

40
2016

BLENDER: Новости

Карты освещения в Cycles

Искусство создания текстур

Как проектировать хорошие игры?

Лучшие свободные движки

Игровая фотография:
новое направление в искусстве

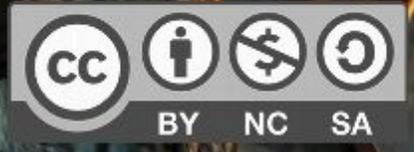
+ многое другое!

О
О
О

Blender 2016

чего ожидать в ближайшем будущем?

Независимый электронно-познавательный журнал.
Издается с 2008 г. Доступен по CC-BY-NC-SA





FPS

№40

*FPS – бесплатный, свободно распространяемый
электронный журнал, посвященный
компьютерному творчеству.*

FPS охватывает широкий круг тем: на страницах журнала рассматриваются вопросы программирования игр с использованием разнообразных движков и графических библиотек, публикуются материалы по двумерной и трехмерной компьютерной графике, включая уроки по популярным графическим редакторам, игровые обзоры, а также статьи по игровой теории, геймдизайну и современным мультимедийным формам искусства.

Журнал издается с января 2008 г.
и выходит раз в два месяца.

© 2008-2016 Редакция журнала «FPS». Некоторые права защищены. Все названия и логотипы являются интеллектуальной собственностью их законных владельцев и не используются в качестве рекламы товаров или услуг. Редакция не несет ответственности за достоверность информации в материалах издания и надежность всех упоминаемых URL-адресов. Мнение редакции может не совпадать с мнением авторов. Материалы издания распространяются по лицензии **Creative Commons Attribution Noncommercial Share Alike (CC-BY-NC-SA)**, если явно не указаны иные условия.

Главный редактор: **Тимур Гафаров**
Дизайн и верстка: **Наталья Чумакова**
Обложка: **Тимур Гафаров**

Наш сайт: <http://fps-magazine.cf>

По вопросам сотрудничества обращайтесь по адресу:

gecko0307@gmail.com

● Blender

- :: Новости
- :: Blender 2016
- :: Секреты и хитрости
- :: Карты освещения в Cycles
- :: Обзор дополнений. Выпуск 18

● 2D-графика

- :: Новости
- :: Искусство создания текстур

● Программирование

- :: Язык D: новости «с Марса»
- :: Создание гистограммы в dlib
- :: Как проектировать хорошие игры?
- :: Лучшие свободные движки

● Linux-гейминг

- :: Игровые новости из мира Linux

● Игровая фотография

- :: Новое направление в искусстве

● Linux

- :: Полезные команды

В ЭТОМ НОМЕРЕ



Blender

Новости

Минувшие месяцы оказались невероятно богаты на анимационные релизы – причем, особенно радует тот факт, что Blender-мультипликаторы сегодня работают в самых разных стилях и направлениях. Общим для этих фильмов являются «законы жанра» короткометражной анимации: простой сюжет, харизматичный главный герой и неожиданная концовка с эмоциональным посылом...

В этот раз действие происходит в заснеженной Патагонии, где наш герой сталкивается с конкурентом в лице маленького пингвина. Фильм длится всего две с половиной минуты, но даже в таком сжатом формате авторам удалось показать забавную и поучительную историю с добрым концом. Технически фильм снят просто блестяще – рекомендуем всем!

<https://www.youtube.com/watch?v=SkVqJ1SGeL0>

Еще один впечатляющий короткометражный фильм – «**Optimist**» от режиссера Джона Мелвина. Это необычная история в еще более необычном визуальном воплощении: все модели, включая персонажей, полностью выполнены в низкополигональном стиле!

<https://vimeo.com/150188598>



29 января состоялся долгожданный выход «**Caminandes 3: Llamigos**» – третьей по счету анимационной короткометражки от режиссера Пабло Васкеза, повествующей о приключениях ламы, которой постоянно приходится преодолевать различные препятствия в поисках корма.



Фильм «Globe» от польского режиссера Павла Ольсжевского – необычный взгляд на простой предмет интерьера: один день из жизни обитателя стеклянного «снежного шара».

<https://www.youtube.com/watch?v=4ALPuOdzch0>



Не отстают от западных коллег и отечественные анимационные студии, использующие в Blender: в декабре прошлого года состоялась онлайн-преьера короткометражного фильма «Чемодан» от студии **MultLabs** и режиссера Михаила Солюянова. Это ироничная история о девушке с большим чемоданом – настолько тяжелым, что ей не хватает сил поднять его на лестницу.

<https://www.youtube.com/watch?v=0aq3nau96Bg>

Также вышли трейлеры невероятно красивого мультфильма «**Etulu**» (релиз намечен на 2017 год) и игровой VFX-картины «**Circle**».

Не заглох и проект «**Wires for Empathy**» от Бассама Курдали: недавно режиссер рассказал о проделанной за последние годы работе.



Переходим к игровым анонсам: космосим **Helium Rain**, о котором мы писали в одном из предыдущих номеров журнала, прошел Greenlight-голосование и вскоре, по заверению разработчиков, будет доступен в рамках Steam Early Access. Весь контент для Helium Rain, как известно, создается при помощи Blender, сама игра работает на основе Unreal Engine.

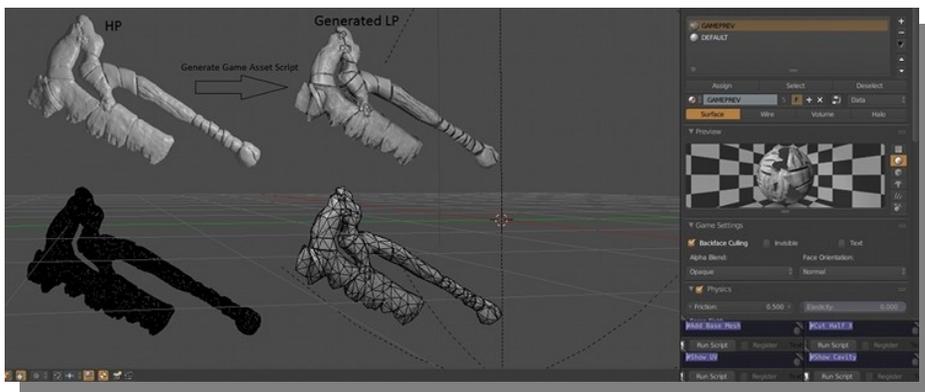
Очень любопытный проект представили на Steam немецкие инди-разработчики Blue Label Studio – это **I Will Escape**, сюрвайвл от третьего лица, в котором вы должны сбежать из тюрьмы. Игра полностью разработана на Blender GE.

Фан-игра Zelda Awakening о приключениях эльфа Линка, о которой мы писали в FPS №33 '14, все еще находится в активной разработке – недавно вышел очередной **видеоотчет** по прогрессу.

Кроме того, игра обзавелась репозиториумом на GitHub: https://github.com/Dynamique-Zak/Zelda_BlenderGame, так что теперь все желающие могут присоединиться к проекту и помочь игре увидеть свет.

Любопытную модификацию Blender представили французские разработчики – это Dark Blender, набор инструментов для быстрой и удобной подготовки игровых ресурсов (в частности, скрипты для конвертирования высокополигональных моделей в низкополигональные, для рисования текстур и т.д.). Скачать Dark Blender можно здесь:

<http://linko.projects.free.fr/dblender.zip>



Ну и, конечно, вышла новая версия открытого фреймворка для создания браузерных 3D-приложений Blend4Web 16.01. В этой версии появилась поддержка процедурной отрисовки линий, начата работа над библиотекой материалов, добавлены новые узлы редактора логики.

Напомним, Blend4Web предназначен для создания трехмерного интерактивного контента, работающего в браузерах без использования плагинов. Пакет тесно интегрирован с Blender, который используется в качестве основного инструмента редактирования 3D-сцен. Воспроизведение контента осуществляется средствами WebGL, Web Audio и других браузерных технологий. Нарботки проекта распространяются под лицензией GPLv3.

<https://www.blend4web.com>

Памятка читателю

В Интернете часто можно встретить вопросы о том, где скачать старые номера нашего журнала. Отвечаем: архив всех номеров «FPS» (с 2008 по 2016 гг.) можно найти сразу на нескольких сервисах:

На файловом хостинге **DropBox**:

https://www.dropbox.com/sh/b7lgxxh6nxbxre9/uVvzqU8_j-

В **Документах Google**:

<https://docs.google.com/folderview?id=0B1BlzRb1uMv-bnpHNDhwZTI4eHc>

В электронном издательстве **Issuu.com**:

<http://issuu.com/tgafaroff/docs>

Для тех, кто предпочитает скачивать с торрентов – журнал также есть на **PyТрекере**:

<http://rutracker.org/forum/viewtopic.php?t=4403193>



BLENDER 2016

чего ожидать в ближайшем будущем?



По сложившейся традиции, публикуем ежегодный «хит-парад» новых разработок, которые совсем скоро станут частью Blender. OpenGL 2.1 во вьюпорте, Fracture Modifier, Object Nodes, пользовательские манипуляторы – читайте обо всем этом и не только в нашем сегодняшнем обзоре.



Переход на OpenGL 2.1

Должен состояться уже в Blender 2.77. В Blender 2.8 планируется перейти на OpenGL 3.2 и OpenGL ES 3.0. Это должно повысить производительность и качество отображения объектов во вьюпорте, а в перспективе – помочь реализовать поддержку OpenSubdiv под OSX. Но, вместе с этим, прекратится поддержка старого «железа», на котором Blender работал много лет. Кроме того, будет потеряна совместимость с аддонами, которые рисовали во вьюпорте через OpenGL.

Fracture Modifier

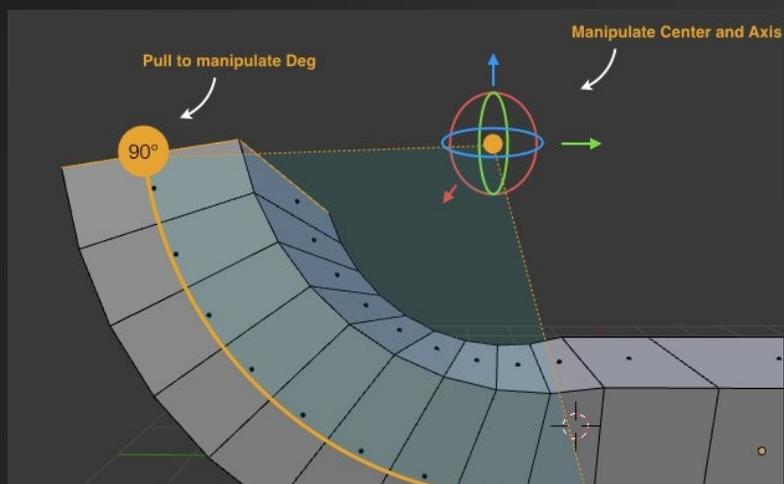
Этот невероятно интересный модификатор позволит автоматически разбивать любой меш на кусочки, и анимировать их при помощи физики твердых тел. В настоящее время для достижения такого эффекта приходится разбивать объекты вручную, что не очень удобно – к тому же, подобная операция необратима.

На самом деле, Fracture Modifier уже полностью готов, но его не принимают в основную ветку Blender по той причине, что планируется рефакторинг других компонентов пакета, от которых модификатор зависит. Будем надеяться, что все технические проблемы будут разрешены в ближайшем будущем.



Object Nodes

Новый граф зависимостей, успешно внедренный в Blender 2.76, открывает дорогу множеству интересных инструментов, например – узлам для анимации, частиц, ограничений и т.д. Идея «все на узлах» постепенно воплощается в жизнь!



Пользовательские манипуляторы

Возможность создавать новые средства ввода во вьюпорте должна сделать Blender еще привлекательнее для профессиональных дизайнеров. Например, можно будет создавать своеобразные «3D-виджеты» для интуитивного управления анимацией персонажа, для генеративного моделирования и т.д.



Интеграция OpenVDB

OpenVDB – это тулkit от Dreamworks, предназначенный для управления объемными дискретными данными, например, симуляциями дыма или облаков. Мы писали об этом проекте в FPS №22 '13. Начальная поддержка OpenVDB планируется уже в Blender 2.77.

Менеджер ресурсов

В последние годы было предпринято несколько попыток реализовать менеджер ресурсов для Blender на скриптах. Тем не менее, для полноценной реализации этой возможности необходимы изменения и в самом Blender: пакет будет «из коробки» поддерживать управление библиотеками внешних ресурсов.

Видео в VGE

Активно разрабатывается ветка Blender, в которой реализуется поддержка потокового видео в игровом движке.



Улучшения из Gooseberry

Некоторые улучшения, сделанные сотрудниками Blender Institute в ходе работы над фильмом «Космическая прачечная», до сих пор не включены в основную ветку — это, в первую очередь, улучшения рендеринга волос и поддержка Alembic.

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%). 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 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: 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 и пр.) могут быть как прикреплены к письму, так и загружены на какой-либо надежный сервер (убедительная просьба не использовать RapidShare, DepositFiles, Letitbit и другие подобные файлообменники — загружайте файлы на свой сайт, блог или ftp-сервер и присылайте статические ссылки). Все материалы желательно архивировать в формате zip, rar, 7z, tar.gz, tar.bz2 или tar.lzma.



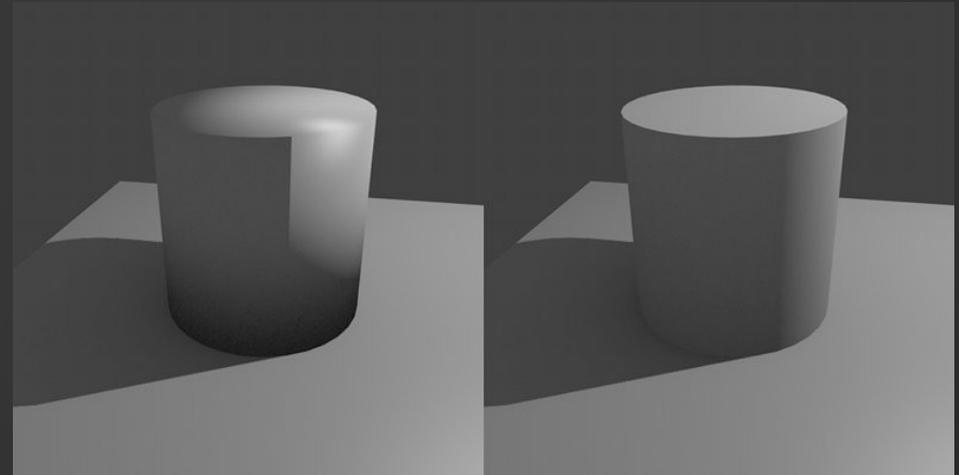
Blender. Секреты и хитрости

полезные «горячие клавиши» и многое другое!

Blender всегда имел репутацию достаточно «хардкорной» программы для изучения начинающими. И, несмотря на значительный пересмотр интерфейса в последних версиях, сделавший типовые операции гораздо более интуитивными, очень многие полезные функции Blender – в частности, горячие клавиши и «секретные» меню – остаются малоизвестными, о них редко пишут в учебниках и не упоминают в видеоуроках. Приводим нашу коллекцию хитростей – «must have» для любого, кто хочет научиться работать быстрее и эффективнее!

1. Острые грани

Добавьте цилиндр и сделайте его гладким (**Smooth**). Заметили неприятный артефакт при рендеринге? Из-за сглаживания нормалей на состыкованных гранях, ребра цилиндра визуально теряют остроту, и получается черт знает что. Это делает моделирование гладких тел вращения довольно трудоемким занятием. Но, к счастью, есть средство, которое помогает решить эту проблему – модификатор **Edge Split**. Он разъединяет грани у острых ребер, и модель приобретает корректные нормали при сглаживании.

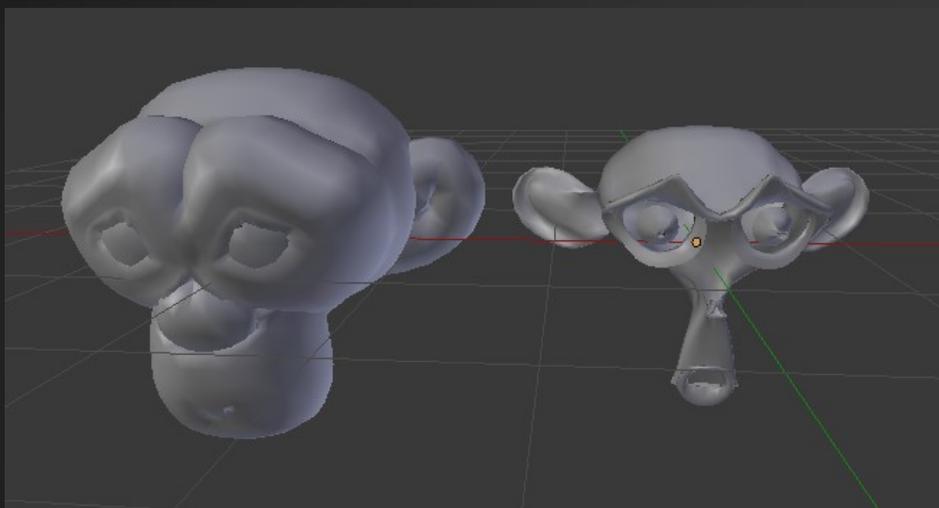


При этом можно указать пороговый угол остроты между гранями, либо пометить грани как «острые» (**Sharp**) на левой панели инструментов.



2. Раздувание/сужение

Очень полезный инструмент, о существовании которого мало кто знает. Выделите набор вершин и нажмите **Alt+S** – двигая мышью, можно «раздуть» объект или, наоборот, «сузить» его.



3. Выделение объектов

В сложных сценах с множеством объектов бывает нужно сосредоточиться на какой-то определенной группе объектов, а остальные скрыть, чтобы не мешали. Это можно сделать при помощи **Shift+N**. А чтобы снова показать все объекты, нажмите **Alt+N**.

А с этим сочетанием клавиш выделение нужного объекта на сложной сцене становится еще проще: **Alt+ПКМ**. Она вызывает меню, в котором вы можете уточнить, как объект хотите выделить.

Правда, эта комбинация может не работать в некоторых дистрибутивах Linux, где Alt связана с оконным менеджером, и Alt+ПКМ вызывает меню управления активным окном. Эту проблему можно решить настройкой системных горячих клавиш – заменой Alt, например, на клавишу Windows.

4. Фокусы с 3D-курсором

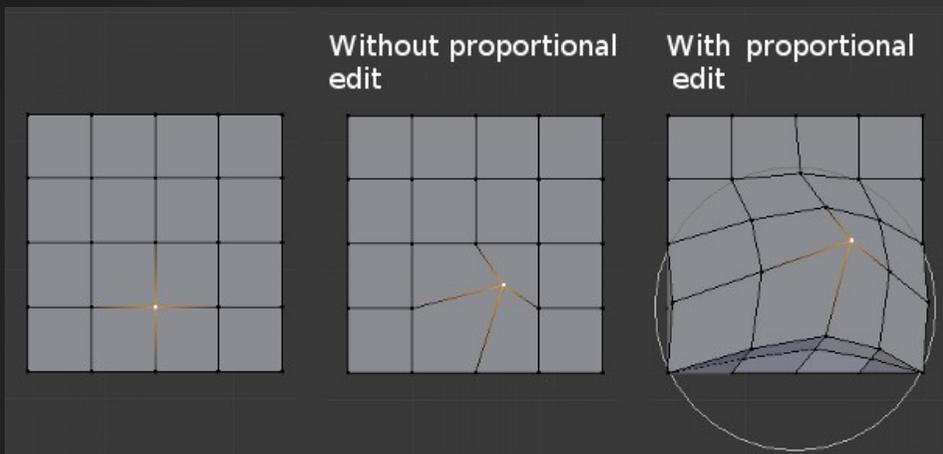
3D-курсор – отличная штука, однако мало кто умеет правильно управлять им. Комбинация **Shift+S** вызывает весьма полезное меню, в котором можно переместить 3D-курсор в центр сцены, в центр выделенного объекта или вершины/ребра/грани, а также выровнять положение курсора по сетке.

Из этого же меню можно переместить выделенный объект к 3D-курсорю или выровнять объект по сетке.

Кстати, есть и более быстрый способ поместить 3D-курсор в центр координатной системы – комбинация **Shift+C**. Она же изменит вид вьюпорта так, чтобы в него попали все объекты сцены, что очень удобно.

5. Пропорциональное редактирование

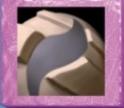
В режиме пропорционального редактирования вы можете менять радиус зоны влияния во время трансформации при помощи **колесика мыши**.

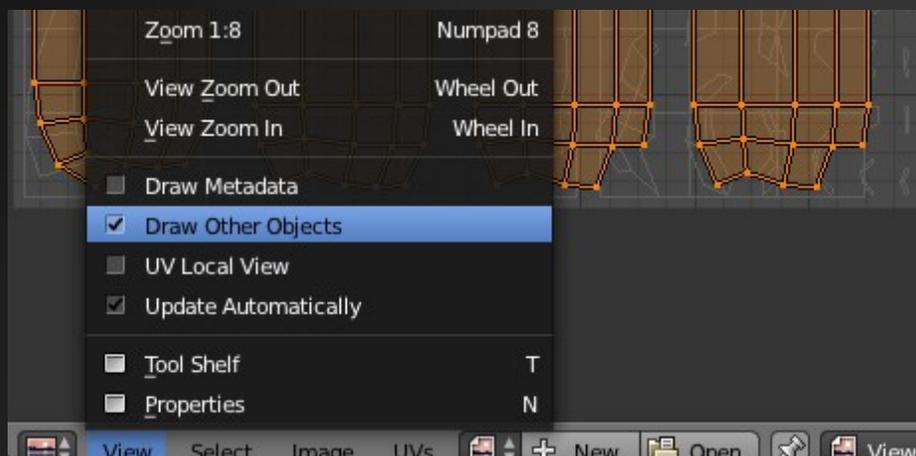


6. Лепка

Знаете ли вы, что в режиме лепки **цифровые клавиши** переключают тип скульптурной кисти? Правда, кистей больше 10, поэтому для переключения на **Nudge**, **Pinch** и остальные используются комбинации **Shift+1**, **Shift+2** и т.д.

Blender Sculpting Cheat Sheet

	Blob 1		Mask 0 or M
	Clay 2 or C		Nudge Shift + 1
	Clay Strips 3		Pinch/Magnify P or Shift + 2
	Crease 4 or Shift + C		Sculpt Draw Shift + 4 or X
	Fill/Deepen 5		Scrape/Peaks Shift + 3
	Flatten/Contrast 6 or Shift + T		Smooth Shift + 5 or S
	Grab 7 or G		Snake Hook Shift + 6 or K
	Inflate 8 or I		Thumb Shift + 7
	Layer 9 or L		Twist Shift + 8



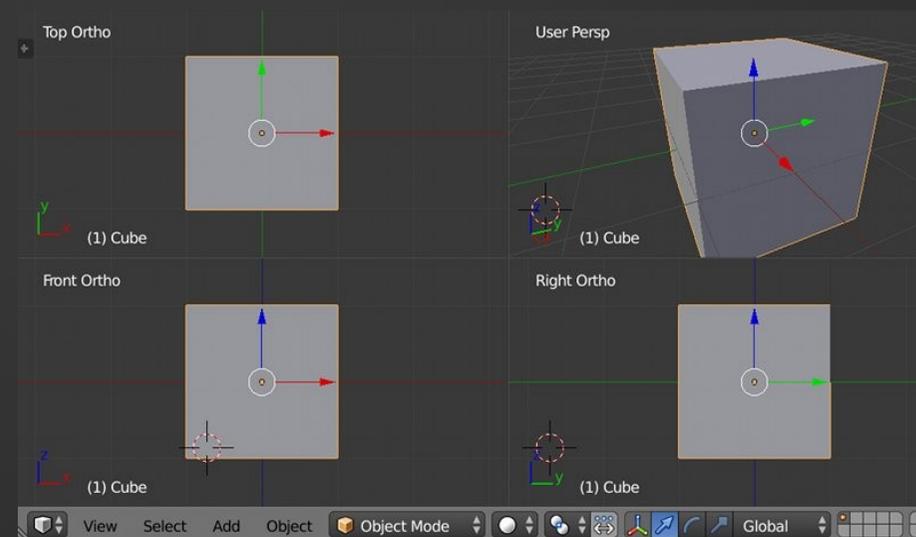
7. UV-развертка

Опция **View → Draw Other Objects** в редакторе UV/изображений включает режим отображения разверток других выделенных объектов, использующих текущую текстуру. Это очень полезно, если вы создаете развертки нескольких мешей с учетом того, что они будут использовать одну и ту же текстуру. Например, текстура автомобиля может содержать изображения одновременно и для кузова, и для колес.

8. Камера и вид

Точно настроить положение и поворот камеры – достаточно трудоемкая задача, если не знать один маленький секрет: на панели свойств во вьюпорте (клавиша **N**) во вкладке **View** поставьте галочку напротив **Lock Camera to View**.

Это синхронизирует активную камеру с видом, что позволит быстро установить нужный ракурс для рендеринга. Не забудьте только переключиться на вид из камеры (меню **View → Camera** или **Numpad 0**).



Кстати, в режиме Quad View можно синхронизировать масштаб и перемещение вида в трех ортогональных проекциях (**Top, Front, Right**). Для этого на той же панели свойств на вкладке **Display** поставьте галочку напротив опции **Box**.

Еще одна малоизвестная фишка вьюпорта – **Zoom to region**, то есть, приближение выделенной зоны: нажмите **Shift+B** и выделите нужную зону, которую хотите рассмотреть поближе.

9. Режим прогулки

Моделируете интерьер или уровень для игры? Тогда вам будет очень полезна возможность поперемещаться по вашей сцене клавишами WASD и мышью, как в компьютерных играх от первого лица. Данный режим активизируется комбинацией **Shift+F**. Выйти из него можно клавишей **Escape** или **ЛКМ**. При этом **колесиком мыши** можно менять скорость движения, а **пробелом** – телепортироваться к объекту, на который вы смотрите. Для полного «эффекта присутствия» можно включить гравитацию клавишей **Tab**. Со включенной гравитацией можно даже прыгать, как в игре – клавишей **V**.

10. Анимация

Знаете, для чего нужна кнопка **«Запись»** (красный кружок) на временной шкале? Она активизирует автоматическое добавление ключевых кадров при изменении параметров объектов – чтобы не нужно было постоянно нажимать **I**.

Если у вас есть свой набор любимых приемов и малоизвестных хитростей – милости просим писать нам на gecko0307@gmail.com, и мы обязательно опубликуем продолжение!

Ищем авторов!

Если вы – околокомпьютерный журналист (неважно, начинающий или опытный) и хотите расширить свою аудиторию, поделиться опытом и знаниями с тысячами читателей, найти единомышленников или просто высказать свою точку зрения – напишите нам! Мы будем рады опубликовать на страницах «FPS» любой материал, соответствующий тематике журнала. Это могут быть статьи, уроки, обзоры, интервью, художественная литература, авторские фото, арт, исходный код и т. д.

Вы можете выбрать одну из следующих тем: разработка компьютерных или приставочных игр, программирование, компьютерная графика, рендеринг, шейдеры, цифровой звук и музыка, геймдизайн, написание игровых сценариев, обзоры программ, движков и игровых конструкторов, отзывы и рецензии на игры, вопросы выбора лицензии, издания, дистрибьютинга или продвижения проектов, репортажи с различных конференций, выставок, встреч разработчиков и демо-пати, интересные факты из истории игр, обзоры «железа» и игровых платформ, обзоры полезных сайтов и сервисов.

Материалы принимаются в любое время на почтовый ящик редакции: gecko0307@gmail.com.

Прикрепленные файлы желательно архивировать в форматах zip, rar, 7z, tar.gz или tar.bz2. В письме или тексте материала укажите Ваши реальные имя и фамилию.

**Любой Читатель может стать Автором!
Желаем Вам творческих успехов!**



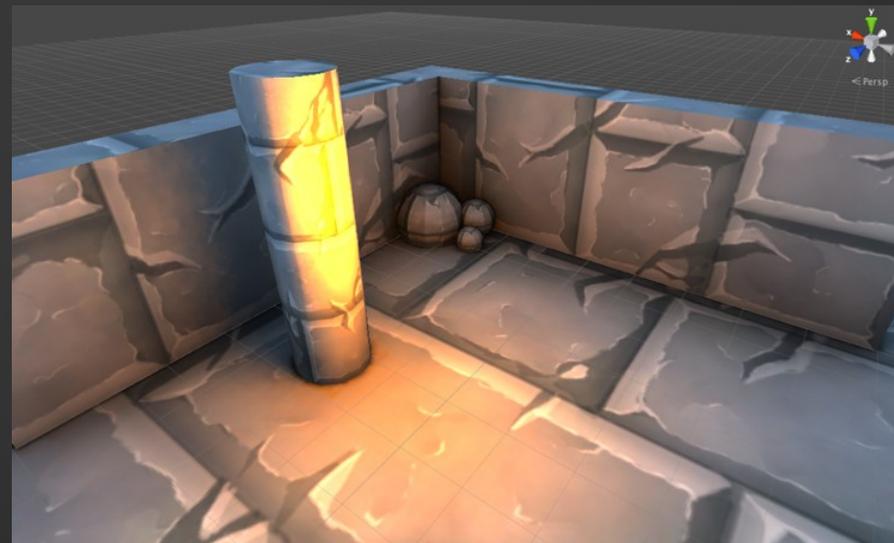
Карты освещения в Cycles

запекаем *global illumination* для игрового уровня

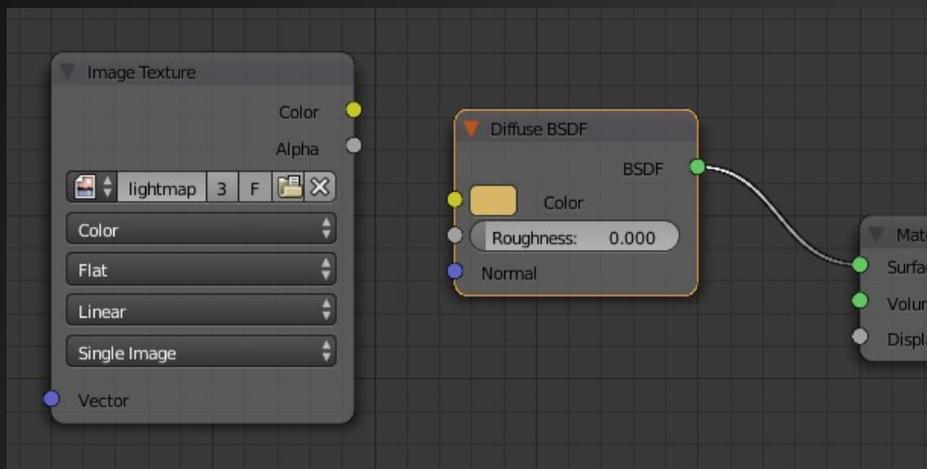
Карты освещения (light mapping, лайтмэппинг) – достаточно старая и распространенная техника, позволяющая рендерить в играх очень качественное статическое освещение, с тенями и множеством источников света. Зачастую при этом используется глобальное освещение, которое позволяет сделать игровое окружение максимально реалистичным. Сегодня мы рассмотрим, как создать такие карты в Blender при помощи рендер-движка Cycles.

Поддержка запекания в текстуру в «Суслике» появилась впервые в Blender 2.71 – в тандеме с таким замечательным инструментом, как **Texture Atlas**, она становится поистине мощным и полезным инструментом для левел-дизайнеров.

О Texture Atlas мы уже не раз писали на страницах нашего журнала – это аддон, который автоматизирует UV-развертку мешей под лайтмэппинг. В Blender есть встроенный алгоритм развертки, оптимальный для запекания карт освещения, но он работает с индивидуальными объектами – в то время, как разумнее было бы использовать одну и ту же карту для нескольких объектов сразу.



Выражаясь языком программиста, нам нужно создать текстурный атлас, и Texture Atlas, как ясно из названия, как раз помогает решить эту проблему. Первым делом активируйте Texture Atlas в менеджере аддонов (он давно уже встроен в Blender, но не активирован по умолчанию). В свойствах рендеринга должна появиться вкладка **Texture Atlas**. Выделите ваши объекты, которые хотите «запечь», и нажмите кнопку с изображением плюса. Дайте новой карте название и нажмите **OK**.

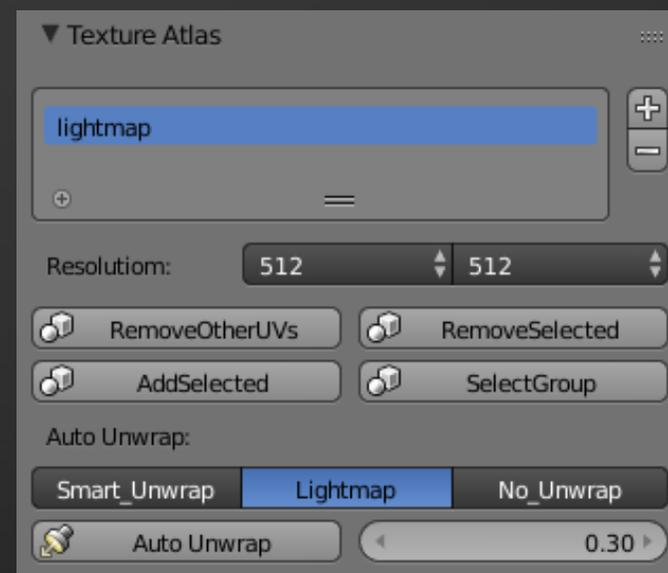


Переключите рендер-движок на Cycles, если еще этого не сделали. Не лишним будет переключить режим отображения объектов во вьюпорте на Texture, чтобы видеть результат запекания. Добавьте объектам материалы, настройте источники света. Каждому материалу в редакторе узлов добавьте узел **Image Texture**, а в нем выберите наше изображение – будущий лайтмэп. Присоединять этот узел ни к чему не надо, он нужен только для запекания.

Нажмите **AddSelected**, чтобы добавить выделенные объекты в карту. Можно также изменить разрешение карты – я обычно ставлю 512x512.

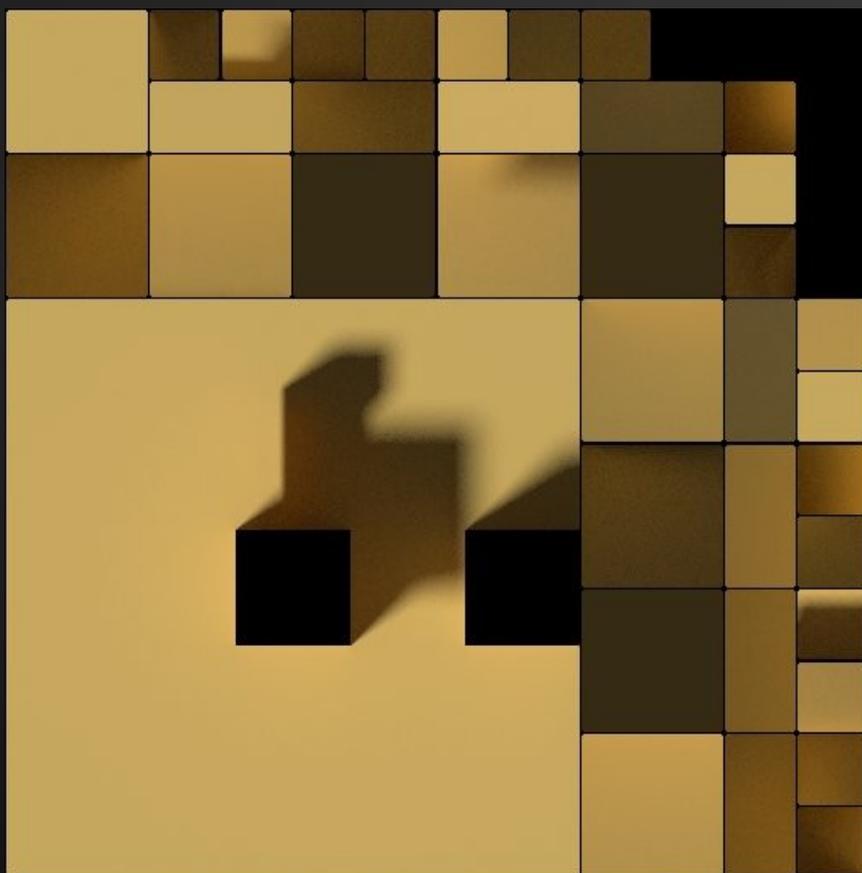
Затем переключите тип развертки на **Lightmap** и нажмите **Auto Unwrap**. Эту кнопку придется нажимать довольно часто – каждый раз, когда вы будете менять геометрию или добавлять новые объекты. Не очень удобно, конечно, но жить можно.

В результате для ваших объектов сгенерируется второй набор UV-координат, имеющий то же название, что и карта – в нем и будет развертка под карту освещения. Аддон также создаст новое изображение заданных размеров, в которое можно смело запекать освещение, чем мы сейчас и займемся.



Вернитесь в свойства рендеринга и найдите вкладку **Bake**. Обратите внимание на параметр **Margin** – он контролирует толщину «обводки» вокруг полигонов на карте (то есть, область вокруг полигона или «острова», которая получит тот же цвет, что и край этого полигона).

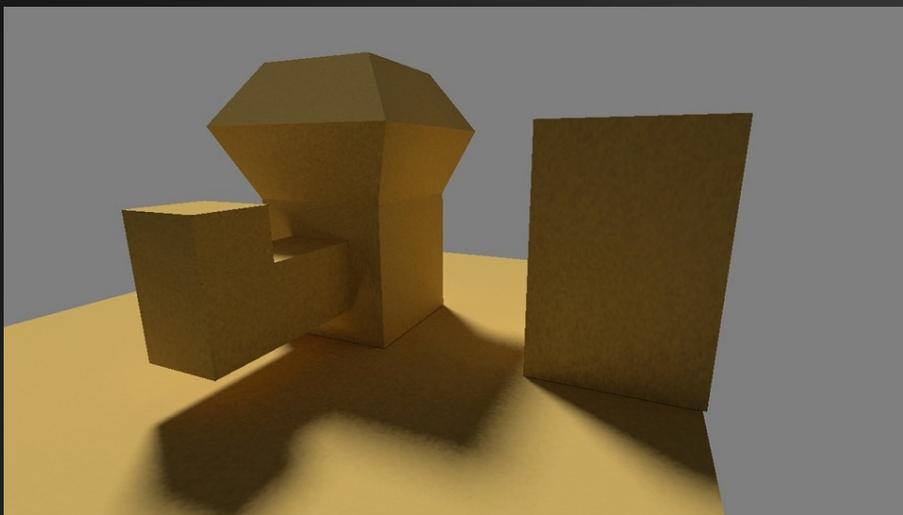
Это нужно для корректного рендеринга карты с текстурной фильтрацией – если сделать Margin равным нулю, вы, скорее всего, увидите неприятные черные края вдоль швов. Кстати, его неплохо корректировать вместе с параметром **Unwrap Precision** на вкладке **Texture Atlas**, которым задается минимальный интервал между «островами» на развертке. Чем больше интервал, тем больший Margin можно задать – и, соответственно, тем качественнее получится лайтмэп.



Однако слишком сильное увеличение интервала делает развертку менее плотной, и, как следствие, не даст использовать текстурное пространство с наибольшей эффективностью. Тут многое зависит от сложности сцены, количества объектов и разрешения карты – экспериментируйте, чтобы подобрать подходящие для вас значения. В простейших случаях Unwrap Precision можно выставить равным 0.3, а Margin – равным 1.



Теперь осталось нажать **Bake**, подождать несколько секунд и наслаждаться результатом! Чтобы сделать лайтмэп менее зернистым, можно повысить количество сэмплов на вкладке **Sampling (Samples → Render)** – начинайте с 50-100 и повышайте, пока не будет достаточно. В принципе, зернистость отрицательно сказывается на качестве картинки только по краям теней – внутри теней этот артефакт даже несколько добавляет объектам реалистичности.



Главный вопрос теперь заключается в том, как экспортировать модель с двумя наборами текстурных координат для рендеринга карт освещения в игре вместе с обычными диффузными текстурами. Практически любой современный движок поддерживает мультитекстурирование – но, к сожалению, далеко не каждый формат моделей.

Как вариант, можно сделать две копии уровня – с первым и вторым набором по отдельности – затем экспортировать их, например, в две модели OBJ, которые будут загружаться и необходимым образом интерпретироваться игровым движком. Но лучше всего, конечно, разработать свой собственный формат и написать экспортер для него. Но это уже совсем другая история...

Тимур Гафаров

Уважаемые читатели!

Наш журнал регулярно выходит на протяжении 8 лет – с февраля 2008 года. Все эти годы он оставался бесплатным изданием, предлагая публике эксклюзивный контент с минимумом рекламы. Мы всегда работали на совесть – не ради денег, а на благо наших читателей. «FPS» был и остается проектом энтузиастов и полностью независимым изданием – мы не защищаем интересы корпораций или политиков, мы пишем о том, что считаем нужным и важным. Мы стоим за свободу слова и творчества, за обмен информацией и знаниями: все материалы журнала можно беспрепятственно копировать, распространять и использовать в любых производных работах.

И мы надеемся, что так будет продолжаться и дальше. Но на создание новых номеров у авторов уходит достаточно много сил и времени, которые никак материально не компенсируются. Поэтому, если вам нравится журнал, и вы хотели бы, чтобы он жил, развивался, становился больше и качественнее, просим **поддержать его электронной валютой** – при помощи **WebMoney**, **PayPal** или **Яндекс.Денег**, любой суммой на ваше усмотрение. Для нас важен любой, даже маленький вклад!

Наш WMR-кошелек: **R120156543694**

Номер кошелька Яндекс.Денег: **410012052560079**

Адрес PayPal: **gecko0307@gmail.com**

Заранее благодарны!



Обзор дополнений Blender

Выпуск 18

Благодаря удобному и мощному API для языка Python, Blender поддается практически неограниченному расширению. В этом выпуске мы представляем дополнения, которые пригодятся при создании пейзажей и природных сцен.

Если вы разрабатываете собственное дополнение или просто нашли в Интернете чей-то интересный проект, будем очень рады, если вы напишете нам об этом и поделитесь ссылкой. Пишите на gecko0307@gmail.com.

Dynamic Sky

Аддон, облегчающий настройку реалистичного освещения уличных сцен в Cycles – для этого больше не нужно вручную создавать лампы, настраивать узлы Environment и т.д.

Dynamic Sky объединяет все настройки неба и солнечного света в простой и интуитивный интерфейс. Дополнение распространяется бесплатно.

Разработчик: Dragoneex
<http://www.dragoneex.com/downloads/dynamic-skyadd-on>

Easy Clouds

Пейзажи не будут достаточно реалистичными без облаков – и этот коммерческий аддон поможет вам моделировать их.

Easy Clouds позволяет как рисовать реалистичные объемные облака от руки, так и генерировать их из мешей. Цена – \$20.00.

Разработчик: AFX Lab
<https://cgcookiemarkets.com/all-products/easy-clouds>



SnowFX

Второй замечательный аддон от разработчиков Easy Clouds – это генератор снега SnowFX, отличный инструмент для создания зимней сцены. Снег генерируется на поверхности любых объектов, можно контролировать объем и степень таяния. Цена – \$44.99.

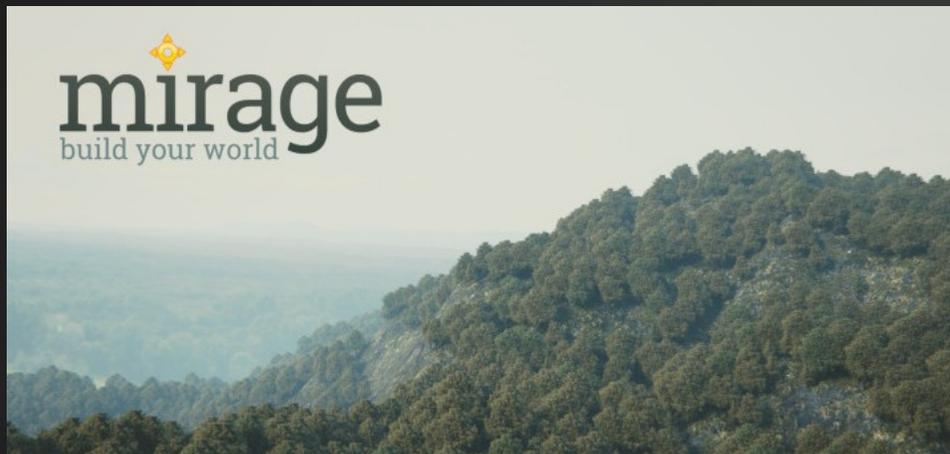
Разработчик: AFX Lab
<https://cgcookiemarkets.com/all-products/snowfx-tool>



The Grove

Генератор деревьев, очень мощный и функциональный – позволяет создавать реалистичные деревья любой породы. При этом можно контролировать процесс роста дерева во всех аспектах, добавлять объекты-препятствия и многое другое. Цена – \$109.00.

Разработчик: Wybren van Keulen
<https://www.thegrove3d.com>



Mirage

Mirage – это генератор ландшафтов из карт высот, который к тому же поддерживает распределение по ним деревьев. Все функции аддона снабжены удобным интерфейсом. Цена (включая пожизненные обновления) – \$25.00.

Разработчик: Diego Gangl
<https://cgcookiemarkets.com/all-products/mirage>



2D-графика: НОВОСТИ

Krita 3.0 pre-alpha

Вышел предварительный релиз свободного графического редактора Krita 3.0. Долгожданная третья ветка пакета примечательна переходом на Qt 5. Из других улучшений можно отметить появление полноценных инструментов анимации, режима мгновенного предпросмотра изменений, функции выделения/дублирования/редактирования сразу нескольких слоев, импорта/экспорта кистей из файлов в форматах GBR и GIN. Также в новой версии значительно улучшена производительность и эффективность работы с памятью.

Ветка Krita 3.0 позиционируется как экспериментальная – после доведения программы до стабильного состояния будет выпущена версия 3.1.

Напомним, Krita – это свободный графический редактор, предназначенный, в основном, для цифровой живописи, хотя и включающий все стандартные инструменты редактирования изображений. Krita распространяется по лицензии GPL, работает под управлением Windows, а также множества Unix-систем.

<https://krita.org>

MyPaint 1.2.0

После трех лет разработки увидел свет новый релиз свободного графического редактора MyPaint 1.2.0. В новой версии добавлена поддержка GTK+3, добавлены новые инструменты, новый набор кистей, улучшен интерфейс, добавлена поддержка векторных слоев. Напомним, MyPaint – это одна из лучших свободных программ для цифровой живописи, рисования при помощи планшета. Распространяется программа по лицензии GPLv2, есть сборки для Windows, Linux и OS X.

<http://mypaint.org>

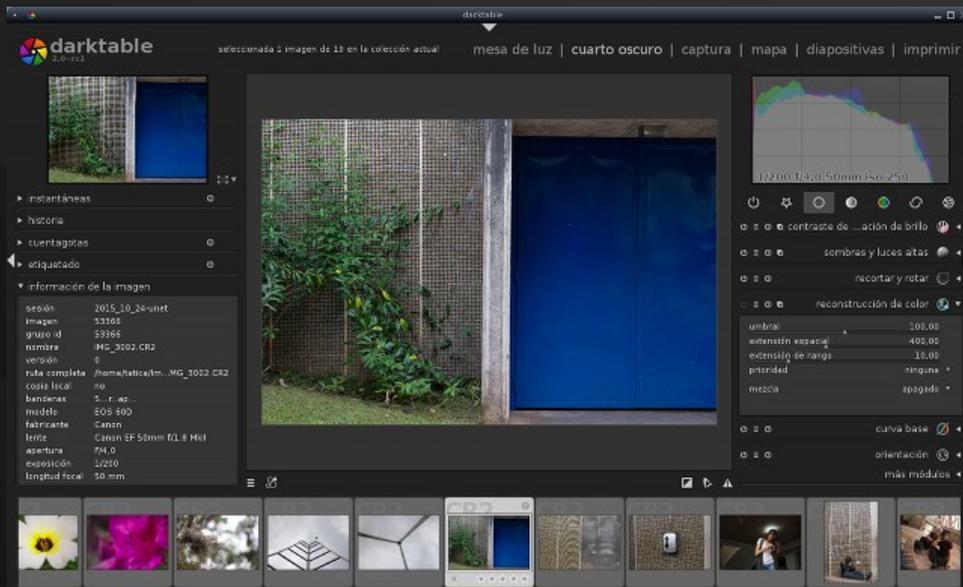


Darktable 2.0

Вышла новая версия программы для RAW-проявки Darktable 2.0, выступающей в роли свободной альтернативы Adobe Lightroom и Apple Aperture. Darktable предоставляет большую подборку модулей для выполнения всевозможных операций по обработке RAW-снимков, позволяет вести базу исходных фотографий, осуществлять наглядную навигацию по имеющемуся снимкам, сохраняя всю историю операций с ними.

В новой версии интерфейс программы портирован на GTK3, реализован новый кэш миниатюр, переработан экран работы с цветами, добавлена поддержка текстовых водяных знаков, реализован экспорт в PDF и многое другое.

<http://www.darktable.org>



Scribus 1.4.6

Увидела свет новая версия свободной программы для верстки Scribus – 1.4.6. Помимо исправления ошибок, в новом выпуске добавлена поддержка режимов смешивания SVG, улучшен экспорт в PDF и расстановка переносов, добавлены цветовые модели CIE LAB и CIE LCH в дополнение к CMYK и sRGB.



Напомним, Scribus является электронной издательской системой, свободным аналогом Adobe InDesign и QuarkXPress, предоставляя профессиональные средства для предпечатной подготовки изданий, генерирует PDF, полностью совместимые с печатными машинами, поддерживает работу со CMYK, плашечными цветами и ICC. Работает на Linux, Windows, OSX и Haiku. Исходный код проекта располстраняется по лицензии GPLv2.

<http://www.scribus.net>

Искусство СОЗДАНИЯ текстур

Текстуры играют важнейшую роль в 3D-играх – они даже несколько важнее, чем геометрия моделей: во времена ранних игр модели были очень условными и низкополигональными, и лишь качественные текстуры могли сделать игру визуально привлекательной. Даже сегодня хорошие текстуры могут «спасти» иную грубую модель с некачественной сеткой. Наконец, правильно расставленные декали, bump и parallax mapping влияют на восприятие сцены, как ничто другое...

В этой статье поделился своим опытом и секретами мастерства художник по текстурам Давид Михальчук, работавший над такими известными тайтлами, как Unreal 2, Warpath и Dungeons & Dragons Online.

Когда я начинал создавать свои первые текстуры для научно-фантастических сцен, это были простые ржавые поверхности – впоследствии я продолжил использовать такой же визуальный стиль, так как он как нельзя лучше вписывался в угловатый дизайн фантастической техники и интерьеров в играх. Меня довольно часто спрашивают о моем рабочем процессе – он, на самом деле, весьма линеен.

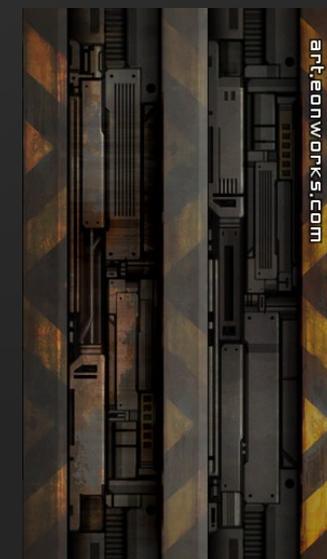


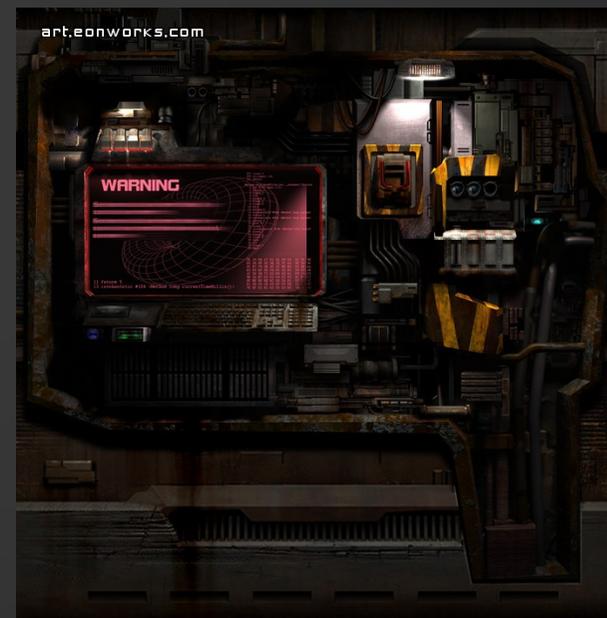
Сначала я определяю **функцию** будущей текстуры – будет ли это потолок, пол, стена и т.д. Затем изучаю **среду**, в которой текстура будет использована: например, текстура для потолка в старом грузовом космическом корабле будет сильно отличаться от текстуры потолка в каком-нибудь сверкающем кокпите в духе «Звездного пути».

Затем я разрабатываю **цветовую палитру** и **структуру изображения**. Очень важно, чтобы текстура гармонировала с другими элементами своей среды – цвета и стилистика должны соответствовать.



Потом я просматриваю свой архив текстур, которые создал за годы – вдруг я уже создавал что-нибудь похожее? Если да, то я использую готовые наработки. В противном случае, создаю новую основу. **Основа** для текстуры – это изображение с простейшей структурой. Например, для уже упомянутой текстуры потолка это будет изображение простой металлической поверхности с незначительными деталями.





Затем я начинаю рисовать основные **контуры рисунка** при помощи прямых линий. Получив рисунок, я намечаю основные **светотени**. На этом же этапе задаются нужные цвета. Затем рисуются более **мелкие детали**: заклепки, ржавчина, элементы рельефа и т.д. Все детали должны располагаться в подходящих местах и в разумных объемах – к примеру, излишек ржавчины и других дефектов на металлических поверхностях выглядит не очень реалистично, как и полное их отсутствие.

Последний этап – **цветокоррекция** всех элементов текстуры с тем, чтобы они гармонировали друг с другом и с другими текстурами сцены. **Визуальная гармония** – на мой взгляд, самая важная характеристика текстуры.

Могу раскрыть один небольшой секрет: при создании текстур используйте не только 2D-редакторы, но и **3D-пакеты**. Некоторые элементы довольно трудно нарисовать от руки – гораздо проще и быстрее их смоделировать в трехмерном варианте, а затем отрендерить. Еще одна хорошая практика – развивать наблюдательность, запоминать текстуры в реальном мире, изучать их природу.

Рисование текстур – очень увлекательный процесс, в котором вы можете полностью раскрыть свой творческий потенциал. И, конечно, приятнее всего увидеть конечный результат – как ваши текстуры в итоге рендерятся игрой...

Давид Михальчук
<http://www.art.eonworks.com>



Язык D

Новости «с Марса» свежие релизы и обновления

Если вы разрабатываете проект, связанный с языком D и хотите рассказать о нем миру, найти новых пользователей, контрибьюторов или тестеров, сообщите об этом нам! Мы готовы регулярно публиковать ваши анонсы со ссылкой на репозиторий и/или страницу проекта. Сообщения принимаем, как обычно, на ящик редакции: gecko0307@gmail.com

ИНФРАСТРУКТУРА

DMD 2.070

Вышла новая версия референсного компилятора D – DMD 2.070.0. Из языковых изменений стоит отметить появление псевдонимов (alias) для литералов функций, в стандартной библиотеке Phobos появился модуль std.experimental.ndslice с реализацией многомерных массивов и диапазонов, расширен модуль core.sys.windows, TypeInfo.init переименован в TypeInfo.initializer.

<http://dlang.org/download.html>

LDC 0.17.0

Вышла LDC 0.17.0 – новая версия компилятора D с LLVM в качестве бэкенда. Релиз основан на фронтеде и рантайме 2.068.2, включает поддержку LLVM 3.8. Обработка исключений больше не задействует сборщик мусора, появилась опция -static для создания статически слинкованных программ под Linux. LDC теперь поддерживает NetBSD.

<https://github.com/ldc-developers/ldc>

Системная разработка

PowerNex

PowerNex – это новый проект по созданию ядра, а затем и полной операционной системы на D. Исходники проекта распространяются по лицензии Mozilla Public License 2.0.

<https://github.com/vild/PowerNex>

Trinix

Еще одна ОС на D. Опирается на стандарт POSIX, создается для x86_64. Поддерживает GRUB Multiboot 2, многозадачность, VFS, включает драйвер файловой системы Ext2.

<https://github.com/Bloodmanovski/Trinix>

Terminix

Эмулятор терминала для Linux на основе GTK3. Особенностью Terminix является тайловый дизайн – то есть, можно разбить основное окно на несколько частей, в каждой из которых будет отображаться отдельная консоль.

<https://github.com/gnunn1/terminix>



матграфика в **dlib**

Создаем гистограмму

Гистограмма – это график распределения пикселей по их яркости, в котором по горизонтальной оси представлена яркость, а по вертикали – относительное число пикселей с конкретным значением яркости. Гистограмму можно составить либо для каждого канала по отдельности, либо для суммы всех каналов. Изучив гистограмму, можно получить общее представление о экспозиции, контрасте и цветовом насыщении снимка.

Вот изображение и гистограмма, которую мы в итоге должны из него получить.



В ходе увлекательнейшей работы над одним из проектов, внутри нашей коллаборации LightHouse Software началось повальное увлечение цифровой обработкой изображений. Один из любопытных разделов этой области – анализ изображений, и в этой статье мы покажем, как при помощи библиотеки dlib сгенерировать для изображения гистограмму.

Для построения гистограммы нам сначала нужно загрузить любое доступное изображение (в чем нам поможет мощная и компактная библиотека dlib), после чего пройтись по каждому пикселю и рассчитать его яркость, полагая, что она лежит в интервале от 0 до 255. В dlib яркость пикселя (luminance) дает интервал от 0 до 1, поэтому нам потребуется операция перенормировки.

После расчета яркостей нужно посчитать количество каждого вычисленного значения яркости, для чего используется обычный массив-счетчик, в начале процедуры заполненный нулями.

Вот так выглядит мой вариант построения гистограммы, учитывающий возможность выбора пользователем цветового канала (ALL – суммарная яркость, RED – яркость по красному каналу, GREEN – яркость по зеленому каналу, BLUE – яркость по синему каналу, ALPHA – «яркость» по каналу прозрачности):

```

enum ColorChannel {
    ALL, RED, GREEN, BLUE, ALPHA
};

auto createHistogram(SuperImage source,
    ColorChannel colorChannel,
    Color4f color = Color4f(0.0f, 0.0f, 1.0f))
{
    SuperImage histogram = image(256, 256);
    histogram.fillColor(Color4f(1, 1, 1, 1));
    float[256] quantity = 0.0f;
    size_t index = 0;
    for (int i = 0; i < source.width; i++)
    for (int j = 0; j < source.height; j++)
    {
        final switch (colorChannel) with (ColorChannel)
        {
            case ALL:
                index = cast(size_t)(255 * source[i,j].luminance);
                break;
            case RED:
                index = cast(size_t)(255 * source[i,j].r);
                break;
            case GREEN:
                index = cast(size_t)(255 * source[i,j].g);
                break;
            case BLUE:
                index = cast(size_t)(255 * source[i,j].b);
                break;
            case ALPHA:
                index = cast(size_t)(255 * source[i,j].a);
                break;
        }
        quantity[index]++;
    }
    quantity[] /= 255.0f;

    foreach(number, colorPeak; quantity)
    {
        histogram.drawLine(color,
            cast(int)number, 255,
            cast(int)number, 255 - cast(int)colorPeak);
    }
    return histogram;
}

```

Для выбора канала используется перечисление (enum), а необычная конструкция final switch обеспечивает гарантию того, что весь диапазон входных значений будет обработан (в данном случае такая надежность хоть и является избыточной, но необходимой).

Обращаю внимание на то, что полученные гистограммы слегка отличаются от традиционных форм гистограмм, как минимум, двумя качествами. Во-первых, нормирующий делитель (то есть, то, на что делим для получения процентного отношения по яркостям) отличается от традиционного варианта. Я выбрал значение 255, а в традиционном варианте в качестве делителя используется общее количество всех пикселей изображения. Какой вариант является правильным – сложный вопрос. И второе – отсутствие эквализации: по идее, гистограмма не должна содержать разрывов, но в нашем варианте она потенциально может их содержать.

Несмотря на эти два отличия, гистограммы, создаваемые этой нехитрой процедурой, представляют собой очень ценный и полезный инструмент – в частности, с их помощью легко можно создавать черно-белые изображения, вычлняя нужные для такого преобразования пороги.

Получился в целом простой и занятный инструмент, с помощью которого можно анализировать изображения, что открывает неплохие пути для дальнейших забав и ваших экспериментов.

Олег Бахарев
<http://lhs-blog.info>

Как проектировать хорошие игры?

Когда Atari выпустила свою первую игровую консоль в 70-х, компания была не очень популярной. Ситуация изменилась, когда она разработала игру Space Invaders – продажи резко выросли, и то же самое случилось после выхода Pacman. Для GameBoy от Nintendo подобную роль сыграл Tetris. Так что же особенного в этих играх, и почему именно они стали хитами – и не просто хитами, а настоящими символами индустрии и целым культурным явлением?

*Примерно ту же ситуацию можно наблюдать и на ПК: некоторые игры приносят своим создателям миллионы, в то время как другие обречены на провал, даже если внешне представляют собой практически то же самое, что их более успешные аналоги. На вопрос, как разрабатывать хорошие игры, в 2007 году попытался ответить Марк Овермарс, разработчик Game Maker и известный специалист в области мультимедиа и вычислительной геометрии из Утрехтского университета в Нидерландах. Его тезисы как нельзя актуальны и сегодня – приводим его статью **«How to design good games»** в переведенном и сокращенном виде.*



Что есть игра?

Перед тем, как говорить о хороших играх, мы должны дать определение игре. Как ни странно, по поводу этого простого, на первый взгляд, вопроса есть множество мнений и теорий. Поэтому легче будет для начала сказать, что игрой не является. Итак:

Фильм – это не игра

Кинематограф не предусматривает непосредственного участия зрителя – то же самое можно сказать о литературе или театре. В сущности, было бы даже странно, если бы зритель мог каким-то образом повлиять на исход событий в фильме или литературном произведении.

Головоломка – это не игра

Очевидно, что многие игры содержат элементы головоломок. Однако главное отличие между этими двумя видами развлечений – в наличии или отсутствии динамики. Кроссворд, sudoku или ребус можно разгадать только один раз – игра же предполагает многократное воспроизведение. Чем больше в игре возможных сценариев и путей к выигрышу, тем интереснее в нее играть (так называемая аддиктивность – степень привлекательности многократной игры). Логично предположить, что идеал – это игра, в которой нет статически определенного содержимого, и все ситуации задаются случайным образом. Конечно, в абсолютном смысле это недостижимо, но, так или иначе, все аддиктивные игры содержат в себе элементы случайности, позволяющие каждый раз проходить ее по-разному.



В играх все наоборот: в них зритель-игрок обязательно принимает участие в событиях – и, следовательно, чем больше исход игры зависит от игрока, тем интереснее игра.

Игрушка – это не игра

Разница между игрой и игрушкой достаточно тонка, но все же четко различима. Ребенок, играющий с игрушкой, сам придумывает свои правила и цели. В играх же задачи и цели определены создателями, и игрок не может их изменить. Очень многие интерактивные развлечения, которые относят к играм, на самом деле больше близки к игрушкам: это, например, разного рода симуляторы, «песочницы» и т.д. – в них нет четко установленных целей, игрок сам устанавливает их для себя.

Немного неоднозначно обстоит дело с ролевыми играми, в особенности онлайн-овыми. Сюда же можно отнести, например, серию Sims. Во многих из них нет глобальных целей (либо ее необязательно достигать) – вместо нее есть множество маленьких «будничных» задач, решая которые, игрок развивает своих персонажей. Впрочем, можно сказать, что глобальные цели здесь все же имеются – к примеру, стать самым сильным воином, накопить богатство, выполнить все квесты и т.д. Практика показывает, что такая модель зачастую более удачна, чем жестко заданные условия выигрыша/проигрыша.

Таким образом, компьютерную игру можно определить как программное обеспечение, организующее интерактивное взаимодействие одного или нескольких игроков с виртуальным миром, в процессе которого игроки совершают определенные действия для достижения установленной цели.



Важные аспекты игровой механики

Компьютерная игра во многом родственна играм традиционным (спортивным, настольным и т.д.), однако предполагает несравнимо более развитый аспект интерактивности. Традиционная игра основывается на социальном факторе, она полностью контролируется игроками, судьями и ведущими – она перестает существовать, когда игроки не предпринимают никаких действий. А компьютерная игра живет своей жизнью, даже когда игрок бездействует – чем сильнее эта иллюзия «объективной» виртуальной реальности, тем интереснее игра.

Кроме того, компьютерная игра способна на то, что недоступно для игры традиционной – это самостоятельная адаптация к игроку, изменение сложности в зависимости от его опыта. Машинная логика позволяет динамически делать врагов умнее, а головоломки – сложнее: ведь никому не будет интересно играть в игру, которая стала чересчур легкой.

Другой важный аспект – принятие решений. Игра должна предоставлять игроку выбор, и чем богаче этот выбор, тем интереснее играть. Как ни странно, во многих играх выбор игрока никак не влияет на игровой процесс. Это, например, некоторые гонки, где автомобили зачастую отличаются только внешне, или шутеры со множеством однотипных видов оружия. Подобное, конечно, негативно сказывается на впечатлении от геймплея. Игрок должен чувствовать, что его успех зависит от него самого, от его решений. Именно поэтому следует избегать ситуаций, когда успех или неудача зависят от случайного фактора. Рассмотрим следующий пример. Допустим, есть игра, в которой оружие на уровнях появляется случайным образом. Игроку может не повезти, и есть вероятность, что он не сможет уничтожить сильного врага. Или наоборот – избыток случайно найденного мощного оружия сделает геймплей слишком легким, игрок не будет ощущать удовлетворенность. Намного интереснее, если мощное оружие спрятано в опасном месте – и игрок будет вынужден принять интересное решение: идти на босса со слабым оружием или рискнуть, попытавшись добыть более мощное.

Баланс

Для хорошей игры важен баланс. Под балансом обычно понимают равновесие между ситуациями и характеристиками. Например, мощное оружие должно быть более редким и ценным, чем слабое. Если есть выбор персонажа, то среди них не должно быть однозначно идеального: скажем, большой и сильный персонаж должен медленнее передвигаться, а маленький, соответственно, быстрее.



Решения, которые принимает игрок, не должны быть однозначно выигрышными и однозначно проигрышными (в текстовых квестах часто неправильный ход ведет к гибели – это довольно обидно и не очень справедливо по отношению к начинающим). Скорее, к выигрышу должны приводить комбинации решений, осознанно спланированные и взвешенные игроком.

Леннарт Нэке, преподаватель геймдизайна, приводит три основных условия баланса:

- **Нет доминирующей стратегии**

Несколько разных стратегий могут привести к победе

- **Умение важнее везения**

Случай не должен нивелировать опыт игрока

- **Нет ничего невозможного**

Сложность игры должна быть в пределах способностей играющего.

Плохой баланс довольно часто приводит к возникновению инфляции, что делает игру либо слишком легкой, либо слишком сложной. Например, в играх с экономической составляющей, если игрок имеет возможность быстрого и неограниченного обогащения, то валюта обесценивается, и игра теряет значительную долю аддиктивности.

В онлайн-играх с возможностью торговли между игроками это приводит к росту цен, в результате чего торговля становится возможна только между богатой «элитой», устанавливающей эти цены сообразно своим средствам. Начинающие же игроки, не обладающие нужными средствами, не имеют никакой возможности участвовать в экономических процессах.

Марк Овермарс
(перевод – Тимур Гафаров)

Лучшие свободные игровые движки

Статус самого популярного игрового движка для инди-разработчиков в наши дни приобрел Unity, и подавляющее большинство новых проектов создается именно на нем. Но как быть, если вы хотите написать полностью свободную 3D-игру, без проприетарных компонентов? К счастью, существует не один и не два свободных движка, которые, конечно, не во всем аналогичны Unity, но, тем не менее, могут стать хорошей основой для вашего проекта. Все перечисленные проекты находятся в актуальном состоянии, поддерживаются изначальными авторами или сообществом.

OGRE

Один из самых известных и развитых свободных движков. Разрабатывается с 2001 года Стивом Стриитингом. Название OGRE расшифровывается как Object-Oriented Graphics Rendering Engine. Движок поддерживает Windows, Linux и OS X, рендерит графику через OpenGL и DirectX, включает уникальную скриптовую систему управления материалами, поддержку скелетной анимации, шейдеров (HLSL, GLSL, Cg) и различных впечатляющих спецэффектов.



Загружает текстуры в формате PNG, JPEG, TGA, BMP или DDS. Использует собственный формат для хранения моделей, для которого есть экспортеры ко всем популярным 3D-пакетам. OGRE предназначен для C++, однако существуют порты движка для Python, Perl, Java, .NET и т.д.

Лицензия – MIT.

<http://www.ogre3d.org>

Irrlicht

Достаточно старый и известный движок, разрабатывается с 2003 года. Начался как персональный проект немецкого программиста Николауса Гебхардта, сейчас разрабатывается целой командой. Irrlicht написан на C++ (хотя использовать его можно из десятка других языков, включая Python, Java, .NET-языки и пр.), выводит графику через OpenGL, DirectX и программный растеризатор, поддерживает много разных платформ (официально Windows, Linux и OS X, неофициальные порты существуют для FreeBSD, Xbox, PlayStation Portable, Symbian, iPhone и Native Client). Слово «Irrlicht» является немецким названием так называемых «блуждающих огней» – призраков, которых, по легенде, можно встретить на кладбище.



Технологически движок находится на достаточно высоком уровне, поддерживает OpenGL, DirectX 8, 9, и 11. Есть полная поддержка шейдеров на разных языках (HLSL, GLSL, Cg), загрузка моделей различных распространенных форматов (плюс возможность писать плагины для поддержки новых), скелетная и вершинная анимация, BSP, стеновые тени, система частиц, динамическая вода, GUI, загрузка ресурсов из ZIP-архивов и многое другое. Из известных игр, созданных на Irrlicht, можно отметить Octodad: Dadliest Catch и SuperTuxKart (использует модифицированную версию Irrlicht под названием Antarctica).

Лицензия – zlib.

<http://irrlicht.sourceforge.net>



GLScene

Очень старый и де-факто стандартный 3D-движок для Delphi/Lazarus. Разработка GLScene началась в 1999 году Майком Личке, начиная с версии 0.5 движок стал открытым, и с тех пор лидером проекта является Эрик Гранж. По нынешним меркам GLScene уже не столь технологически развит, но включает практически все, что нужно для создания типовых 3D-приложений: статические и анимированные модели разных форматов, материалы, текстуры, шейдеры, простые системы частиц, HUD-объекты, средства рисования воды и ландшафта, встроенная физика с использованием движка ODE, 3D-звук и многое другое. Рендеринг, как ясно из названия, осуществляется через OpenGL.

Лицензия – Mozilla Public Licence.

<http://glscene.sourceforge.net>



Godot

Очень перспективный свободный движок, который начал свою жизнь в качестве свободного проекта относительно недавно – в 2014 году, а до этого долгое время развивался как внутростудийный инструмент своих разработчиков, компании OKAM.

В отличие от вышеперечисленных движков, имеет собственный эксклюзивный редактор сцен и игровых объектов, в котором можно создавать как 2D, так и 3D-игры.

Godot – не просто графический, а полноценный игровой движок, во многом аналогичный Unity: он включает рендеринг, вывод звука, физику, сеть, различные устройства ввода, а также собственный скриптовый язык GDScript, похожий на Python.

Модели для Godot можно экспортировать из любого 3D-редактора, который поддерживает формат Collada (Blender в их числе).

Графическая часть движка основана на OpenGL ES 2.0, поддерживает шейдеры, динамические тени, HDR, эффекты пост-процессинга, систему частиц, воспроизведение видео и многое другое.

Лицензия – MIT.

<http://godotengine.org>

Blender Game Engine

Или сокращенно BGE. Это встроенный игровой движок популярного пакета 3D-моделирования Blender. Фактически, если вы умеете работать в Blender, вы можете создать игру полностью в нем, не прибегая к другим инструментам (исключая, конечно, 2D-редакторы и программы звукозаписи). BGE кроссплатформенный, работает везде, где запускается сам Blender – а именно, на Windows, Linux, OS X и FreeBSD. Развивается он, конечно, уже не так активно, как прежде, но, тем не менее, поддерживает практически все, что нужно для разработки инди-проекта современного уровня.

BGE выводит графику через OpenGL, поддерживает шейдеры, пост-процессинг, динамические мягкие тени, скелетную анимацию, физику (используется физдвижок Bullet) и звук. Игры можно создавать как без программирования, при помощи специального редактора логики, так и на языке Python (что позволяет расширять возможности BGE за счет сторонних Python-библиотек).

Единственная проблема BGE – достаточно высокая ресурсоемкость и слабая оптимизация, движок стабильно работает только на мощном «железе».

Лицензия – GNU GPL v2.

<http://blender.org>

Blend4Web

Если вы – пользователь Blender и мечтаете создавать мультимедийные веб-приложения, то этот движок буквально создан для вас. Blend4Web – это фреймворк профессионального качества для разработки интерактивного браузерного 3D-контента. Вывод графики осуществляется через WebGL, для создания сцен используется Blender. Движок поддерживает шейдеры, каскадные тени, динамические отражения, скелетную и вершинную анимацию, 3D-звук, физику твердых тел и колесного транспорта, частицы, постпроцессинг, реалистичный рендеринг воды, травы и деревьев, стереоизображение и многое другое.

Особенно интересно, что проект разрабатывается в России – по сути, эта отечественная разработка является основным конкурентом Unity в сфере веба.

Blend4Web существует в двух вариантах: бесплатная версия SDK, распространяющаяся по лицензии GPL и не позволяющая создавать с ее помощью проприетарные приложения, и коммерческая, не требующая раскрытия исходников производного продукта (стоимость – 35900 руб).

Лицензия – GNU GPL v3, dual-licensing.

<https://www.blend4web.com>

Panda3D

Этот движок разработан компанией Disney и Университетом Карнеги-Меллон. Достаточно серьезная профессиональная разработка – движок работает под всеми основными платформами (Windows, Linux, OS X, FreeBSD), в качестве ЯП использует Python.



Поддерживает шейдеры, 3D-звук, обнаружение столкновений, физику (ODE, PhysX и Bullet), различные устройства ввода, GUI, сеть, искусственный интеллект. Контент для Panda3D можно создавать в Blender, 3ds Max или Maya, есть также поддержка популярных форматов моделей (Collada и др.). В дистрибутив движка включено множество утилит, конвертеров, есть богатый инструментарий для отладки и профайлинга приложений.

Лицензия – BSD.

<http://panda3d.org>

idTech 4

Неплохой выбор для поклонников продукции id Software и тех, кто имеет опыт моддинга этих игр. Движок изначально создавался для Doom 3, впоследствии на его основе вышли Quake 4, Prey, Enemy Territory: Quake Wars и др. В 2011 году исходники движка были размещены в свободный доступ под GNU GPL.

idTech 4 был революционной на момент своего появления технологией: в далеком 2004 году, когда вышел Doom 3, игр с подобным уровнем графики еще не было. Именно движок от id Software задал планку всем играм AAA-класса – в нем активно использовалось попиксельное освещение, normal mapping, теневые объемы, сложная скелетная анимация и т.д.

Сегодня, конечно, уровень idTech 4 уже не назовешь передовым, но, тем не менее, для инди-проекта его возможностей более чем достаточно.

Движок предназначен для C++, поддерживает Windows, Linux, OS X, Xbox 360 и PlayStation 3.

Лицензия – GNU GPL v3.

<https://github.com/Timo/doom3.gpl>

Продолжение следует...



Мобильный FPS



Теперь любимый журнал всегда с вами!

Читайте FPS на мобильных устройствах:
скачайте приложение для Android или iOS!



Available on the
App Store



ANDROID APP ON
Google play

Разработчик приложения: цифровое издательство St.Appler <http://www.stappler.org/>

Linux-гейминг

Игровые новости из мира Linux

Открываем обзор большой новостью: консорциум Khronos Group наконец-то опубликовал спецификацию Vulkan 1.0. Напомним, Vulkan – это графический API нового поколения, который должен преодолеть недостатки OpenGL и Direct3D, сократив прослойку между пользовательским кодом и видеодрайвером, что позволит более глубоко оптимизировать приложения.



Vulkan подразумевает более эффективную работу с GPU, минимизируя накладные расходы, вносимые драйвером, и предоставляя встроенную поддержку многопоточной обработки команд. Также новый API предполагает вынос генерации команд GPU на сторону приложения (в противовес компиляции шейдерных программ на стороне драйвера, как это происходит в OpenGL) путем использования универсального байт-кода – промежуточного представления SPIR-V.



Еще одна особенность Vulkan – унифицированный API для всех платформ. Как известно, существует три разных стандарта OpenGL для трех основных платформ – OpenGL, OpenGL ES и WebGL. Vulkan же будет охватывать все доступные платформы и предоставляет единый API для настольных, мобильных и web-систем.

Одновременно компании Intel, AMD, NVIDIA, Qualcomm и Imagination выпустили новые видеодрайверы с поддержкой Vulkan. Для разработки игр на основе Vulkan компания Valve опубликовала LunarG SDK для Linux и Windows. Компания Google подготовила SDK для Android. Поддержка Vulkan вскоре ожидается и в Mesa.

Поле десяти лет разработки доступна игра SuperTux 0.4 – свободный 2D-платформер в духе Mario с пингвином Туксом в главной роли. Новый выпуск примечателен полной переработкой движка, переходом на OpenGL, OpenAL и SDL2.



Также в игре реализованы новые уровни и персонажи, добавлено много новых графических и звуковых эффектов и элементов геймплея. Для моддеров имеется даже скриптовый API, при помощи которого можно задавать свою логику уровней.

<http://supertuxproject.org>



Еще один подарок для любителей классики – NetHack 3.6.0, новая версия культового текстового «рогалика». Игра представляет собой очень сложный, динамический и непредсказуемый фэнтезийный мир, в котором игрок сражается с различными монстрами, торгует, развивается и продвигается по уровням лабиринта для того, чтобы в итоге завладеть Амулетом Йендора. Успешному прохождению NetHack способствует изучение исходного кода (смысл слова «hack» в названии) и обсуждение стратегии с другими игроками по сети (отсюда слово «net», хотя сама игра сугубо одиночная).

Кроме общих исправлений, в новом выпуске отражена серия фундаментальных изменений в подходе к разработке, представленных обновленной командой разработчиков. Внесены улучшения в игровой процесс, которые были заимствованы из форков UnNetHack и NetHack 4.

<http://www.nethack.org>

На хакерской конференции Chaos Communication Congress группа Fail0verflow продемонстрировала успешный взлом консоли PlayStation 4. Им удалось не только осуществить джейлбрейк, но и установить на PS4 полноценную версию Linux вместо штатной операционной системы Orbis OS, превратив приставку в полноценный компьютер.

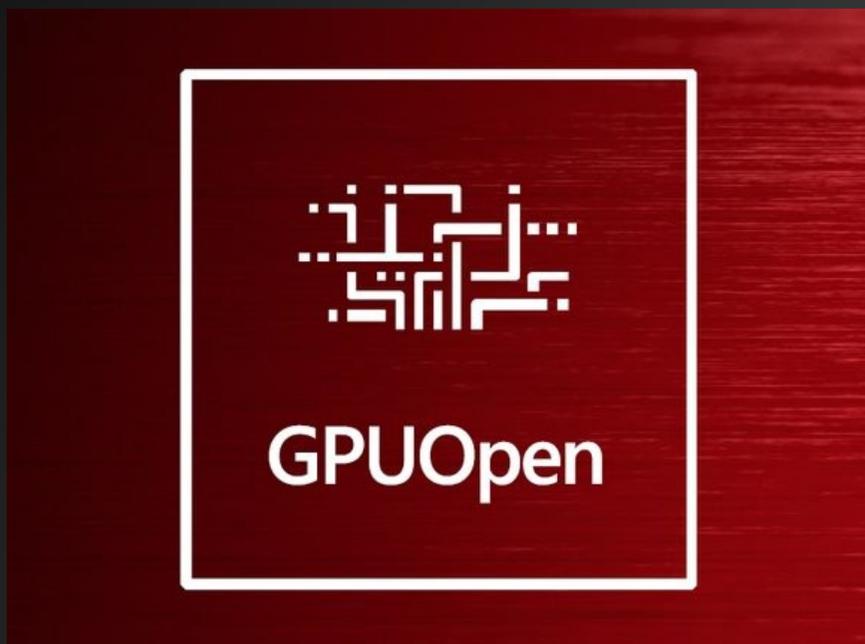


Для обхода ограничений консоли был использован эксплоит уязвимостей в движке WebKit и работающем на уровне ядра драйвере GPU. В загруженном окружении Linux был запущен рабочий стол Xfce, обеспечена работа Wi-Fi, Bluetooth, Ethernet, HDMI-видео и вывода звука. Пока, впрочем, не работает 3D-ускорение, HDMI-звук и порты USB.

В середине декабря 2015 года увидела свет новая версия Mesa – 11.1.0. Выпуск имеет экспериментальный статус – после проведения окончательной стабилизации кода будет выпущена стабильная версия 11.1.1. В новой ветке основное внимание сосредоточено на развитии VAAPI, EGL и второстепенных драйверов. Значительных изменений в поддержке новых версий OpenGL 4.x не отмечено, драйверы RadeonSI (HD 7700-7900, HD 8000, Rx 240-290, Rx 300) и Nouveau (GeForce 400/500/600 на базе GPU Fermi и Kepler) как и в прошлом выпуске поддерживают OpenGL 4.1, а драйвер Intel i965 поддерживает специфичные расширения OpenGL 4.2, но пока реализованы не все расширения OpenGL 4.0 и 4.1.

Компания AMD, между тем, анонсировала новые планы в отношении разработки открытых драйверов и инструментов для разработчиков графических приложений под свои GPU. Новая стратегия подразумевает распространение разработок в форме открытых проектов, что, конечно, не может не радовать.

Кроме ранее представленной инициативы Boltzmann, в рамках которой ведется разработка открытых средств для гибридных вычислений на CPU и GPU, компания представила новый проект GPUOpen, в котором планируется подготовить альтернативу проприетарному набору библиотек GameWorks от NVIDIA, делая упор на открытый исходный код и совместную разработку, рассматривая их как ключевые конкурентные преимущества.



Инициатива позиционируется как мультиплатформенная и охватывает как разработку приложений для Linux с использованием OpenGL и Vulkan, так и для Windows с использованием DirectX. Среди целей GPUOpen отмечается упрощение портирования игр с консолей Xbox One и PlayStation 4 на персональные компьютеры.

Для задействования вычислительных возможностей GPU компания AMD опубликовала ряд наработок в области гетерогенных вычислительных систем. Наиболее интересным является компилятор HCC C++, который позволяет использовать при выполнении приложения ресурсы GPU и предоставит средства для трансляции кода с расширениями CUDA.

Что касается драйверов, то проприетарные драйверы AMD в скором времени будут переведены на общие с открытыми драйверами компоненты AMDGPU, в которые компания постарается перенести как можно больше кода – это упростит сопровождение, решит проблему с адаптацией проприетарного драйвера для новых версий ядра Linux и ускорит перенос новых возможностей в открытый драйвер.

А исследователи из Бингемтонского университета (США) сообщили о первом успешном внедрении открытого GPU, разрабатываемого в рамках OpenHardware-проекта Nyuzi (бывший Nyami), наработки которого распространяются под свободной лицензией Apache 2.0. Графические процессоры, основанные на архитектуре Nyuzi, задействованы в серии экспериментов с целью сравнения влияния изменения аппаратной и программной конфигурации на производительность схемы.

Nyuzi позиционируется как многопоточный гибрид CPU и GPU (GPGPU), построенный с использованием полностью программно определяемой архитектуры, не привязанной к фиксированным аппаратным реализациям и способной выполнять различные виды вычислений в разных потоках.

Аппаратная реализация синтезируется из схем SystemVerilog для выполнения на типовых FPGA, таких как Altera Cyclone IV.

ИГРОВАЯ ФОТОГРАФИЯ

Новое направление в искусстве?

Фотография прочно вошла в нашу жизнь и окружает нас буквально повсюду – особенно заметно это стало в последние два десятилетия, с появлением дешевых цифровых фотоаппаратов и разного рода мобильных устройств, при помощи которых фотографирование стало обыденной (если не сказать машинальной) частью нашей повседневности. Хорошо это или плохо – сказать трудно: с одной стороны, мир стал более детально и тщательно документироваться миллионами людей, ежеминутно иллюстрирующих нашу реальность в Интернете, но с другой – фотография стала обесцениваться, терять свою «магию».



Обычными портретами или пейзажами сегодня никого уже не удивить: поскольку сейчас каждый сам себе «фотограф», профессионалам приходится идти на разного рода ухищрения, применять специализированную аппаратуру, изобретать необычные эффекты и искать редкие кадры. Впрочем, для того, чтобы снять шедевр, вовсе необязательно покупать дорогую камеру и ехать в экзотические страны: зарождающееся новое направление фотоискусства не требует ничего, кроме компьютера с установленными играми!

Речь идет о «внутриигровой фотографии» (in-game photography, IGP) – иными словами, искусстве создания скриншотов. Вы, конечно, не раз делали скриншоты из любимых вам игр – но пробовали ли вы при этом мыслить и действовать, как фотограф? Найти удачный ракурс и интересный свет, подобрать композицию, выгадать момент и нажать PrintScreen...

Конечно, все это может показаться несколько наивным – ведь игровые вселенные сами по себе являются чьим-то произведением, результатом работы художников и программистов. А смоделировать и отрендерить можно все, что душе угодно, и для этого вовсе не требуется игра как таковая. Следовательно, у создания скриншотов, казалось бы, мало преимуществ перед «обычным» CG. Но ведь то же самое можно сказать и по поводу физической фотографии: зачем тратить деньги на фототехнику и силы на поиск нужной природы, если художники могут реализовать абсолютно любой творческий замысел с нуля?



Ответ, конечно, прост – работа художника стоит намного дороже. Фотография, как ни крути – более дешевое и доступное искусство, а IGP – соответственно, еще более дешевое и доступное. Но они от этого отнюдь не перестают быть искусством: если через игровые скриншоты можно выражать мысли и эмоции, то почему бы, действительно, не делать этого?

Игры, в сущности, исследованы гораздо меньше, чем кажется. Большая часть скриншотов, которые можно видеть в пресс-релизах и обзорах – не более чем технологическая демонстрация, они всего лишь показывают качество графики. Но истинную красоту виртуальных миров можно оценить, только посвятив той или иной игре достаточно много времени: и вот здесь талант фотохудожника мог бы здорово помочь – как игрокам, так и разработчикам. Ведь главная задача художника – передать «дух» изображаемого объекта, его внутреннюю сущность. А это куда важнее, чем бездумно «щелкать» сцены со спецэффектами и красивыми девушками.



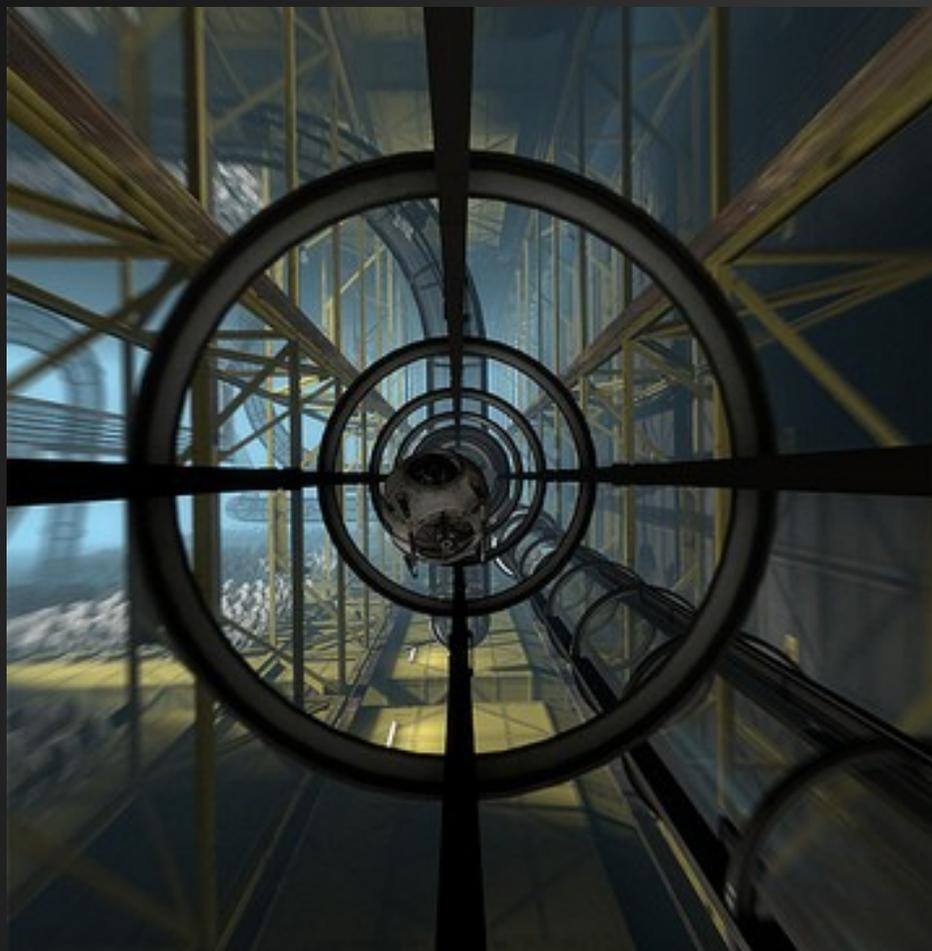
Хотя, в общем-то, красивые девушки тоже могут быть предметом игрового фотоискусства – это успешно доказывает **Дункан Харрис** и его творческие исследования Rise Of The Tomb Raider или Guilty Gear Xrd. Сделать качественный игровой портрет гораздо сложнее, чем пейзаж или натюрморт (как, собственно, и в традиционном искусстве), и Харрису это блестяще удается. Писатель Алекс Гарланд так отзывается о его скриншотах: «В своих исключительных работах Харрис находит и великолепно передает пейзаж, детали, настроение и эфемерные моменты. Он заставляет зрителя увидеть игру свежим взглядом».

Джеймс Поллок, однофамилец знаменитого абстракциониста и полный тезка другого известного Поллока-художника (простая случайность?) – образцовый представитель пейзажной IGP. В **своем Tumblr** он выкладывает уличные сцены GTA IV, урбанистические композиции из Mirror's Edge, природные виды Skyrim и даже пейзажи из Minecraft. Особую роль в его творчестве играет пост-обработка: Поллок активно использует обесцвечивание, виньетирование и ломографические эффекты. Его скриншоты невероятно жизненны и как нельзя более близки к физической фотографии. В тех играх, где нет простого способа делать снимки экрана (например, на консолях), Поллок применяет реальный фотоаппарат – этим во многом и объясняется «физичность» его работ.



Весьма интересны и пейзажи Йена Эндрюса, геймера из Австралии – он тоже публикует их [в Tumblr](#). Объектом его внимания являются Max Payne 3, Dishonoured, Team Fortress 2, Assassin's Creed Revelations, RAGE и многие другие известные тайтлы. Эндрюс – мастер деталей и необычной компоновки кадра, его скриншоты отличаются особым колоритом, богаты сложной фактурой и светоненевыми отношениями. Особое внимание он уделяет текстурам поверхностей: старое дерево, ржавчина, грязь, кирпичные стены – все эти обычно неприметные глазу детали в скриншотах Эндрюса начинают жить собственной жизнью и выстраиваться в целостное повествование.





Джош Тейлор работает целыми сериями: например, «A Distant Sadness» – это фотоотчет о событиях Battlefield 3, практически неотличимый, внешне и по духу, от реальных фотографий с мест военных действий. Особый интерес представляет серия «Gods of the Heavens», посвященная культовому Eve Online – коллекция пейзажей бескрайнего космоса. И, наконец, «Faces of the Apocalypse» – цикл портретов персонажей RAGE.

IGP занимаются не только геймеры, но и вполне профессиональные фотографы и дизайнеры – например, **Леонардо Санг**, который называет свое творчество «фотографией виртуальной реальности» (virtual reality photography). Его творчество включает пейзажи из Battlefield 3 и 4, Bioshock Infinite, Star Wars Battlefront, а также излюбленный всеми GTA IV. Скриншоты Санга характерны классической строгой композицией в духе художников Ренессанса, монументальностью образов, атмосферностью и изящным колоритом.

IGP – конечно, только зарождающееся направление искусства, но у него есть высокий потенциал: игры с каждым годом становятся все более реалистичными, появляется много открытых миров, которые могут стать поистине неисчерпаемым источником вдохновения для «виртуальных фотографов». Кто знает – может быть, наступит время, когда виртуальная реальность станет для художников даже более интересным объектом, чем реальность физическая?..

Тимур Гафаров

Linux: ПОЛЕЗНЫЕ КОМАНДЫ

Давненько я не делился с вами, уважаемые читатели, новыми командами и bash-скриптами! В последний раз это было, кажется, в 2014 году. Надо исправить это недоразумение – тем более, что за два года моя коллекция значительно пополнилась...

Работая над одним веб-проектом, понадобилось записать двоичные данные в CSS – для этого используется кодирование Base64:

```
$ base64 -w 0 inputFile > outputFile
```

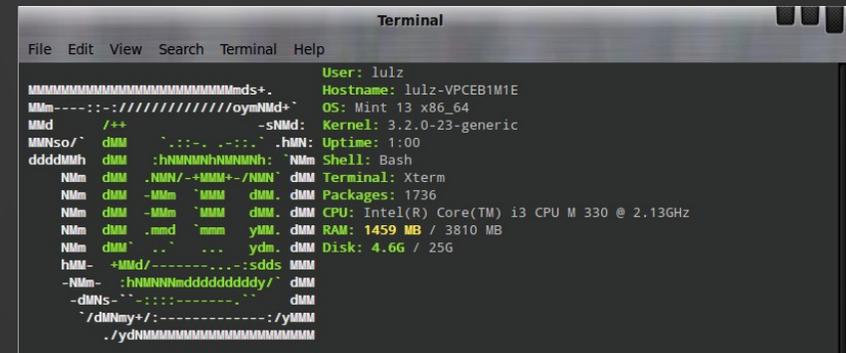
Если у вас, как у жителей Острова Невезения, нет календаря, но есть linux-терминал, можно набрать короткую команду

```
$ cal
```

...и увидеть календарь на текущий месяц.

Как-то раз пришлось конвертировать много RAW-фотографий из камеры Sony. Это можно сделать при помощи программы dscraw – очень полезно, когда под рукой нет профессионального RAW-проявщика, либо нет времени/желания запускать его:

```
$ dscraw DSC01269.ARW
```



Любой линуксоид регулярно использует команду ls, выводящую список файлов в текущем каталоге. А вот как заставить ее показывать скрытые файлы (это те, что с точкой в начале) – знает, наверное, не каждый:

```
$ ls -a
```

Очень часто приходится удалять с диска однотипные файлы. Ни в коем случае не тратьте свое время, не делайте это вручную – у вас же есть всемогущий find! Вот пример использования этой команды для рекурсивного удаления файлов *.рус в текущем каталоге:

```
$ find . -name "*.рус" -exec rm -rf {} \;
```

Напоминаю, что find также можно использовать для того, о чем говорит его название – для поиска файлов:

```
$ find ./ -name filename
```

А вот этот шедевр bash выполняет узкоспециализированную, но необычайно полезную для меня задачу – ищет заданный текст в ODT-файлах. Дело в том, что я часто имею дело с крупным архивом документов LibreOffice Write, который никак автоматически не индексируется – и, если нужно найти в нем документ с определенным известным текстом, без скрипта не обойтись:

```
#!/bin/bash
if [ $# -ne 2 ]; then
    echo "Usage: search searchpath searchterm"
    exit 1
fi
find $1 -name "*.odt" | while read file
do
    unzip -ca "$file" content.xml | grep -qli "$2"
    if [ $? -eq 0 ]; then
        echo "Found keyword in: " $file
    fi
done
```

Забыли, что делали час назад? Не беда – вам поможет утилита history, которая выводит историю введенных вами консольных команд:

```
$ history
```

Тимур Гафаров

Наши проекты

Cook

Программа автоматизации сборки проектов на языке D. В отличие от аналогичных инструментов (Make, CMake, Scons, Jam, DSSS и др.), Cook не требует конфигурационного файла: всю информацию о проекте она получает самостоятельно, сканируя модули (файлы *.d). При этом программа отслеживает прямые и обратные зависимости между модулями: если модуль был изменен, необходимо скомпилировать заново не только его, но и все модули, которые от него зависят (это важно, если был изменен внешний интерфейс модуля: объявления классов, семантика шаблонов и т.д.). Для этого Cook производит лексический анализ модулей - но не всех, а только тех, которые были изменены со времени последнего анализа. Данные анализа кэшируются в файл для повторного использования (кэш автоматически обновляется при пересборке). Cook работает в Windows и Linux.

<http://github.com/gecko0307/cook2>

dlib

Коллекция библиотек «на все случаи жизни» для D, которая может быть использована в игровых движках и других мультимедийных приложениях. Написана на D2 с использованием Phobos, не имеет никаких других внешних зависимостей. Разработка dlib пока находится на ранней стадии – API нестабилен и может измениться в любой момент, если появится возможность улучшить общую архитектуру.

<http://github.com/gecko0307/dlib>

Это все!

Надеемся, номер вышел интересным. Если Вам нравится наш журнал, и Вы хотели бы его поддержать – участвуйте в его создании! Отправляйте статьи, обзоры, интервью и прочее на любые темы, касающиеся игр, графики, звука, программирования и т.д. на gecko0307@gmail.com.



<http://fps-magazine.cf>