

#7
2009

FPS

От редакции

Здравствуйте, уважаемые читатели!

На экране перед Вами — долгожданный очередной выпуск «FPS». Почти целый год журнал был, можно сказать, «в коме» — кто-то уже было решил, что его история закончилась. Но, я надеюсь, теперь все будет по-старому и регулярный выход «FPS» в скором времени наладится.

Хотелось бы поблагодарить читателей за теплые отзывы. Спасибо всем, кто не забывает нас и с нетерпением ждет каждого номера! Зная, что твоя работа кому-то нужна и что выпуски не уходят «в пустоту», намного легче надеяться на лучшее.

Многим, вероятно, будет интересен тот факт, что редакция журнала полностью переходит на открытый софт. Данный номер верстался уже не в Microsoft Office Publisher 2002, как раньше, а в OpenOffice.org 3.0. Вы можете оценить все преимущества данного пакета, бесплатно скачав его на официальном сайте: www.openoffice.org.

не забывайте заглядывать на наш сайт: xtreme3d.narod.ru и, разумеется, почаще писать на наш ящик: clocktower89@mail.ru.

Приятного чтения!

Gecko

В этом выпуске:

▶ **LOOXIS Faceworx**

3D-портрет из фотографии!.....3

▶ **Основы Blender**

Горячие клавиши.....6

▶ **Интервью с Dr. BEST**

Разработчик знаменитого Ultimate3D.....8

▶ **Rextro 3D**

Аркадные гонки от украинских игроделов.....12

▶ **OpenGL для начинающих**

Основы Basic4GL.....13

▶ **Битва гигантов**

Xtreme3D vs Ultimate3D.....17

▶ **Радио будущего**

Советский писатель предсказал Интернет?.....20

LOOXIS Faceworx

3D - портрет из фотографии!

Без сомнения, моделирование головы человека — наиболее сложное дело в 3D-графике. Если для всех остальных моделей мелкие погрешности не заметны (а если и заметны, всегда можно «прикрыться», мол, так и было задумано), то в отношении головы все куда жестче. Мелкие погрешности могут повлиять на сходство (если портрет делается с реального человека), а то и вовсе сделать лицо некрасивым и неестественным. Для тех, кто еще не моделировал голову, а в некоторых случаях и для более опытных моделлеров, желающих сэкономить время, существует бесплатная программа **LOOXIS Faceworx**.

Скачать ее можно на сайте LOOXIS: <http://www.looxis.com>. Размер 11,5 Мб.

Для работы в Faceworx Вам понадобится две фотографии головы — в фас и профиль (рис. 1). Профиль должен «смотреть» вправо. Желательно, чтобы голова была хорошо освещена (но без резких контрастных теней), прическа не должна скрывать лицо. Исходные фотографии будут использованы для автоматической генерации текстуры, поэтому стоит учесть этот факт при начальной обработке снимков.

Принцип моделирования следующий: в программе уже имеется готовая голова, и Вы должны подогнать ее под фотографию, изменяя расположение ключевых точек, контуров и плоскостей, соответствующих частям лица и деталям черепа. При этом все контуры представляют из себя кривые, автоматически меняющиеся при перетаскивании отдельных точек, поэтому редактировать придется далеко не все точки, формирующие полигональную сетку (и слава Богу!). Сразу оговорюсь, что, тем не менее, это не так просто, как кажется. Неплохо бы знать анатомию и строение черепа, а также иметь возможность рассмотреть моделируемую голову «в натуре». Можно сделать еще несколько снимков в разных ракурсах — не для Faceworx, а чисто для себя, чтобы время от времени сверяться.

Шаг 1

Загружаем исходные снимки. *Model > Load front...* для фаса и *Model > Load side...* для профиля (см. рис. 2). Да, негусто... :) Надо подогнать общий контур головы (в терминологии художников — овал лица). Для этого используется ограничивающая рамка (bounding box), правка которой является базовым уровнем редактирования модели — обозначена пунктирным прямоугольником. Растяните и переместите ее, как необходимо, потом сделайте то же для профиля (см. рис. 3). Теперь модель выглядит лучше, но все еще довольно убого.



Рис.1 Фотографии в фас и профиль



Рис. 2



Рис. 3

Шаг 2

Настало время редактировать элементы лица. Включите отображение элементов *View > Features* и отключите рамку *View > Box*. Видно, что глаза, рот, ноздри, уши все не там, все надо подгонять. Совет: Вы можете двигать как отдельные точки, так и кривые целиком (для этого наведите курсор на пунктирную линию). Линии, не имеющие точек это очертания основных лицевых мышц и соединений костей черепа. Особо внимательно отнеситесь к профилю подчас он более важен, чем фас.

Шаг 3

Необходимо подправить некоторые углы? Добавьте детализации: *View > Points > More*. Появятся новые точки, и Вы сможете добавить точности глазам, форме носа, ушей и т.д. Для сверхточной правки можно добавить еще точек (опять *View > Points > More*). Но я на данном этапе закончу (см. рис. 4).

Шаг 4

Экспортируем работу в файл: *File > Export...* Программа поддерживает экспорт в OBJ. Вы можете указать детализацию модели и качество текстуры. После экспорта получится очень высокополигональная модель (у меня - 3664 полигонов), что не есть хорошо, особенно если полученную голову надо посадить на туловище, а уж тем более, если использовать в игре (см. рис 5). Как уменьшить детализацию, т.е. оптимизировать модель? Можно воспользоваться популярным редактором **Milkshape**. Для этого загрузите модель в редактор и откройте *Tools > DirectX Mesh Tools...* и уменьшите количество полигонов при помощи ползунка (см. рис. 6). Я в два прохода уменьшил детализацию до 220 полигонов (см. рис. 7).

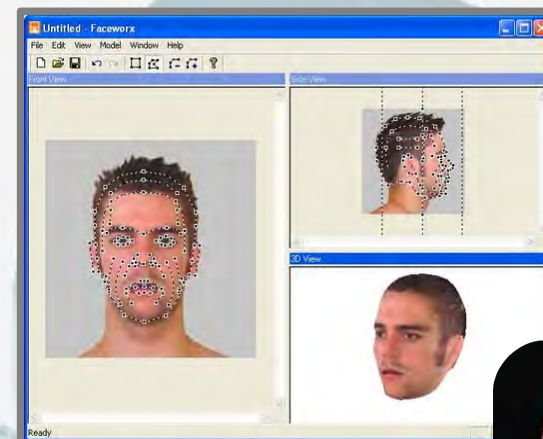


Рис. 4

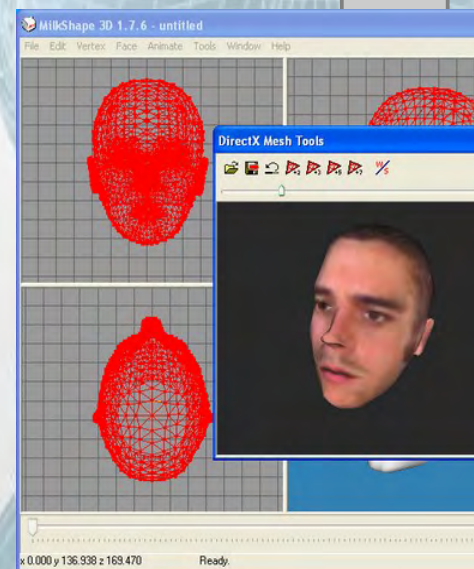


Рис. 6

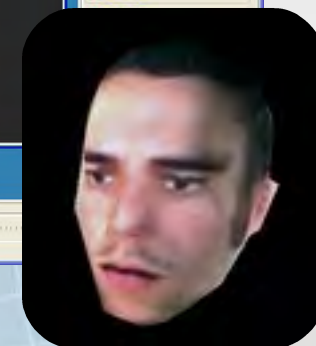
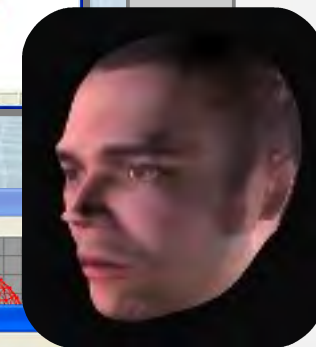


Рис. 7

Итог: результат, может, и далек от идеального, но тут уже дело в том, насколько точно Вы подгоняли модель при редактировании. Да, волос нет. Да, на затылке наблюдается некорректный UV-маппинг. Но волосы вручную смоделировать легче, чем голову, а вторая проблема решается при помощи UV-редактора. Текстуру можно подправить, убрать излишнюю «искусственность». Да и вообще, если подходить творчески, то можно не фотографировать, а рисовать исходные проекции. Тогда получатся более оригинальные и выразительные типажи. Как Вам, например, идея сделать гоблина или ведьму? Включайте воображение! Высокополигональные головы тоже можно пустить в дело, к примеру, при создании артворка, спрайтов, аватаров и т.д. Вариантов использования Faceworkx много, и данная утилита пригодится каждому. Тем более, что программа находится в активной разработке, и будущие версии, возможно, предложат пользователю более широкий арсенал функций.

Напоследок замечу: наши кичливые снобы-профессионалы, возможно, станут кричать, что использовать подобные программы - ламерство, что если не умеешь моделировать голову «нормальным» способом, то и не берись и т.п. Но я в своей жизни не сталкивался с каким-либо универсальным, «нормальным» способом моделирования вообще чего-либо, который подошел бы для каждого - да что говорить, такого способа и не может быть, так как каждый художник-моделлер ставит перед собой свои задачи, и, соответственно, изобретает свои собственные способы. В умелых руках любое программное средство станет невероятно мощным и функциональным, настоящий профессионал не лагает все подряд и во всем находит что-то полезное.

Gecko
clocktower89@mail.ru



В своих бесконечных блужданиях по просторам Интернета я обнаружил несколько интересных сайтов, которые могут заинтересовать читателей.

Полезные сайты

Seventh Sanctum

<http://www.seventhsanctum.com/>

Serendipity

<http://nine.frenchboys.net/>

Генераторы случайных имен, названий, описаний локаций и т.д. Очень полезно, когда нужно вдохновение.

CoolText

<http://cooltext.com/>

Генератор заголовков. Вы выбираете стиль, шрифт, фон, цвета и т.д., и система отрендерит Вам результат. Также на сайте имеется нехилая библиотека шрифтов.

FlamingText

<http://www.flamingtext.com/>

Аналогичный сайт

Hero Machine

<http://www.ugo.com/channels/comics/heroMachine2/>

Генератор персонажа-супергероя (основан на Flash). Вы выбираете все, включая одежду, символику, цвета и пр.

ОСНОВЫ Blender



<Продолжение. Начало в #6>

Горячие клавиши

Горячие клавиши в Blender в разы увеличивают скорость работы. Действия, на которые уходят несколько кликов мыши, можно ускорить до одного-единственного нажатия клавиши — разумеется, если знать, что нажимать.

[TAB] - переключение между режимом редактирования (редактирование вершин) и объектным режимом. Если Вы находитесь в режиме редактирования объекта и создаете новый объект - он будет объединен с этим объектом.

"O" (не ноль) - запускает режим пропорционального редактирования вершин, если Вы находитесь в режиме редактирования.

"A" - в режиме редактирования эта опция используется для выделения всех вершин, повторное нажатие этой кнопки приведет к снятию выделения со всех вершин.

"B" - Вызывает инструмент "Выделение прямоугольником", который используется для выделения нескольких объектов. В режиме редактирования с помощью этого инструмента можно выделять несколько вершин. Повторное нажатие клавиши **"B"** вызывает инструмент "Выделение окружностью", радиус которого вы можете изменять с помощью колесика мышки.

[Space Bar] - вызывает меню инструментов, в котором вы можете добавить меш-объект, камеру, лампу и т.д.

[NumPad] - Управление режимами просмотра: **7** - вид сверху, **1** - вид спереди, **3** - вид с боку, **0** - вид из камеры, **5** - перспектива, **."** - изменение фокусного расстояния, относительно выбранного объекта, **"+"** и **"-"** - изменение фокусного расстояния в большую и меньшую сторону соответственно.

Мышь - левая кнопка мыши предназначена для манипулирования, правая - для выделения, средняя - для вращения точки обзора и изменения фокусного расстояния. Если Вы будете одновременно удерживать **[Shift]** и среднюю кнопку мышки, сможете, двигая мышкой, передвигать проекцию экрана.

[Shift] - удерживайте, для активации мульти-выделения правой кнопкой мыши.

Стрелки - используются для установки кадров анимации. Правая стрелка увеличивает количество кадров на 1, левая - уменьшает на 1 кадр, стрелка вверх - увеличивает количество кадров на 10, а стрелка вниз - уменьшает на 10.

"R" - вращение выделенных объектов или вершин.

"S" - масштабирование выделенных объектов или вершин.

"G" - перемещение выделенных объектов или вершин.

"P" - в режиме редактирования отделяет выделенные вершины в отдельный меш-объект.

[Shift+D] - копирование выделенных объектов или вершин.

"E" - в режиме редактирования позволяет экструдировать (вытягивать) выделенные вершины.

"U" - В объектном режиме вызывает меню единого использования, вы можете в нем сделать уникальным Материал, Анимацию (IPO) и т.д. В режиме редактирования работает как отмена команды, так что вы можете вернуться на несколько шагов назад.

"M" - В объектном режиме перемещает выбранные объекты на другой слой. В режиме редактирования работает как инструмент зеркального отображения объекта.

"Z" - переключение каркасного отображения объекта в сплошное.

[Alt+Z] - переключение сплошного отображения объекта в светотеневое.

"P" - в объектном режиме запускает режим игры.

[**Alt/Ctrl**+**"P"**] - используется для того, чтобы создать (либо разорвать) связь "родитель-потомок". Для создания связи, удерживая [**Shift**] выберите объект, который будет "потомком", а затем объект, который будет "родителем", и нажмите [**Ctrl**+**"P"**]. Для разрыва связи, выполните ту же процедуру, но в конце нажмите [**Alt**+**P**].

"N" - вызывает информационное окно выделенного объекта. Данные в этом окне можно изменять.

[**Ctrl**+**J**] - служит для объединения выбранных объектов в одно целое.

[**Alt**+**A**] - проигрывает анимацию в выбранном окне. Ваш курсор должен находиться в этом окне для начала проигрывания.

"F" - в режиме редактирования создает полигон между выделенными вершинами. Для создания полигона допустимо выделение только трех или четырех вершин.

"W" - в режиме редактирования вызывает специальное меню. В объектном режиме вызывает меню булевых операций.

"X"/[Delete] - удаляет выбранный объект, вершину или полигон.

"I" - используется для вставки ключевого кадра анимации.

[**Alt**+**U**] - глобальная отмена команды. Нажав [**Alt**+**U**], Вы увидите список команд, которые можно отменить. По умолчанию количество шагов отмены - 32, но Вы можете изменить это значение в настройках.

[**Alt**+**C**] - используется для того, чтобы конвертировать меш-объекты, текст и кривые. Например, если Вы создали текст и хотите его преобразовать в меш-объект, с помощью [**Alt**+**C**] вы сможете превратить его в кривую, а повторное нажатие [**Alt**+**C**] превратит кривую в меш-объект.

[**Shift**+**Space**] - разворачивает активное окно на весь экран.

[**Ctrl**+**0**(ноль)] - очень удобная команда, если Вы используете в Вашей сцене несколько камер. Выделите одну из камер, а затем нажмите [**Ctrl**+**0**], и эта камера станет главной.

Функциональные клавиши:

F1 - загрузка файла; **F2** - сохранение файла; **F3** - сохранение картинки; **F4** - кнопки редактирования ламп; **F5** - кнопки редактирования материалов; **F6** - кнопки редактирования текстур; **F7** - кнопки анимации; **F8** - кнопки игрового движка; **F9** - кнопки редактирования; **F10** - кнопки дисплея; **F11** - последний рендер; **F12** - запуск рендеринга.

<продолжение следует...>

Джеймс КРОНИСТЕР, «Основы *Blender*»
В переводе **Юлии КОРБУТ**

Вы разрабатываете перспективный проект? Открыли интересный сайт? Хотите «раскрутить» свою команду или студию? Мы Вам поможем!

Спецпредложение!

«FPS» предлагает уникальную возможность: совершенно БЕСПЛАТНО разместить на страницах журнала рекламу Вашего проекта!! При этом от Вас требуется минимум:

- **Соответствие рекламируемого общей тематике журнала.** Это может быть игра, программное обеспечение для разработчиков, какой-либо движок и/или SDK, а также любой другой ресурс в рамках игростроя (включая сайты по программированию, графике, звуку и т.д.). Заявки, не отвечающие этому требованию, рассматриваться не будут

- **Готовый баннер или рекламный лист.** Для баннеров приемлемое разрешение: 800x200 (формат JPG, сжатие 100%, Progressive). Для рекламных листов: 1000x700 (формат JPG, сжатие 90%, Progressive). Содержание — произвольное, но не выходящее за рамки общепринятого и соответствующее грамматическим нормам. Совет: к созданию рекламного листа рекомендуем отнестись ответственно. Если не можете сами качественно оформить рекламу, найдите подходящего художника. !! «Голый» текст без графики и оформления не принимается.

- Краткое описание Вашего проекта и - обязательно - **ссылка на соответствующий сайт** (рекламу без ссылки не публикуем).

- Заявки со включенными **дополнительными материалами для журнала** (статьи, обзоры и т.д.) не только приветствуются, но даже более приоритетны.

Заявки на рекламу принимаются на почтовый ящик редакции: clocktower89@mail.ru (тема: «Сотрудничество с FPS», а не просто «Реклама», так как ее может отсеять спам-фильтр). Прикрепленные материалы (рекламный лист, информация, статьи и пр.) могут быть как прикрепленными, так и загруженными на какой-либо надежный сервер или файлохранилище (но не RapidShare или DepositFiles!). Все материалы желательно архивировать в формате ZIP, RAR или 7Z. Вас интересует, зачем мы это предлагаем? Да еще и бесплатно, хотя могли бы заработать? Мы это делаем из дружеских побуждений, чтобы в этом мире царили не только деньги, но и профессиональная солидарность, честь и достоинство. Если реклама в нашем журнале может кому-то реально помочь, почему бы ее не опубликовать? А деньги — чем меньше с ними имеешь дело, тем полезнее для здоровья...

Я думаю, нет такого пользователя Game Maker, который не слышал бы о популярном движке Ultimate3D. Поэтому не буду распространяться о всех его достоинствах, они всем известны и без меня. А вот что действительно может заинтересовать читателей: как сумел немецкий школьник в одиночку написать движок профессионального уровня? Кто он такой, легендарный Dr. BEST, создатель Ultimate3D?

FPS: Многие знают вас как создателя Ultimate3D. Расскажите о себе. Когда вы начали работать с GM? Каковы были ваши цели?

Dr. Best: Мое настоящее имя - Кристофф Петерс, мне семнадцать лет (к моменту взятия интервью - прим. ред.) и живу я в немецком городе Кологне. Ник Dr. Best произошел от названия немецкого производителя зубных щеток =) Я открыл для себя Game Maker в двенадцать лет, и мне он сразу понравился. Поначалу я сделал несколько клонов Invaders, Pong и Moorhen Jagged. GML стал моим первым языком программирования. В это время был выпущен Game Maker 5.0, который еще не располагал 3D-функциональностью. Я узнал про Gadget3D - технологию создания 3D-графики из двумерных спрайтов (вроде как в Doom). Меня поразила сама мысль о том, что вся 3D-графика - не что иное, как проекция на 2D-плоскость, и я сильно заинтересовался. Я начал модифицировать и расширять Gadget3D, и достиг в этом определенных результатов, но картинка все равно смотрелась ужасно - по сравнению с любой коммерческой 3D-игрой. По этой причине я решил изучить C++. После пяти месяцев изучения, я начал работать над Ultimate3D.

Dr. BEST: «Это сложнейшая работа, поэтому она мне и нравится»

Мой изначальный план состоял в создании чего-то, что сделает разработку несложных 3D-игр такой же простой, как и 2D. На тот момент единственным способом добиться хорошей 3D-графики была библиотека Xtreme3D, которая не пользовалась большой популярностью вследствие сложности в использовании. Вероятно, поэтому первая версия Ultimate3D, выпущенная в октябре 2004 г., получила множество положительных отзывов. Это побудило меня продолжить разработку. Ultimate3D стал моим крупнейшим (и, исключая различные мелкие или незавершенные работы, единственным) проектом. И, разумеется, я многому научился, создавая его и читая различную дополнительную литературу. После выхода Ultimate3D 1.31, разработка движка была остановлена на пару месяцев, так как я увлекся созданием игры (которая так и не была завершена). Разработка этой игры открыла мне глаза на некоторую ограниченность Ultimate3D, и мне тогда пришлось многое изучить. Поэтому я начал новую версию движка, где многое решил принципиально изменить. В конце-концов, новая версия, получившая номер 2.0, была переписана полностью, так что предыдущие и рядом не стояли - настолько все стало сложнее и мощнее. Разработка заняла у меня два года. Ultimate3D для меня очень важен. Это моя крупнейшая работа, к которой так или иначе причастны сотни пользователей. Не говоря уже о том, что движок стал хорошей основой для других моих проектов. Моя цель сегодня - сделать Ultimate3D более завершенным продуктом, законченным решением для разработки трехмерных игр. И, несомненно, мне предстоит еще долгий путь к достижению этой цели.



FPS: *Так как же начинался Ultimate3D? И насколько сильно он изменился со временем?*

Dr. Best: Ultimate3D начался с мечты - мечты дать людям возможность создавать хорошие 3D-игры в Game Maker. С выходом Game Maker 6.0 эта изначальная цель несколько изменилась, так как базовая трехмерная графика была уже встроена в этой версии. Несмотря на простоту и скорость, Ultimate3D и Xtreme3D перестали быть единственным решением. Так или иначе, я просто продолжил разработку. У U3D были свои фанаты, появилось несколько многообещающих проектов, которые действительно были готовы использовать любые средства, которые я предлагал. Со временем ранних версий движков сильно изменился, с Ultimate3D 2.0 их не сравнить. Почти все возможности старых версий теперь составляют лишь раздел для начинающих - а это только вершина айсберга. Ultimate3D идет к тому, чтобы стать движком профессионального качества. Надо сказать, далеко не вся его мощь может быть реализована в Game Maker, но, если не хотите бросать работу в этой программе, Ultimate3D - лучшее, что вы можете использовать.

FPS: *Сумели ли вы уложиться в свои планы с выходом Ultimate3D 2.0?*

Dr. Best: Финальный продукт стал тем, чем должен был стать, и я им полностью удовлетворен. Где-то он даже лучше, чем я предполагал. Появились несколько особенностей, которые не были запланированы вначале. Внутренняя архитектура была полностью переделана и стала намного более стабильной и гибкой. Список возможностей движка становился все длиннее и длиннее по мере того, как я над ним работал. За четыре месяца разработки все возможности, которые я изначально хотел задействовать, были задействованы. С возрастающей сложностью продукта, его старая архитектура становилась все менее надежной и практичной. Поэтому я начал переписывать одну часть за другой, и в результате переписал все. По этой причине разработка затянулась на целый год дольше, чем ожидалось. Но оно того стоило.

FPS: *Сколько же труда было вложено в U3D? Вы можете назвать количество написанных строк кода?*

Dr. Best: Много =) Ultimate3D 2.0, который, как я уже сказал, был полностью переписан, оставался в разработке около полутора лет до выхода первой беты, и потребовал еще полгода до выхода кандидата в релизы, включавшего несколько важных нововведений. Разработка Ultimate3D - мое хобби, я занимаюсь ей в свободное время, которого, впрочем, было затрачено немало. Исходники Ultimate3D 2.0 final включают около 28000 строк и занимают 1 Мб чистого кода.

FPS: *Насколько сложной была работа по U3D 2.0? Можете ли вы сказать, что пользовательская статистика поддержала вас?*

Dr. Best: Да если это не было сложно, стал бы я этим заниматься? =)) Это самая сложная работа, поэтому она мне и нравится. Количество вещей, которые мне пришлось изучить, просто огромно. Тот факт, что конечный продукт используется множеством людей, стал настоящей побудительной силой и придает моей работе некоторую степень важности. И, что еще более важно, в разработку включилось Сообщество Ultimate3D. Я получаю немало полезных предложений и сообщений об ошибках. Очень здорово иметь такое место, где все используют твой продукт =)



FPS: Какая ваша любимая игра на основе вашего движка? И почему?

Dr. Best: Вообще, завершённых игр на основе Ultimate3D не так уж много, и все по большей части очень маленькие. Но одна мне особенно нравится: это Ruins. Автор игры - skarik - настоящий маньяк спецэффектов. У нас с ним есть что-то общее =). Ещё есть Power Gates за авторством gameDeveloper. Если бы не многочисленные баги, я бы мог назвать её своей любимой игрой на основе Ultimate3D. Самая большая и известная - это Edge of Infinity (ранее известная как Project Cynthillia) за авторством Despellanion. Если этот проект выживет и реализует свой потенциал, это будет нечто. Время покажет...

FPS: Достиг ли Ultimate3D своей цели? Можем ли мы ожидать новых возможностей в будущем? Если да, то каковы ваши планы в этом направлении?

Dr. Best: Ultimate3D 2.0 стал тем, чем должен был, так что, можно сказать, определённо, достиг. Но в целом, U3D не закончен и нуждается в дальнейшей разработке. Я долгое время работал, не получая никаких отзывов, и теперь оказалось, что многое ещё предстоит сделать, исправить. Наиболее важным нововведением будет совершенно новая система обнаружения столкновений и управление ею. Она будет основана на сферах вместо лучей, что более надёжно и правдоподобно, и будет встроенной; это упростит её использование и поднимет скорость обработки, так как потребуются гораздо меньше кода GML. Также планируется множество небольших улучшений, которые принесут новые возможности: например, будет возможность применять фоновые изображения с шейдерными эффектами.

FPS: Вы все ещё остаетесь верны Game Maker, или имеете виды на что-то ещё?

Dr. Best: Я не считаю себя пользователем Game Maker. Я не использую его на протяжении нескольких лет, и мне было достаточно трудно снова начать работать с ним для создания технологической демы. Все, что я делаю в GM, это интерфейсные скрипты для Ultimate3D. Вы, скорее всего, никогда больше не увидите ничего, сделанного мною на Game Maker, что не связано с Ultimate3D. Если мне захочется написать игру, я возьму C++ и Ultimate3D - разумеется, не Ultimate3D.dll с интерфейсом для GM, а другую библиотеку, которую я не планирую выпускать. Но, в любом случае, вы можете рассчитывать, что я продолжу писать интерфейсы Ultimate3D для Game Maker, так что обязательно будут выходить новые версии движка.

FPS: Как насчёт статистики по скачиванием? Какая версия стала более популярна?

Dr. Best: Да я не веду особо точной статистики. Но, мне кажется, вторая версия превзошла первую. Первая бета Ultimate3D 2.0 вышла довольно давно, и, пожалуй, с тех пор 2.0 стала более популярна, чем все старые версии. С тех пор, как я открыл новый сайт движка в мае 2007 г., стартовая страница была посещена более 43000 раз (к моменту взятия интервью - прим. ред.). Разница более ощутима в Сообществе Ultimate3D. В течение 20 месяцев до релиза было отправлено 2305 сообщений, в течение 14 месяцев после релиза - 12000 сообщений.

FPS: С каким количеством багов пришлось столкнуться при разработке? Какой был самый неприятный?

Dr. Best: Во время работы неизбежно возникают ошибки, так что багов не избежать. Достаточно отвлечься на мгновение - и появится какая-нибудь мелочь, незаметная на первый взгляд. Количество багов, которые я исправил после выхода первой беты Ultimate3D 2.0, достигает двухсот. Наиболее значимые, приводящие к критическим ошибкам, обычно исправляются почти сразу, а те, что поменьше - проходит не меньше месяца, пока с ними кто-нибудь столкнется и сообщит об этом. Если мне о них сообщают, исправление багов представляется довольно легким, так что, на самом деле, я не могу сказать, какой из них был наиболее неприятным.

FPS: Что бы вы могли посоветовать пользователям Ultimate3D?

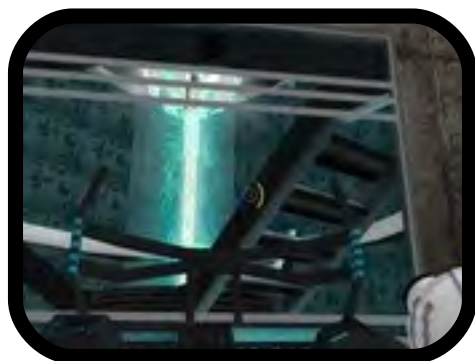
Dr. Best: Много чего. У меня есть множество разных хитростей и советов, и большинство из них можно встретить в документации к Ultimate3D, доступном в качестве CHM-файла, его онлайн-версии, а также Википедии Ultimate3D. Трудно посоветовать что-то конкретное, ведь не существует таких трех супер-правил, которые гарантированно сделали бы любой проект успешным. Есть много вещей, которые нужно знать, и я сделал все возможное, чтобы собрать их все вместе. Например, если вам требуется повысить производительность графики, советы, касающиеся этого, можно найти в соответствующей главе документации.

Единственное, что я хотел бы здесь сказать: не ожидайте слишком многого. В Ultimate3D можно добиться отличных спецэффектов и эффективно управлять большими объемами геометрии, но существует две причины, по которым вы вряд ли сможете добиться графики уровня коммерческих игр, даже не самых новых. Во-первых, Ultimate3D предоставляет далеко не все, что для этого требуется. Во-вторых, Game Maker - это инструмент для создания простых игр, не обеспечивающий достаточную надежность и производительность, необходимых для разработки крупных проектов. Да он никогда и не был ориентирован на это. Так что будьте креативнее, не пытайтесь, как говориться, "прыгнуть выше головы" =)

FPS: Огромное спасибо за интервью! Что вы можете сказать читателям на прощание?

Dr. Best: Что ж, Ultimate3D проделал длинный путь. Разработка версии 2.0 была серьезным шагом вперед, но этот путь еще не закончен. В последнее время у меня было мало времени на работу по движку, но теперь я могу ее продолжить. Запаситесь терпением и ждите обновлений!

Оригинал интервью принадлежит журналу [Game Maker Tech](http://gamemakertech.info)
<http://gamemakertech.info>
Перевод: Gecko



www.Ultimate3D.org





"Добро пожаловать в уникальный мир Rextro 3D - трехмерной игры, которая впечатлит Вас оригинальностью и веселой атмосферой! Цель проста: уничтожить как можно больше оппонентов, набирая очки. На первый взгляд, ничего особенного, но... Персонажи - забавные пилоты маленьких машинок - великолепно озвучены. Вы не поверите, но стреляют не из бластеров и даже не файрболлами, а... овцами! Игра наполнена юмором. Скачайте и поднимите себе настроение!"

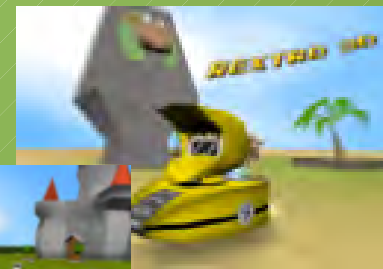
Таким вот образом описывается данная игра на известном портале gmarcade.com. Rextro 3D - весьма качественное изделие из братской Украины. Сразу вспоминаются незабвенные MicroMachines, Crash Team Racing... Но обо всем по порядку.

Сюжета, во всяком случае в последней виденной мною бете, нет как такового. Да и нужен ли сюжет аркаде, спросил бы я. Сев за руль одной из разноцветных машинок, что игра предлагает нам выбрать после запуска игры, вы вряд ли почувствуете себя профессиональным водителем, тем более, что трасс в игре как таковых тоже нет. Есть две арены, по которым можно свободно перемещаться - в Египте и подворье средневекового замка.

В игровой вселенной Rextro3D, похоже, страдают от переизбытка мелкого рогатого скота. Иначе как объяснить овец в качестве оружия? Но, вообще, "овечья идеология" присуща многим играм - и как только над бедными барашками не издеваются, вот теперь заставили и кидаться ими в оппонентов. Ну да ладно, оставим это на совести поставщиков некачественного "источника вдохновения" (ну, вы поняли, о чем я). Шутка =)

Правую руку на мышшь, левую - в область WASD, и вперед! Противники обладают завидным искусственным интеллектом (который, кстати, можно отрегулировать в игровых опциях), всюю обстреливая вас блестящими снарядами. Кстати, овцы имеют талант здорово рикошетить от препятствий, поэтому стоит смотреть внимательнее, куда стреляете - не попасть бы в самого себя =) В сами препятствия врезаться тоже не советую, чревато. К счастью, на аренах попадаются power-up'ы для ремонта.

Название: Rextro3D
Разработчик: dmitko
Жанр: аркада
Год выпуска: 2008



О графике. Трехмерные игры в GM - давно не редкость, и мало кто из доблестных игроделов не попробовал свои силы в создании 3D-игры стандартными средствами конструктора. Многие принципиально не используют dll-библиотеки типа Xtreme3D или Ultimate3D, предпочитая ставшие уже традиционными функции d3d_. Хорошо это или плохо - спорный вопрос, но во всяком случае, пока одни ведут бесконечные дебаты на форумах, другие занимаются делом, и это достойно уважения. А уважать есть за что. Графика в Rextro3D далеко не самая худшая, наоборот: вас ждут красочные декорации, забавные персонажи и качественные спецэффекты.

Звук в игре достоин всяческих похвал. Нечасто в GM-проекте встретишь настоящую актерскую озвучку, да и музыка не подкачала.

В общем, несколько часов хорошего настроения вам действительно обеспечено. С точки зрения геймдизайна - неплохой, где-то, может, недопиленный, но в целом вполне качественный казуальный продукт. Зря мы ругаем украинцев, хорошие они ребята =)

Графика: 7/10
Звук: 8/10
Геймплей: 7/10
Итого: 7,3/10

Gecko
clocktower89@mail.ru



OpenGL

для начинающих

Часть I

Так как же писать программы для OpenGL? Неужели придется ставить громоздкие компиляторы, изучать сложные языки программирования и разбираться с кучей библиотек, только для того, чтобы элементарно загрузить текстуру? Да, конечно :) Но в целях обучения я предлагаю более легкий путь.

Скачав и установив бесплатную среду Basic4GL с сайта <http://www.basic4gl.net>, вы сможете легко и быстро программировать под OpenGL на родном и знакомом языке Basic, безо всякой рутинной работы с подготовкой окна и средств ввода-вывода. Можно даже создавать небольшие игры, ведь Basic4GL поддерживает сборку в *.exe! Помимо встроенной поддержки OpenGL, этот язык обладает всеми необходимыми инструментами. Здесь поддерживаются многомерные массивы, вектора, матрицы, пользовательские структуры данных и многое другое. На мой взгляд, Basic4GL гораздо мощнее GML, хоть и тоже является интерпретируемым. Basic4GL компилирует программы в собственный байт-код, благодаря чему достигается высокая скорость исполнения приложений, необходимая для работы с низкоуровневыми командами OpenGL.

За годы работы в конструкторах игр (главным образом, в Game Maker) у вас наверняка возник вопрос: а как вообще пишутся профессиональные 3D-игры? Ведь не на гейм мейкере же! Я, конечно, пока не собираюсь отказываться от любимого конструктора и пытаюсь написать новый Doom 3, да и вам не советую. Но попробовать освоить азы не помешает.



Будем считать, что вы все установили и готовы начать писать свою Первую Программу на OpenGL. Но сначала я вас удивлю. Помните встроенный 3D-движок Game Maker (сокращенно GM d3d)? Там надо было вручную рисовать каждую вершину модели:

```
d3d_primitive_begin(pr_trianglelist);
    d3d_vertex(1,1,0);
    d3d_vertex(1,2,0);
    d3d_vertex(1.5,1.5,2);
    d3d_vertex(1,2,0);
    d3d_vertex(2,2,0);
    d3d_vertex(1.5,1.5,2);
    d3d_vertex(2,2,0);
    d3d_vertex(1,1,0);
    d3d_vertex(1.5,1.5,2);
    d3d_vertex(1,1,0);
    d3d_vertex(1,2,0);
    d3d_vertex(2,2,0);
d3d_primitive_end();
```

На OpenGL то же самое выглядит так:

```
glBegin(GL_TRIANGLES)
    glVertex3f(1.0,1.0,0.0)
    glVertex3f(1.0,2.0,0.0)
    glVertex3f(1.5,1.5,2.0)
    glVertex3f(1.0,2.0,0.0)
    glVertex3f(2.0,2.0,0.0)
    glVertex3f(1.5,1.5,2.0)
    glVertex3f(2.0,2.0,0.0)
    glVertex3f(1.0,1.0,0.0)
    glVertex3f(1.5,1.5,2.0)
    glVertex3f(1.0,1.0,0.0)
    glVertex3f(1.0,2.0,0.0)
    glVertex3f(2.0,2.0,0.0)
glEnd()
```

Немного истории...

Стандарт OpenGL (Open Graphics Library) был разработан в 1992 г. компанией Silicon Graphics Inc. на основе библиотеки IRIS GL как открытая платформонезависимая спецификация, определяющая программный интерфейс для написания приложений, использующих двумерную и трехмерную компьютерную графику.

Развитие OpenGL определяется консорциумом компаний, поддерживающих этот стандарт. В консорциум, получивший название ARB (Architecture Review Board) сегодня входят Intel, AMD, Apple, nVidia, id Software, Sony, Sun, Creative Labs и др.

На сегодняшний день OpenGL имеет широчайшее распространение и является главным конкурентом DirectX

Найдите 10 отличий! :)) Оказывается, принцип везде одинаковый. А если вы с GM d3d не знакомы, подскажу: и Direct3D, и OpenGL главным образом умеют рисовать только многоугольники - полигоны. Полигоны могут состоять из трех или четырех вершин. Трехвершинные полигоны называются треугольниками (кто бы мог подумать?), а четырехвершинные - квады. Модель, составленная из полигонов, называется примитив. Перед тем, как рисовать примитив, надо задать трансформацию. Сначала мы задаем перемещение, потом поворот. Если сделать наоборот, повернется не сам примитив, а система координат, в которой перемещен примитив. Получится, как будто земля вокруг солнца. Это тоже иногда нужно. Вспомните систему родителей/потомков Xtreme3D - она уходит корнями именно в это свойство OpenGL.

Вот вам простейшая OpenGL-программа для Basic4GL, запустив которую, вы увидите белый полигон-треугольник:

```
while true
  glClearColor(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT)
  glLoadIdentity()
  glTranslatef(0, 0, -5)
  glBegin(GL_TRIANGLES)
    glVertex3f(-1,-1, 0)
    glVertex3f( 1,-1, 0)
    glVertex3f( 0, 1, 0)
  glEnd()
  SwapBuffers()
wend
```

Что мы имеем? Цикл while-wend, в котором происходит вызов команд OpenGL:

`glClearColor(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT)`
Очищает буферы цвета и глубины. Буфер цвета содержит набор пикселей, задающих цвет в каждой точке области рисования OpenGL. Буфер глубины - это тоже набор пикселей, указывающих, насколько далеко точки в буфере цвета расположены от наблюдателя. Благодаря буферу глубины (его еще называют z-буфером) OpenGL всегда знает, в каком порядке полигоны перекрывают друг друга. А очищать эти буферы необходимо, потому что в противном случае данные о цвете и глубине будут просто наслаиваться, создавая полный венецет.

`glLoadIdentity()`
Функция, аналогичная `d3d_transform_set_identity()` в GML. Сбрасывает текущее состояние трансформации на значения по умолчанию. Если этого не сделать, при повторении цикла вся трансформация будет совершена относительно текущей.

`glTranslatef(0, 0, -5)`
Функция, аналогичная `d3d_transform_set_translation(x,y,z)` в GML. Совещает перемещение (трансляцию) системы координат в заданный вектор. В нашем случае мы переместились на -5 по оси Z. Это значит, что все следующие вектора и координаты будут задаваться относительно точки (0,0,-5).

```
glBegin(GL_TRIANGLES)
  glVertex3f(-1,-1, 0)
  glVertex3f( 1,-1, 0)
  glVertex3f( 0, 1, 0)
glEnd()
```

На GML это выглядело бы так:

```
d3d_primitive_begin(pr_trianglelist);
  d3d_vertex(-1,0,-1);
  d3d_vertex( 1,0,-1);
  d3d_vertex( 0,0, 1);
d3d_primitive_end();
```

Мы нарисовали примитив - полигон, заданный тремя точками. Если посмотрите внимательно, то заметите отличия между координатными системами OpenGL и Direct3D. Ось Z в Direct3D соответствует оси Y в OpenGL, и наоборот.

`SwapBuffers()`
Эта функция копирует значения из вторичного буфера в первичный. Во вторичном буфере (back buffer) происходит вычисление картинки, а первичный буфер выводится на экран.

Белый треугольник выглядит скучновато, поэтому попробуем его раскрасить:

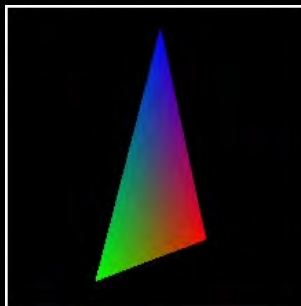
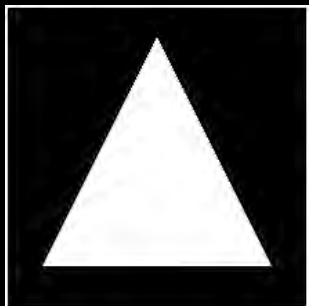
```
while true
    glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    glTranslatef(0, 0, -5)
    glBegin(GL_TRIANGLES)
        glColor3f(1,0,0): glVertex3f(-1,-1, 0)
        glColor3f(0,1,0): glVertex3f( 1,-1, 0)
        glColor3f(0,0,1): glVertex3f( 0, 1, 0)
    glEnd()
    SwapBuffers()
wend
```

`glColor3f(R,G,B)`

Эта функция задает текущий цвет, которым должны быть раскрашены отрисованные далее вершины. Хотя сама по себе вершина не имеет цвета, ведь это всего лишь воображаемая геометрическая точка без объема. Цвет вершины влияет на окраску полигона. Если мы зададим собственный уникальный цвет для каждой вершины, то полигон, как по волшебству, красиво закрасится трехцветным градиентом.

Цвет RGB в OpenGL задается не привычными нам значениями от 0 до 255, а от 0 до 1. Это позволяет производить такие фокусы, как модуляция цветов, которая нужна для эффектов затенения. Например: $(0.1, 0.4, 0.2) * (1, 0, 0) = (0.1, 0, 0)$

В то время, как при диапазоне [0,255] та же операция дала бы совсем иной результат, поскольку 1 соответствует 255.



А теперь сделаем так, чтобы треугольник поворачивался вокруг своей оси:

```
dim angle#
while true
    glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    glTranslatef(0, 0, -5)
    glRotatef(angle#, 0, 1, 0)
    glBegin(GL_TRIANGLES)
        glColor3f(1,0,0): glVertex3f(-1,-1, 0)
        glColor3f(0,1,0): glVertex3f( 1,-1, 0)
        glColor3f(0,0,1): glVertex3f( 0, 1, 0)
    glEnd()
    SwapBuffers()
    angle# = angle# + 0.5
wend
```

`wend`

Вы, должно быть, уже догадались, что мы сделали: ввели переменную `angle#` (решетка в конце указывает на то, что это вещественное значение) и, увеличивая ее на 0.5 при каждом шаге цикла, вызывали функцию

`glRotatef(angle#, 0, 1, 0)`

...которая и поворачивает текущую систему координат на угол `angle#` по оси Y. Обратите внимание: если бы мы вызвали `glRotatef` перед `glTranslatef`, перемещение произошло бы после поворота, а значит - треугольник вращался бы не вокруг своей оси, а вокруг исходной оси (0,Y,0).

На этом первый урок по изучению OpenGL можно считать завершенным. Я намеренно не касался особенностей синтаксиса Basic4GL и вопросов написания OpenGL-приложений на C++, так как цель данного цикла статей – дать общее понятие об OpenGL как о независимой от языка программирования среде. Вы вправе сами выбрать, на основе какого языка изучать OpenGL, так как большого значения это не имеет.

«Битва гигантов»: Xtreme3D vs Ultimate3D

Никому не дает покоя извечный вопрос: какой движок мощнее – Xtreme3D или Ultimate3D? Что ж, попробуем разобраться и решить, наконец, эту проблему. Я оценил оба движка по десяти контрольным критериям по десятибалльной шкале и в конце суммировал баллы, получив окончательный результат. Но, хотелось бы заранее отметить, я вовсе не имею в виду, что какой-то из этих движков явно хуже, а другой лучше. Нет. Оба весьма достойны и сопоставимы по возможностям. Я лишь постарался объективно оценить технические достоинства и недостатки, чтобы читатели приняли их к сведению и извлекли максимальную выгоду из этой информации. Хотя я и причисляю себя к приверженцам X3D, это отнюдь не очередной холивар.

Объектная ориентация

U3D использует встроенную объектную модель GM поверх своей. Это удобно далеко не всем, хотя в некоторых случаях имеет свои преимущества. В то же время X3D изначально предлагает собственную внутреннюю объектную модель. Программист вправе остаться на ней или же использовать объекты GM. Справедливости ради стоит отметить, что U3D более оптимизирован для использования с объектной моделью GM и глубже интегрируется в него на уровне скриптов.

В U3D начисто отсутствует встроенная иерархия объектов для трансформации локальных координатных систем. Ее необходимо писать самостоятельно при помощи матриц и векторов. Подобные манипуляции – не самые производительные, поэтому исполняться они должны ну никак не на уровне скриптового языка. В X3D, где реализация иерархии присутствует изначально, эти операции выполняются на низком уровне команд OpenGL, поэтому снижения производительности не происходит. Но, в то же время, недостатком X3D является отсутствие в нем функций для работы с матрицами и векторами как с математическими объектами.

Ввиду отсутствия иерархии, в U3D нет и объекта класса Empty (Pivot, Dummycube), который широко распространен в других движках, включая X3D.

X3D: 9/10

U3D: 4/10

Материалы и текстуры

Один из наиболее существенных недостатков U3D – небольшой выбор функций для работы с материалами. Например, цвет материала задается только через Diffuse:RGBA; Emission:RGBA, в то время, как в X3D вы не привязаны к модели RGB, а цвет материала задается через Ambient:CA; Difuse:CA; Specular:CA; Emission:CA (где C – color, A – alpha).

Система мультитекстурирования U3D ограничена восемью текстурными слоями и объединена с операциями смешивания, то есть, например, суммирование цветов может быть применено только к отдельному текстурному слою материала, но не к самому материалу. В X3D есть по крайней мере три независимые системы мультитекстурирования, использующие отдельные материалы в качестве текстурных слоев, а опции смешивания применяются сугубо к самим материалам отдельно от мультитекстурирования.

Кроме того, в X3D можно настроить масштаб, фильтрацию, формат и аппаратное сжатие текстур, режим отображения полигонов и многое другое. Все это в U3D без использования шейдеров невозможно.

И U3D, и X3D поддерживают environment mapping. Но в X3D, на мой взгляд, эта технология реализована более вменяемо.

X3D: 9/10

U3D: 3/10

Шейдеры и спецэффекты

Главное преимущество U3D – поддержка пиксельных и вершинных шейдеров (1.4, ассемблер). Благодаря этому можно написать множество эффектов, изначально отсутствующих в движке. X3D, к сожалению, не поддерживает программирование шейдеров. Хотя в библиотеке GLScene, на которой основан X3D, имеется полная поддержка шейдеров на ассемблере, Cg и GLSL (OpenGL 2.0). Так что, вполне возможно, в будущем наметятся кое-какие подвижки в этом направлении.

В U3D есть встроенный эффект bump mapping, работающий через шейдеры. В X3D он тоже есть, причем, благодаря программной эмуляции на уровне OpenGL, bump mapping реализуется даже на старых видеокартах, не поддерживающих шейдеры. Кроме того, в X3D есть встроенные аппаратные эффекты cel shader, outline shader, hiddenline shader. Множество красивых эффектов реализуются посредством мультитекстурирования, а также с использованием специальных объектов, например, реалистичная вода, lensflare, motion blur, trail и т.д. Большая часть подобных эффектов в U3D изначально отсутствует.

В обоих движках есть реалтаймовые стеновые тени, а также системы частиц. В U3D система частиц более мощная, так как позволяет изменять вид и поведение частиц модификаторами и гравитационными излучателями. Зато в X3D есть уникальная система, моделирующая молнии.

Оба движка располагают возможностью совершать рендеринг в текстуру.

X3D: 6/10

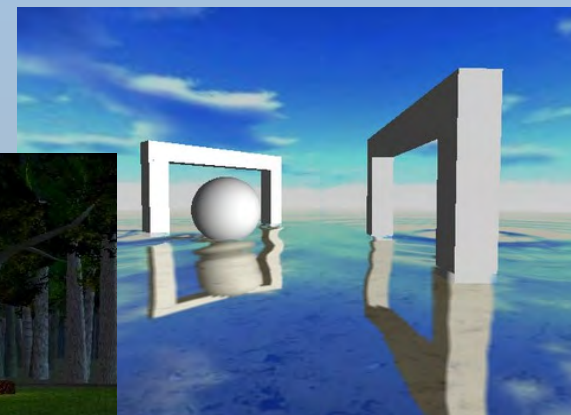
U3D: 9/10

Обнаружение столкновений

В U3D есть только один метод обнаружения столкновений: трассировка лучей. В X3D, помимо нее, имеется множество других методов. Можно находить пересечение примитивов (сфера, куб, меш, точка), использовать встроенный движок обнаружения столкновений DCE (Dynamic Collision Engine), в котором все вычисления производятся автоматически, а вам остается лишь назначить сталкивающиеся объекты. Кроме того, есть отдельная система для создания игры от первого лица (FPS Manager), а также интегрированный физический движок ODE, имеющий собственную систему проверки столкновений. Правда, интеграция с ODE далека от полной стабильности и имеет несколько неприятных багов.

X3D: 9/10

U3D: 3/10



Графические примитивы

В X3D, несомненно, гораздо более широкий выбор примитивов: плоскость, куб, сфера, цилиндр, конус, тороид, полный цилиндр, диск, трехмерный текст (объемный и плоский), полигоны, спрайт (биллборд). В U3D из вышеперечисленного есть только плоскость, куб, биллборд и полигоны. Стоит также отметить, что в U3D для вершин полигонов можно указать координаты UV, а в X3D – нельзя.

X3D: 9/10

U3D: 5/10

Легкость в освоении

X3D вовсе не так сложен для начинающих, как все думают. Просто X3D, в отличие от U3D, не ориентирован на тех, кто раньше работал со встроенной 3D графикой GM. Но вот изначальное наличие подробной документации и уроков дало U3D преимущество и с тех пор U3D де-факто признан более легким в освоении, нежели X3D. Сегодня эта ситуация кардинально меняется, во многом благодаря выпуску справочного материала к X3D, наличию примеров, уроков и т.д.

X3D: 6/10

U3D: 9/10

Кроссплатформенность

Несмотря на то, что оба движка распространяются только под Windows, имеется кое-какая возможность запустить игры, созданные с их применением, под ОС Linux. Существует альтернативная реализация Windows API – WINE, позволяющая запускать большинство приложений Windows в среде Linux. При этом OpenGL поддерживается намного лучше, чем Direct3D, поэтому шанс запустить игру на базе X3D куда выше, чем игру на основе U3D.

X3D: 7/10

U3D: 3/10

Оптимизация производительности

Кроме общеизвестного frustum culling, в обоих движках есть система порталов. Несомненным плюсом X3D является встроенная реализация quadtree и octree, благодаря которым можно с высокой скоростью отрисовывать сцены со многими тысячами объектов.

X3D: 8/10

U3D: 5/10

Отрисовка ландшафта (terrain renderer)

Присутствует в обоих движках. В U3D получить качественный ландшафт намного легче; плюс есть возможность сгенерировать детализированную карту освещения для ландшафта. Кроме того, в X3D нет явного способа наложить на ландшафт несколько текстур в режиме сплаттинга.

X3D: 3/10

U3D: 9/10

2D-графика

В U3D есть функции для рисования изображений и текста на экране. Примерно то же самое вы найдете и в X3D, поэтому в этом отношении движки одинаковы.

X3D: 6/10

U3D: 6/10

Итог

X3D: 72/100

U3D: 56/100

«Радио будущего»

Советский поэт-футурист предсказал появление Интернета?

Виктор Владимирович (Велимир) Хлебников (1885 – 1922) известен прежде всего как поэт-футурист. Он был своеобразным художником, оказавшим определенное влияние на творчество ряда советских поэтов. Но он вовсе не рвал полностью связей с традиционной литературой – например, призывал изучать русские сказки. Хлебников искренне верил, что может предвидеть будущее...

Очерк «Радио будущего», который еще двадцать лет назад сочли бы научно-фантастическим, относится к 1921 г. За шестьдесят лет до возникновения Интернета Хлебников предсказал передачу текста, изображений, звука, новостные и социальные сервисы, сетевые игры, обмен мгновенными сообщениями и многое другое. Хлебников называет всемирную сеть «Радио». Радио рассылает информацию по проводам в самые дальние населенные пункты, духовно обогащая людей и формируя глобальное информационное сообщество.

То, что сегодня кажется обыденностью, в двадцатые годы казалось утопической сказкой. Возможно, нам стоит воспринимать научную фантастику более серьезно, ведь она, в какой-то мере, формирует наш мир и предопределяет наше будущее.

«Радио будущего» перепечатывается нами из журнала «Техника молодежи», № 9 за 1985 г.

РАДИО БУДУЩЕГО Велимир Хлебников (1885 – 1922)

Радио будущего – главное дерево сознания – откроет ведение бесконечных задач и объединит человечество... Малейшая остановка работы Радио вызвала бы духовный обморок всей страны, временную утрату ею сознания.

Радио становится духовным солнцем страны, великим чародеем и чарователем.

Вообразим себе главный стан Радио: в воздухе паутина путей, то погасающих, то зажигающихся вновь, переносящихся с одного конца здания на другой... Из этой точки земного шара ежедневно, похожие на весенний пролет птиц, разносятся стаи вестей из жизни духа.

Дела художников пера и кисти, открытия художников мысли (Мечников, Эйнштейн), вдруг переносящие человечество к новым берегам...

Советы из простого обихода будут чередоваться со статьями граждан снеговых вершин человеческого духа. Вершины волн научного моря разносятся по всей стране к местным станам Радио, чтобы в тот же день стать буквами на темных полотнах огромных книг, ростом выше домов, выросших на площадях деревень, медленно переворачивающих свои страницы.

Эти книги улиц – читальни Радио! Своими великанскими размерами обрамляют села, исполняют задачи всего человечества.

Радио решило задачу, которую не решил храм как таковой, и сделалось так же необходимым каждому селу как теперь училище или читальня.

Задача приобщения к единой душе человечества, к единой ежесуточной духовной волне, проносимой над страной каждый день, орошающей страну дождем научных и художественных новостей, – эта задача решена Радио с помощью молнии. На громадных теневых книгах деревень Радио отпечатало сегодня повесть любимого писателя, статью о дробных степенях пространства, описание полетов и новости соседних стран. Каждый читает, что ему любо...

Железный рот самогласа пойманную и переданную ему зыбь молнии превратил в громкую разговорную речь, в пение и человеческое слово.

Все село собралось слушать.

Из уст железной трубы громко несутся новости дня, дела властей, вести о погоде, новости из бурной жизни столиц.

Кажется, что какой-то великан читает великанскую книгу дня. Но это железный чтец, это железный рот самогласа; сурово и четко сообщает он новости утра, посланные в село маяком главного стана Радио.

Но что это? Откуда этот поток, это наводнение всей страны неземным пением, ударом крыл, свистом и цоканьем и целым серебряным потоком дивных, безумных колокольчиков, хлынувших оттуда, где нас нет, вместе с детским пением и шумом крыл?

На каждую сельскую площадь страны льются эти голоса, этот серебряный ливень. Дивные серебряные бубенчики вместе со свистом хлынули сверху. Может быть, небесные звуки – духи – низко пролетели над хаткой. Нет...

Мусоргский будущего дает всенародный вечер своего творчества, опираясь на приборы Радио, в просторном помещении от Владивостока до Балтики, под голубыми стенами неба... В этот вечер, ворожа людьми, причащая их своей душой, а завтра – обыкновенный смертный! Он, художник, околдовал всю страну, дал ей пение моря и свист ветра! Каждую деревню, каждую лачугу посетят божественные свисты и вся сладкая нега звуков.

Почему около громадных огненных полотен Радио, что встали как книги великанов, толпятся сегодня люди отдаленной деревни? Это Радио разослало по своим приборам цветные тени, чтобы сделать всю страну и каждую деревню участницей выставки художественных холстов далекой столицы. Выставка перенесена световыми ударами и повторена в тысячи зеркал, по всем станам Радио. Если раньше Радио было мировым слухом, теперь оно – глаза, для которых нет расстояния. Главный маяк Радио послал свои лучи, и Московская выставка холстов лучших художников расцвела на страницах книг читален каждой деревни огромной страны, посетив каждую населенную точку...

Подойдем ближе... Гордые небоскребы, тонущие в облаках, игра в шахматы двух людей, находящихся на противоположных точках земного шара, оживленная беседа человека в Америке с человеком в Европе...

Вот потемнели читальни; и друг донеслась далекая песня певца, железными горлами Радио бросило лучи этой песни своим железным певцам: пой железо! И к слову, выношенному в тиши и одиночестве, к его бьющим ключам, причастилась вся страна.

Покорнее, чем струны под пальцами скрипача, железные приборы Радио будут говорить и петь, повинуюсь его волевым ударам.

В каждом селе будут приборы слуха и железные голоса для одного чувства и железные глаза для другого.

И вот научились передавать вкусовые ощущения – к простому, грубому, хотя и здоровому, обеду Радио бросит лучами вкусовой сон, призрак совершенно других вкусовых ощущений... Сытный и простой обед оденет личину роскошного пира... Это даст Радио еще большую власть над сознанием страны...

Даже запахи будут в будущем покорны воле Радио, глубокой зимой медовый запах липы, смешанный с запахом снега, будет настоящим подарком Радио стране.

Современные врачи лечат внушением на расстоянии по проволоке. Радио будущего сумеет выступать и в качестве врача, исцеляющего без лекарства...

И, наконец, – в руки Радио переходит постановка народного образования. Верховный совет наук будет рассылать уроки и чтение для всех училищ страны – как высших, так и низших.

Учитель будет только спутником во время этих чтений. Ежедневные перелеты уроков и учебников по небу в сельские училища страны, объединение ее сознания в единой воле.

Так Радио скует непрерывные звенья мировой души и сольет человечество.

Это все!

Надеемся, номер вышел интересным. Если так, поддержите FPS! Отправляйте статьи, обзоры, интервью и прочее на любые темы касательно игр, графики, звука, программирования и т.д. на ящик редакции: clocktower89@mail.ru.

*Главный редактор журнала **Gecko***



© 2008-2009 Clocktower Games. Все названия и логотипы в журнале являются собственностью их законных владельцев и не используются в качестве рекламы продуктов или услуг. Редакция не несет ответственности за корректность и достоверность информации в статьях и надежность всех упоминаемых URL-адресов. Использование материалов журнала разрешено только с согласия авторов. Любое коммерческое использование журнала или его материалов без согласия редакции и авторов материалов считается незаконным.

Директор, главный редактор и оформитель: Гафаров Т. А. aka **Gecko**

По вопросам сотрудничества обращаться по адресу clocktower89@mail.ru.
Официальный сайт журнала: <http://xtreme3d.narod.ru>