ждемся завершения исполнения *Action*'а:

:SYS.WAIT = ACTION.«Play DV»

//запускаем *Action*, удаляющий экран из сцены:

:ACTION.START = «Hide TV»

//ожидание завершения исполнения *Action*'а, удаляющего экран из сцены:

:SYS.WAIT = ACTION.«Hide TV»

//снова фокусируем камеру на актёре:

:ACTION.START = «Go to Close Up»

:SYS.WAIT = ACTION.«Go to Close Up»

9.3.7 Создание кнопок для выполнения определённых *Action*'ов

Имея библиотеку готовых *Action*'ов, можно создать кнопочные панели, привязанные к *Hotset*'у, и непосредственно используемые в рабочем режиме в качестве основного интерфейса управления сценой.

С *Action*'ами в библиотеке можно обращаться как с обычными файлами в папке: копировать, удалять и так далее. Причём эти действия можно производить и над группой *Action*'ов, выбрав её стандартным способом – стрелками, удерживая зажатой клавиши **Shift** или **Ctrl**.

Выделите в библиотеке необходимую группу *Action*'ов и положите её в *Hotset*, при этом для них создастся новый *Hotbar*. Каждой кнопке *Hotbar*'а соответствует ссылка на определённый *Action*, по умолчанию их названия совпадают. На любой *Hotbar* можно добавлять кнопки, перетаскивая непосредственно на него *Action*'ы из библиотек *Action*'ов.

Таким образом, *Hotset*'ы – наборы *Hotbar*'ов. Они удобны тем, что все принадлежащие им *Hotbar*'ы можно прятать, показывать или перемещать одновременно.

В диалоге *Properties* для нового *Hotbar*'а можно поменять его имя и представление кнопок на нём, а в диалоге *Properties* для любой кнопки *Hotbar*'а можно установить или поменять иконку для неё. Подробнее работа с кнопками, *Hotbar*'ами и *Hotset*'ами описана в соответствующих главах руководства пользователя *HotActions*. Примерный результат работы приведён на Рис. 10.

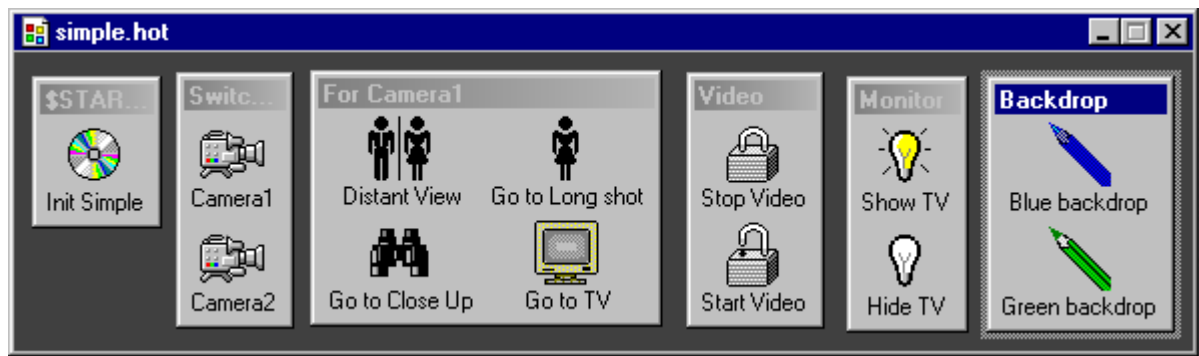


Рис. 10. Готовый *Hotset* и его *Hotbar'ы* – основные инструменты работы в приложении *HotActions*.

10 Экспорт сцен из 3D Studio MAX в файл *.3D

После окончания работы по созданию сцены необходимо экспортировать файл формата *3D Studio MAX* (с расширением *.MAX) в файл рабочего формата сцен (с расширением *.3D) для использования в приложении *HotActions*.

10.1 Установка Plug-in'a для 3D Studio MAX (MAYA)

Для преобразования формата сцены, созданной в *3D Studio MAX* в формат сцены для *HotActions*, требуется установить специальный модуль (plug-in) экспорта из *3D Studio MAX*, входящий в состав программного обеспечения виртуальной студии.

Если приложение *3D Studio MAX* установлено на компьютере, то установка plug-in'a для экспорта происходит автоматически при инсталляции программного обеспечения студии, в папку с соответствующими модулями приложения *3D Studio MAX* (как правило, это папка называется plugins). Также при инсталляции в директорию, в которую устанавливается приложение *HotActions*, создаётся папка *Exporters* с модулями для экспорта из графического редактора *3D Studio MAX* (поддерживаются версии с 4.0 по 3DSMax 2009) и различных версий *Maya*: **MayaExporter4.5**, **MayaExporter5.0** и так далее. Как правило, это директория **C:\Program Files\Focus Software\HotActions\Exporters**, если студия установлена на диск «C:» компьютера.

Если требуется установить модули для экспорта из графического редактора на другой компьютер, где установлен *3D Studio MAX* или *Maya*, то нужно скопировать файл **VSExporter.dlu** из папки *MaxExporter* или **VSExporter.mll** из папки с номером соответствующей версии *Maya* в папку для plug-in'ов *3D Studio MAX (MAYA)* этого компьютера. Альтернативным способом встраивания модуля является установка приложения *HotActionsDesign*.

Для проверки успешной установки plug-in'a откройте диалог *Plug-In Info* (Рис. 11) с помощью команды **Summary Info...** в меню **File** приложения *3D Studio MAX*.

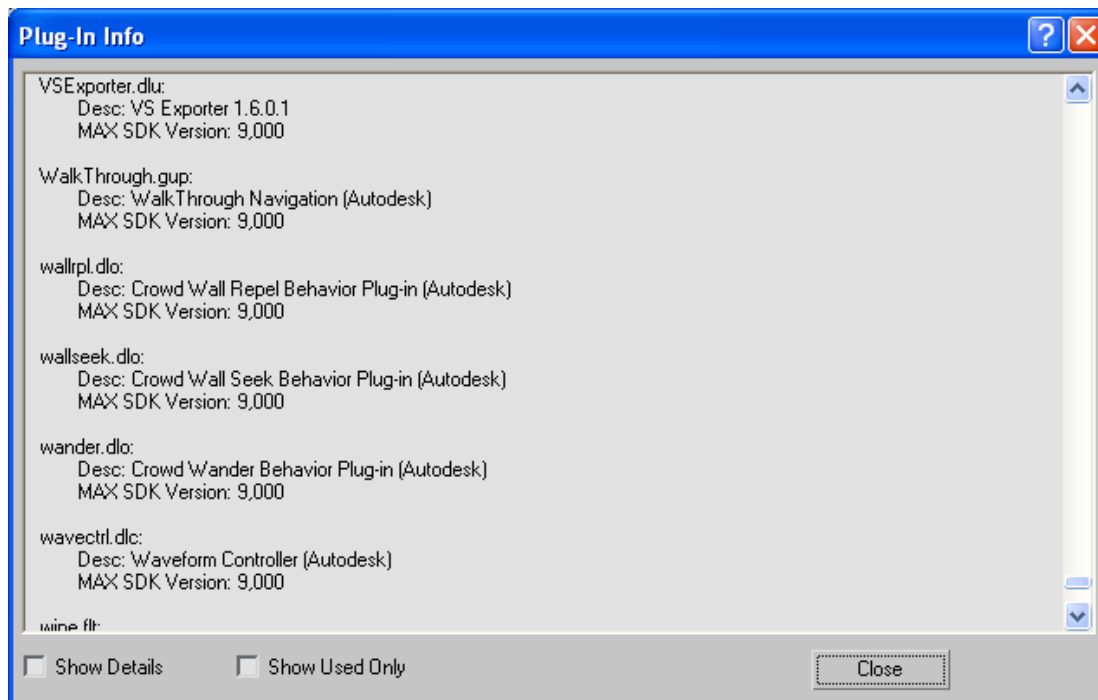


Рис. 11. Диалог *Plug-In Info* в приложении *3D Studio MAX*.

Если файла **VSExporter.dlu** в списке plug-in'ов нет, проверьте, правильно ли указан полный путь к директории, в которой эти файлы находятся. Для этого в меню **Customize** (Рис. 12) выберите команду **Configure Paths...**, а в появившемся диалоге *Configure Paths* закладку *Plug-Ins* (Рис. 13). Если там нет установки для *Additional 3D Studio MAX plug-ins*, то добавьте путь, щелкнув на кнопку **Add...**

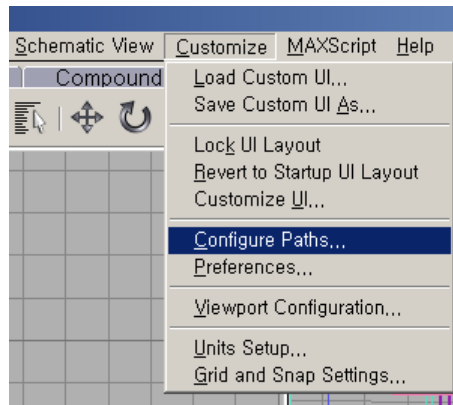


Рис. 12. Меню *Customize*.

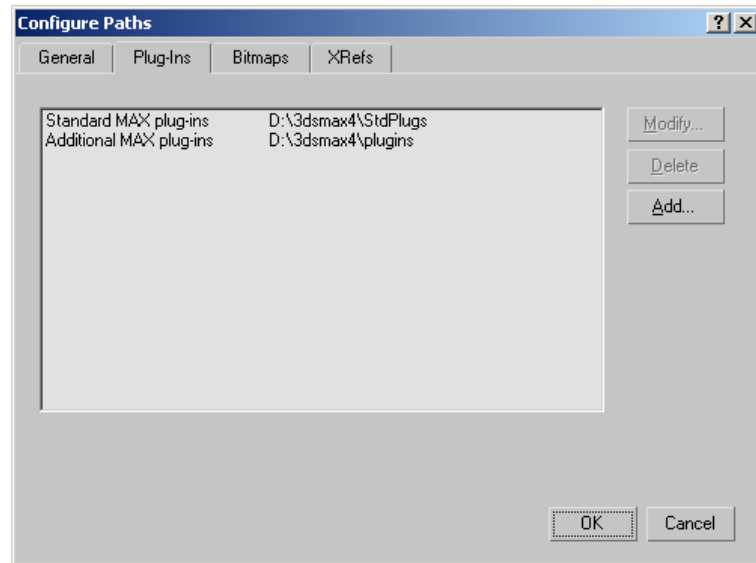


Рис. 13. Панель *Plug-Ins* в диалоге *Configure Paths*.

10.2 Стандартная утилита экспорта

Для экспорта сцены выберите команду **Export...** из меню **File**. Откроется диалог, как на рисунке ниже. Выбрав папку, введите имя файла в поле **Имя файла:**, нажмите **Сохранить**, чтобы завершить экспорт.

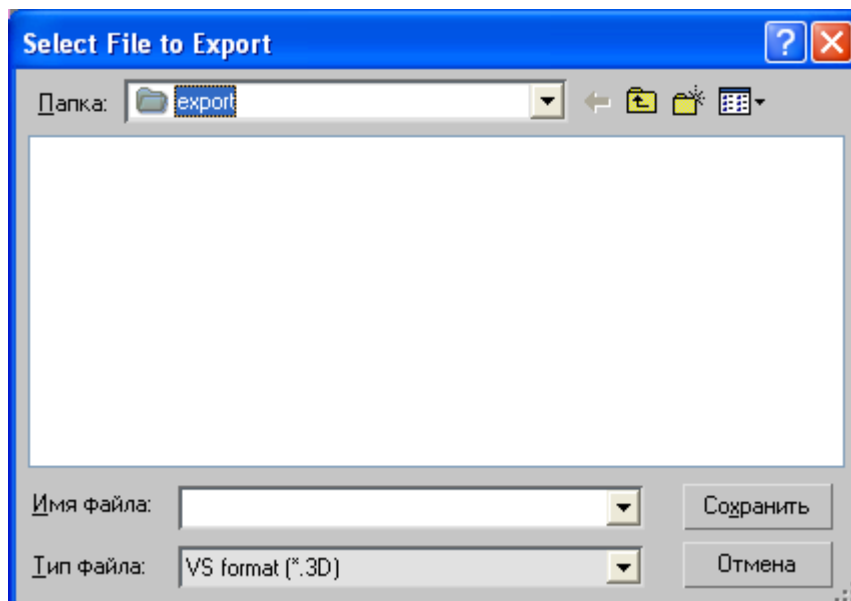



Рис. 14. Диалог экспорта файлов.

11 Использование VS Exporter

Использование *VS Exporter* (Рис. 15, справа) позволяет предварительно просматривать и сохранять сцену в 3D-формате без выхода из приложения *3D Studio MAX*.

11.1 Загрузка VS Exporter

Раскройте панель команд **Utilities**  и нажмите кнопку *VS Exporter* (Рис. 15, справа). Если такой кнопки нет, нажмите **More ...** и, выбрав *VS Exporter* в диалоге, нажмите **OK**. На панели команд **Utilities** появится кнопка приложения *VS Exporter* (Рис. 15, слева).

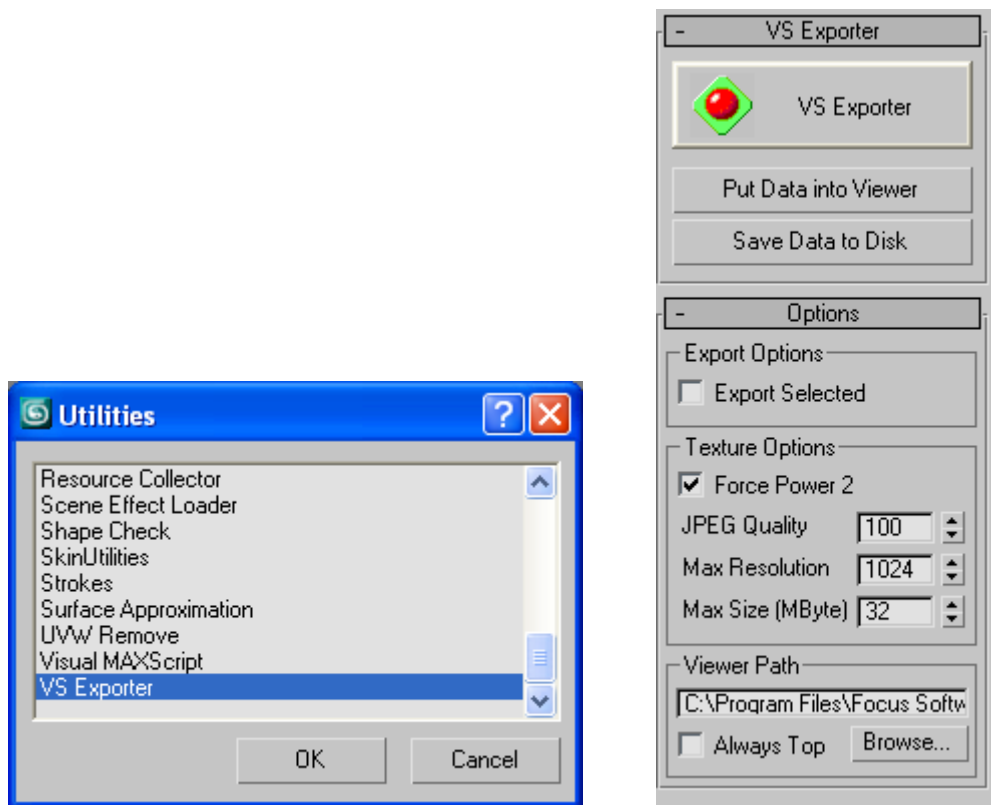


Рис. 15. Список дополнительных приложений 3Ds MAX и диалог *VS Exporter*.

В панели **Options** *VS Exporter* указываются параметры экспорта объектов (раздел *Export Options*), текстур (раздел *Texture Options*) и путь к программе *HotActions.exe* (раздел *Viewer Path*) для её запуска при предварительном просмотре сцены

Включение опции **Export Selected** раздела **Export Options** позволяет экспортировать не все объекты сцены, а только предварительно выделенные.

Опция **Force Power 2** определяет масштабирование всех текстур сцены при экспорте к размерам 2^N текселов, где N – целое число (см. раздел 2.1.10). Если опция будет выключена при экспорте, размеры текстур не будут преобразовываться и сохранятся назначенными в сцене 3DS MAX. В таком случае перемасштабирование текстур к размерам 2^N текселов будет происходить в дальнейшем каждый раз при загрузке сцены в приложение *HotActions*. Таким образом, включение опции **Force Power 2** позволяет осуществить процедуру перемасштабирования текстур один раз, за счёт чего загрузка сцены в приложение *HotActions* в дальнейшем будет осуществляться быстрее.

При экспорте можно указать качество текстур – параметр **JPEG Quality**, что также влияет на размер экспортируемого файла, определить максимально возможное разрешение текстур **Max Resolution**, что может быть актуально при экспорте видеофайла в качестве текстуры, или задать максимально допустимый размер текстуры **Max Size** (в мегабайтах).

Предварительный просмотр сцены в приложении *HotActions* перед экспортом описан в следующем разделе, но чтобы осуществить такой просмотр, необходимо, чтобы в поле *Viewer Path* был указан путь к *HotActions.exe*. Изменить путь можно в диалоге **Открыть**,

вызываемом при нажатии кнопки **Browse...** Установленная опция **Always Top** указывает на то, что рабочие окна приложения *HotActions* всегда будут занимать на рабочем столе положение поверх окон других приложений.

11.2 Просмотр сцены с помощью *VS Exporter*

Чтобы увидеть результаты последних изменений сцены в приложении *HotActions*, нужно нажать кнопку **Put Data into Viewer**. Это запустит *VS Viewer* с загруженными данными текущей сцены.

Если среди текстур сцены есть видеофайл, размер которого превышает значение максимально возможного размера текстуры **Max Size**, указанное в параметрах экспорта (Рис. 15), то как при просмотре сцены в *VS Viewer*, так и при экспорте (см. следующий раздел) появится сообщение с предложением уменьшить размер видеофайла, изменив размер кадра, либо сократив длительность видеофайла (количество кадров) (Рис. 16).

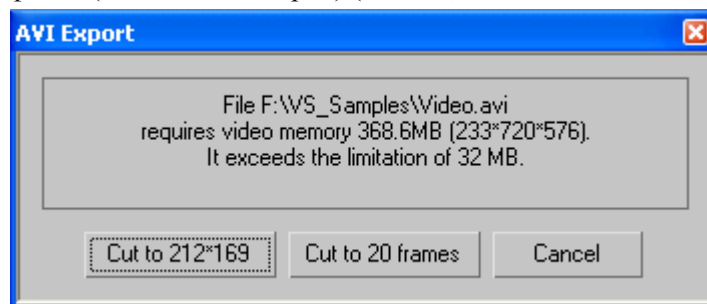


Рис. 16. Сообщение о превышении размера текстуры назначенного значения при экспорте сцены.

Экспорт сцены или подготовка к предпросмотру может занимать некоторое время, при этом количество экспортированных данных показано в нижней части приложения 3DS MAX (Рис. 17).

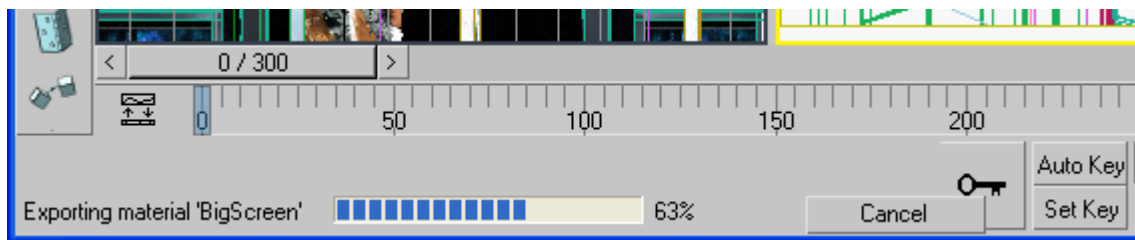


Рис. 17. Панель экспорта сцены из 3DS MAX в *HotActions*.

Есть некоторые ограничения на то, что может быть отображено в средстве просмотра сцены перед экспортом, – сцена не должна быть пуста и должна включать виртуальную камеру (Рис. 18).

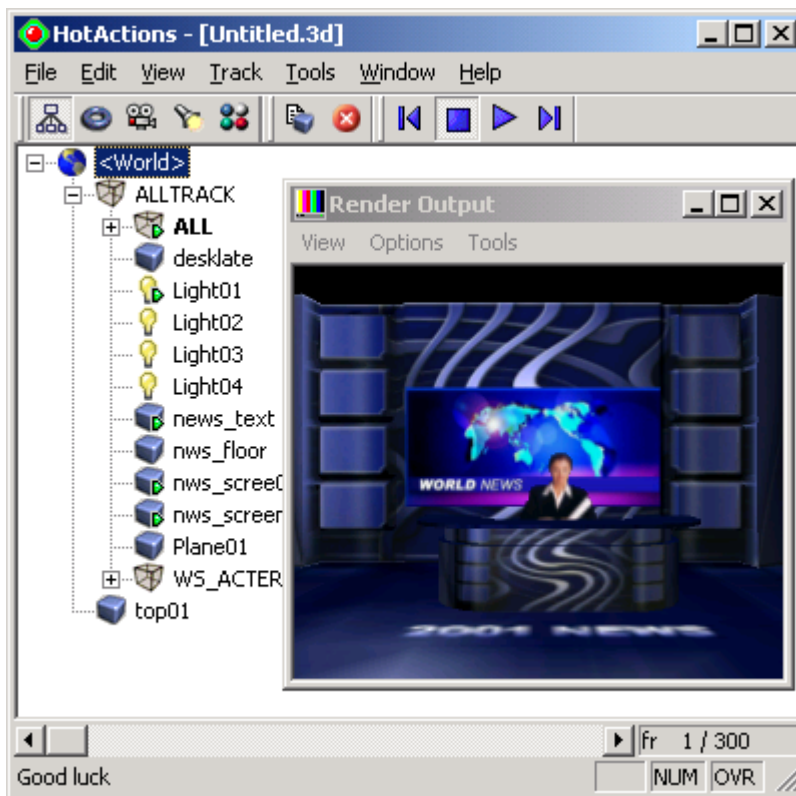




Рис. 18. Состояние окна приложения после запуска *VS Viewer*.

В *VS Viewer* составляющие сцены распределены по типам между соответствующими панелями.

Переключаться между этими панелями можно с помощью кнопок .

Кнопки  служат для воспроизведения и остановки общего трека сцены. Подробнее работа со сценой описана в разделе 3.4 главы 3 Руководства пользователя *HotActions*.

11.3 Сохранение сцены в формате *HotActions* с помощью *VS Exporter*

Для сохранения сцены выберите команду **Save as...** в меню **File** на инструментальной панели приложения *HotActions* (Рис. 18). Откроется поле диалога, позволяющее определить директорию для сохранения и имя файла. Сохранить сцену в файл с расширением *.3d без предварительного просмотра можно, нажав на кнопку **Save Data to Disk** на панели команд **Utilities** (Рис. 15).