

# FPS

№ 11  
2010

Blender 2.5x  
SIGGRAPH 2010  
Дополненная реальность  
Талисманы машинной графики  
Tr3s Lunas

Модели освещения  
Ваш 3D-формат  
Creative Commons  
OpenShot  
... и многое другое



# FPS

<http://fps-magazine.narod.ru/>

## Нас читают:

[gcup.ru](http://gcup.ru)

Всё для начинающего  
и профессионального  
разработчика игр

[xbobr.at.ua](http://xbobr.at.ua)

Создание игр,  
программ, музыки

[make-games.ru](http://make-games.ru)

Портал создания игр

[creat-game.ru](http://creat-game.ru)

Все для создания игр

[portalgame.net.ru](http://portalgame.net.ru)

Игровой портал

[gamer-club.ucoz.com](http://gamer-club.ucoz.com)

Все для создания игр без  
программирования  
и не только

[gamesfpscreator.at.ua](http://gamesfpscreator.at.ua)

Сайт для помещений игр  
и обсуждаловок

[gdev.tabaru](http://gdev.tabaru)

Сообщество  
разработчиков игр

[dapf.us](http://dapf.us)

Форум для дизайнеров  
и программистов

[www.mizzystic.ru](http://www.mizzystic.ru)

Крупнейший  
информационный  
Game Maker портал

[gameshaker.ucoz.ru](http://gameshaker.ucoz.ru)

Все для редактирования и  
создания игр.

- **Blender 2.5x**  
Скелетная анимация.....3  
Горячие клавиши.....7
- **Sintel**  
Новый фильм от Blender Foundation.....8
- **SIGGRAPH 2010**  
Всемирная выставка компьютерной графики..11
- **Дополненная реальность**  
Технология будущего.....16
- **«Талисманы» машинной графики**  
Чайник Юта и другие.....21
- **Tr3s Lunas**  
Музыкальная реальность Майка Олдфилда....25
- **Модели освещения**  
Диффузное рассеивание по Ламберту.....31
- **Ваш собственный 3D-формат**  
Из Blender – в OpenGL.....35
- **Creative Commons**  
"Конструктор" лицензий.....42
- **OpenShot**  
Редактор видео под Linux.....45

© 2008-2010 Clocktower Games. Все названия и логотипы 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. Материалы издания распространяются согласно условиям лицензии Creative Commons Attribution Noncommercial Share Alike (CC-BY-NC-SA).

Главный редактор: Тимур Гафаров  
Дизайн и верстка: Тимур Гафаров

По вопросам сотрудничества обращаться по адресу [clocktower89@mail.ru](mailto:clocktower89@mail.ru) или [gecko0307@gmail.com](mailto:gecko0307@gmail.com).  
Официальный сайт журнала: <http://fps-magazine.narod.ru/>





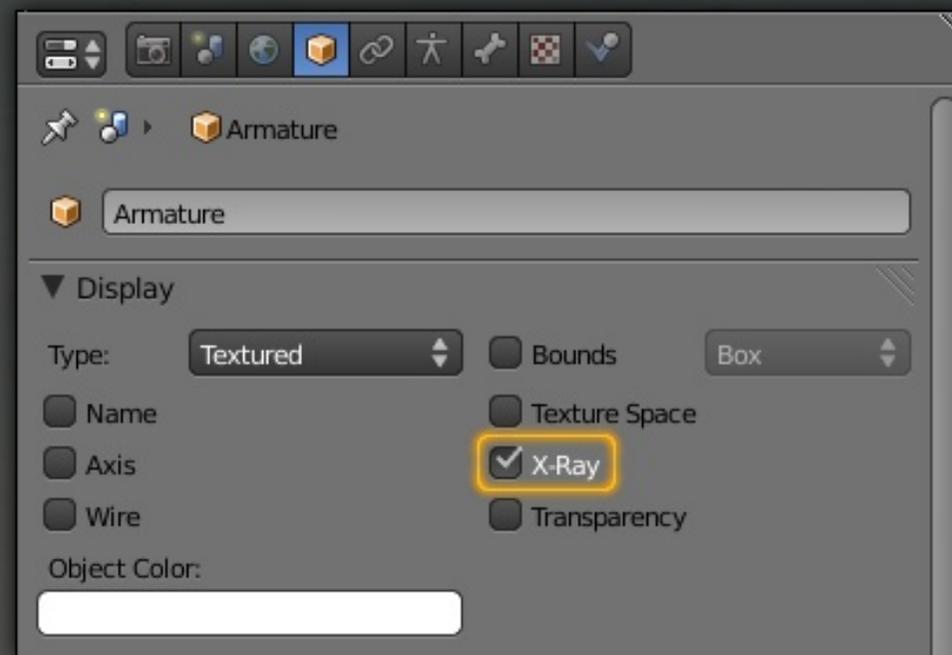
# Blender 2.5

## Скелетная анимация

Скелетная анимация (или, в терминологии Blender, риггинг) имеет репутацию довольно сложного дела. Действительно, чтобы создавать грамотную анимацию, недостаточно быть хорошим моделлером — нужно обладать дополнительными знаниями и быть достаточно наблюдательным, чтобы запоминать реальные движения и с точностью повторять их виртуально (ситуация упрощается при использовании Motion Capture — но это уже совсем другая история). Однако сам принцип создания скелета довольно прост. Ну а в Blender 2.5 — прост до неприличия.

1. Нам необходима модель персонажа для создания анимации. Можете создать свою или взять готовую.

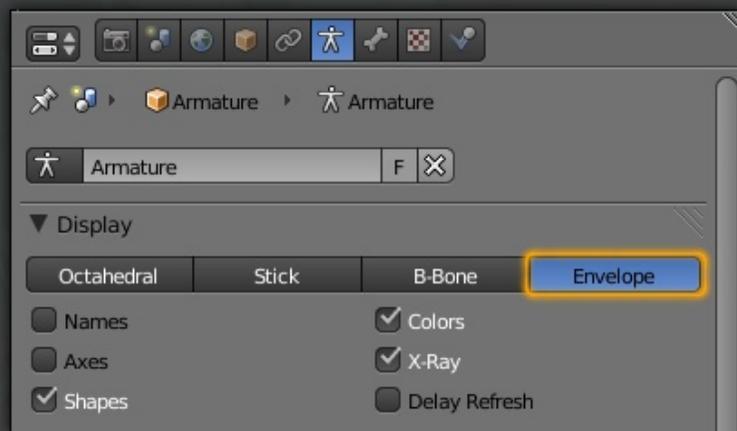
2. Переключитесь на вид спереди (3D View > View > Front или Numpad 1). Добавьте новую арматуру (Info > Add > Armature > Single Bone) и сразу включите режим отображения X-Ray ("Рентген"): Properties > Object > Display > X-Ray.



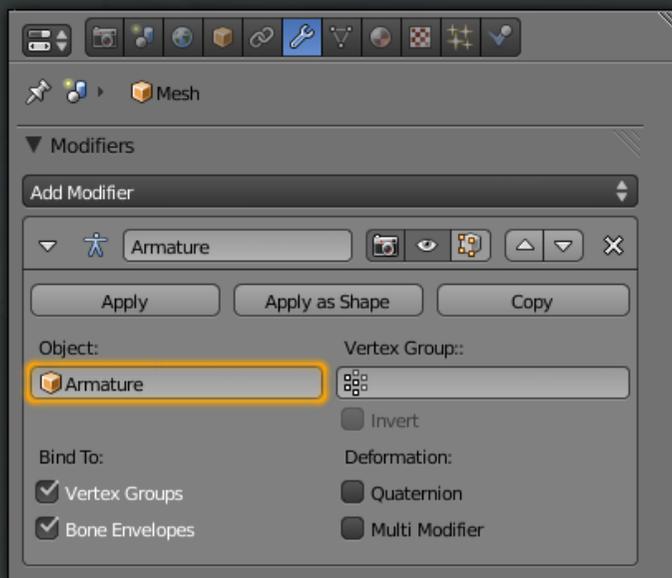
3. Войдите в режим редактирования (Tab). Выделите верхушку кости и нажмите E. Потяните курсор, чтобы удлинить созданную кость. Аналогичным способом создайте остальные кости для всех частей тела. Здесь не лишним будет знание пластической анатомии. Обратите внимание, что в данном уроке в качестве образца используется модель монстра из Doom 3, анатомия которого несколько отличается от человеческой. Особенно это касается ног.

4. Отдельный вопрос - нужно ли создавать кости для пальцев. В шутерах от первого лица (да и вообще в любых играх AAA-класса), конечно, анимация отдельных пальцев просто необходима. Однако, если создается низкополигональная модель (скажем, для MMORPG или стратегии), то пальцы, как правило, сжаты в кулак - соответственно, анимация им не требуется. В некоторых играх ладонь разделяется только на две анимированные части: большой палец и соединенная группа из остальных четырех. Правда, она при этом становится похожа на рукавицу.

5. В Blender предусмотрено несколько способов схематичного отображения костей. Лично мне, например, нравится Envelope (Properties > Object Data > Display > Envelope). Активировав его, можно изменить цвет или текстуру объекта, чтобы скелет выделялся более контрастно.

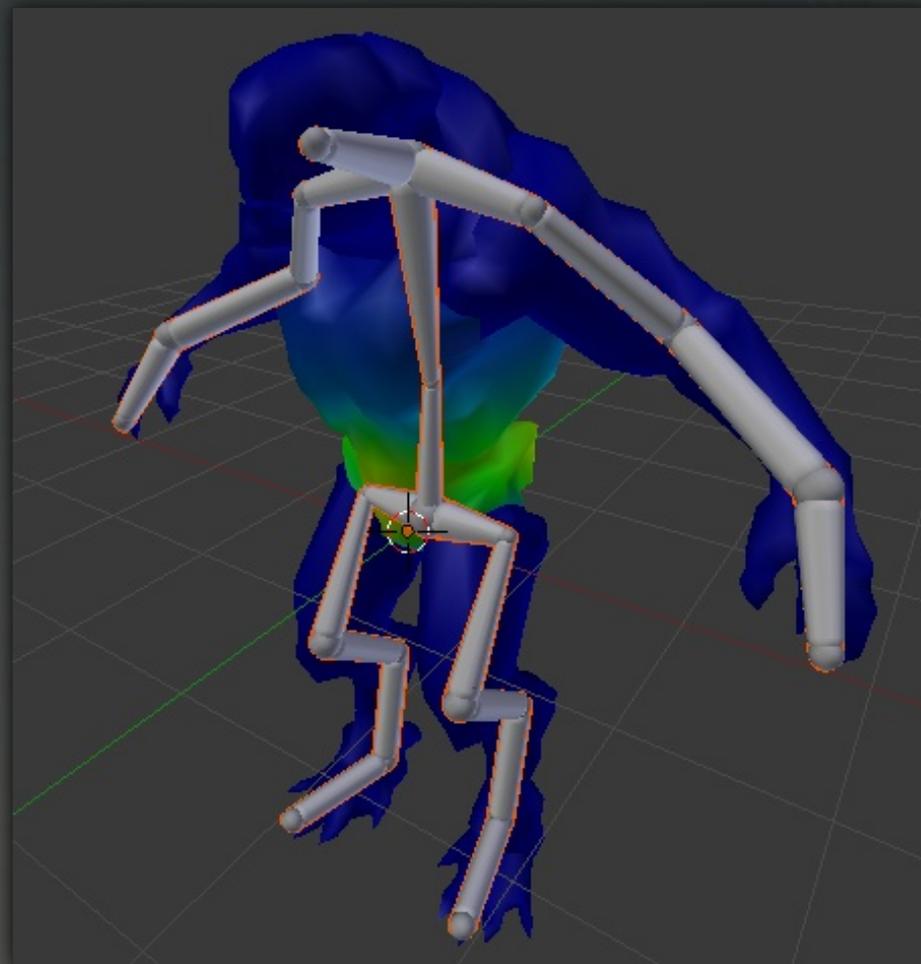


6. Выходите из режима редактирования арматуры (Tab) и выделите модель. Добавьте ей модификатор Armature (Properties > Modifiers > Add Modifier > Armature). Не забудьте в поле Object выбрать объект-арматуру.



7. Не снимая выделение, перейдите в режим рисования развесовки (3D View > Weight Paint). Развесовка — это свойство вершин модели реагировать на трансформацию костей скелета. Любая кость может в той или иной степени влиять на любые вершины. В Blender это влияние визуально представлено цветом (чем теплее, тем сильнее влияние: синий — самое низкое, красный — самое высокое). Можно «рисовать» развесовкой прямо на модели (отсюда и название режима).

Обычно кость влияет только на ближайшие к ней вершины — а значит, можно рассчитать их развесовку математически. В Blender есть специальная функция для этого. Кликните правой кнопкой мыши на любой кости и нажмите W. В выпадающем меню выберите Automatic. Повторите эту операцию для всех остальных костей.



8. Перейдите в объектный режим (3D View > Object Mode) и снова выделите арматуру. Перейдите в режим позирования (3D View > Pose Mode). В нем вы, наконец, сможете увидеть скелет в действии. Выделяйте кости и поворачивайте их, нажимая R.

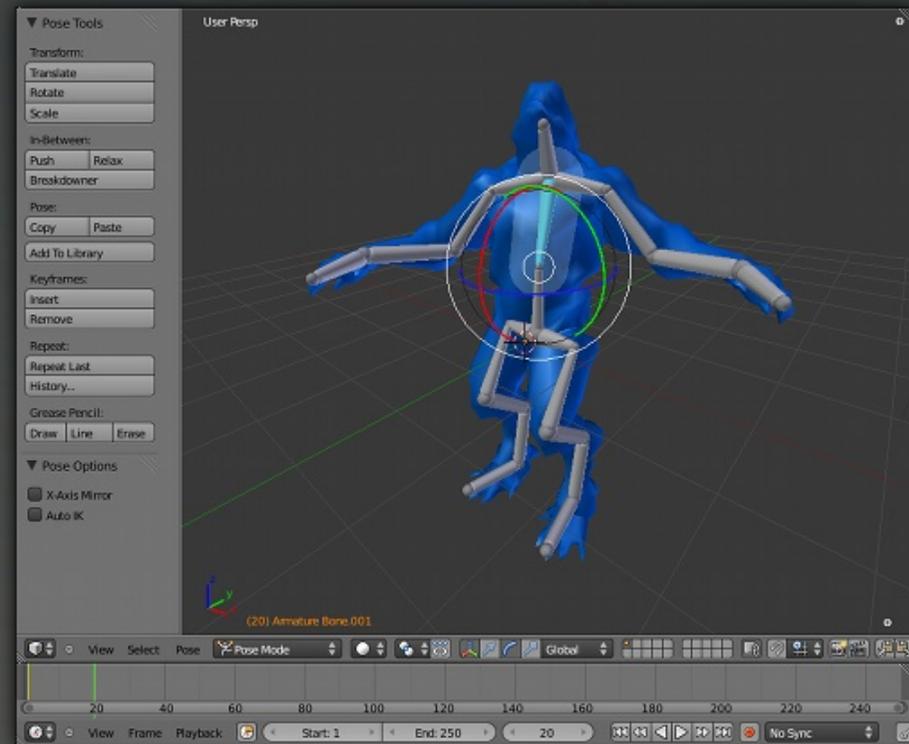
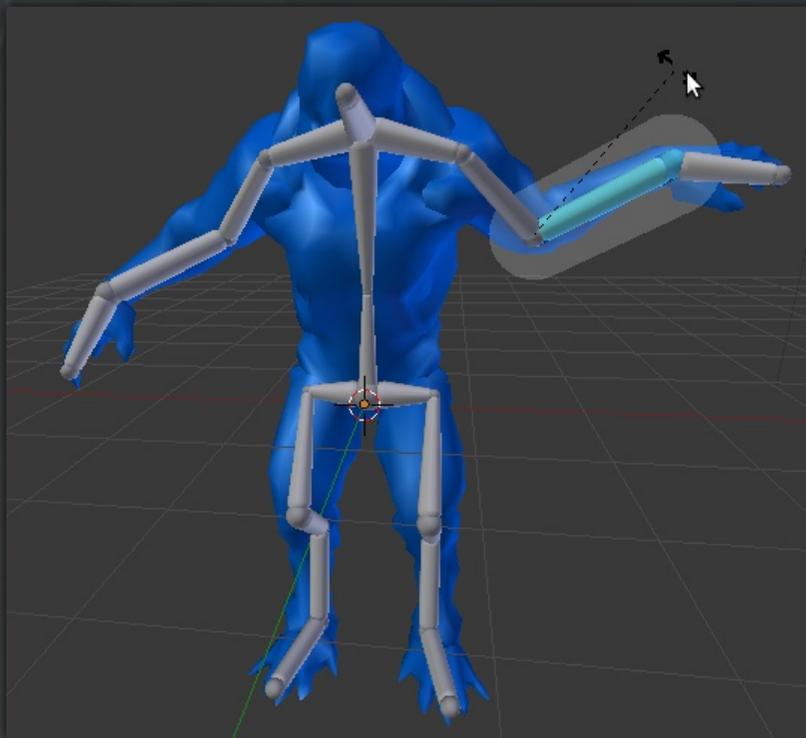
9. Попробуем создать анимацию. Не выходя из режима позирования, нажмите клавишу I. В выпадающем меню выберите Rotation. Это добавит ключевой кадр с текущей трансформацией арматуры. Перед этим убедитесь, что текущий кадр — первый.

Откройте в какой-нибудь горизонтальной панели временную шкалу (Timeline) и переключитесь, например, на 20-й кадр (можно горячими клавишами: два раза «вверх» и один - «назад»). В Blender 2.5, кстати, появились удобные кнопки для перемещения по временной шкале (прямое и обратное воспроизведение, переход к следующему/предыдущему ключевому кадру и т.д.)

Измените позу персонажа и добавьте новый ключевой кадр.

10. Перейдите обратно к первому кадру и запустите воспроизведение анимации.

**Gecko**  
*gecko0307@gmail.com*





# Blender 2.5

## Основные горячие клавиши

The image shows the Blender 2.5 interface with a monkey head model in the center. A keyboard overlay is positioned in front of the interface, with keys color-coded to match the Blender hotkey scheme. The overlay includes:

- Function Keys (F1-F12):** F1 Браузер + Shift, F2 Логика, F3 Узлы, F4 Консоль, F5 3D-окно, F6 Граф, F7 Свойства, F8 Монтаж, F9 Объекты, F10 UV/Изобр., F11 Текст, F12 Аним.
- Navigation and Editing:** Esc, Print, Scroll, Pause, Ins, Home, Up, Del, End, Down, Num Lock, /, \*, - (Отдалить), 7 Вид сверху, 8 Поворот вперед, 9 + (Приблизить), 4 Поворот влево, 5 Персп.-ва, 6 Поворот вправо, 1 Вид спереди, 2 Поворот назад, 3 Вид сбоку, 0 Камера, . По центру, Enter.
- Layer and Object Tools:** 1 Слой 1 + Shift + Alt, 2 Слой 2, 3 Слой 3, 4 Слой 4, 5 Слой 5, 6 Слой 6, 7 Слой 7, 8 Слой 8, 9 Слой 9, 0 Слой 10, Backspace, Tab Режим + Ctrl, Q, W Спец. + Tab, E Экструд., R Повор. + Ctrl, T Инструм. + Ctrl, Y Ось, U Разделить + Tab, I Польз.-ль, O Ключ. кадр, P Пропорц.-но, (, ), Enter.
- Selection and Manipulation:** Caps Locks, A Выделить, S Масш. + Ctrl, D, F Грань + Shift + Ctrl, G Перем. + Ctrl, H Скрыть + Shift + Alt, J Соедин. + Alt, K, L Локальн., ;, ', /, #, Shift, \, Z Ось Затенение + Alt, X Ось Удалить, C Копир. Объемн. выдел.-е, V Встав. Верш. + Alt, B Выдел. по контуру, N Свойств., M Слои, ', ., /, Shift +function.
- Control and Navigation:** Ctrl, Win, Alt +function, Space Поиск операции, Alt Gr, Win, Menu, Ctrl +function, ^ След. 10 кадров, < Пред. кадр, V Пред. 10 кадров, > След. кадр.

The background interface shows the 3D viewport with a monkey head model, the Outliner panel on the right, and the Properties panel at the bottom right.



# SINTEL

30 сентября 2010 г. состоялась долгожданная онлайн-премьера «Синтел» — третьего по счету открытого анимационного фильма Blender Foundation, при создании которого было использовано исключительно открытое программное обеспечение, преимущественно — 3D-редактор Blender (два предыдущих проекта — «Elephants Dream» и «Big Buck Bunny»). Работа над этим короткометражным фильмом (общая длительность с титрами — около 15 мин.) велась в течение последних полутора лет; в ней приняли участие талантливые художники и аниматоры со всего мира. В течении первой же недели после выхода фильм набрал более 1 млн. просмотров на YouTube.

В основу сюжета легли скандинавские легенды. Молодая девушка по имени Синтел, сирота, живущая на улицах фэнтезийного города, находит маленького дракона с перебитым крылом, который становится ее лучшим другом. Дракончик выздоравливает и вновь обретает способность летать. Синтел дает ему имя — Scales, что можно перевести как «Чешуйчик». Однажды, во время очередной прогулки, друзья сталкиваются со взрослым драконом, который похищает Чешуйчика. Синтел отправляется на его поиски, не подозревая, чем они могут закончиться...



Необходимо отметить, что фильм снят весьма профессионально. Персонажи и окружение смоделированы предельно качественно: особенно впечатляют сложная фактура драконьей кожи, моделировка волос и меха, микрорельеф поверхности одежды на персонажах. Отдельных похвал заслуживают спецэффекты — в фильме можно наблюдать в действии систему симуляции огня и дыма, вошедшую в число нововведений Blender 2.5. К недостаткам можно причислить разве что несколько неестественную анимацию некоторых движений героини. Однако не стоит забывать, что реалистичная анимация в современной трехмерной графике достигается преимущественно путем применения технологии Motion Capture, а данное чудо техники за пределами Голливуда могут позволить себе далеко не все.



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

## Спецпредложение от «FPS»!

«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 игрового мира (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%). Для рекламных листов: 1000x700 (формат JPG, сжатие 90%). Содержание — произвольное, но не выходящее за рамки общепринятого и соответствующее грамматическим нормам. Совет: к созданию рекламного листа рекомендуем относиться 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](mailto:clocktower89@mail.ru) или лично главному редактору: [gecko0307@gmail.com](mailto:gecko0307@gmail.com) (просьба в качестве темы указывать «Сотрудничество с FPS», а не просто «Реклама», так как письмо может отсеять спам-фильтр).

Прикрепленные материалы (рекламный лист, информация, статьи и пр.) могут быть как прикреплены к письму, так и загружены на какой-либо надежный сервер или файлохранилище (но не RapidShare или DepositFiles!). Все материалы желательно архивировать в формате zip, rar, 7z, tar.gz или tar.bz2.

«Синтел» можно посмотреть на YouTube, а также совершенно бесплатно и легально скачать на официальном сайте, причем доступны варианты в различных форматах и разрешениях. Для получения полноценного впечатления рекомендуется не ограничиться YouTube'ом и скачать фильм хотя бы в минимальном HD-качестве (96 Мб). На сайте вы также найдете тексты субтитров на нескольких языках, включая русский.

На данный момент идет подготовка «Синтел» для стереоскопического показа на киноэкранах, а другая команда параллельно занимается разработкой игры по мотивам фильма. В качестве основного инструмента, как нетрудно догадаться, используется все тот же Blender.

Официальный сайт «Синтел»: <http://www.sintel.org>

Сайт игры по мотивам фильма: <http://sintelgame.org>



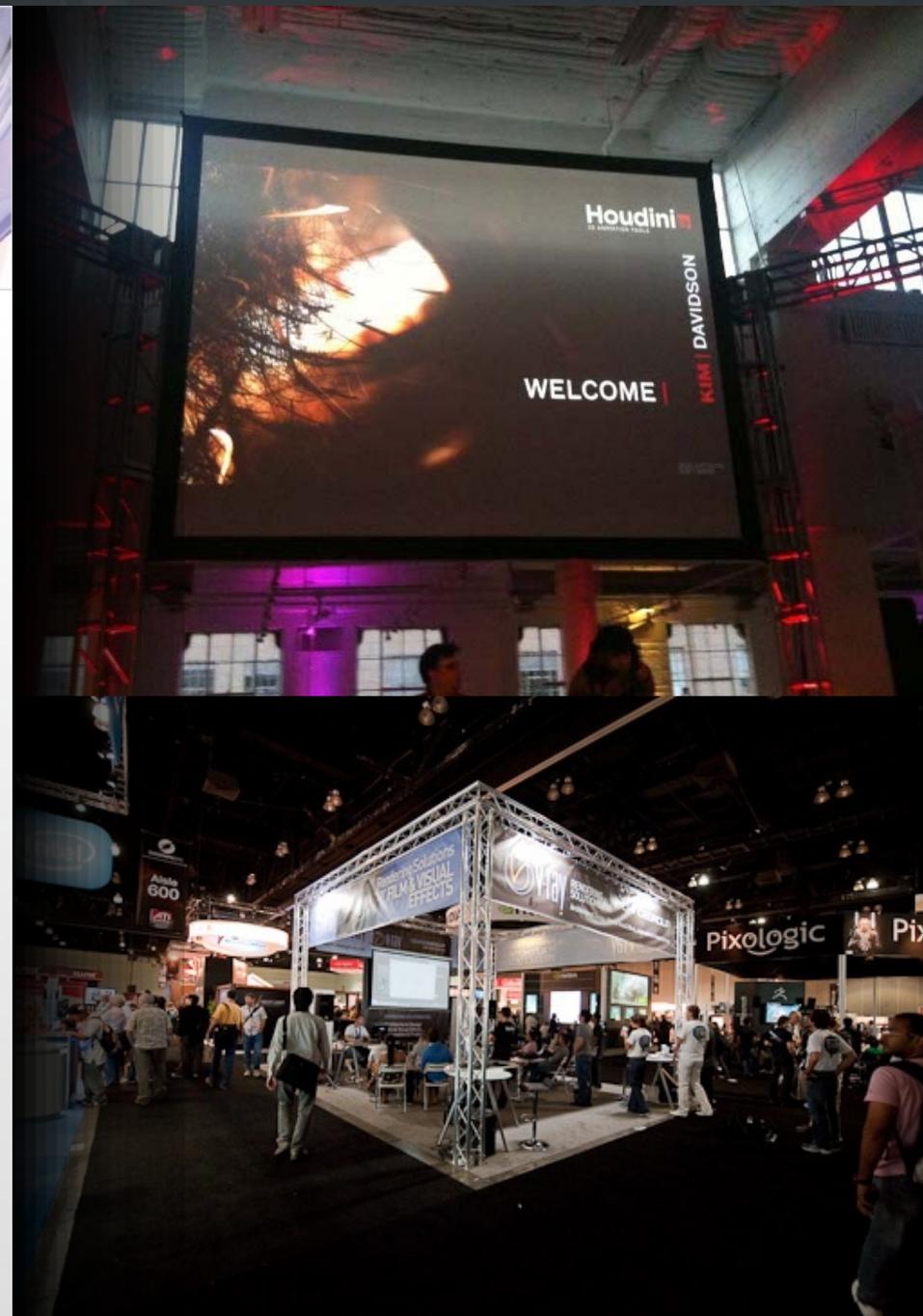


**27 июля 2010 г. в Лос-Анджелесе стартовала самая известная и самая престижная ежегодная конференция и выставка компьютерной графики SIGGRAPH 2010.**

Аббревиатура SIGGRAPH расшифровывается как Special Interest Group on Graphics and Interactive Techniques («Специальная Группа по Вопросам Графических и Интерактивных Технологий»). Выставка проводится организацией ACM SIGGRAPH с 1974 г. в Лос-Анджелесе, Новом Орлеане, Сиэтле, Далласе, Бостоне, Сан-Диего и других городах США. На нее приезжают тысячи людей со всего мира, а в последние годы к ним прибавилась многомиллионная аудитория тех, кто наблюдает за ходом выставки по Интернету.

На SIGGRAPH съезжаются представители крупнейших компаний, разрабатывающих программы и технологии, которые касаются машинной графики и анимации. На выставке демонстрируются новейшие разработки в этой области. Кроме этого, на SIGGRAPH приезжают исследователи из различных НИИ, которые представляют статьи на темы, имеющие отношение к компьютерной графике. Наконец, SIGGRAPH — это еще и место, где встречаются, делятся друг с другом опытом и проводят мастер-классы лучшие 3D-художники. В рамках SIGGRAPH также проводится фестиваль анимации и множество других мероприятий.

«Технологический контент, который демонстрируется на SIGGRAPH, с каждым годом впечатляет все сильнее. Участники выставки стараются превзойти самих себя — это нелегкая задача, но цель оправдывает средства», — утверждает Терренс Мэйссон, представитель Северо-восточного Университета США (Бостон, шт. Массачусетс).



На SIGGRAPH 2010 свои разработки представили более 200 компаний со всего мира. Среди них AMD, NVIDIA, Autodesk, Blue Sky Studios, Chaos Group, Cebas Computer, DAZ Productions, E-on Software, Canon, Google, Maxon Computer, NewTek, Pixologic, Pixar Animation Studios, Intel, Next Limit Technologies.

### OpenGL 4.1

Организация Khronos Group представила новую версию спецификации OpenGL 4.1 и языка описания шейдеров GLSL 4.1. Она обеспечивает полную совместимость с OpenGL ES 2.0, что должно упростить портирование мультимедийных приложений между настольными и мобильными платформами. Появилась возможность запрашивать и загружать скомпилированную версию объектов шейдерных программ для сокращения времени перекомпиляции, добавлена возможность индивидуального связывания программ с программными ступенями для улучшения гибкости их программирования. Кроме того, была добавлена поддержка 64-битных компонентов для вершинных шейдеров, благодаря чему повысилась точность обработки.

### ZBrush 4

Компания Pixology представила новую версию пакета трехмерной лепки ZBrush 4. За какие-то несколько лет эта программа уверенно вошла в разряд необходимых инструментов для 3D-художников всего мира. Ее используют почитатели самых разных трехмерных редакторов, поскольку она является отличным дополнением к любому из них. В числе фильмов, авторы которых использовали ZBrush, значатся такие всемирные блокбастеры, как «Аватар», «300 спартанцев», «Пираты Карибского моря: Сундук мертвеца», «Железный человек 2», «Властелин Колец».

Среди нововведений ZBrush 4 следует отметить новые типы кистей и режимы смешивания, анимированные текстуры, инструменты Matchmaker и Spotlight, новые материалы и шейдеры, включая подповерхностное рассеивание, блики по Фонгу, преломление/отражение по Френелю.



## Дополненная реальность (Augmented Reality)

Токийский Университет представил на выставке новый способ лазерного сканирования трехмерных объектов без использования стереоскопической камеры, а исследователи из Kyushu предложили свой метод для определения позиции объекта в пространстве.

### NVIDIA

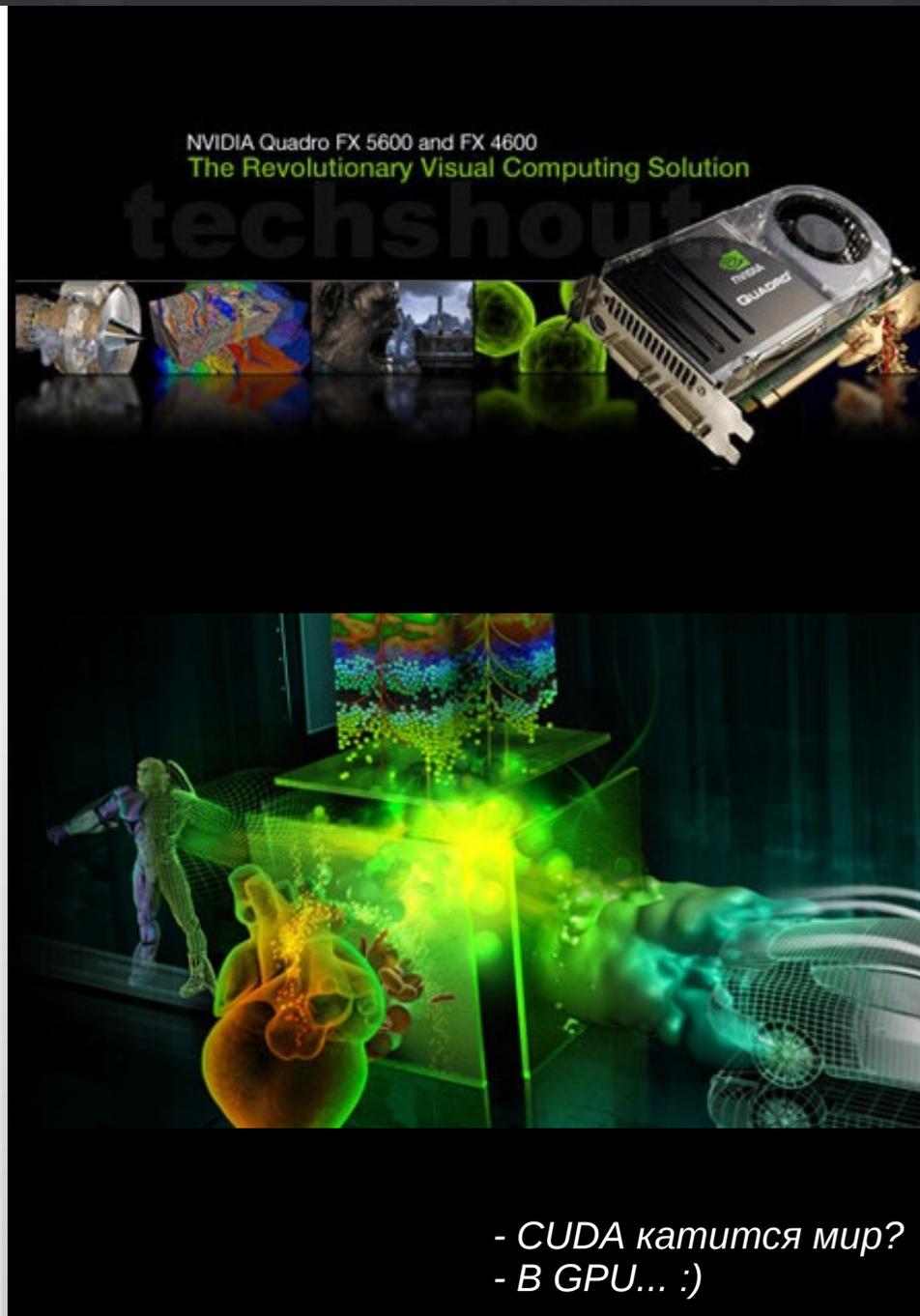
Корпорация NVIDIA представила целую россыпь новых продуктов, которые ориентированы в первую очередь на профессиональных пользователей из числа инженеров, дизайнеров, компьютерных аниматоров, архитекторов и учёных, работающих со сложными детализированными 3D-моделями.

Новые графические решения NVIDIA семейства Quadro построены на передовой архитектуре Fermi и совместимы с DirectX 11, OpenGL 4.0, DirectCompute и OpenCL. В них использованы такие фирменные разработки NVIDIA, как Scalable Geometry Engines и Application Acceleration Engines (AXE), а также технология параллельных вычислений NVIDIA CUDA.

Теперь разработчики программного обеспечения могут создавать профессиональные приложения нового поколения для успешного решения таких ресурсоёмких задач, как трассировка лучей, симуляция физических эффектов, вычислительная динамика жидкостей и обработка видеоэффектов в режиме реального времени.

Решения Quadro поддерживают работу с новой системой NVIDIA 3D Vision Pro, которая включает в себя активные затворные очки и радиочастотную систему связи, обеспечивающую высококачественную стереоскопическую картинку для самых разных сценариев использования, включая индивидуальный просмотр трёхмерного изображения на ЖК-панелях и групповой просмотр на одном или нескольких проекторах или видеостенах.

NVIDIA Quadro FX 5600 and FX 4600  
The Revolutionary Visual Computing Solution



- CUDA катится мир?  
- В GPU... :)

## Blender

В августе прошлого года, на SIGGRAPH 2009, Blender Foundation (организация, занятая разработкой и продвижением бесплатного и открытого пакета трехмерного моделирования Blender) приоткрыла завесу тайны над проектом Blender 2.5 и показала результаты нескольких лет разработки: полностью настраиваемый графический интерфейс — логическое развитие «фирменного» GUI Blender, а также новую систему анимации. В этом году была проведена официальная встреча сообщества разработчиков и пользователей Blender — «Birds of a Feather», в рамках которой Тон Роозендаал обсудил результаты работы над Blender 2.5 за последний год и представил демонстрацию готовящегося к выходу «Sintel» — третьего по счету открытого анимационного фильма Blender Foundation. Участники встречи также провели короткие презентации собственных проектов.

Кстати, выражение «birds of a feather» (его русским аналогом может служить фраза «два сапога пара» или «братья по разуму») на сленге компьютерщиков означает неформальное собрание на конференциях, где участники собираются в группы на основе общих интересов и проводят дискуссии без какого-либо заранее запланированного расписания. Термин произошёл от «Birds of a feather flock together».

## Демосцена

На SIGGRAPH 2010 были представлены демонстрации новейших достижений в области 3D-графики реального времени. Помимо компьютерных игр, эта область включает средства интерактивной визуализации данных, симуляцию сложных физических процессов, произведения информационной эстетики и искусства.

### Agenda Circling Forth

Уникальное демо, вдохновленное творчеством импрессионистов: трехмерная сцена построена не из традиционных полигонов, а из частиц (particles), «живых» точек, обладающих определенными свойствами. Частицы просчитываются на GPU. Данная работа также участвовала на демо-пати Breakpoint 2010, заняв первое место. Демо можно скачать с FTP-сервера Breakpoint:

[ftp://ftp.untergrund.net/breakpoint/2010/pc\\_demo/agenda.zip](ftp://ftp.untergrund.net/breakpoint/2010/pc_demo/agenda.zip)



## **GlyphSea**

Демонстрация современных методов трехмерной визуализации статистических данных. Один из сценариев работы GlyphSea моделирует процессы, протекающие в земной коре при землетрясении.

## **God of War III**

Новая игра серии God of War для Playstation 3 виртуозно объединяет новейшие технологические достижения и качественный детализированный графический контент для создания неповторимой атмосферы древних мифов и эпических сражений.

## **Making Love**

Полностью процедурная игра, в которой все, включая модели и текстуры, генерируется «на лету» при помощи специальных алгоритмов.

## **Real-Time Particle-Based Liquid Simulation**

Симуляция жидкости на основе частиц, выполняемая полностью на GPU. Демонстрация просчитывает 128000 частиц при скорости 60 кадров в секунду.

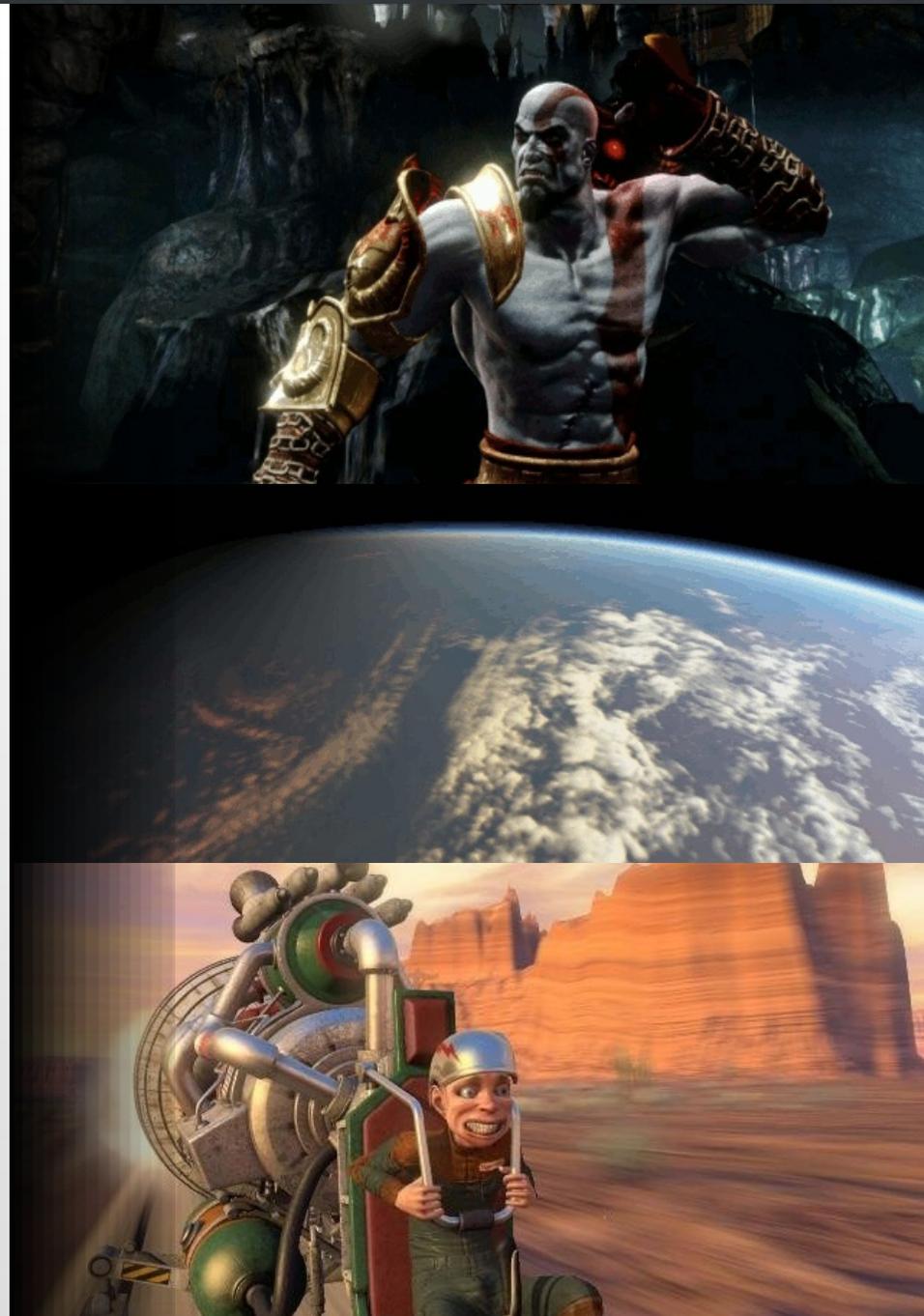
## **Proland**

Реалистичная виртуальная модель Земли с физически корректным рендерингом и анимацией океана, атмосферы, облаков и ландшафта.

## **Supersonic Sled**

Симулятор вождения мощного сверхскоростного транспортного средства с использованием PhysX, CUDA и DirectX 11.

**Gecko**  
[gecko0307@gmail.com](mailto:gecko0307@gmail.com)



# Дополненная реальность

*Дополненная реальность (Augmented Reality, AR) — общее название класса информационных технологий, направленных на объединение реального и виртуального миров для создания новых условий визуализации, где физические и цифровые объекты сосуществуют и взаимодействуют в режиме реального времени. Сочетание реальности, дополненной реальности и виртуальной реальности открывает путь новейшим технологиям в областях развлечения, бизнеса, обучения, искусства и коммуникации.*

Дополненная реальность впервые появилась в первой половине 90-х годов и сегодня широко используется в фильмах и телевизионных программах. Например, часто синтезируется изображение ведущего на фоне погодной карты или природных пейзажей. Технология прогрессирует и как способ решения военных задач — таких, как совершенствование приборов ночного видения, помощь боевым пилотам в отслеживании цели и проведение технического обслуживания техники. Несколько лет назад началось практическое применение дополненной реальности в сфере медицины.

В скором времени AR вполне может стать неотъемлемой частью повседневной жизни. Этому в немалой степени поспособствуют новые алгоритмы обработки изображений, более производительные микропроцессоры и графические чипы. Как отмечает разработчик библиотеки ARToolkit профессор Хироказу Като (Hirokazu Kato) из японского Института наук и технологий Нара, "даже iPhone предоставляет большую вычислительную мощность, чем рабочие станции 1990-х". Уже сейчас на рынок выходят первые потребительские продукты на основе AR, а в течение нескольких лет тенденция захватит домашнюю электронику, переворачивая типичное представление о ней.

Так что же такое AR? Это технология, расширяющая воспринимаемую человеком действительность за счёт дополнения видимого и ощущаемого мира цифровой информацией в реальном времени. Фундаментальное отличие между дополненной реальностью и виртуальной (VR), активно развивавшейся в последние несколько десятков лет, заключается в соотношении данных из реального мира и обрабатываемых компьютером. Если VR стремится заменить настоящий мир практически полностью, то AR только расширяет представление о происходящих в нём явлениях.

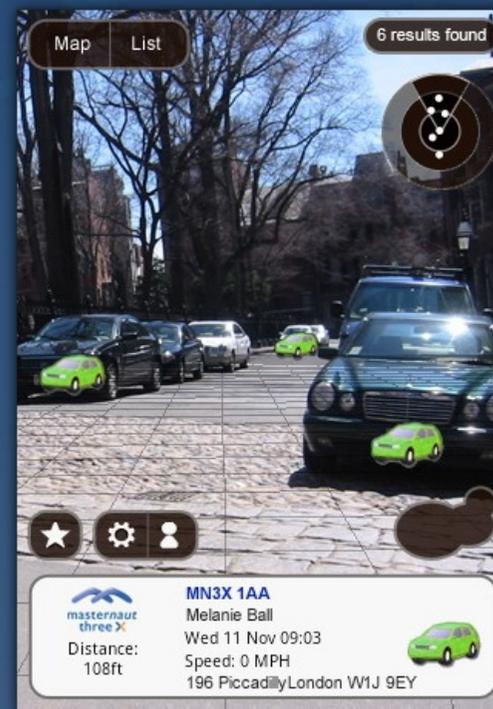
Чтобы реализовать AR на практике, необходимо соответствующее оборудование: компьютеры, сенсоры, интерфейсы и устройства вывода информации. В зависимости от возможностей каждого компонента и методов представления результатов его работы открываются широкие перспективы использования AR для решения прикладных задач. Например, если невидимое невооруженным глазом изображение человека в инфракрасном спектре или полученное с помощью магнитно-резонансной системы наложить на реальный вид, становится возможным наблюдение кровеносных сосудов, работы органов и тому подобное, что пришлось бы весьма кстати в улучшении качества медицинского обслуживания. Но такие системы пока сложны в реализации, а вот появляющиеся перед взглядом подсказки или инструкции относительно работы с тем или иным компонентом оборудования, его ремонта – это функциональность сегодняшних прототипов.

AR может означать дополнительное осязание или звук, но преимущественно разработки ведутся в сфере визуальной информации. Подобные интерфейсы дополненной реальности могут быть разделены на четыре группы по типу устройства вывода: компьютерные экраны, мобильные телефоны, встроенные в очки дисплеи и проекторы. Во всех случаях изображение фиксируется камерами, затем обрабатывается и выводится в изменённом виде.

Активно развивается ПО для мобильных платформ, которое в зависимости от местонахождения аппарата и положения его экрана в пространстве показывает соответствующую информацию о снимаемом на камеру объекте. Данные предоставляют сетевые сервисы, но ситуация в корне отличается от непосредственного доступа к ним пользователя через Интернет: нет нужды в мышке, клавиатуре, вводе запроса в поисковую строку и выбора из ссылок подходящей. Смартфон может показать направление, в котором искать потерявшегося в толпе товарища, или указать путь к ближайшим ресторанам, накладывая на реальное изображение улиц стрелочные указатели и дистанцию до заведений. Так, AR-приложения для iPhone дополняют реальную картину мира различными данными, полученными с помощью GPS, компаса, Интернет-соединения, акселерометра, микрофона и, собственно, телефона. Вы смотрите на мир сквозь камеру устройства, а на экране возникают подсказки-описания окружающих объектов. При помощи телефона на базе Android и сервиса Layar (так называемого «браузера реальности») можно в реальном времени получить доступ к информации об окружающем мире через камеру – это могут быть данные о кафе, ресторанах, гостиницах, памятниках архитектуры, автомобилях на улице и т.д. Layar даже позволяет увидеть, как выглядело то или иное место в прошлом: для этого используются реконструкции и старые фотографии.

Если проекторы и встроенные в очки дисплеи с технологией дополненной реальности будут популярны, они изменят жизнь не меньше, чем мобильные телефоны: ведь им доступна схожая функциональность, но с более интуитивным визуальным интерфейсом – например, проецирование клавиатуры на руку для набора номера или текста сообщения. А простой лист бумаги может превратиться в книгу с анимированными персонажами.

Продвинутые системы AR связаны не только с выводом изображения. Уже упомянутый профессор Като исследует метод манипулирования виртуальными предметами с помощью рук. По его словам, хотя сенсорные панели более интуитивны, чем мышь или клавиатура, но "вы всё равно не касаетесь информации". С помощью дополненной реальности взаимодействие с объектами цифрового мира становится более реалистичным.



Что касается домашней электроники, отдельные эксперты предсказывают закат некоторых видов устройств и замену их системами с AR. Например, кнопки станут виртуальными и будут проецироваться устройством на ладонь. Виртуализация интерфейсов взаимодействия может быть продемонстрирована на примере разработки CRISTAL (Control of Remotely Interfaced Systems using Touch-based Actions in Living spaces – Управление дистанционно связанными системами с помощью сенсорно фиксируемых движений в жилом помещении), представленной на конференции SIGGRAPH 2009. При помощи камер и проекторов на столе создаётся изображение всей комнаты, и когда пользователь касается проекции какой-либо техники, активируется режим управления им – подобно значкам на рабочем столе компьютера.



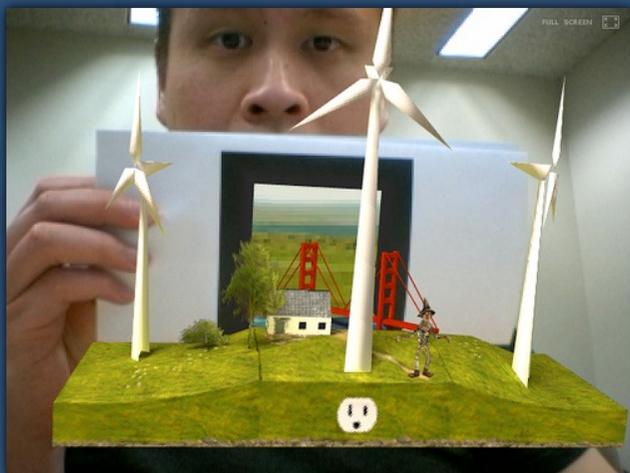
Отличным примером недорогой системы дополненной реальности может служить интерфейс "Шестое чувство" (SixthSense), созданный Пранавом Мистри (Pranav Mistry). В него входят камера, карманный проектор, смартфон и цветные ленты для пальцев, которые выступают в роли маркеров. Их перемещение в пространстве фиксируется и соответствующим образом меняется выводимое проектором изображение. Так, если сомкнуть большие и указательные пальцы рук в виде рамки и "щёлкнуть затвором", будет сделано фото. При определённом положении ладони на неё проецируется клавиатура мобильного телефона, а работать в графическом редакторе можно почти на любой поверхности, просто показывая знаки руками в пространстве.

В последние годы AR активно используется в рекламном бизнесе. При помощи дополненной реальности можно увидеть трехмерную модель автомобиля, расположив перед веб-камерой рекламный буклет с его изображением. Или рассмотреть товары в интернет-магазине, показав веб-камере объявление в газете об их продаже. Одной из первых масштабную кампанию с применением дополненной реальности запустила американская сеть пиццерий Papa John's. На 30 миллионах коробок с пиццей напечатан квадрат с машиной. Показав его веб-камере, потребитель увидит автомобиль Camaro 1972 года выпуска, легенду предприятия (именно эту машину пришлось продать основателю Papa John's, чтобы открыть сеть пиццерий).

Другую кампанию запустила почтовая служба США. У нее на основе AR создан полезный сервис, который позволяет положить реальный предмет в виртуальную посылку. Размеры виртуальных посылок соответствуют настоящим, так что все желающие могут заранее понять, поместятся отправляемые вещи в коробку или нет.



Огромное внимание привлекла кампания General Electric, став одним из самых популярных образцов использования дополненной реальности. Один из разделов сайта General Electric рассказывает о технологии экономии электроэнергии Eco Smart Grid. Демонстрацию их дополненной реальности посмотрели на YouTube более миллиона человек, а еще 250 тысяч посетили сайт и провели на нем не менее пяти минут, залюбовавшись ветряными электростанциями, вырастающими из листа бумаги.



Технически для создания виртуального рекламного окружения необходимо наличие веб-камеры, распечатанного изображения-маркера и доступа к серверу, который создает AR. Веб-камера берет снимок обычной реальности, а специальная программа добавляет в него виртуальные элементы. Обычно в качестве маркера используется квадрат - система видит, что он показан под углом, и соответственно поворачивает сцену. Например, квадрат на листе бумаги в программе может соответствовать одной из граней куба. Тогда пользователь, повернув лист, повернет и нарисованный на экране куб, "приклеенный" к бумаге.

Нет никаких сомнений, что дополненная реальность серьезно повлияет и на игровую индустрию. Уже сегодня многие разработчики успешно внедряют эту технологию в свои проекты.

### Invizimals

Популярная игра для портативной платформы PSP. Общая концепция напоминает нашумевших покемонов: игрок ловит фантастических существ (инвизималов), тренируя их затем для поединков — с искусственным интеллектом или с другими игроками через беспроводную связь. Ключевая особенность в том, что инвизималы «существуют» в реальном мире, и увидеть их помогает дополненная реальность. При помощи камеры PSP и специального маркера в виде фигурной пластинки игра распознает ракурс обзора и положение объектов в пространстве, соответствующим образом выравнивая виртуальные модели персонажей.



## Ghostwire

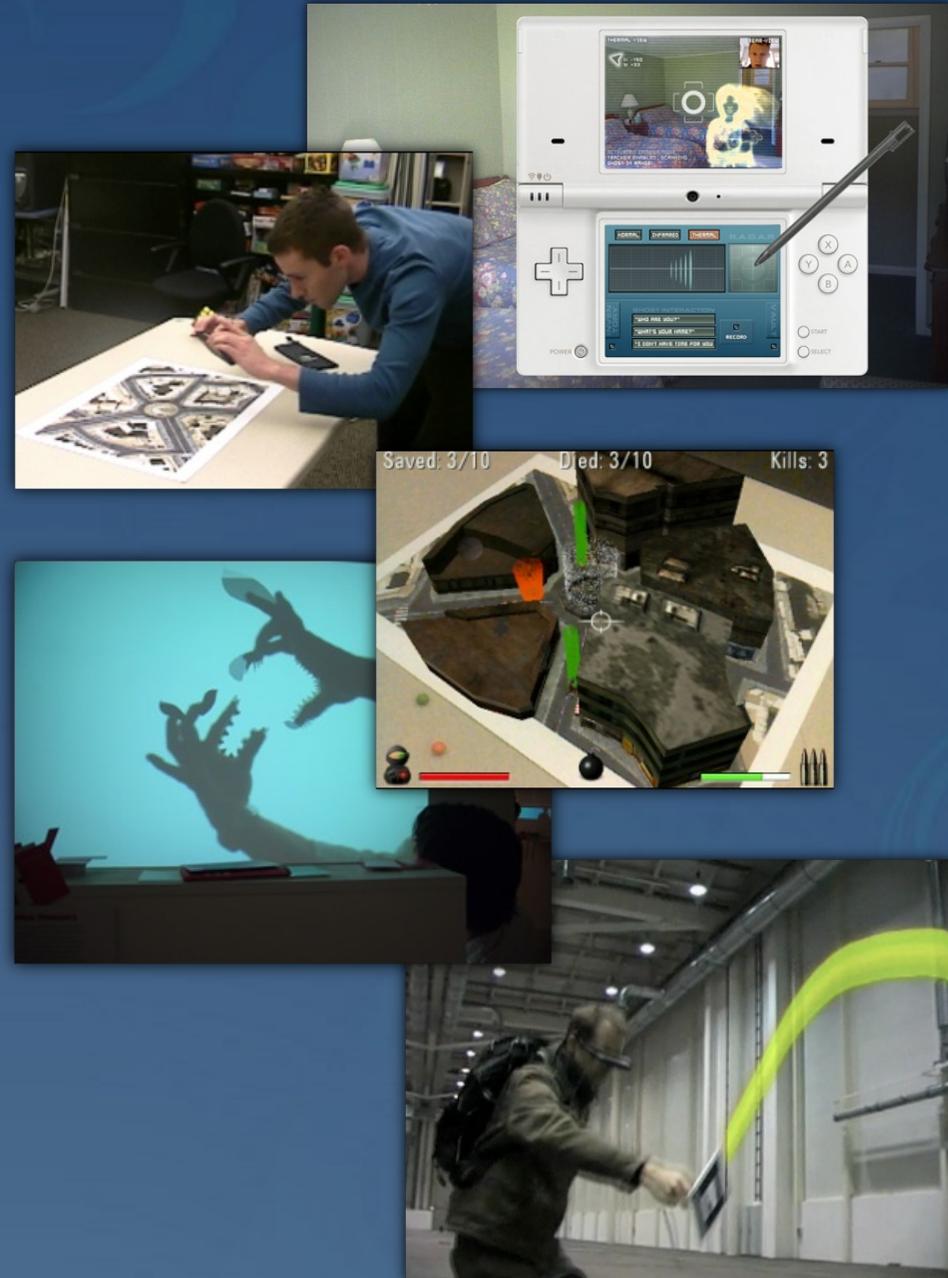
Издательство Majesco анонсировало игру Ghostwire: Link to the Paranormal для Nintendo DSi (для игры используется встроенная камера, которая есть в DSi, но отсутствует в обычной Nintendo DS). Консоль используется для поиска призраков и духов в реальном окружении. На результат охоты влияет время суток и уровень освещенности. Игрок может заносить найденных призраков в каталог и выяснять, почему они задержались в нашем мире, а также решать головоломки, помогая привидениям обрести покой.

## Arhrrrr!

В экспериментальном шутере Arhrrrr! игрок «находится» в вёртолёте, парящем над заброшенным городом и отстреливает злобных зомби. Виртуальный город на экране выравнивается по маркеру в виде карты этого города, распечатанной на бумаге, а в качестве «оружия» можно использовать разноцветные конфетки «Skittles». Игра реализована на платформе nVidia Tegra. Arhrrrr! уже окрестили настоящей революцией в жанре шутеров.

Дополненная реальность используется также современными художниками-экспериментаторами. Некоторые наиболее интересные результаты, получившиеся в ходе разработок в этом направлении, заслуживают отдельного упоминания. Так, инсталляция Филипа Вортингтона Shadow Monsters «превращает» тени от человеческих рук в силуэты забавных монстров, добавляя рога, клыки и уши. Известный немецкий граффитист Daim использует камеры, отслеживающие движения его руки — данные передаются специальному программному обеспечению, и на экране возникает виртуальное трехмерное граффити.

**Gecko**  
[gecko0307@gmail.com](mailto:gecko0307@gmail.com)





# «Талисманы» МАШИНОЙ ГРАФИКИ

В любой области человеческой деятельности есть свои исторически сложившиеся стандарты, традиции, даже суеверия. Компьютерная графика — еще достаточно молодая отрасль, но и в ней уже сменилось не одно поколение; а значит, сформировались и определенные традиции. В этом небольшом обзоре речь пойдет о наиболее распространенных и ставших эталонными объектах — «талисманах» компьютерной графики. Если у Вас есть что добавить или Вы можете рассказать о других, не упомянутых здесь «талисманах», просим писать на почтовый адрес редакции: [clocktower89@mail.ru](mailto:clocktower89@mail.ru) (или лично главному редактору: [gecko0307@gmail.com](mailto:gecko0307@gmail.com)).

## Чайник Юта (The Utah Teapot)

Чайник Юта (или чайник Ньюэлла) стал одним из эталонных объектов в трехмерной компьютерной графике. Это простая, округлая, сплошная и частично вогнутая математическая модель обычного заварного чайника.

Модель была создана в 1975 году Мартином Ньюэллом, участником программы исследований в компьютерной графике в Университете Юты (США). Ньюэлл нуждался для своей работы в умеренно простой математической модели знакомого объекта. Жена Ньюэлла, Сандра, предложила смоделировать их чайный сервис, так как в тот момент они пили чай. Мартин взял миллиметровку и карандаш и зарисовал весь сервис на глаз, затем, вернувшись в лабораторию, вручную ввел контрольные точки Безье на трубке памяти Tektronix. Хотя вместе со знаменитым чайником были оцифрованы чашка, блюдце и чайная ложка, один лишь чайник добился повсеместного использования. Считается, что также был смоделирован молочник, но данные о нем были утеряны.

Модель чайника состоит из 32 порций бикубической поверхности Безье, координаты опорных точек которых и являются исходным описанием модели. Точки образуют массив из 306 элементов. Основной объём чайника (корпус) образован из 12 порций бикубической поверхности Безье, ручка — из следующих четырех, следующие четыре порции формируют носик. На крышку чайника ушло восемь порций. Оставшиеся четыре образуют донышко.

Эти данные широко распространены среди специалистов по трехмерной компьютерной графике и уже много лет используются для демонстрации работы и при проверке алгоритмов. Чайник часто можно встретить в числе других стандартных примитивов в различных графических пакетах. Он даже появлялся в известном анимационном фильме «История игрушек».





### Стэнфордский кролик (The Stanford Bunny)

Другую популярную тестовую модель создали в 1994 г. сотрудники Стэнфордского Университета (США) Грег Турк и Марк Левой в ходе исследований по оцифровке трехмерных объектов путем объемного сканирования (соответствующая статья была представлена на SIGGRAPH '94). В качестве исходного объекта для сканирования была использована керамическая статуэтка кролика.

Оригинал модели распространяется в формате .ply в четырех вариантах различной степени детализации. Наиболее высокополигональный вариант состоит из 69451 треугольника.

Стэнфордский кролик, как правило, используется в тестовых и демонстрационных программах для произведения различных манипуляций с вершинами (таких, как тесселяция и сглаживание), а также для проверки шейдеров и алгоритмов рендеринга (рельеф, отражение и преломление света, мех и т.д.)





### Сюзанн (Suzanne)

Обезьянка Сюзанн хорошо знакома всем пользователям популярного трехмерного редактора Blender. Она входит в состав стандартных примитивов программы начиная с версии 2.25 — это был последний релиз Blender как закрытого продукта холдинга NaN в 2002 г. Модель изготовил Вилльм-Пауль ван Овербругген, а свое имя обезьянка получила в честь орангутана из фильма «Джей и Молчаливый Боб наносят ответный удар».

Модель состоит всего из 500 полигонов и используется в качестве оригинальной альтернативы традиционному чайнику Юта. Наиболее распространенное применение — демонстрация материалов и эффектов освещения.

Кстати, на ежегодном фестивале анимации Blender вручается награда под названием Suzanne Awards.

### Лена (Lena)

Специалисты в области обработки изображений часто используют в качестве тестовой картинкой фотографию Лены — симпатичной молодой девушки в шляпке, смотрящую на вас вполборота. Этот снимок знаком почти всем — однако о том, кто такая эта Лена и как она появилась на страницах самых уважаемых научных журналах, знают немногие.

Всё началось с того, что в начале 1970-х исследователи из Университета Южной Калифорнии искали тестовое изображение для использования в работе, которая послужила в будущем основой для стандартов JPEG и MPEG. Один из сотрудников лаборатории принёс свежий номер журнала «Playboy» с фотографией Лены Содерберг (Lena Sjöblom), Мисс «Playboy» за ноябрь 1972 года. На фотографии — красивая девушка в шляпке и сапожках, которая стоит вполборота к зеркалу. Лена всем понравилась, и разворот «Плэйбоя» поместили в сканер. Так как сканер был барабанным, с разрешением 100 линий на дюйм, и инженерам была нужна картинка 512x512, то они отсканировали только верхние 5.12 дюйма снимка. Как выяснилось позже, «неизвестным исследователем», предложившим отсканировать фотографию Лены из «Playboy», был не кто иной, как Уильям К. Прэтт, автор бессмертной книги «Цифровая обработка изображений».

Фотография Лены очень легко прижилась, стала широко распространённым и узнаваемым, и многие исследователи даже не задумывались, откуда оно взялось. «Лена, просто Лена» стала настолько привычной и родной, что многие считают её Мадонной информационной эпохи. Что только не делали с этим изображением — сжимали, размывали, выделяли границы, корректировали цвета, деформировали и разбивали на регионы...

Главный редактор журнала «IEEE Transactions of Image Processing», Дэвид Мунсон, отчасти объяснил, почему изображение Лены такое удачное. Во-первых, оно содержит хорошую смесь деталей, однотонных областей и текстур, что служит прекрасным материалом для тестирования алгоритмов обработки изображений. Во-вторых, это фотография очаровательной девушки — не удивительно, что сообщество исследователей, в большинстве своем мужское, так держится за это изображение.

А тем временем «Playboy» даже не подозревал о том, что фрагмент их изображения гуляет широкими тиражами по другим журналам - до момента, пока пресловутая фотография не появилась на обложке «Optical Engineering». Тут проснулся владелец копирайта, и начал грозить судом научным журналам, в том числе и «IEEE Transactions of Image Processing», одному из наиболее уважаемых журналов в области обработки изображений. Произошедшее вызвало оживленные дебаты в научном сообществе - никто не хотел отдавать Лену!

К счастью, с редакцией «Playboy» удалось договориться и ситуация разрешилась мирно: изображение с тех пор можно получить и использовать совершенно легально. Более того, журнал «Playboy» стал гордиться своей ролью, помог разыскать Лену (она так и жила в Швеции) и пригласить ее на юбилейную 50-ую конференцию Society for Imaging Science and Technology, проходившую в 1997 году, где ее представили как «Первую Леди Интернета».



Лена Содерберг, ныне солидная замужняя дама, до сих пор живет в Швеции с семьей и тремя детьми. Интернетом она никогда не пользовалась и поэтому не подозревала о своей славе — до тех пор, пока ей не рассказал об этом репортер, принимавший у нее интервью. Лена была приятно удивлена тем, что случилось с ее фотографией прошлых лет - так она впервые узнала о своей огромной популярности в научных кругах. Для нее было большим сюрпризом, что на конференции серьезные с виду ученые мужи толпятся вокруг нее, ожидая автографа и надеясь с ней сфотографироваться. «Как же я, наверно, надоела вам за эти годы..» — сказала смущенная Лена в своей речи.

У Лены теперь есть свой сайт, на котором много интересного: и рассказы о жизни, и неожиданная слава, и, конечно, принесшая ей славу легендарная фотография в полный рост.

А «Playboy» за ноябрь 1972 г. стал самым продаваемым номером в истории журнала (7161561 экземпляр).

# Tr3s Lunas

## Музыкальная реальность Майка Олдфилда

Майк Олдфилд (полное имя — Майкл Гордон Олдфилд) — легендарный британский композитор и мультиинструменталист, работающий в самых разнообразных жанрах: нью-эйдж, эмбиент, прогрессив-рок, арт-рок, фолк и др. Главным произведением Олдфилда считается концептуальный альбом «Tubular Bells» (1973). Новаторское использование модифицированных католических трубчатых колоколов и уникальное звучание его немного перегруженных гитар наряду с огромным количеством монтажных склеек и наложений при записи сделало стиль Олдфилда узнаваемым и стало основой его музыкального пути. Отдельные композиции с альбома были использованы при записи саундтрека к знаменитому фильму «Изгоняющий дьявола». В 2002 году «Tubular Bells» был включен в рейтинг The 25 Most Influential Ambient Albums Of All Time («25 лучших альбомов всех времён в стиле эмбиент»). Годом позже пластинка была заново перезаписана и издана в несравненно лучшем качестве. Среди песен, написанных Майком, наиболее известны «Moonlight Shadow» и «To France», с вокалом Мэгги Рейли.

Музыкальную карьеру Майк начал в 14 лет: вместе с сестрой Салли он организовал фолк-дуэт Sallyangie. Группа записала несколько синглов и альбом, после чего последовал небольшой концертный тур. В 1969 Майк присоединяется в качестве гитариста к Кэвину Эйрсу и группе The Whole World, в составе которой работает над четырьмя альбомами.

В 1971 году группа распалась и Олдфилд начал сольную карьеру. Сейчас в дискографии Олдфилда уже 25 студийных альбомов.

Tubular Bells вышел в июне 1973 года. Альбом был представлен в лондонском Queen Elizabeth Hall. Критики и публика приняли абсолютно новую музыку с большим энтузиазмом, и продажи альбома подтвердили это — свыше 2 миллионов копий за год и около 16 миллионов копий на сегодняшний день. Альбом стал самым продаваемым в Англии в 1973 году, сразу попав на верхние строчки мировых чартов.

Другой ранний альбом композитора — вышедший в 1975 году «Ommadawn» — был провозглашен «шедевром века», а Майка пресса нарекла «Волшебником тысячи наложений». Впервые Олдфилд позволил принять участие в создании своей музыки другим исполнителям.

Премьера альбома «Tubular Bells II» состоялась 4 сентября 1992 г. в старинном эдинбургском замке. Комбинация великолепной музыки, энтузиазм фанатов и вдохновляющий финал с огромным фейерверком сделали этот концерт самым запоминающимся в карьере музыканта. Шоу было записано на видео и с тех пор неоднократно транслировалось по всему миру.

Два года спустя «The Songs Of Distant Earth» сломал все шаблоны музыкальной индустрии — он стал первым коммерческим альбомом, включившим в себя элементы виртуальной реальности. Мультимедийное приложение к альбому позволяло слушателю решить небольшую головоломку и разблокировать секретные видеоклипы. Майк пошел дальше и начал работу над полноценным интерактивным видеоальбомом.

Программное обеспечение для воссоздания виртуального окружения разрабатывалось на мощных станциях Silicon Graphic. Персональные компьютеры того времени были еще недостаточно мощны для обработки сложной трехмерной графики в реальном времени, и вышло так, что задумки Олдфилда опережали время. Первоначальные разработки в этом направлении так и не увидели свет — этому помешали не только существующие технические ограничения, но и нежелание Олдфилда привязывать потенциальных пользователей к какому-то конкретному оборудованию (притом, достаточно дорогому и громоздкому).

К началу «нулевых» данная проблема решилась сама собой со стремительным развитием 3D-ускорителей для компьютеров. Олдфилд выпускает Tres Lunas (в переводе с испанского — «Три Луны») — первую компьютерную игру для PC в рамках проекта, получившего название MusicVR — «музыкальная виртуальная реальность» (изначально — SonicVR, или «звуковая виртуальная реальность»).

Майк неоднократно обращался в издательские фирмы, но ни одна компания игрой не заинтересовалась. Все посчитали, что игра, в которой не нужно стрелять, не вызовет интереса у покупателей. И тогда Майк принимает решение сделать игру частью своего нового альбома. Альбом «Tres Lunas» вышел в начале 2002 г. и состоит из двух дисков. Первый диск — собственно музыка, второй — CD-ROM с демо-версией игры. В этом же году вышел бутлег «Tres Lunas II», куда вошли треки из игры.

Год спустя вышел перезаписанный с помощью цифровых технологий «Tubular Bells 2003». Этот ре-релиз приурочен к тридцатилетней годовщине выхода легендарного альбома, а так же юбилею самого Майка. Альбом вышел в нескольких вариантах, в том числе и в формате DVD-Audio 5.1.

Пластинка послужила вдохновителем к созданию новой игры из серии MusicVR — Maestro (изначально она называлась The Tube World). Наряду с эксклюзивным музыкальным контентом, в игру вошли композиции с классического «Tubular Bells».

Как и Tres Lunas, Maestro поддерживает мультиплеерный режим. Игроки при этом имеют личные 3D-аватары, а также возможность общаться друг с другом во встроенном чате. Иногда к игре даже подключается сам Олдфилд — фанаты получают уникальную возможность пообщаться с мастером «вживую».

Полные версии игр сейчас абсолютно бесплатно можно скачать на фан-сайте tubular.net:

[http://tubular.net/dist/setup\\_treslunas.exe](http://tubular.net/dist/setup_treslunas.exe)

[http://tubular.net/dist/setup\\_maestro.exe](http://tubular.net/dist/setup_maestro.exe)



## Tres Lunas

**Разработчик:** Oldfield Interactive Ltd.

**Платформа:** PC

**Дата релиза:** 07.09.2002

**Жанр:** квест / виртуальная реальность

**Системные требования:** Intel Pentium III 500 МГц,  
64 Мб ОЗУ, видеокарта 16 Мб с поддержкой OpenGL,  
ОС Windows (работает также в окружении Wine под Linux)

Игра совмещает музыку Олдфилда и сюрреалистическое окружение — вулканы, космические пейзажи, пещеры, причудливый ландшафт. Играющий путешествует в этом мире в поисках семи магических золотых колец. Цель чисто символическая и условная, так как основной смысл «музыкальной виртуальной реальности» — погружение в искусственно созданную Вселенную, исследование и познание ее законов. Это не похоже на обычную компьютерную игру: от игрока не требуется спасти мир или бороться со злом. Вы просто путешествуете, слушаете музыку и наслаждаетесь неповторимой атмосферой. Tres Lunas часто сравнивают с Myst.

В игре нет оружия, врагов, опасностей. Встречающиеся на пути существа и интерактивные объекты могут переносить вас в пространстве и времени, а также оказывать другие полезные действия. Некоторые из них созданы просто для красоты.

К 2006 г. в проект было вложено, в общей сложности, 26 млн. фунтов стерлингов. Tres Lunas разрабатывалась только для PC — несмотря на то, что Олдфилд использует компьютеры Apple для записи музыки. Игра была создана при помощи специально разработанной визуальной среды Newlook. Совместно с Майком над проектом работали художник Ник Кэтчсайд и программист Колин Дули, сотрудник Silicon Graphics. «Обычно я отправлял ему e-mail и просил добавить очередную кнопку — например, чтобы создать область вокруг 3D-объекта и выбрать звуковой файл, который должен в ней воспроизводиться.» — вспоминает Майк, — «Представьте несколько тысяч таких кнопок, и вам станет ясно, как все это делалось.»



Игра неотрывно связана с музыкой Олдфилда и составляет с ней единое целое. Красной нитью через нее проходит символика творчества и внутреннего мира Майка — изогнутые трубчатые колокола, гитары, пианино, лошади, самолеты, даже космический корабль... Известно, что Майк увлекается авиацией (он неплохой пилот и парашютист), а также научной фантастикой и лошадьми. А слон на длинных тонких ногах — явная дань искусству Сальвадора Дали.

Название альбома иногда пишется как «Tr3s Lunas» — на обложке литера «e» заменена на цифру 3.



Вот как сам Олдфилд прокомментировал композиции альбома:

### 1. Misty (03:58)

*«Мисти — имя одной из моих лошадей. У меня много свободного времени, которое я обычно посвящаю своим хобби. Мне давно хотелось держать красивых животных. Мисти — это молодая арабская лошадь, я купил её, когда ей было четыре месяца. Сейчас [на помент взятия интервью — прим. ред.] ей уже год, мы даже отпраздновали её день рождения. Ей подарили морковный «торт» с одной свечкой. Мне нравится ухаживать за животными, я люблю их.*

*«Misty» — это первая песня, где я использовал новую гитарную технику. Я называю её «саксофон-гитарный» звук, получается почти как новый инструмент. Он есть в нескольких композициях альбома. Я люблю гитару, техника игры, вместе с клавишными и компьютером позволяют получить звучание виртуального саксофона, смешанного с гитарными аккордами.»*

### 2. No Mans Land (06:06)

*«Я люблю пустыню. Я много раз бывал в пустынях, когда жил в Калифорнии, ездил в Марокко и Мексику. Мне нравится чувство отчуждения, ощущение того, что ты далек от всего, сделанного рукой человека. Мне нравится эта пустота — это особое чувство. «No Mans Land» — это музыкальное воплощение пустынных прерий Соединенных Штатов. В игре MusicVR пустыня представлена двумя частями, каждая из которых открывает перед игроком множество действий.»*



### 3. Return To Origin (04:39)

*«Это один из лейтмотивов — как альбома, так и игры. «Origin» — это центральная точка трехмерного мира Tr3s Lunas. Иногда необходимо вернуться к началу — только так можно пройти некоторые эпизоды игры. Для меня это отражение возвращения к первоначальному состоянию, духовный взгляд на то, чем мы, люди, являемся на самом деле, взгляд в центр. Иногда вы должны забыть материальные ценности жизни, потому что они могут быть причиной стрессов и несчастий. Мы должны постараться возвратиться к началу или, по крайней мере, пытаться делать это как можно чаще.»*

#### **4. Landfall (02:17)**

«Это почти эмбиент. Музыка соответствует моменту игры, когда ты можешь покинуть пустыню, собрав достаточно колец, чтобы попасть на крошечный островок по середине одного из морей. «Landfall» – это старинное выражение моряков, таких как Христофор Колумб, которым они называли возвращение к земле после долгого морского путешествия.»

#### **5. Viper (04:32)**

«Это первая композиция альбома, которая полностью меня удовлетворила. Она достаточно ритмична, в ней есть особое настроение, которое мне хотелось передать этим альбомом. Это песня одной из змей в игре, гадюки. В Tubular Bells III тоже есть змеи. Может быть, это моя личная страсть, так как, согласно китайскому гороскопу, я тоже змея.»

#### **6. Turtle Island (03:41)**

«Эта композиция несомненно связана с Ибицей и «Café Del Mar». Это очень простая, легко привязывающаяся мелодия. Не могу представить лучшей музыки для бара, где ты сидишь, выпиваешь — просто отдыхаешь, наслаждаясь жизнью.

Эта песня занимает особое место в игре. Когда попадаешь на остров, то видишь черепаху. Приблизившись к ней, слышишь «Turtle Island», и внезапно вокруг тебя появляются множество маленьких черепашек, которые начинают играть жемчужинами и устрицами. Если положить три жемчужины в черепаший панцирь, то получишь возможность поплавать с дельфином. В этой игре все взаимосвязано.»

#### **7. To Be Free (04:22)**

«Сначала это была простая акустическая гитарная тема, но потом я решил усилить ритм и прибавить ударные. Затем мы решили добавить вокал. Я договорился с вокалисткой и написал слова за два часа до её приезда в студию. Нам повезло, мы сработали всё быстро и легко. Нам понравилось великолепное звучание, и мы решили сделать эту песню первым синглом.»

«Сочиняя слова, я думал о своих детях. У меня трое детей, сейчас они как раз решают, чему посвятить себя в жизни. Я знаю многих людей разных возрастов у которых та же ситуация. По моему, «To Be Free» – это гимн для тех, у кого в жизни много проблем. В песне говорится о тех моментах, когда в самый неожиданный момент все вокруг тебя изменяется самым необычным образом.»

#### **8. Firefly (03:46)**

«Это очень необычная песня, она легко создает образ: ты прямо видишь светлячков вокруг. «Светящаяся» - лучшее определение этой песни. Она состоит из очень простых гитарных аккордов, но играл я по особому, будто бы лаская струны. Я давно мечтал добиться такого звучания, и наконец-то получил его. Похожую технику я использовал в Tubular Bells, но сейчас звук получился более ясным, выделяясь на передний план.

В игре светлячкам отведена довольно важная роль. Они вылетают из пещер и кратеров, помогая найти пути в другие миры виртуальной реальности.»

#### **9. Tres Lunas (04:35)**

«Это, пожалуй, главная песня альбома. Когда меня просят объяснить, почему я выбрал такое название, я вспоминаю время, когда жил на Ибике. Там есть великолепный ресторан, его называют «Las Dos Lunas», это между Сан-Антонио и Ибика-сити. Владелец этого ресторана одолжил мне сборник, который мы часто слушали в те дни. Там было множество простых гитарных мелодий, я даже узнал там некоторые свои работы. Тогда я и решил записать незатейливый альбом.

Мне пришлось перезаписывать композицию пять или шесть раз, прежде чем она меня полностью устроила. Главным было сделать её как можно проще. Мне хотелось передать ощущения человека, который живет на необычной планете и вдруг обнаруживает на небе три луны вместо одной...»

### **10. Daydream (02:15)**

*«Эта песня навевает на меня воспоминания. Меня вдохновил уик-энд в Венеции два года тому назад. Тогда я побывал на острове Бурано, я нашел там очень необычное кладбище, очень тихое. Я видел могилы Игоря Стравинского и Сергея Дягилева (Дягилев был хореографом у Стравинского). Эта особая атмосфера очень впечатлила меня тогда. Меня всегда притягивало искусство конца XIX – начала XX в. Я, например, обожаю сюрреализм и работы Сальвадора Дали. В то же время на «Daydream» меня вдохновила восхитительная музыкальная пьеса Эрика Сати «Gymnopédies.»»*

### **11. Thou Art In Heaven (05:23)**

*«У этой песни длинная история. Основная идея возникла, когда я получил предложение написать музыку для новогоднего концерта в Берлине, она превратилась в альбом «The Millennium Bell». Но тогда эту идею не возможно было реализовать до конца. Сейчас она очень хорошо подходит для проекта. На название «Thou Art In Heaven» меня вдохновил стих из молитвы.»*

### **12. Sirius (05:47)**

*«Эта песня о звезде под названием Сириус, которая всегда меня очаровывала. Это очень большая звезда, бриллиант, – идеально подходит для Трех Лун. Сириус – это также имя одной из моих лошадей. Это тоже арабский конь, как и Мисти.»*

Композиция «Thou Art In Heaven», кстати, впервые прозвучала на знаменитом тысячелетнем концерте Олдфилда в Берлине 31 декабря 1999 г., который сопровождался грандиозным лазерным шоу. Название «Art In Heaven» («Искусство на небесах») принадлежало компании, которая занималась организацией концерта. Трек начинается темой из «In the Beginning» с альбома «The Songs of Distant Earth» и заканчивается «Одой радости» Бетховена...

**Gecko**  
[gecko0307@gmail.com](mailto:gecko0307@gmail.com)

# Модели освещения

## Диффузное рассеивание по Ламберту

Свет - это слишком сложная система, чтобы смоделировать ее в совершенстве. Именно поэтому мы довольно редко можем встретить созданные компьютером трехмерные изображения, которые были бы по-настоящему фотореалистичны. Во всех случаях, чем сложнее и реалистичнее создаваемая виртуальная сцена, тем больше вычислений необходимо произвести, и тем медленнее она будет воспроизводиться. Как программист, вы должны будете решить, чем вы больше готовы пожертвовать: качеством изображения или скоростью его просчета на компьютере.

Свет состоит из мельчайших частиц, называемых фотонами. Фотон имеет свойства, присущие как волнам, так и частицам. Фотоны отрываются от источника энергии и прямолинейно распространяются в пространстве, пока не произойдет столкновение с внешним объектом в пространстве.

Фотонов очень много. Так много, что обычно говорят — их неопределенно много. Исходя из этого, мы можем пренебречь фактом, что свет состоит из единичных фотонов и рассмотреть свет как непрерывный поток энергии. В этом случае к свету можно применить статистические законы, и полученные результаты будут достаточно точны именно благодаря огромному количеству вовлеченных фотонов. Таким образом, свет может быть смоделирован на компьютере.

Взаимодействие светового потока с окружающими предметами (объектами) позволяет нам видеть их. Свет исходит из источника световой энергии. Триллионы фотонов вырываются и с огромной скоростью уносятся от источника, взаимодействуя с предметами, ударяя каждую мельчайшую деталь окружающей обстановки. Небольшое количество из них попадает в маленькое темное пятно в середине нашего глаза. Это зрачок. Глаз устроен таким образом, что он несколько подправляет направление движения фотона перед тем, как он достигнет задней части глаза. Здесь фотон поглощается светочувствительными рецепторами. Эти рецепторы дают соответствующие сигналы нашему мозгу. Мозг интерпретирует поступившую последовательность сигналов и снабжает нас подробной информацией о нашем окружении. Изображение, которое мы видим на самом деле, не является соответствующим ему набором физических предметов. Все, что мы получаем, на самом деле лишь его энергетический отпечаток, который прошел огромное количество сложнейших преобразований в нашем мозге. Синий объект — не есть в действительности синий. Он считается синим потому, что мы интерпретируем свет, пришедший от него, как синий. В этом заключается разница между Данными (Data) и Информацией (Information). Данные — это световой образ, формирующийся на сетчатке глаза. Информация — это интерпретация этого образа нашим мозгом.

Разнообразие моделей освещения, применяемых в процессе формирования изображений компьютером, — это попытка увеличить количество информации, которую мозг сможет извлечь. В данном цикле статей будут рассмотрены распространенные модели освещения, которые могут быть эффективно реализованы современными средствами графики реального времени.

Световая энергия, падающая на поверхность, может быть поглощена, отражена или пропущена. Частично она поглощается и превращается в тепло, а частично отражается или пропускается. Объект можно увидеть, только если он отражает или пропускает свет; если же объект поглощает весь падающий свет, то он невидим и называется абсолютно черным телом. Количество поглощенной, отраженной или пропущенной энергии зависит от длины волны света. При освещении белым светом, в котором интенсивность всех длин волн снижена примерно одинаково, объект выглядит серым. Если поглощается почти весь свет, то объект кажется черным, а если только небольшая его часть - белым. Если поглощаются лишь определенные длины волн, то у света, исходящего от объекта, изменяется распределение энергии и объект выглядит цветным. Цвет объекта определяется поглощаемыми длинами волн.



Свойства отраженного света зависят от строения, направления и формы источника света, от ориентации и свойств поверхности. Отраженный от объекта свет может также быть диффузным или зеркальным. Диффузное отражение света происходит, когда свет как бы проникает под поверхность объекта, поглощается, а затем вновь испускается. При этом положение наблюдателя не имеет значения, так как диффузно отраженный свет рассеивается равномерно по всем направлениям. Иными словами, такое освещение *изотропно*.

## Как отражается свет

### Белый свет

- цвет падающего света

- цвет поверхности

### Красный свет

- цвет падающего света

- цвет поверхности

Комплиментарные (дополняющие) цвета НЕ отражают друг друга (только поглощают) - на выходе образуется черный цвет.  
Черная же поверхность ничего не отражает.

Простейшая модель освещения - идеальное диффузное рассеивание. Освещенность в точке поверхности определяется только интенсивностью отраженного света, а она линейно зависит от косинуса угла между направлением света и нормалью к поверхности (закон Ламберта):

$$I = I_t k_d \cos\theta_l$$

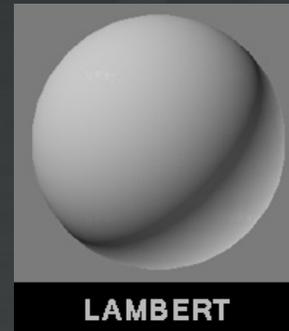
где  $I$  – интенсивность отраженного света,  $I_t$  – интенсивность точечного источника света,  $k_d$  – коэффициент диффузного отражения,  $\theta_l$  - угол между направлением света и нормалью к поверхности.

Коэффициент диффузного отражения зависит от материала и длины волны света, но в простых моделях освещения обычно считается постоянным.

Предполагается, что источник света точечный, поэтому объекты, на которые не падает прямой свет, кажутся черными. Однако на объекты реальных сцен падает еще и рассеянный свет, отраженный от окружающей обстановки, например от стен комнаты. Рассеянному свету соответствует распределенный источник. Поскольку для расчета таких источников требуются большие вычислительные затраты, в графике реального времени они заменяются на коэффициент рассеяния - константу, которая входит в формулу в линейной комбинации с членом Ламберта:

$$I = I_a k_a + I_t k_d \cos\theta_l$$

где  $I_a$  - интенсивность рассеянного света,  $k_a$  - коэффициент диффузного отражения рассеянного света.



Косинус угла между нормалью к поверхности в точке и направлением на источник света соответствует скалярному произведению этих двух векторов (при условии, что векторы нормализованы, т.е. имеют единичную длину):

$$\cos\theta_l = (n, l)$$



*В природе нет диффузных материалов, стопроцентно отражающих весь падающий свет. Светоотражение самой белой ламинированной бумаги не превышает 85%.*

*Даже самый яркий искусственный материал — спектралон — находящий применение в оптике и космонавтике, имеет светоотражение 99% в видимой части спектра. Диффузное рассеивание спектралона почти полностью соответствует закону Ламберта.*

## Реализация на GLSL\*

Вершинная программа:

```
varying vec4 V_eye;
varying vec4 L_eye;
varying vec4 N_eye;
void main(void)
{
    V_eye = gl_ModelViewMatrix * gl_Vertex;
    L_eye = gl_LightSource[0].position - V_eye;
    N_eye = vec4(gl_NormalMatrix * gl_Normal, 1.0);
    gl_Position = ftransform();
    V_eye = -V_eye;
}
```

Фрагментная программа:

```
varying vec4 V_eye;
varying vec4 L_eye;
varying vec4 N_eye;
void main(void)
{
    vec4 Ca = gl_FrontMaterial.ambient;
    vec4 Cd = gl_FrontMaterial.diffuse;
    vec3 L = normalize(vec3(L_eye));
    vec3 N = normalize(vec3(N_eye));
    float diffuse = clamp(dot(L, N), 0.0, 1.0);
    gl_FragColor = Ca + (Cd*diffuse);
    gl_FragColor.a = 1.0;
}
```

Gecko  
gecko0307@gmail.com

\* - предполагается, что в приложении уже включены и настроены соответствующие параметры OpenGL (источник света, материал и др.)

# Ваш собственный 3D-формат

Загрузка внешних моделей в движок — одна из наиболее актуальных проблем для начинающего игрового программиста (если, конечно, не используется готовый движок вроде Irrlicht или Ogre — тогда никаких проблем нет, но это немного иной случай).

Существует ряд более-менее стандартных 3D-форматов, которые «понимают» практически все крупные пакеты моделирования. Однако нет ни одного универсального: одни предназначены больше для обмена данными, чем для игр (например, Collada), другие заточены под определенную программу, третьи — под определенный движок... К сожалению, на данный момент еще нет открытого стандарта для легковесного, универсального формата моделей (хотя работы ведутся). Поэтому, если у вас нет желания писать загрузчики для существующих форматов, остается разработать свой.

При создании своего формата моделей обычно сначала пишут его экспортер для какой-либо программы моделирования. В этом отношении весьма дружелюбен Blender: встроенный Python API предоставляет доступ к внутренней структуре трехмерной сцены, включая интерфейс на уровне объектов, мешей, полигонов и отдельных вершин. Для Blender уже написано довольно много скриптов экспорта в разнообразные форматы — но они довольно сложны, начинающему изучать API по ним не очень удобно. А простого, наглядного скрипта, на основе которого можно писать свой экспортер, почему-то нет. Вот именно эту ситуацию мне и хотелось бы исправить.

Приведенный скрипт предназначен для Blender 2.54b. Его необходимо поместить в директорию 2.54/scripts/io (относительно каталога, в который был распакован дистрибутивный архив программы). Обладая минимальными знаниями языка Python, вы легко сможете в нем разобраться.

```
#"""  
#Name: 'Custom model format (.model)...'  
#Blender: 254  
#Group: 'Export'  
#Tooltip: 'Export to custom format'  
#  
#"""  
  
bl_addon_info = {  
    "name": "Export to custom format (.model)",  
    "author": "Timur Gafarov aka Gecko",  
    "version": (0, 0, 1),  
    "blender": (2, 5, 4),  
    "api": 31847,  
    "location": "File > Export > Custom (.model)",  
    "description": "Export to custom format (.model)",  
    "warning": "",  
    "wiki_url": "",  
    "tracker_url": "",  
    "category": "Import/Export"}  
  
import bpy,struct,math,os,time,sys,mathutils  
  
class modelSettings:  
    def __init__(self, savepath):  
        self.savepath = savepath
```

```

def save_model(settings):
    print("Exporting selected object...")
    bpy.ops.object.mode_set(mode = 'OBJECT')
    bpy.ops.object.shade_smooth()
    obj = bpy.context.selected_objects[0]
    if ((obj.type == 'MESH') and ( len(obj.data.vertices.values()) > 0 )):
        print( "Processing mesh: " + obj.name )

        buffer = "MODEL \"%s\"\n" % (obj.name)

        mesh = obj.data.vertices.id_data

        buffer = buffer + "VERTICES %i\n" % (len(obj.data.vertices.values()))
        buffer = buffer + "FACES %i\n" % (len(obj.data.faces.values()))

        for vertex in mesh.vertices:
            buffer = buffer + "VERTEX %f %f %f\n" % (vertex.co[0],vertex.co[1],vertex.co[2])

        for vertex in mesh.vertices:
            buffer = buffer + "NORMAL %f %f %f\n" % (vertex.normal[0],vertex.normal[1],vertex.normal[2])

        for face in obj.data.faces:
            buffer = buffer + "FACE %i %i %i " % (face.vertices[0],face.vertices[1],face.vertices[2])
            buffer = buffer + "UV1 %f %f " % (obj.data.uv_textures.active.data[face.index].uv[0][0], obj.data.uv_textures.active.data[face.index].uv[0][1])
            buffer = buffer + "UV2 %f %f " % (obj.data.uv_textures.active.data[face.index].uv[1][0], obj.data.uv_textures.active.data[face.index].uv[1][1])
            buffer = buffer + "UV3 %f %f\n" % (obj.data.uv_textures.active.data[face.index].uv[2][0], obj.data.uv_textures.active.data[face.index].uv[2][1])

        model_filename = settings.savepath + ".model"
        if (model_filename != ""):
            try:
                file = open(model_filename, 'w')
            except IOError:
                errmsg = "IOError"
            file.write(buffer)
            file.close()
            print( "saved mesh to " + model_filename )
        else:
            print( "No model file was generated." )

```

```

from bpy.props import *
class ExportMODEL(bpy.types.Operator):
    """Export to custom format (.model)"""
    bl_idname = "export.model"
    bl_label = 'Export to custom format'

    logenum = [("console", "Console", "log to console"),
               ("append", "Append", "append to log file"),
               ("overwrite", "Overwrite", "overwrite log file")]

    filepath = StringProperty(subtype = 'FILE_PATH', name="File Path", description="Filepath for exporting", maxlen= 1024, default= "")

    def execute(self, context):
        settings = modelSettings(savepath = self.properties.filepath)
        save_model(settings)
        return {'FINISHED'}

    def invoke(self, context, event):
        WindowManager = context.window_manager
        WindowManager.add_fileselect(self)
        return {"RUNNING_MODAL"}

    def menu_func(self, context):
        default_path = os.path.splitext(bpy.data.filepath)[0]
        self.layout.operator(ExportMODEL.bl_idname, text="Custom format (.model)", icon='BLENDER').filepath = default_path

    def register():
        bpy.types.INFO_MT_file_export.append(menu_func)

    def unregister():
        bpy.types.INFO_MT_file_export.remove(menu_func)

if __name__ == "__main__":
    register()

```

В результате работы скрипта вы получите файл \*.model примерно такого содержания:

```
MODEL "Model name"
VERTICES 8
FACES 12
VERTEX 1.000000 1.000000 -1.000000
VERTEX 1.000000 -1.000000 -1.000000
...
NORMAL 0.333323 0.666646 -0.666646
NORMAL 0.816492 -0.408246 -0.408246
...
FACE 4 0 7 UV1 0.000000 0.000000 UV2 1.000000 0.000000 UV3 0.000000 1.000000
FACE 0 3 7 UV1 1.000000 0.000000 UV2 1.000000 1.000000 UV3 0.000000 1.000000
...
```

Слова «FACE», «VERTEX», «NORMAL» и пр. используются только для наглядности — при желании их можно сократить соответственно до «f», «v», «n» и т.д., если вы хотите уменьшить размер файла.

Однако это еще не все: необходим загрузчик для нашего формата в игровой движок. Я реализовал загрузчик на примере OpenGL, в виде класса, ответственного и за загрузку, и за рендеринг, и за удаление модели. Рендеринг осуществляется через список отображения (Display List), но можно использовать и другие методы вывода (обычными функциями OpenGL, массивом вершин или через VBO).

```
#ifndef _MODEL_H
#define _MODEL_H
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <string>
#include <GL/gl.h>
```

```
using namespace std;
```

```
typedef GLfloat vec2_t[2];
typedef GLfloat vec3_t[3];
typedef GLfloat vec4_t[4];
```

```
class Model
{
    public:
    Model();
    ~Model();
    int Load(string filename);
    void Render();

    protected:
    char ModelName[128];
    int Vertices;
    int Faces;
    vec3_t* normalArray;
    vec2_t* texcoordArray;
    vec3_t* tangentArray;
    vec3_t* vertexArray;
    GLuint* vertexIndices;
    bool ModelLoaded;
    GLuint list;
};
```

```
#endif
```

Model.h

```

#include "Model.h"

int LoadFile(char* filename, char** result)
{
    int size = 0;
    FILE *f = fopen(filename, "rb");
    if (f == NULL)
    {
        *result = NULL;
        return -1;
    }
    fseek(f, 0, SEEK_END);
    size = ftell(f);
    fseek(f, 0, SEEK_SET);
    *result = (char*)malloc(size+1);
    if (size != fread(*result, sizeof(char), size, f))
    {
        free(*result);
        return -2;
    }
    fclose(f);
    (*result)[size] = 0;
    return size;
}

Model::Model()
{
    sprintf(ModelName, "Empty");
    Vertices = 0;
    Faces = 0;
    normalArray = NULL;
    texcoordArray = NULL;
    tangentArray = NULL;
    vertexArray = NULL;
    vertexIndices = NULL;
    ModelLoaded = false;
    list = 0;
}

```

```

Model::~Model()
{
    glDeleteLists(list, 1);
    if (normalArray) delete normalArray;
    if (texcoordArray) delete texcoordArray;
    if (tangentArray) delete tangentArray;
    if (vertexArray) delete vertexArray;
    if (vertexIndices) delete vertexIndices;
}

int Model::Load(string filename)
{
    char* Source;
    int Length = LoadFile((char*)filename.c_str(), &Source);
    if (Length <= 0)
    {
        delete Source;
        return Length;
    }

    char buf[512];
    for (int bufi = 0; bufi < 512; bufi++) buf[bufi]=0;
    char currentchar;
    int chars = 0;
    int index = 0;

    int valerator = 0;
    int nalterator = 0;
    int talterator = 0;
    int vilterator = 0;

    while (chars<Length)
    {
        currentchar = Source[chars];
        if (currentchar=="\n") currentchar = ' ';
        if (currentchar!="\012") {buf[index]=currentchar; index++;}
        else
        {

```

```

for (int ii=index; ii<512; ii++) {buf[ii]=0;}
index = 0;

float x,y,z;
float u1,v1,u2,v2,u3,v3;
int xInd, yInd, zInd;

if (sscanf (buf, "MODEL %s",ModelName) == 1) { }

if (sscanf (buf, "VERTICES %d",&xInd) == 1)
{
    vertexArray = new vec3_t[xInd];
    normalArray = new vec3_t[xInd];
    Vertices = xInd;
}

if (sscanf (buf, "FACES %d",&xInd) == 1)
{
    texcoordArray = new vec2_t[xInd*3];
    vertexIndices = new GLuint[xInd*3];
    Faces = xInd;
}

if (sscanf (buf, "VERTEX %f %f %f",&x,&y,&z) == 3)
{
    vertexArray[valterator][0] = x;
    vertexArray[valterator][1] = y;
    vertexArray[valterator][2] = z;
    valterator++;
}

if (sscanf (buf, "NORMAL %f %f %f",&x,&y,&z) == 3)
{
    normalArray[nalterator][0] = x;
    normalArray[nalterator][1] = y;
    normalArray[nalterator][2] = z;
    nalterator++;
}

```

```

if (sscanf (buf, "FACE %d %d %d
UV1 %f %f
UV2 %f %f
UV3 %f %f",
&xInd,&yInd,&zInd,
&u1,&v1,
&u2,&v2,
&u3,&v3) == 9)
{
    vertexIndices[vilterator+0] = xInd;
    vertexIndices[vilterator+1] = yInd;
    vertexIndices[vilterator+2] = zInd;

    texcoordArray[talterator+0][0] = u1;
    texcoordArray[talterator+0][1] = v1;
    texcoordArray[talterator+1][0] = u2;
    texcoordArray[talterator+1][1] = v2;
    texcoordArray[talterator+2][0] = u3;
    texcoordArray[talterator+2][1] = v3;

    vilterator+=3;
    talterator+=3;
}
}
chars++;
}

ModelLoaded = true;
delete Source;

list = glGenLists(1);
glNewList(list, GL_COMPILE);

```

```

for (int i=0; i<Faces; i++)
{
    glBegin(GL_TRIANGLES);
    for (int j = 0; j < 3; ++j)
    {
        glTexCoord2f(texcoordArray[i*3+j][0],
                    texcoordArray[i*3+j][1]);

        glNormal3f(normalArray[vertexIndices[i*3+j]][0],
                  normalArray[vertexIndices[i*3+j]][1],
                  normalArray[vertexIndices[i*3+j]][2]);
        glVertex3f(vertexArray[vertexIndices[i*3+j]][0],
                  vertexArray[vertexIndices[i*3+j]][1],
                  vertexArray[vertexIndices[i*3+j]][2]);
    }
    glEnd();
}
glEndList();

return 1;
}

void Model::Render()
{
    if (list!=0) glCallList(list);
}

```

Model.cpp

**Gecko**  
**gecko0307@gmail.com**

# «Конструктор» лицензий



Creative Commons — это ряд свободных лицензий, выпущенных одноименной американской некоммерческой организацией, основанной в 2001 году. Creative Commons создает переводы основных лицензий CC на другие языки и производит адаптацию их содержания для местного законодательства различных стран мира.

В отличие от различных других свободных лицензий (например, GNU GPL), лицензии Creative Commons более удобны для фотографий, рисунков, коротких текстов и т. д., так как они не требуют распространения вместе с произведением сопроводительного текста с условиями лицензии — достаточно буквенного кода из базовых элементов.

Лицензии Creative Commons состоят из следующих базовых элементов, которые можно комбинировать желаемым образом в соответствии с условиями распространения произведения:

**Attribution (BY):** требуется упоминание авторства использованной работы способом, указанным автором или, при отсутствии такового, предусмотренным лицензией. При отсутствии других элементов произведение и производные работы могут распространяться под другой лицензией, например, более закрытой.

**Noncommercial (NC):** запрещается использование в целях получения прибыли.

**No Derivative Works (ND):** запрещается делать производные работы.

**Share Alike (SA):** распространение и модификация произведения разрешается на том условии, что производные работы также будут распространяться на условиях этой же лицензии (всех ее элементов) — то есть, запрещено добавлять элемент ND и, при отсутствии такового, NC.

Традиционно используются следующие шесть вариантов комбинаций. Все они требуют указания авторства (ранее использовались комбинации, не требующие этого, но Creative Commons перестала их поддерживать). Условия лицензий являются неформальными, в них легко разобраться неспециалистам. Создатель Creative Commons, Лоуренс Лессиг, называет их «human readable» («понятные человеку»). Но они не имеют юридической силы, поэтому для каждой из них подготовлены корректные формулировки в рамках местного законодательства: по выражению Лессига — «lawyer readable» (англ. понятные юристу). Таким образом, система лицензий CC становится удобна и для пользователей, и для юристов.

CC-BY				
CC-BY-SA				
CC-BY-ND				
CC-BY-NC				
C-BY-NC-SA				
C-BY-NC-ND				

**Attribution (BY)** — лицензия «с указанием авторства». Позволяет другим распространять, перерабатывать, исправлять и развивать произведение, даже в коммерческих целях, при условии указания автора произведения.

**Attribution - Share Alike (BY-SA)** — лицензия «с указанием авторства - с сохранением условий». Позволяет другим перерабатывать, исправлять и развивать произведение даже в коммерческих целях при условии указания авторства и лицензирования производных работ на аналогичных условиях. Эта лицензия является копилефт-лицензией. Все новые произведения, основанные на лицензированном под нею, будут иметь аналогичную лицензию, поэтому все производные будет разрешено изменять и использовать в коммерческих целях.

**Attribution - No Derivative Works (BY-ND)** — лицензия «с указанием авторства - без производных работ». Позволяет свободно распространять произведение, как на коммерческой, так некоммерческой основе, при этом работа должна оставаться неизменной и обязательно должно указываться авторство.

**Attribution - Noncommercial (BY-NC)** — лицензия «с указанием авторства - некоммерческая». Позволяет другим перерабатывать, исправлять и развивать произведение на некоммерческой основе, и хотя для производных работ сохраняются требования указания авторов и некоммерческого использования, не требуется предоставления третьим лицам аналогичных прав на производные от нее.

**Attribution - Noncommercial - Share Alike (BY-NC-SA)** — лицензия «с указанием авторства - некоммерческая - с сохранением условий» . Эта лицензия позволяет другим перерабатывать, исправлять и развивать произведение на некоммерческой основе, до тех пор пока они упоминают оригинальное авторство и лицензируют производные работы на аналогичных условиях. Все новые произведения, основанные на этом, будут иметь одни и те же лицензии, поэтому все производные работы также будут носить некоммерческий характер.

**Attribution - Noncommercial - No Derivative Works (BY-NC-ND)** — лицензия «с указанием авторства - некоммерческая - без производных». Позволяет другим получать и распространять произведение, до тех пор пока они упоминают автора и ссылаются на него, но они не могут ни под каким видом изменять произведение и использовать его в коммерческих целях.

<http://creativecommons.org/>

**Геско**  
(на основе материалов Википедии)  
[gecko0307@gmail.com](mailto:gecko0307@gmail.com)

# OpenShot

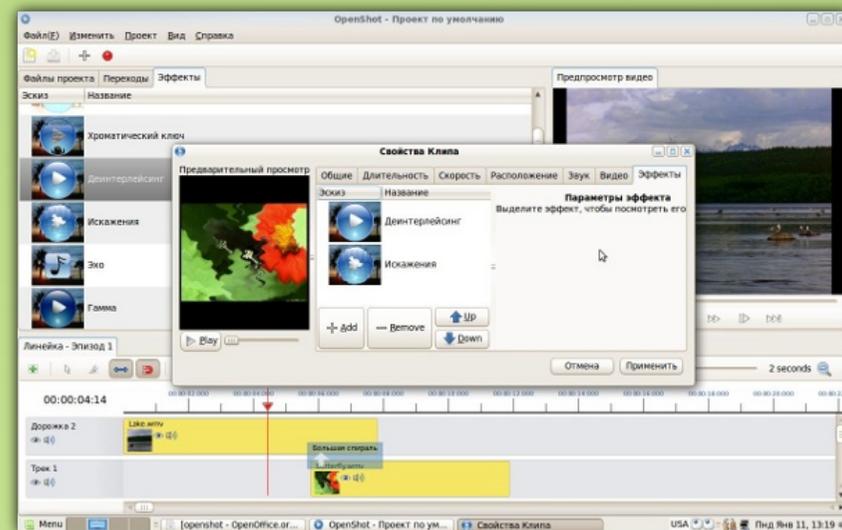
## Редактор видео под Linux

Не секрет, что на сегодняшний день выпущено немало прикладного софта для Linux. Думаю, все согласятся, что ряд повседневных задач (работа с файлами, редактирование документов, воспроизведение мультимедийных файлов, веб-серфинг, чтение почты) в современных десктопных дистрибутивах Linux выполняется не хуже, а порой даже лучше и быстрее, чем в Windows. Однако не стоит забывать, что на дворе двадцать первый век, и многие пользователи ПК являются счастливыми обладателями таких чудес техники, как цифровые фотоаппараты и видеокамеры. Причем каждому хочется запечатлеть дорогие сердцу воспоминания в виде красиво оформленного фильма или слайдшоу. Вот тут-то и пришлось бы грустно развести руками, так как аналогов известным программам для видеомонтажа вроде Adobe Premiere или Sony Vegas под Linux нет. Пришлось бы — до недавнего времени.

Речь идет о молодом, но весьма перспективном редакторе OpenShot.

Альтернативой коммерческим «монстрам» его, конечно, не назовешь — данный продукт рассчитан прежде всего на домашнего пользователя. Однако и ставить его в один ряд с поделками вроде Windows Movie Maker тоже как-то некорректно. Судите сами: поддержка почти всех форматов видео, аудио и графических изображений, неограниченное количество дорожек, огромная коллекция эффектов и переходов, возможность применять несколько эффектов к одному видеофрагменту, экспорт в любые форматы и кодеки, включая mp4, 3gp, flv, mov, ogg и др. Согласитесь, неплохо для бесплатного редактора. И это только начало!

Автор этой замечательной программы с самого начала поставил целью создание дружелюбного, но в то же время мощного свободного редактора видео, который мог бы стать своего рода стандартом в своей нише — как, скажем, GIMP или OpenOffice.org. Действительно, интерфейс программы весьма дружелюбен и интуитивен. Есть ряд тем оформления и локализация на несколько десятков языков. Окно программы, на первый взгляд, напоминает тот же Movie Maker: справа список мультимедийных файлов, слева — экран предпросмотра, внизу — временная шкала и дорожки. Но на этом сходство заканчивается.



Дорожки — это «сердце» нелинейного редактора. На них располагаются видеотреклеты, из которых собирается фильм. Во многих любительских программах видеомонтажа пользователь ограничен в лучшем случае тремя-четырьмя дорожками, в худшем — всего одной. При этом нередко случается так, что одна или две дорожки зарезервированы для аудиозаписей, то есть аудио нельзя поместить на дорожку для видео, и наоборот. С OpenShot вы навсегда забудете о таких ужасных условиях. Дорожек здесь может быть столько, сколько вам нужно, и на любую из них вы свободно можете помещать любые данные, будь то аудио, видео, изображения или титры.

Новичкам видеомонтажа обычно непременно требуется наличие переходов и эффектов для видео. OpenShot и здесь не подведет. В вашем распоряжении около 50 переходов («часовые стрелки», «спирали», «звезды», «жалюзи» и др.) и 30 эффектов (размытие, эффект карандашного рисунка, «синий фон», искажение, инверсия, «эффект старины», сепия, собел и т.д.) Такая коллекция вполне способна удовлетворить аппетиты любого начинающего пользователя. Даже если вам чего-то не хватает, нет проблем: переходы в OpenShot реализованы как простые черно-белые изображения, и ничто не мешает вам создать свои — разумеется, при наличии опыта работы в графическом редакторе. Просто загляните в каталог, где хранятся переходы (`/usr/share/openshot/transitions`) и поймете, о чем я. Эффекты же представлены XML-файлами в каталоге `/usr/share/openshot/effets`, в которых прописаны соответствующие параметры.

Любой эффект можно настраивать в окне свойств видеотреклета. Там же, кстати, задаются параметры трансформации клипа: можно изменить масштаб, расположение, скорость воспроизведения, длительность с точностью до миллионных долей секунды, прозрачность, плавное появление и исчезание, а также многие другие.

Полезной особенностью OpenShot также является поддержка SVG — стандарта векторной графики. Благодаря ей вы можете добавить в фильм любые титры, заголовки и логотипы, и всегда будете уверены в качестве полученной картинки: векторная графика, как известно, масштабируется без потери качества. В программу встроен простенький редактор титров, который поможет быстро создать красиво оформленный текст, если под рукой нет Inkscape.

Поддержка форматов видео в OpenShot реализована через библиотеку ffmpeg. Перед записью проекта в конечный файл, программа попросит вас настроить параметры экспорта. Это можно сделать простым способом, который не требует глубоких знаний особенностей кодеков и видеорежимов. Вы просто выбираете тип проекта (файл, Blu-Ray, DVD, интернет-видео и др.), формат видео внутри выбранного проекта (например, для интернет-видео это могут быть форматы для YouTube, Vimeo, Metacafe и др.), один из поддерживаемых профилей для выбранного формата, и, наконец, качество — низкое, среднее или высокое.

Для опытных же пользователей есть вкладка «Дополнительно», где открывается доступ ко всем параметрам экспорта, как говорится, «от и до»: все мыслимые видеорежимы (включая HD), форматы-контейнеры (даже такие «экзотические», как matroska, ogg video, ipod и др.), громадный набор кодеков, битрейт, настройки звука. Вы даже можете экспортировать фильм не в видео, а в последовательность изображений, для которых тоже можно выбрать желаемый формат (bmp, png, jpg, jpeg2000 и т.д.)

Редактор довольно стабилен и «повесить» его во время работы над фильмом не так-то просто, но в некоторых специфических ситуациях он все же может упасть. Например, при экспорте с неверно указанными параметрами. Так что, советую вам на всякий случай почаще сохранять проект, чтобы ненароком не потерять проделанные изменения.

Как и все программы для работы с видео, OpenShot потребляет довольно много памяти (от 30 до 400 Мб в зависимости от сложности проекта), отчего вполне вероятно, что на старой машине он просто не запустится. Однако любой современный компьютер должен потянуть его без особых проблем.

На сайте OpenShot <http://launchpad.net/openshot> или <http://www.openshot.com>) вы найдете все необходимые файлы для установки программы на Linux-машину. Например, если вы являетесь пользователем дистрибутива семейства Ubuntu (или на ее основе), то скачивайте deb-пакеты, а также архив с зависимостями для вашей версии ОС. Для установки OpenShot также могут понадобиться следующие пакеты: `libgoocanvas3`, `libgoocanvas-common` и `python-pygoocanvas`. Их можно найти в официальной репозитории Ubuntu. OpenShot распространяется по лицензии GNU GPL v3.

**Gecko**  
***gecko0307@gmail.com***

# Это все!

Надеемся, номер вышел интересным. Если так, поддержите FPS! Отправляйте статьи, обзоры, интервью и прочее на любые темы, касающиеся игр, графики, звука, программирования и т.д. на ящик редакции: [clocktower89@mail.ru](mailto:clocktower89@mail.ru) или лично главному редактору: [gecko0307@mail.ru](mailto:gecko0307@mail.ru).

*Редакция*

