

35
2015

BLENDER:
Новости
Планы на 2015 год

OpenAL
Бинауральный 3D-звук

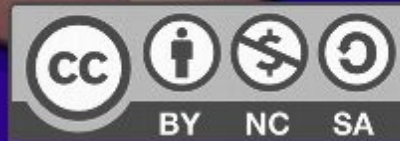
Как стать хакером?

Проблема инструментария:
чего не хватает инди-разработчикам?

VULKAN
Графический API
нового поколения

+ многое
другое!

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





FPS

№35

FPS — бесплатный, свободно распространяемый электронный журнал, посвященный разработке игр и другим видам цифрового творчества.

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 пакетам и редакторам, а также различные статьи по теоретическим вопросам, дизайну и философии компьютерных игр.

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

© 2008-2015 Редакция журнала «FPS». Некоторые права защищены. Все названия и логотипы являются интеллектуальной собственностью их законных владельцев и не используются в качестве рекламы продуктов 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 2015: планы на ближайший год
- :: Обзор дополнений, выпуск 14

• 2D-графика

- :: Новости

• Кодинг

- :: OpenAL: бинауральный 3D-звук
- :: Проблема инструментария
- :: Трудный выбор
- :: Как стать хакером?

• Linux-гейминг

- :: Игровые новости из мира Linux
- :: Vulkan — новый API на смену OpenGL

• От Монополии до Warhammer

- :: Магия настольных игр

• «Конструируя чувства...»

- :: Техно-симфония Виктора Аргонова



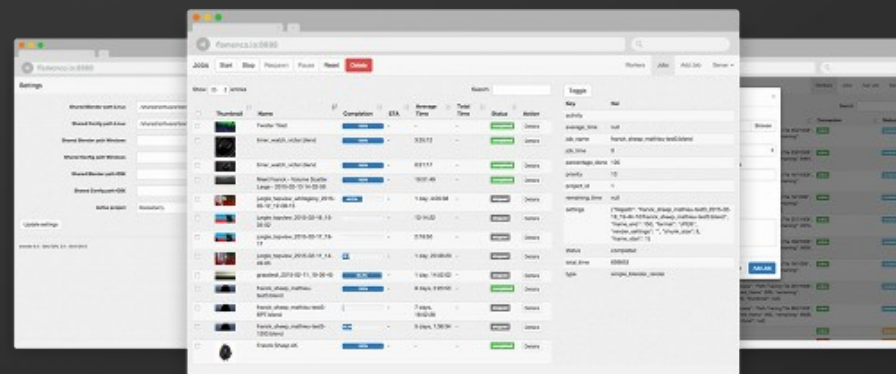
Blender

Новости

Увидел свет Blender **2.74**. Одно из ключевых нововведений – новая система композитинга в окне проекции сцены, которая поддерживает такие эффекты, как Ambient Occlusion и Depth of Field. В будущем планируется добавить поддержку узлов для создания новых эффектов. Еще одна важная новинка – долгожданная поддержка редактируемых нормалей: есть API для Python, который позволяет загружать и сохранять нормали мешей, а также базовый модификатор для автоматического переназначения нормалей.

Также в эту версию включены улучшения, сделанные в рамках проекта **Gooseberry**: в основном, это обновление системы рендеринга волос. Так, добавлена начальная поддержка обнаружения столкновений волос с мешами, динамика волос с учетом их объема, новый режим деформации волос («Спираль») и многое другое.

Изменения также коснулись рендер-движков Cycles и Greestyle, а также режима лепки, рисования и Grease Pencil. Скачать Blender 2.73 для Windows, Linux и Mac OS X можно на странице <https://www.blender.org/download>



«Побочные результаты» Gooseberry дают о себе знать уже достаточно давно: так, еще в феврале команда проекта представила альфа-версию своего рендер-менеджера **Flamenco**. Это клиент-серверная система, написанная на Python, которая позволяет отслеживать и управлять различными процессами студийного конвейера: задачами рендеринга, симуляции, кодирования и т.д. Система рассчитана на быстрое развертывание на любом Python-сервере и легкую интеграцию в сторонние инструменты.

Flamenco является свободным ПО, исходники доступны на **GitHub** по лицензии GPL 2.0. Подробности – на официальном сайте: <http://www.flamenco.io>



А вот еще один открытый сетевой рендер-менеджер – **BlendWorks**. Обеспечивает подключение 50 клиентов, включает веб-интерфейс, поддерживает смартфоны и планшеты.

Исходники можно найти здесь:

<https://github.com/timoschwarzer/blendworks>

Начало весеннего сезона ознаменовалось также появлением нового ресурса для сообщества Blender – сайта **Blender Hub** (<http://www.blenderhub.com>). Это платформа для публикации видеоуроков – пользователи могут размещать видео с YouTube и Vimeo с указанием уровня сложности материала (beginner, intermediate, advanced).

Авторы Blender Hub надеются, что сайт поможет начинающим пользователям быстрее овладеть программой, а авторам видеоуроков – занять многочисленную аудиторию.

По сложившейся традиции, не обойдем стороной и новости мира рендер-движков. Так, недавно произошло большое событие: студия Pixar, как и было обещано на SIGGRAPH 2014, выпустила бесплатную версию своего знаменитого рендер-движка **RenderMan**!

Доступны версии для Windows, Linux и Mac OS X (только для 64-битных систем). Есть готовые экспортеры для Maya и Katana, в разработке также – для Houdini и Cinema4D. Движок также может работать в качестве самостоятельного инструмента и рендерить сцены в формате RIB.

RenderMan Free полнофункционален, но только для некоммерческого использования – для извлечения прибыли от создаваемых в нем работ, придется приобрести лицензию (\$495).



Напомним, RenderMan – это не просто рендер, подаривший миру анимационные шедевры Pixar, это де-факто стандарт в области 3D-визуализации: его формат данных RIB и язык описания шейдеров SL широко используются крупнейшими голливудскими продакшен-хаусами. Так, RenderMan был использован при создании фильмов «Аватар», «Пираты Карибского моря», «Человек-паук 3» и многих других.



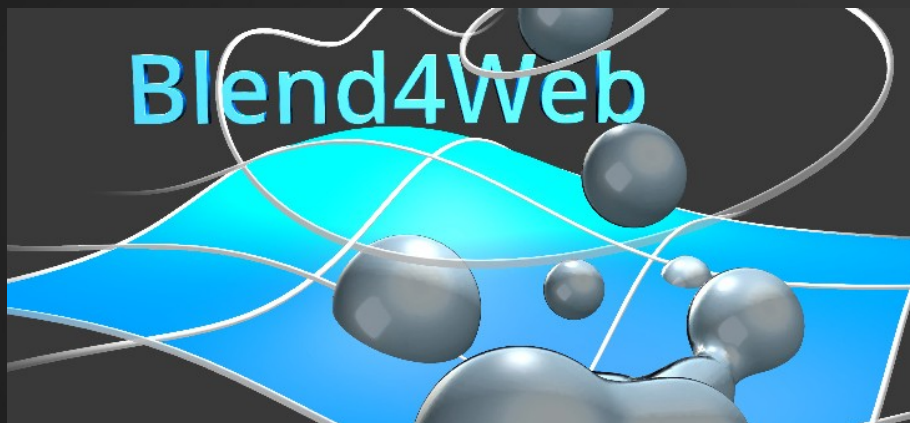
Поддержка RenderMan в Blender реализована в лице проекта **RIB Mosaic**, полнофункционального экспортера в RIB (формат сцен RenderMan).

Скачать его можно на странице проекта на SourceForge:

<http://sourceforge.net/projects/ribmosaic>



А ранее увидела свет новая версия свободного физически корректного рендера **LuxRender** – 1.4. Добавлена поддержка новых материалов, текстур и типов освещения, улучшена real-time интеграция в Blender. Подробности – на сайте проекта: <http://www.luxrender.net>.



Также недавно состоялся релиз **Blend4Web 15.03** – свободной платформы для создания браузерных 3D-приложений.

В этой версии в состав SDK был добавлен инструмент для создания аннотаций, API для создания копий объектов в рантайме, базовая реализация профиля Blend4Web для самого Blender.

Напомним, Blend4Web предназначен для создания трехмерного интерактивного контента, работающего в браузерах без использования плагинов: воспроизведение осуществляется средствами WebGL, Web Audio и других браузерных технологий.

Движок тесно интегрирован с Blender, который используется в качестве основного инструмента редактирования 3D-сцен. Нарботки проекта распространяются под лицензией GPLv3, при желании возможно также приобретение коммерческой лицензии.

<https://www.blend4web.com/ru>

Кстати, понемногу продвигается работа по портированию Blender на **Android** – на официальном сайте появились APK-сборки проекта: <http://download.blender.org/demo/android>

Напомним, что впервые этот порт был представлен в 2012 году – эту работу проделал Александр Кузнецов в рамках одного из проектов GSoC 2012.



Интересный проект с использованием Blender анонсировали в феврале наши соотечественники: запущена в продакшн визуальная аниме-новелла **«2032»**, создаваемая по мотивам известной техно-оперы Виктора Аргонова «2032: Легенда о несбывшемся грядущем».



Графический контент готовится в Blender и Inkscape, а сам проект работает на основе движка Ren'Py (как альтернатива рассматривается также Unreal Engine 4). Кстати, инициатор проекта – автор опенинга и тизера культовой интерактивной новеллы «Бесконечное лето», так что мы имеем все шансы увидеть очередную шедевр в рамках жанра!

Подробную рецензию на саму техно-оперу, а также на остальное творчество Аргонова читайте ниже в этом номере журнала.

Сайт-блог проекта: <http://2032ls.blogspot.ru>

Еще один игровой проект, который активно разрабатывается на Blender – это сюрвайвл от первого лица **The Island** от автора Stars и Tom's Adventures.

По жанру игра напоминает культовую серию Stranded, но отличается более реалистичной графикой и физикой, а также уникальной системой конструирования: здесь вы можете построить из подручных материалов все, что угодно, а не только запрограммированные разработчиком вещи.

Как и в других играх подобного плана, игрок вживается в роль «Робинзона»: вы можете охотиться, разводить огонь, создавать орудия труда – все, чтобы выжить на необитаемом острове.



Проект еще далек от завершения, но уже сейчас выглядит многообещающе – было бы здорово увидеть в итоге аналог The Forest или, на худой конец, улучшенный Stranded 2.



Жанр survival, будучи относительно новым, в последнее время оказался в центре внимания многих разработчиков. Так, полным ходом идет разработка **Stranded 3!**

Blender

НАСТОЛЬНАЯ КНИГА

В третьей версии знаменитого «симулятора Робинзона» игроков ждет многопользовательский режим, современная графика, поддержка языка Lua для написания скриптов, поддержка Linux и Mac OS X. Контент для игры создается в Blender.

Напомним, Stranded – это серия симуляторов выживания на необитаемом острове. Персонаж, за которого вы играете от первого лица, имеет параметры голода, жажды и усталости, которые периодически нужно восстанавливать – искать пропитание, пить пресную воду и спать (желательно не на голой земле). Для этого необходимо создать орудия труда, оружие, построить шалаш или хижину – вы исследуете остров, развивая полезные навыки и разрабатывая собственную стратегию выживания.

Подробности – на сайте проекта: <http://www.stranded3.com>



В марте вышел 46 номер журнала «BlenderArt» – самого популярного англоязычного электронного издания, посвященного Blender.

В этом выпуске читателей ждет много полезной информации: уроки, статьи жара «making-of», галерея работ и другие материалы.

<http://www.blenderart.org>

Журнал «FPS» отслеживает все самые свежие новости из мира Blender, моделирования, анимации и рендеринга! В следующем номере ждите очередную подборку новостей – оставайтесь с нами и держите руку на пульсе последних событий!

«Blender. Настольная книга» – это проект от журнала «FPS» по созданию полноценного русскоязычного электронного руководства по основам работы в Blender. Целевая аудитория – начинающие пользователи программы (как перешедшие со старых версий, так и начинающие знакомство с Blender «с нуля»). Книга будет представлять собой сборник статей, охватывающих различные аспекты использования Blender, скомпонованных по принципу «от простого к сложному».

Издание будет распространяться бесплатно, по лицензии Creative Commons BY SA. На данный момент книга находится в процессе верстки.

К работе над проектом приглашаются все желающие! На почтовый ящик редакции (gecko0307@gmail.com) принимаются статьи и уроки, а также общие советы и предложения. Кроме того, книге нужны графические материалы: авторские художественные работы, интересные скриншоты, демонстрационные рендеры, схемы, диаграммы и т.д. Весь Ваш вклад в книгу обязательно будет учтен, и Ваше имя будет указано в списке авторов.





Blender 2015

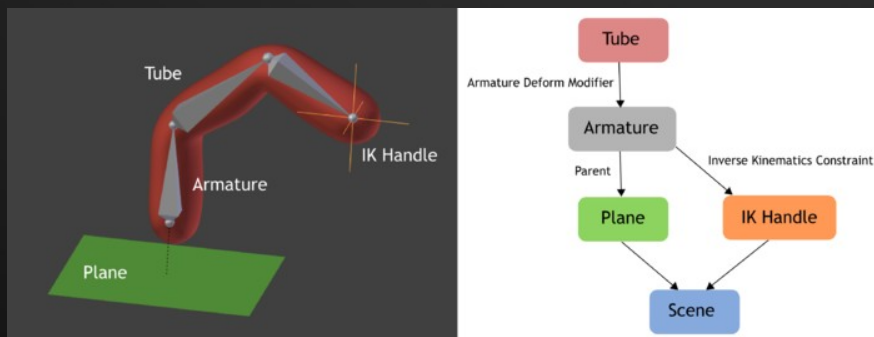
Планы на ближайший год

Кажется, что активность разработки Blender достигает своего апогея – с каждым релизом мы получаем тонны новых функций и полезных возможностей: за последние годы Blender стал не хуже (а то и лучше) коммерческих аналогов. Однако всем нам интересно – что же будет дальше? Тон Розендаль недавно опубликовал подборку наиболее значительных нововведений, которые пользователи могут увидеть в Blender уже в течение этого года.

Граф зависимостей

Разработчики: Сергей Шарыбин, Джошуа Люн
Вероятность завершения в 2015 году: **100%**

Это разрабатываемая система управления связями между различными компонентами Blender, которая должна сделать механизм работы программы более структурированным и надежным. Особенно граф зависимостей важен в области анимации/кинематики, где огромное значение имеет порядок выполнения модификаторов.



Мультипросмотр

Разработчики: Далай Фелито

Вероятность завершения в 2015 году: **100%**

Данная система уже много лет обозначена как TODO. Мультипросмотр позволит более гибко настраивать режимы отображения сцены в интерфейсе Blender – к примеру, добавить поддержку стереоскопического 3D.

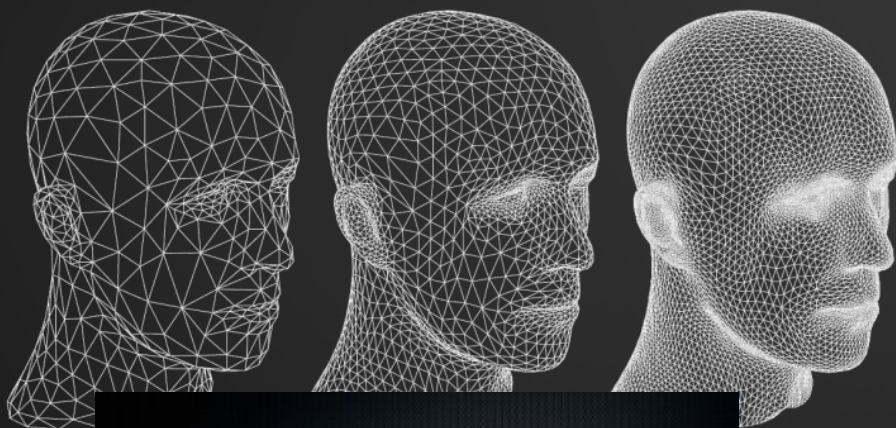


OpenSubdiv

Разработчики: Сергей Шарыбин

Вероятность завершения в 2015 году: 90%

Мы уже не раз писали об этом проекте: OpenSubdiv – это библиотека подрабзиения мешей по алгоритму Кэтмулла-Кларка от Pixar, которая поддерживает ускорение на GPU, что дает значительный прирост скорости и интерактивности вьюпорта. Этот проект был, по сути, готов уже прошлым летом, но интеграция в основную ветку была отложена до нового релиза OpenSubdiv от Pixar во втором квартале 2015 года.



 **Open
Subdiv**

BLENDER



Alembic

Разработчики: Лукас Тонн

Вероятность завершения в 2015 году: 90%

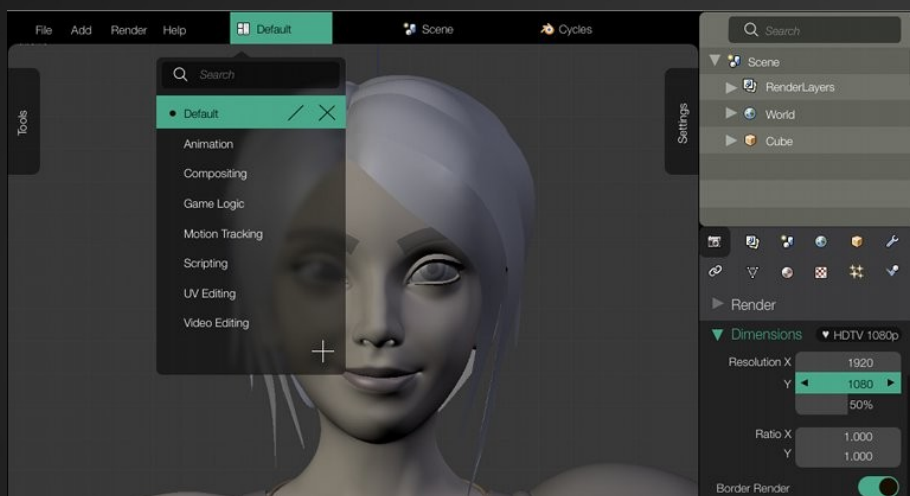
Ведется активная работа по интеграции в Blender формата хранения данных Alembic – открытой альтернативы FBX от ILM и Sony Pictures Imageworks, которая уже не в первый раз радует OpenSource-сообщество своими интересными разработками (первым был язык описания шейдеров OSL). В Blender планируется использовать Alembic для хранения кэша анимации и физики – это сделает возможным конвейерный обмен «запеченными» данными между Blender и другими 3D-редакторами.

Пользовательские манипуляторы

Разработчики: Энтони Рякиотакис

Вероятность завершения в 2015 году: **неизвестно**

Проект по редизайну манипуляторов и других элементов UI в окнах просмотра сцены. В данный момент тестируется новая система виджетов, предназначенная для управления скелетной анимацией.



Workflow

Разработчики: Джонатан Уильямсон

Вероятность завершения в 2015 году: **неизвестно**

Разработчики Blender прислушались к некоторым идеям сообщества о общем улучшении интерфейса Blender – в ближайшем будущем ожидается появление рабочего прототипа новых режимов UI, которые помогут сделать поток работы в Blender более эргономичным.

Новые инструменты моделирования

Разработчики: Кэмпбелл Бартон

Вероятность завершения в 2015 году: **100%**

Не забыты и инновации в области редактирования мешей: в течение этого года должны появиться новые инструменты лепки, ретопологии и UV-развертки.

Окно просмотра сцены

Разработчики: Джейсон Уилкинс, Энтони Рякиотакис

Вероятность завершения в 2015 году: **95%**

Вьюпорт Blender создавался с учетом спецификации OpenGL 1.0 – это было почти 20 лет назад. Теперь мы стоим на пороге важного обновления в этой области: система отображения контента в Blender будет значительно пересмотрена.

Скорее всего, она будет работать полностью на основе средств OpenGL 2.0 (шейдеров GLSL), поэтому новые версии Blender будет невозможно запустить на старых системах. Такой серьезный шаг компенсируется более высокой гибкостью вьюпорта – например, в нем можно будет динамически создавать любые новые элементы интерфейса путем написания скриптов.



Волосы и частицы

Разработчики: Лукас Тонн

Вероятность завершения в 2015 году:

100% для динамики, 50% для узлов

Проект Gooseberry предусматривает значительные улучшения системы рендеринга волос в Blender. В первую очередь, это касается физической симуляции динамики волос – в Blender 2.74 уже появилась обработка столкновений с использованием Bullet. В идеале, система волос должна быть переведена на узлы.



Все на узлах

Разработчики: Лукас Тонн, Кэмпбелл Бартон

Вероятность завершения в 2015 году: неизвестно

Если появятся узлы (ноды) для системы частиц – почему не быть узлам для модификаторов, ограничений, трансформации, генеративного моделирования и т.д.? В этом случае необходимо модернизировать и унифицировать всю систему узлов Blender – но для этого потребуется тщательно разработать дизайн.

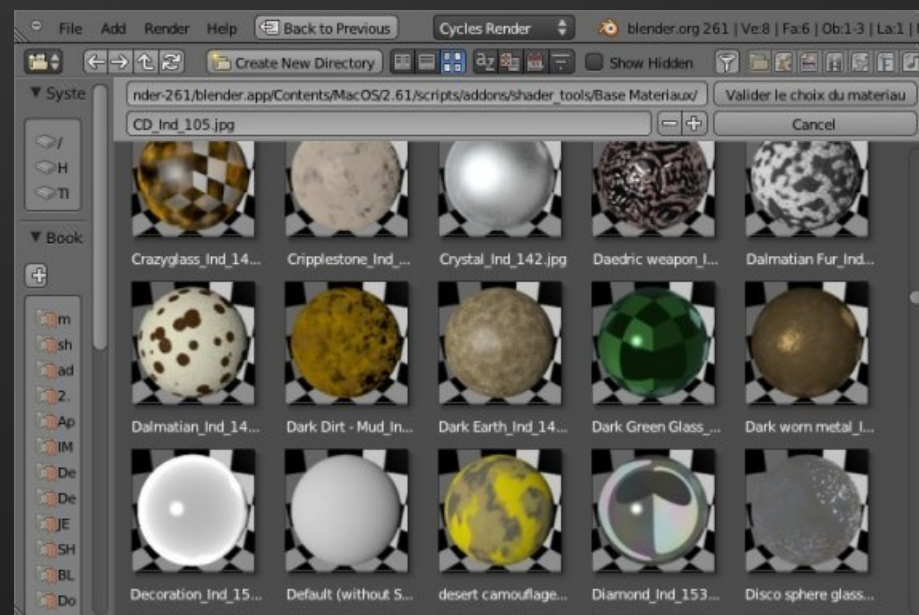
Работа в данном направлении уже ведется – как знать, может быть, замечательная идея «все на узлах» уже совсем скоро станет реальностью?

Менеджер ресурсов

Разработчики: Бастьен Монтань

Вероятность завершения в 2015 году: 98%

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: редактор должен «из коробки» поддерживать управление библиотеками внешних ресурсов. Кстати, вполне реальна интеграция в эту систему клиента облачного сервиса Blender Cloud.



PTex

Разработчики: Николас Бишоп

Вероятность завершения в 2015 году: 90%

PTex – это интересный проект от Disney, библиотека, реализующая динамическое текстурирование мешей без необходимости создавать UV-развертки. Интеграция подобной системы в Blender (с обязательной поддержкой Cycles) сделает пакет на порядок мощнее большинства коммерческих аналогов.

Ускорение Cycles

Разработчики: Сергей Шарыбин, Томас Дингс

Вероятность завершения в 2015 году: 100%

Хоть Брехт ван Ломмель и ушел из Blender Institute, работа над оптимизацией Cycles не прервалась. В этом году ожидается оптимизация BVH, более эффективный сэмплинг, а также поддержка шумоподавления.



Улучшение BGE

Разработчики: неизвестны

Вероятность завершения в 2015 году: 50%

Игровой движок Blender должен получить выигрыш от введения новой системы вьюпорта – появится возможность реализовать сложные модели освещения и затенения а ля Unreal Engine. Также будут обновлены редактор логики и игровая анимация.



Отслеживание движений

Разработчики: Сергей Шарыбин и Кьер Мерль

Вероятность завершения в 2015 году: 90%

Важное обновление ждет и систему трекинга. Как признавался в интервью Сергей Шарыбин (FPS №30 '14), ей недостает только «большой зеленой кнопки «Autotrack» – и, образно говоря, именно над такой «кнопкой» сейчас и работают Сергей и его коллеги: речь идет о полностью автоматическом распознавании движения камеры.



Обзор дополнений Blender

Выпуск 14

Благодаря удобному и мощному API для языка Python, Blender поддается практически неограниченному расширению. Наш журнал отслеживает выход новых полезных дополнений для Blender, которые могут заинтересовать пользователей, использующих программу в качестве инструмента для разработки игр.

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, либо в наше сообщество:

<https://plus.google.com/communities/103327597951489946649>

Blender Effectors

Аналог эффекторов из Cinema 4D – система анимационного дизайна, которая позволяет управлять анимацией полигонов или групп объектов.

Автор: Патрик Кроуфорд
[Ссылка](#)



Commotion

Еще одна система анимационного дизайна для Blender, облегчающая управление F-кривыми, NLA, драйверами и взаимосвязями между объектами.

Автор: Михаил Рачинский
[Ссылка](#)

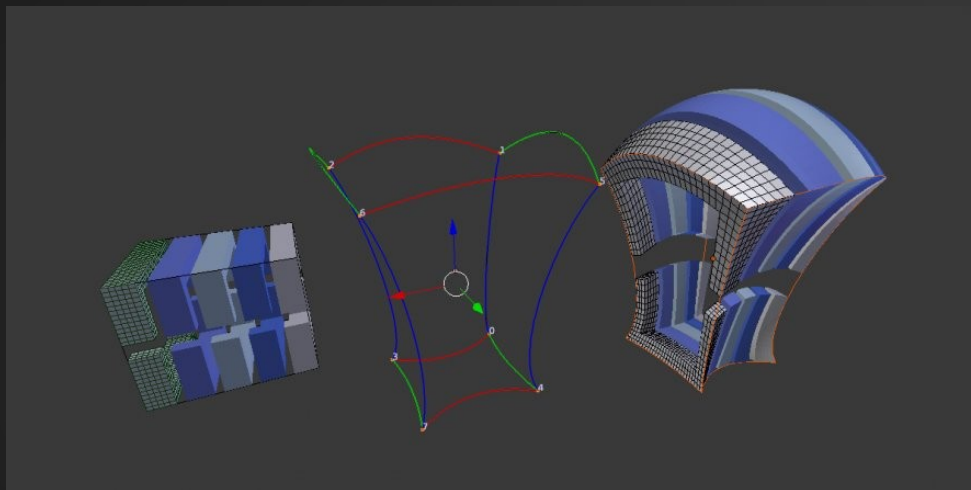


Curve Volume

Данное дополнение является аналогом модификатора Mesh Deform, только вместо полигональной сетки оно использует сплайны Безье. Такая функциональность особенно востребована в промышленном дизайне, где используются плавные искривления объектов. Аддон платный, его можно купить на Blender Market за \$35.

Автор: M. Vincek

[Ссылка](#)

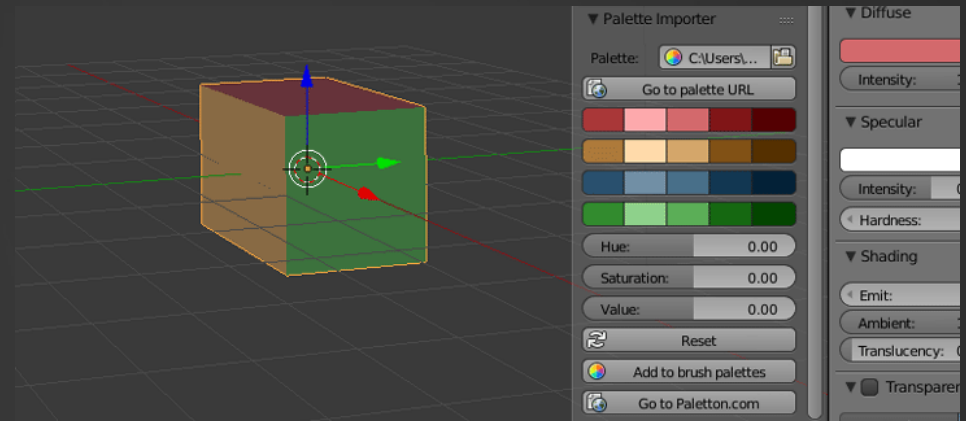


Snapshot Meshes

Невероятно полезный аддон, который реализует историю редактирования для мешей: перед тем, как сделать серию изменений, вы можете создать «снимок» меша и, если понадобится, в один клик вернуться к нему. Механизм аналогичен стеку модификаторов в 3ds Max. Дополнение бесплатное.

Автор: Антонио Мендоза

[Ссылка](#)

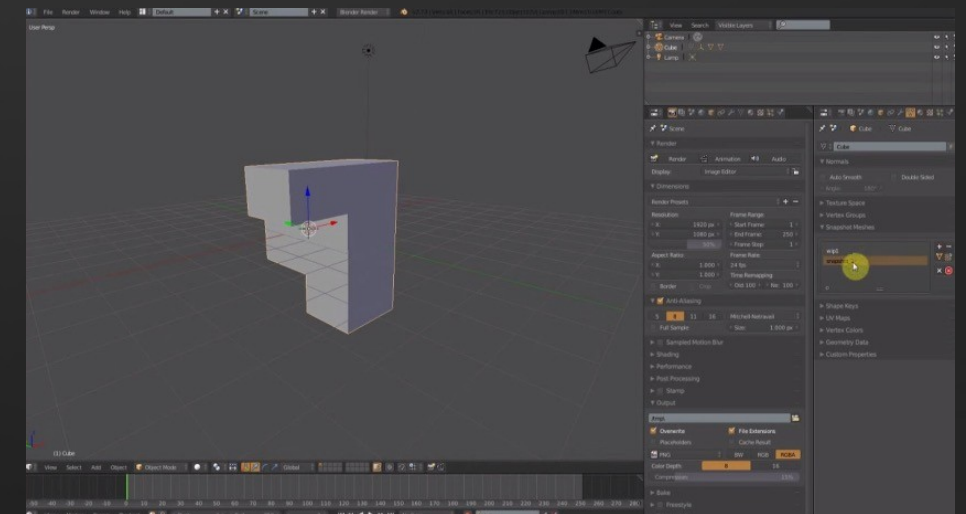


Palette Importer

Импортер для образцов цвета с сайта <http://paletton.com>.
Цена: \$4.00

Автор: NixArt

[Ссылка](#)





2D-графика: НОВОСТИ

Inkscape 0.91

Последние месяцы ознаменовались обновлением сразу трех крупных свободных графических редакторов. Так, в январе состоялся релиз Inkscape 0.91: в этой версии значительно оптимизирована производительность отрисовки (состоялся переход на Cairo), улучшена многопоточность фильтров (на основе OpenMP), кодовая база перенесена на C++, обновлен инструмент рисования текста, улучшена поддержка формата документов Corel DRAW. Появилась сборка под 64-битную Windows.

Напомним, Inkscape – это лучший свободный векторный редактор, аналог Corel DRAW. Программа включает полную поддержку стандарта SVG, а также включает собственный расширенный формат векторной графики на основе SVG.

<https://inkscape.org>

Krita 2.9.0

Увидела свет Krita 2.9.0. В данной версии добавлена начальная поддержка Mac OS X, реализована возможность работы сразу с несколькими изображениями, улучшены инструменты трансформации и деформации, а также средства рисования HDR-изображений.

Кроме того, обновлены инструменты рисования кистями, улучшена совместимость с Photoshop, добавлена поддержка RAW-файлов.

Напомним, Krita – это свободный растровый графический редактор, развиваемый в рамках проекта KDE и входящий в состав офисного пакета Calligra. Проект курируют фонд Krita Foundation и организация KO GmbH (Krita Studio).

<https://krita.org>



Pinta 1.6

Следом за Krita, обновился и графический редактор Pinta – до версии 1.6. Проект ставит целью создать свободный аналог Paint.NET с использованием GTK+. Редактор предоставляет базовый набор возможностей для рисования и обработки изображений, ориентируясь на начинающих пользователей – Pinta поддерживает слои, включает набор инструментов для наложения эффектов и корректировки изображений. Исходники Pinta распространяются под лицензией MIT.

В новой версии программы переработаны инструменты для рисования фигур, во встроенном менеджере дополнений добавлена поддержка репозитория дополнений.

<http://pinta-project.com>



GIMP Magazine #9, #10, #11

Увидели свет подряд три новых номера GIMP Magazine – замечательного англоязычного журнала о GIMP и его пользователях. Журнал, как всегда, радует интересным материалом, красочными иллюстрациями и качественным дизайном.

В свежих выпусках (№№ 9, 10, 11) вы найдете статью по основам современного дизайна, по обработке каллиграфии в GIMP и Inkscape, урок по обработке RAW-снимков при помощи знаменитого плагина-проявщика UFRAPW и многое другое. GIMP Magazine, кстати, теперь выходит ежемесячно.



Журнал можно бесплатно скачать в формате PDF, прочитать онлайн на сервисе Issuu, купить в бумажном виде (\$26) или в специальной версии для iPad (\$5).

PDF-версия журнала рас-пространяется по лицензии CC-BY-SA.

<http://gimpmagazine.org>

OpenAL: бинауральный 3D-звук

К сожалению, в последние годы интерес к звуку в играх заметно низок, особенно на фоне ажиотажа вокруг графики. Звуковые карты с аппаратным ускорением различных интересных эффектов, появившиеся на рубеже тысячелетий, так и не получили широкого распространения, чего нельзя сказать о видеокартах и видеопроцессорах. Между тем, современные вычислительные возможности компьютеров позволяют даже на дешевых акустических системах получить 3D-звук такого качества, что многие люди, впервые услышав его, неизменно задаются вопросом: почему этого до сих пор нет в играх?



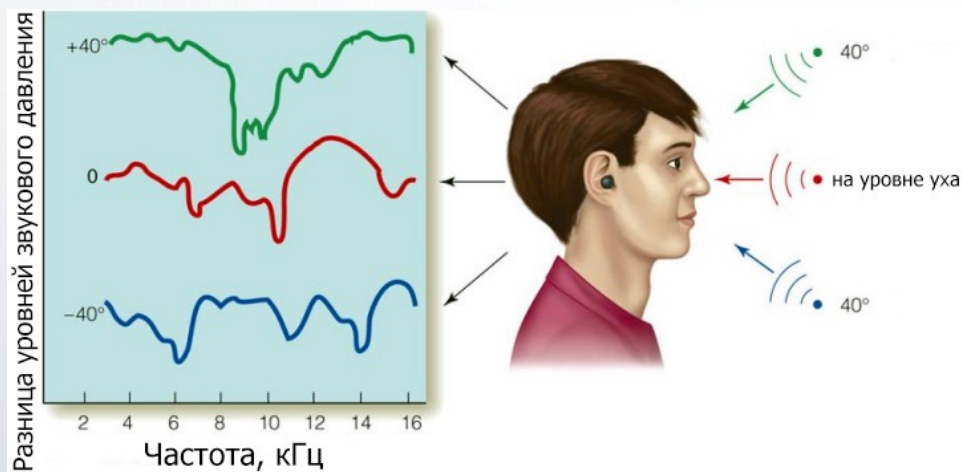
На сегодняшний день существуют два основных метода имитации трехмерного звука. В кинематографе широко используется многоканальный звук (surround) – воспроизведение фонограмм через систему из нескольких динамиков, окружающих слушателя.

Вторая технология менее распространена, но имеет большие перспективы применения в мультимедийных приложениях: это так называемый бинауральный звук, для воспроизведения которого не нужны дорогостоящие акустические системы – достаточно обыкновенных стереонаушников. Бинауральный звук записывается двумя микрофонами через уши специального манекена, повторяющего анатомическое строение головы – вплоть до формы ушных раковин. В итоге получается настоящее чудо – реальный 3D-звук, прослушивая который через наушники, мы можем определить положение источника в пространстве относительно нашей головы!

Чтобы лучше понять, о чем идет речь, послушайте пример бинауральной записи. Одна из самых популярных, которую вы легко найдете в Интернете – это «Virtual Barbershop» («Виртуальная парикмахерская») от компании Starkey. Слушать необходимо через накладные наушники – не «капельки»! Для максимального эффекта рекомендую слушать с закрытыми глазами и не поворачивать при этом голову. По звуку голоса «парикмахера» и его инструментов вы сможете ощутить, как он передвигается вокруг вас и на каком расстоянии находится.



Каким же образом достигается этот удивительный эффект? В первую очередь, он основан на том, что звук поступает в наши уши не одновременно, а с небольшой разницей во времени – ухо, расположенное ближе к источнику, воспринимает звук раньше. Эта разница ничтожно мала, но она все-таки есть, и наш мозг учитывает ее, чтобы «вычислить» расположение источника (так называемая фазовая локализация).



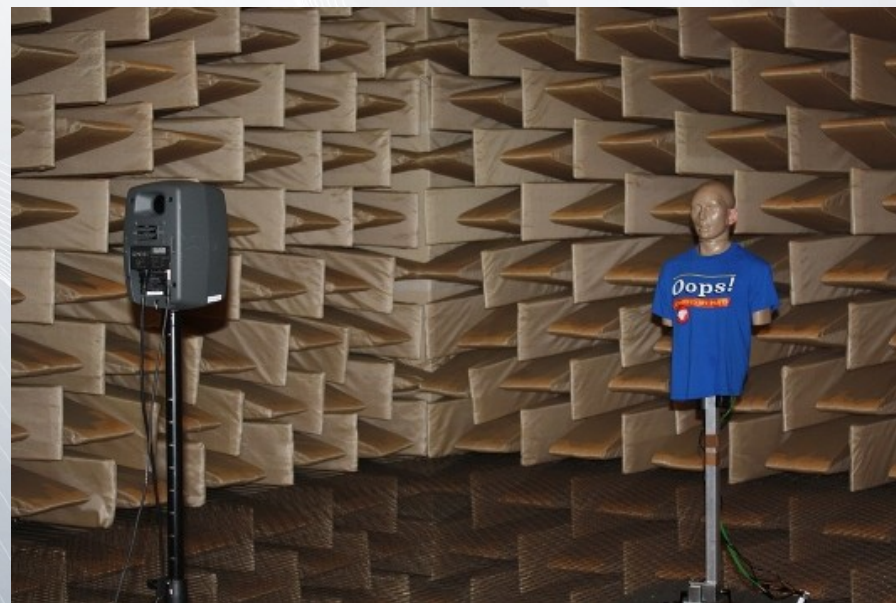
Но это не объясняет, как мы распознаем источники, расположенные на одном и том же расстоянии от обеих ушей – середи, сзади, снизу или сверху. Здесь вступает в действие способность мозга классифицировать звуки по их частотным характеристикам. Все дело в том, что звук, покинув источник, проходит весьма сложную серию преобразований перед тем, как попасть в наши внутренние органы слуха – из-за особенностей формы головы и ушной раковины, мы по-разному воспринимаем один и тот же звук, поступающий сзади и спереди, сверху и снизу (локализация по разнице частотного спектра).

Эти частотные преобразования (передаточные функции головы – head-related transfer function, HRTF) используются компьютерными алгоритмами для синтеза бинауральных звуков – зная HRTF для обеих ушей, достаточно произвести ими свертку монофонического сигнала, и получится реалистичный 3D-звук!

Делается это следующим образом. Из физиологии мы знаем, что звук, улавливаемый нашим ухом, создает некоторое давление на барабанную перепонку. Обозначим это давление как функцию от частоты: $P_1(f)$. Если поместить между источником звука и барабанной перепонкой какое-либо препятствие (в нашем случае это голова с ушами), то звуковое поле искажается за счет дифракции и отражения – тот же источник звука будет создавать несколько другое давление: обозначим его как $P_2(f)$. Тогда передаточную функцию головы можно записать как отношение этих давлений:

$$H(f) = P_2(f) / P_1(f)$$

Чтобы найти $H(f)$, на практике замеряется давление по времени $P_1(t)$ и $P_2(t)$. Затем вычисляется $H(t)$ – коэффициент давления на барабанную перепонку, инвариантный относительно частоты сигнала. Его также называют head-related impulse response (HRIR) – импульсной переходной функцией головы. HRIR переводится в HRTF (в частотную область) путем преобразования Фурье.



Учеными накоплены крупные базы данных по HRTF, которые можно свободно использовать для синтеза бинаурального звука. Наиболее часто применяются замеры с манекена KEMAR, сделанные в 1994 году в Массачусетском технологическом институте (США), а также данные Центра интегрированных акустических информационных исследований (CIAIR) Нагойского университета (Япония).



Из программных средств со встроенной поддержкой HRTF-преобразований можно отметить OpenAL – одну из наиболее популярных на сегодняшний день кроссплатформенных библиотек для вывода звука. OpenAL широко применяется в сфере разработки игр – так, библиотека используется во многих играх класса AAA от id Software и Epic Games, в S.T.A.L.K.E.R., Battlefield, Penumbra и множестве других.

OpenAL был изначально спроектирован фирмой Loki Software (которая также известна как разработчик SDL) для их основного бизнеса – портирования коммерческих игр с Windows на Linux. После того, как Loki обанкротилась, проект не умер и продолжил свое развитие на базе сообщества СПО.

Проект OpenAL Soft – это форк оригинальной OpenAL, который добавляет поддержку множества Unix-систем и различных звуковых интерфейсов (OSS, ALSA, PulseAudio и др.). В отличие от проприетарной версии, эта реализация OpenAL продолжает развиваться: последнее обновление на момент написания статьи (1.16.0) датируется августом 2014 года.

Особенность бинаурального звука OpenAL в том, что он включается системной настройкой – то есть, бинауральное 3D можно включить для любого приложения, использующего достаточно свежую версию OpenAL. Само приложение при этом изменять и пересобирать не требуется.

Для этого используется конфиг `.alsoftrc` в домашнем каталоге пользователя, в котором необходимо прописать следующие ключи:

```
stereo-mode = headphones  
hrtf = true  
hrtf_tables = /path/to/hrtf_table.mhr
```

Последний ключ опционален – это список путей к файлам `*.mhr`, содержащим базы данных по HRTF. Их можно сгенерировать из замеров KEMAR, CIAIR и др. при помощи утилиты `makehrtf`, входящей в состав OpenAL SDK.

При использовании HRTF, наиболее соответствующей вашим биометрическим параметрам, 3D-звук будет намного качественнее. Однако напоминаю, что слушать его надо исключительно через наушники.

Тимур Гафаров

Проблема инструментария

Чего не хватает инди-разработчикам?

Последние 10 лет стали настоящим бумом свободного программного обеспечения. Теперь его используют не только отдельные энтузиасты и хакеры – на СПО переходят крупные корпорации, муниципалитеты и государственные органы, научные лаборатории и вузы.

Весь Web 2.0 полностью построен на базе свободных серверов, фреймворков и CMS – Интернет стремительно уходит от «мутного времени» проприетарных стандартов. В мобильном секторе тоже лидирует свободный Android. Открытая разработка ПО на GitHub и других аналогичных хостингах уже давно переросла среди программистов стадию модного веяния и превратилась в общепринятую норму. Свобода изучать, модифицировать и распространять исходники программ стала одним из основополагающих принципов современной computer science.

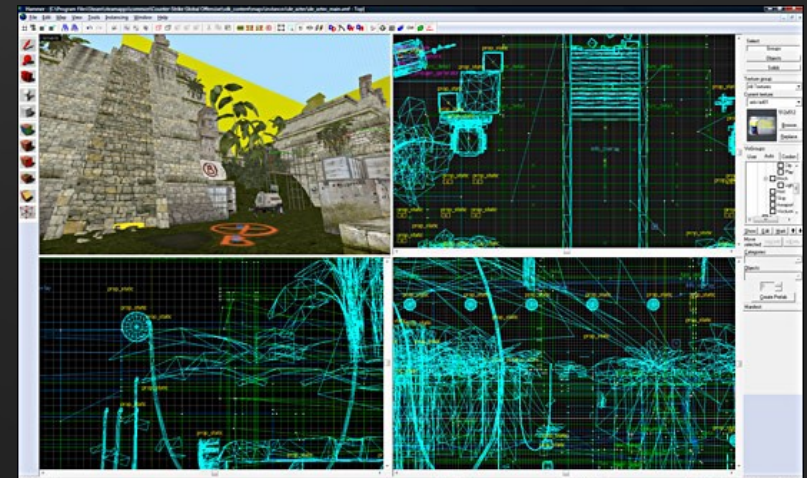
Однако есть еще сферы IT, где СПО не одержало верх – и это не только офисная ниша с ее неизменной связкой Windows + MS Office. И даже не полиграфия, где уже много лет царят Adobe и Corel. Нас, авторов «FPS», в первую очередь удручает отсутствие полного набора свободных инструментов и открытых стандартов в области разработки игр. На протяжении всей истории геймдева самыми популярными оставались проприетарные решения, не оставляющие никаких конкурентных шансов свободным альтернативам.

Сейчас умение работать с проприетарными движками (Unity, Unreal Engine и т.д.) стало своего рода профессиональным навыком, который востребован работодателями – по сути, наблюдается та же ситуация, что и в дизайнерской среде, где вы с умением работать в Blender, GIMP или Inkscape мало кому нужны. Даже если способны выдавать результаты не хуже (а то и лучше), нежели ваши коллеги, «воспитанные» на Photoshop и 3ds Max.

Но если оставить в стороне вопрос трудоустройства и остановиться на нашем любимом инди-секторе, то и здесь до сих пор маловато доступных инструментов – всякому независимому разработчику, не желающему соприкасаться с проприетарным миром, приходится каждый раз «изобретать велосипед» и тратить уйму времени на подготовку игрового тулсета. Рассмотрим самые, на наш взгляд, важные пункты:

- **Трехмерный редактор уровней**

В коммерческих движках это, как правило, основной компонент тулсета. Свободные движки редактор уровней обычно не включают, либо он сам по себе является коммерческим продуктом (как, например, проприетарный IrrEdit в свободном движке Irrlicht).



Конечно, есть сторонние свободные редакторы уровней, заточенные под какой-либо популярный движок или игру (например, GtkRadiant для серии Quake, Doom 3 и других игр на основе id Tech), но они мало годятся для использования с другим движками – для использования карт формата Quake вам придется реализовать в своем движке львиную долю технологий id Software, которые специфичны для indoor-шутеров от первого лица и подходят далеко не для всех других жанров.

Существует несколько проектов по созданию универсального редактора уровней, но все они либо находятся в зачаточном состоянии, либо не кроссплатформенны (например, известный DeleD и недавно открытый ATF LevelEditor от Sony работают только под Windows). Большинство из них стары в техническом отношении, нерасширяемы, плохо масштабируемы и не соответствуют современным реалиям геймдева и возможностям аппаратных средств.

• Формат моделей

До сих пор не существует единого стандартного формата для игрового контента, который покрывал бы нужды большинства, был бы расширяем и достаточно прост для парсинга (XML, как говорится, не предлагать!)

Было время, когда в качестве такого «стандарта» программисты на DirectX использовали *.X, а все остальные – форматы, специфичные для конкретного движка или 3D-редактора (например, MD2, SMD, 3DS, OBJ – соответственно, для Quake 2, Half-Life, 3ds Max и Wavefront TAV). Популярность этих форматов пришла на эпоху повального увлечения моддингом, когда движки Quake и Half-Life считались эталоном и приобрели статус культовых. Однако сейчас эти форматы морально устарели, и им необходима более современная замена.

Работы в этом направлении уже ведутся: так, авторы свободных игр, продолжающих идеи Quake, предложили новый формат IQM (Inter-Quake Model) – универсальный формат для хранения моделей со скелетной анимацией.



IQM компактен, его легко читать и генерировать, у него есть как двоичное, так и текстовое представление. Существует полнофункциональный экспортер IQM для Blender, что делает этот формат лучшим на сегодняшний день выбором для инди-разработчиков – настоящий must-have!

Однако все еще остается открытым вопрос хранения карт, статических моделей, данных игровой физики и логики. Возможно, эту нишу со временем займут форматы на основе OpenDDL (Data Description Language) – текстовой разметки, ориентированной на описание составных структур данных. Другой возможный кандидат – JSON.

● Игровой отладчик

Программистам приходится ограничиваться разношерстным набором из обыкновенного отладчика (GDB, Valgrind), профайлера (если он вообще существует для используемого языка) и разной степени удобства утилитами для анализа видеопамати и отслеживания вызовов графического API (как, например, проприетарный gDEBugger от AMD для OpenGL). Отсутствие хорошего свободного игрового профайлера особенно удручает.

● Кроссплатформенный API для устройств ввода

Открытые графические и звуковые интерфейсы уже «установились» – настало время для какого-нибудь OpenInput. Да, библиотеки наподобие SDL и SFML абстрагируют доступ к устройствам ввода, но им все-таки присущи серьезные ограничения. Например, SDL до последнего времени не поддерживала Force Feedback для джойстиков (в версии 2.0 поддержка появилась, но в системах, отличных от Windows, она все равно требует особых системных конфигураций). О платформонезависимом доступе к сенсорным дисплеям, графическим планшетам и устройствам типа Kinect и говорить не приходится...

● Искусственный интеллект

При огромном выборе графических, физических, звуковых и сетевых движков, универсальной свободной библиотеки для игрового ИИ до сих пор нет. Существует ряд узкопрофильных решений (например, OpenCV – реализация распознавания образов в реальном времени), но, чтобы элементарно заставить врагов и NPC вести себя более-менее реалистично, придется в очередной раз изобрести велосипед.

Тимур Гафаров

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: gesko0307@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.



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/>

Трудный выбор

Почему хакеры выбирают пермиссивные лицензии?

Несмотря на то, в широких массах термин «свободное программное обеспечение» ассоциируется с лицензией GNU GPL, значительный объем современного СПО к этой лицензии никакого отношения не имеет. Например, я публикую весь свой код под лицензией Boost и рекомендую использовать пермиссивные лицензии, а не GPL. Так же поступают многие другие разработчики – наиболее популярными лицензиями на GitHub, например, являются MIT, BSD и Apache License. Почему так? Попробую объяснить.

Для начала – немного «матчасти» для непосвященных. Исторически все свободные лицензии принято делить на две большие категории: пермиссивные и копилефт-лицензии.

Копилефт (copyleft, игра слов от англ. Copyright) – это требование распространять все производные работы на тех же условиях, что и оригинал. Типичный пример копилефт-лицензии – это GPL: ее даже прозвали «вирусной лицензией» из-за того, что любой софт, включающий GPL-исходники, тоже обязан быть под GPL. Автором и главным апологетом этой концепции является Ричард Столлман, создатель GPL. В противоположность, пермиссивные (permissive – разрешительные) лицензии не имеют такого ограничения – исходники под такими лицензиями вполне легально можно использовать в проприетарных коммерческих продуктах.

При всем уважении к Столлману и его вкладу в СПО, я не могу назвать GPL подходящей лицензией для всех свободных проектов. Идеал – это Public Domain, общественное достояние. Чтобы осознать это, задайтесь вопросом – о чем вы думаете, делаясь своей работой со всем миром? Какова ваша мотивация? Наверное, самым привычным ответом будет: «Я хочу, чтобы мои знания и опыт принесли кому-то пользу».

Конечно, я не берусь утверждать, что этот мотив движет абсолютно всеми: для кого-то важнее репутация и уважение, кто-то заполняет свое портфолио, кто-то хочет продемонстрировать другим свой творческий подход к технологиям, для кого-то открытая модель разработки предпочтительна в экономическом плане, а некоторые программы просто технически не могут быть закрытыми (например, скрипты на Bash). Статья «Open-source movement» на Википедии также трактует мотивацию программистов СПО как неоднозначную. Но все же альтруизм, так или иначе, является основной идеей всего движения СПО.

Конечно, как программист, вы можете приносить пользу и работая над несвободными программами. Кто-то может посчитать, что полезность программ оценивается по их качеству – по тому, насколько хорошо они выполняют задачи, для которых были созданы. Отчасти это так, но свобода программы делает ее значительно более полезной, а потому – создание свободных программ больше соответствует альтруистическим принципам.

Это лучше понять от обратного: отнимая у пользователей свободу модифицировать и распространять программы, вы делаете их, в конечном итоге, менее полезными. Сравните две программы одинаково плохого качества – свободную и несвободную. Какая из них предпочтительнее? Ответ очевиден.

Необходимо помнить, что идеальных программ не бывает, все они содержат ошибки. Даже если программа выглядит безошибочной, рано или поздно в ней всплывут критические ошибки. И тогда ее полезность будет стремиться к нулю, если пользователь не будет иметь свободы исправить их.

Впрочем, апологеты GNU и GPL имеют несколько иной взгляд на вопрос свободы – было бы несправедливо не упомянуть о нем. С их точки зрения, свобода модифицировать и распространять программы должна быть одним из неотъемлемых прав человека, существовать естественным образом – как, например, свобода слова. Говоря о несвободе, апологеты GPL не оперируют термином «полезность», вместо этого они акцентируют внимание на том, что несвободные программы являются орудием несправедливой власти – власти имущественных владельцев программ над их пользователями. Такая трактовка, разумеется, вполне имеет право на существование, но у нее есть некоторые недостатки, о которых речь пойдет ниже.

Итак, выбранная лицензия должна обеспечивать максимальную полезность программы. Чем в этом отношении плох копилефт? Казалось бы, он защищает свободу пользователей: есть определенная гарантия (или, по крайней мере, ощущение такой гарантии), что все GPL-программы и их производные всегда будут свободными. Но этот аргумент хорош только в теории.

На практике гораздо чаще наблюдается другая ситуация: программы под GPL подвергаются двойному лицензированию – «по умолчанию» исходники доступны бесплатно под GPL, а альтернативная лицензия для их использования в несвободных продуктах стоит денег. Что же это означает? О чем заботятся в данном случае разработчики – о всеобщем благе или о собственной прибыли?

Посмотрим на это с такой стороны: выбирая GPL (читай – умышленно ограничивая своих пользователей копилефтом), разработчик тем самым подчеркивает, что оставляет за собой исключительное право распоряжаться кодом с имущественной точки зрения – ведь право перелицензирования в данном случае остается только за ним.

Он как бы говорит: «Я разрешаю вам модифицировать и делиться моим кодом, но он все-таки мой». Таким образом, он хочет защитить себя от «воровства» – чтобы посторонние лица не смогли беспрепятственно наживаться на его труде.

Стоп! Не это ли то, с чем, по сути, надо бороться: заблуждение, что информация – это нечто такое, что можно украсть? Не напоминает ли это методы проприетарщиков? Неужели их можно победить, действуя их же оружием?

Я уверен, что нет. Нельзя победить большое зло, используя меньшее зло в качестве оружия. Не может быть условной свободы, тем более в вопросах взаимодействия человека и информации – такая свобода всегда должна быть безусловной. Если, открывая исходники, вы боитесь, что кто-то может использовать их в корыстных целях, нарушить ваши имущественные права, и вы не получите компенсации – значит, вы думаете о полезности программы для себя, а не для всех.

Да, мы живем в неидеальном мире, где отнюдь не каждый действует по принципам альтруизма. Но это не значит, что ваш альтруизм должен быть направлен только на других альтруистов: выпуская программы под копилефт-лицензией, вы подразумеваете, что ваша программа будет полезна только другим апологетам копилефта. Но как же все остальные?

Это похоже на то, как если бы религиозные благотворительные организации помогали только своим единоверцам. Но цена такой «помощи» – грош!

Даже если вы выбираете GPL не из корыстных целей, а просто думая о копилефте так, как это написано на сайте fsf.org – то есть, как о необходимом средстве борьбы за свободу – то, вы, по сути, действуете именно как борец: делите мир на «своих» и «чужих». Апологеты этой борьбы упускают из виду возможность существования безусловного альтруизма – а значит, поступают менее альтруистично.

Я, конечно, не говорю о том, что борьба за свободу не важна – равно как и не оспариваю столлмановское определение свободы. Наоборот, я считаю себя убежденным приверженцем ключевых идей FSF. Мои взгляды отличаются лишь в том, что касается субъектов свободы. Почему, думая о лицензиатах, я должен делить их на «хороших» и «плохих»?

Наука и технологии находятся вне добра и зла – они должны быть одинаково доступны для всех. И, если я хочу бороться с тем, что считаю злом, я буду делать это вне технологий: моя личная позиция не должна никак затрагивать пользователей моих программ. Они не должны иметь меньше свобод только потому, что я решил навязать всем свою идеологию.

Именно поэтому истинно свободная лицензия – это, по сути, ее отсутствие: полный отказ от имущественных прав на код. Иными словами – Public Domain, общественное достояние.

И все же – почему тогда я использую лицензию Boost, а не Public Domain? Ответ прост: из соображений совместимости. Я пишу на D, а практически весь софт на D распространяется под лицензией Boost. Кроме того, не во всех странах, к сожалению, есть правовая возможность авторов делать свои работы общественным достоянием. По этой причине, кстати, была предложена лицензия Creative Commons Zero (CC0) – юридически корректный аналог Public Domain.

Напоследок хочется упомянуть один нелицеприятный факт: огромное количество проектов на GitHub размещены вовсе без указания лицензии. Как ни парадоксально, из-за этого они теряют статус свободных – согласно авторскому праву большинства стран, такой код «по умолчанию» нельзя копировать и модифицировать! Поэтому в последнее время наметилась своеобразная агитация под лозунгом «Выбери хоть какую-то лицензию»: на выбор обычно предлагаются самые популярные – GPL, MIT, BSD и Apache.

Я не буду навязывать никому свои убеждения о свободе и свой выбор лицензии. Я только хочу предостеречь начинающих от того, чтобы выбирать лицензию «наугад» – ведь от этого выбора зависит будущее проекта! Лицензию необходимо выбирать аккуратно и взвешенно, тщательно вчитываясь в текст и сопоставляя с вашими собственными целями и принципами.

Тимур Гафаров

Памятка читателю

В Интернете часто можно встретить вопросы о том, где скачать старые номера нашего журнала. Отвечаем. Архив всех номеров «FPS» (с 2008 по 2014 гг.) можно найти сразу на нескольких сервисах:

На файловом хостинге **DropBox**:

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

В **Документах Google** (для скачивания файлов нужен аккаунт Google):

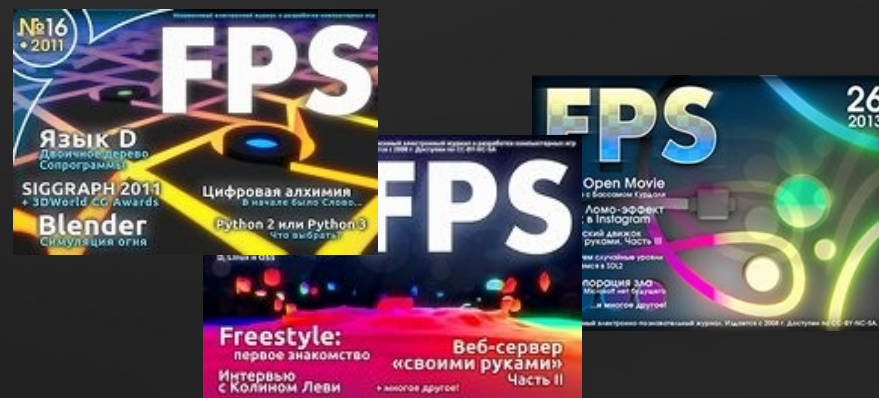
<https://docs.google.com/folderview?id=0B1BlzRb1uMv-bnpHNDhwZTI4eHc>

В электронном издательстве **Issuu.com**:

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

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

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



Как стать хакером?

Советы от Эрика Реймонда

В представлении обывателя, хакер – это компьютерный злоумышленник, взламывающий программы, сайты и банковские счета с целью наживы. На самом деле, конечно, это огромное заблуждение – хоть эти взломщики и называют себя «хакерами», их следует называть «крэкерами». То, что человек способен взломать систему безопасности, не делает его хакером – точно так же, как умение угонять машины не делает вас мастером-автомехаником. Главное различие в следующем: хакеры строят, а крэкеры ломают.

Но кто, в таком случае, настоящие хакеры – и чем они отличаются от обычных программистов, админов и инженеров? На этот вопрос поможет ответить статья известного американского хакера Эрика Реймонда, основателя движения Open Source – мы приводим ее в переведенном и адаптированном для современного читателя виде.

История хакеров уходит корнями во времена мини-компьютеров PDP и ранних цифровых сетей – это культура, объединившая программистов, которые писали первые операционные системы и создавали высокоуровневые языки программирования. Хакеры решают фундаментальные проблемы и придумывают новые идеи – это олицетворяет их от рядовых программистов, которые занимаются реализацией уже существующих идей.

Хакеры создали Интернет и операционную систему Unix, хакеры пишут компиляторы ЯП, создают сетевые протоколы, алгоритмы шифрования и многое другое. Если вы являетесь частью этой культуры, если вы сделали в нее свой вклад, и другие члены этой культуры знают, кто вы, и называют вас хакером, то вы – хакер.

Одно из определений звучит следующим образом: хакер – это человек, наслаждающийся doskonaльным пониманием работы компьютерных систем. Суть хакерства – в том, чтобы получать интеллектуальное удовольствие от решения проблем в области компьютеров, либо от творческого преодоления каких-то ограничений.

Удовольствие такого рода требует массы усилий, и для этого нужна мотивация. Спортсмены черпают мотивацию из своего рода физического удовольствия, получаемого от доведения собственного тела до совершенства, от преодоления пределов своих возможностей. Подобно этому, хакер получает максимум удовольствия от решения проблем, от оттачивания своих навыков, от тренировки своего интеллекта. Деньги, успех, карьера – все это для хакера вторично по отношению к интеллектуальному удовольствию.

Впрочем, хакерский взгляд на мир не ограничивается лишь компьютерами. Есть люди, применяющие хакерский подход и к другим вещам – вы можете встретить их на высших уровнях любой науки или искусства. Компьютерные хакеры признают таких близких по духу людей и тоже могут называть их «хакерами» – многие считают, что хакерская природа на самом деле не зависит от области, в которой человек работает.

Другая важная черта хакера – высокоразвитая способность к обучению. Даже если вы и не знаете всего, что необходимо для решения проблемы, то вам достаточно будет изучить лишь небольшую часть, которой достаточно, чтобы сделать очередной шаг в решении – и так далее, пока все не будет сделано. Вы должны сами поверить, что можете изучить все, что угодно – это и есть один из источников мотивации хакера.

Хакеры, как творческие люди, никогда не должны заниматься скучными вещами и погрязать в рутине монотонной работы. Если это происходит, то это означает, что они не раскрывают свой потенциал, не делают того, что могут делать лишь они – решать новые проблемы. Поэтому настоящий хакер всегда автоматизирует все скучные мелочи настолько, насколько это возможно. Если вы делаете много рутинной работы, подумайте, как можно ее автоматизировать – это один из способов стать хакером.

Хакеры по своей природе анти-авторитарны. Любой, кто может отдавать вам приказы, может помешать вам решать очаровавшие вас проблемы. Так что с авторитарным подходом следует бороться всюду, где вы его встретите, чтобы он не душил вас и других хакеров.

Конечно, это не то же самое, что сражаться с властью – хакер может соглашаться на принятие какого-то рода руководства, когда есть возможность получить то, что нужно и не особо много времени тратиться на выполнение приказов. Но это просто вид разумной сделки, что-то вроде личной уступки.

Кстати, авторитаризм процветает в условиях цензуры. Авторитарные люди не верят в добровольное сотрудничество и деление информацией, им нравится только такое «сотрудничество», которым руководят они. Поэтому, чтобы быть настоящим хакером, вы должны выработать в себе инстинктивное неприятие любой цензуры и секретности.

Хакерский взгляд на жизнь – это важно, но мастерство многократно важнее. Философская позиция не заменит компетентности, и существует определенный набор элементарных навыков, которыми вам необходимо обладать, прежде чем вас теоретически можно будет назвать хакером.

Этот базовый набор со временем потихоньку изменяется – по мере того, как технологии порождают новые навыки и делают ненужными устаревшие. Например, раньше в его числе упоминалось программирование на машинном языке, но в него не входило знание HTML. В сегодняшний набор обязательных навыков вполне определенно входят следующие вещи:

- **Программирование.** Это, конечно же, фундаментальный хакерский навык. Если вы не знаете ни одного языка программирования, рекомендуем начать с Python. Он хорошо документирован и относительно дружелюбен к начинающим. Несмотря на то, что он хорош как первый язык, это не просто игрушка – на самом деле, это очень мощный и гибкий инструмент, хорошо подходящий для крупных проектов.

Но знайте, что вы не достигнете хакерского уровня мастерства (и даже просто уровня хорошего программиста), если будете знать только один язык. Знание какого-то конкретного языка не так важно, как понимание принципов программирования вообще. Чтобы быть настоящим хакером, вам надо достичь уровня, на котором вы сможете выучить любой новый язык за несколько дней (!), соотнося его документацию с тем, что вам уже известно.

Это означает, что вам следует выучить несколько очень разных языков. В хакерских кругах особенно уважаемы C, Lisp, Haskell и Perl. Некоторые также скажут вам, что очень полезно знание языка ассемблера, Forth или других машинно-ориентированных языков. Но, повторимся, знание каких-то конкретных языков – это не самоцель: в руках хакера абсолютно любой язык может стать мощным орудием.

- **Работа с Unix.** Один из важнейших шагов, которые необходимо предпринять для приобретения хакерских навыков – это обзавестись каким-нибудь вариантом Linux или BSD.

Значение unix-систем в хакерском мире трудно переоценить: учиться хакерству на Windows – это все равно что учиться танцевать в гипсе. Unix – это, фактически, операционная система Интернета. Вы, конечно, можете использовать Интернет и не зная Unix, но вы не можете стать интернет-хакером, не понимая Unix. Симбиоз между Unix и Интернетом настолько прочен, что даже сил Microsoft не хватает, чтобы серьезно на это влиять.

Современные версии Linux дружелюбны к начинающим пользователям, существует множество доступной литературы, которая поможет вам овладеть этой ОС.

Однако важно не столько научиться работать с графическим интерфейсом Linux (он не сложнее, чем в Windows), сколько понять общие принципы, характерные для всех unix-систем: например, как составлять конвейеры ввода/вывода, поднимать серверы и назначать права доступа.

- **Работа с WWW и HTML.** Всемирная паутина – это гигантская «игрушка» хакеров, которая изменила наш мир. Лишь по одной этой причине (а также и множеству других) вам следует научиться работать с Вебом. Выучите HTML, создайте свой сайт и наполните его интересным и полезным для других хакеров содержанием. Хорошая практика – создавать сайты для персональных программных проектов, либо для документации и FAQ по чужим проектам. Помните, что хакеры не приветствуют засорение Веба бессмысленной информацией.

Может ли человек сам себя провозгласить хакером? Скорее всего, нет. Хакерская культура строится на репутации. Вы можете решать любые интересные для вас проблемы, но насколько эти проблемы интересны для всего остального мира, и хороши ли ваши решения – могут судить только другие хакеры, равные вам или превосходящие вас. Именно поэтому вы не можете называть себя хакером, пока вас не станут так называть другие.

Вы зарабатываете статус и репутацию не тем, что руководите другими людьми, не тем, что имеете вещи, которые являются предметом вожделения других. Но скорее тем, что раздаете вещи – в частности, вкладывая в хакерскую культуру свое время, свое мастерство и результаты своего творчества. Есть пять основных типов вещей, которые вы можете делать, чтобы вас уважали хакеры:

- **Пишите программы с открытым исходным кодом.** Мы привыкли называть их «свободным программным обеспечением» (free software), но это часто приводит в замешательство непосвященных людей, точно не понимающих, что в данном контексте означает слово «free» (в английском языке оно означает как «свободный», так и «бесплатный»). Поэтому для таких программ многие хакеры предпочитают использовать термин «открытое программное обеспечение» (open source software).

- **Помогайте тестировать и отлаживать программы.** В нашем несовершенном мире мы неизбежно затрачиваем большую часть времени на исправление ошибок. Именно поэтому любой хакер скажет вам, что хорошие тестеры ценятся на вес золота. Если вы начинающий, то найдите программу, которая вам интересна, и станьте ее бета-тестером.

- **Публикуйте полезную информацию.** Отбирайте и накапливайте полезную и интересную информацию на веб-страницах, делая ее общедоступной. Хакеры, пишущие техническую документацию, разного рода FAQ и руководства, практически столь же уважаемы, как и авторы программ с открытым исходным кодом.

- **Помогайте поддерживать работу инфраструктуры.** Имеется масса необходимой работы, которую нужно делать, чтобы поддерживать хакерский мир: администрирование серверов, моделирование новостных групп и форумов, управление большими архивами ПО, разработка технических стандартов. Люди, хорошо делающие подобные вещи, глубоко уважаемы, поскольку каждый знает, что такая работа требует кучу времени и свидетельствует о самоотверженности.

- **Служите хакерской культуре** – распространяйте идеи хакеров. Но этим не стоит заниматься до тех пор, пока вы не «поваритесь» в этом мире достаточное время и не станете хорошо известны благодаря одной из четырех первых вещей.

Вопреки расхожему мифу, чтобы стать хакером, вам не обязательно быть «гиком» или «нердом». Многие хакеры, впрочем, действительно «гики» – статус социального изгоя помогает оставаться сосредоточенным на интеллектуальной работе.

Но если вы способны в достаточной степени сосредоточиться на хакерстве и достигать заметных результатов, но при этом иметь личную жизнь – то это прекрасно.

Если же вас влечет к хакерству по той причине, что у вас нет личной жизни – то и это неплохо. По крайней мере, у вас не будет проблем с концентрацией. А личная жизнь – как знать, может она придет позже!

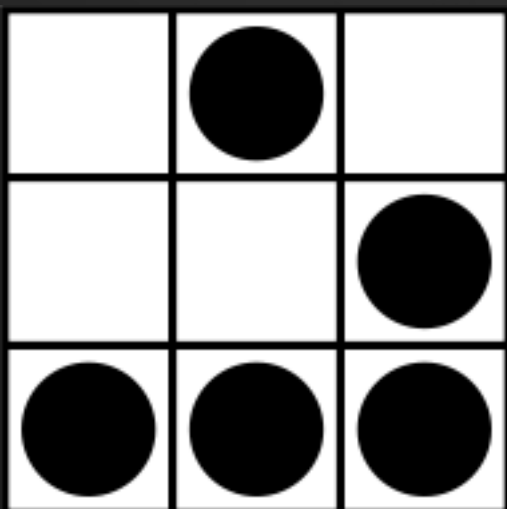
И, наконец, несколько вещей, которые делать не следует:

- Не используйте глупые и напыщенные псевдонимы.
- Не вовлекайтесь в холивары и флеймы.
- Не называйте себя «киберпанком».
- Не отправляйте письма с орфографическими ошибками.

Единственная репутация, которую вы себе создадите, занимаясь любой из этих вещей – это посмешище. У хакеров хорошая память, и вам могут понадобиться годы жизни, чтобы об этом забыли.

Помните, что хакерство – это одновременно и мировоззрение, и особый уровень мастерства, которому вы должны научиться, в основном, сами. Хотя настоящие хакеры и желают вам помочь, они не будут вас уважать, если вы станете просить накормить вас «с ложечки» всем, что им известно. Сначала научитесь чему-нибудь сами – покажите, что стараетесь, что способны к самостоятельному обучению. И уже потом отправляйтесь к хакерам с вопросами. Чтобы быть хакером, вам нужна инициатива, а также способность к самообразованию. Начинать прямо сейчас!

Эрик Реймонд



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, творческие работы, 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!

Linux-гейминг

Игровые новости из мира Linux

Открываем наш сегодняшний обзор настоящей сенсацией: компания Epic Games сделала свой флагманский движок Unreal Engine 4 полностью бесплатным и открытым для всех! Также открыт бесплатный доступ к исходникам, нужен только аккаунт на GitHub и авторизация на сайте Epic. Правда, исходники доступны под несвободной лицензией (EULA), которая не предоставляет вам право распространять модифицированный код конечным пользователям, за исключением других лицензиатов UE4. Инструменты и утилиты, входящие в комплект движка, также несвободны.

Кроме того, UE4 бесплатен, по сути, только для инди-проектов и академической среды – в случае коммерческого использования лицензия предполагает выплату 5% роялти после превышения дохода в \$3000 от ежеквартальных продаж. Тем не менее, этот радикальный шаг, по всей видимости, станет для Epic Games очередным конкурентным прорывом – и, надеемся, подтолкнет остальных разработчиков на аналогичные решения: ждем исходники Unity!

Напомним, Unreal Engine 4 – это 3D-движок AAA-класса, поддерживающий все ведущие ОС и платформы (включая PlayStation 4 и Xbox One). В 2014 году Epic открыла доступ к исходникам продукта для всех желающих по подписке за \$19 в месяц.

Вскоре ответный шаг поступил от корпорация NVIDIA, которая опубликовала исходники своего знаменитого физического движка PhysX. Код движка будет поставляться в составе бесплатного PhysX SDK, подготовленного для Windows, Linux, OS X и Android – ранее SDK включал бинарные библиотеки и поставлялся только для Windows.

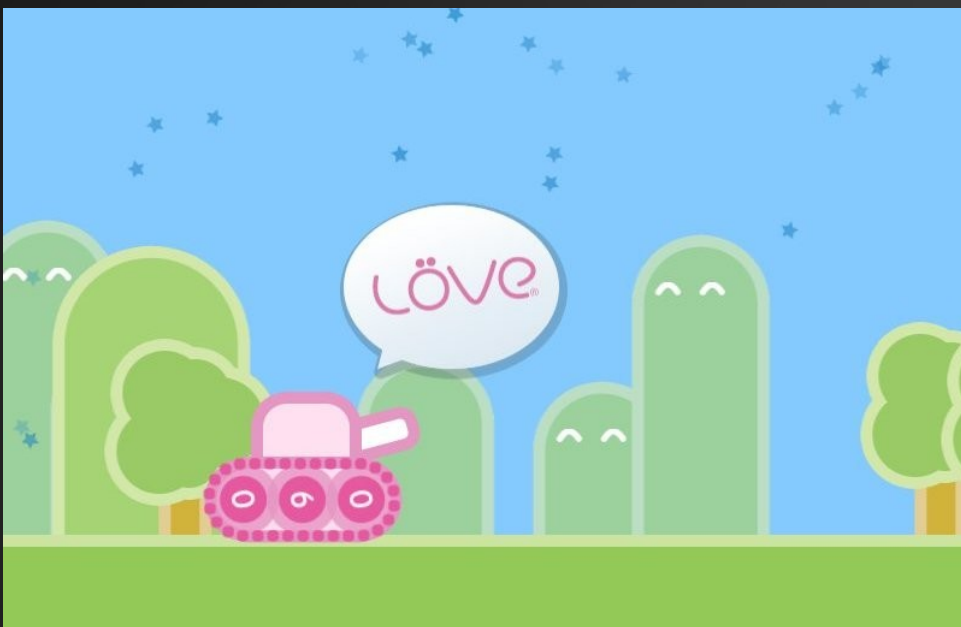


Напомним, PhysX – это один из самых популярных игровых физических движков, он используется в более чем 500 современных играх на различных платформах и поставляется в составе Unreal Engine и Unity. Кстати, NVIDIA и Epic Games достигли договоренности по включению кода PhysX в исходники Unreal Engine 4.

Кстати, недавно стал полностью бесплатным Corona SDK – кросс-платформенный движок для создания игр под iOS и Android. Кроме того, увидел свет очередной релиз популярного движка LÖVE – 0.9.2. В этой версии была реализована улучшенная поддержка UTF-8 в строках Lua, добавлена встроенная возможность сохранять конфигурации клавиш, относительный режим ввода мыши, а также множество более мелких улучшений и исправлений.

Напомним, LÖVE – это свободный кроссплатформенный фреймворк для разработки 2D-игр на языке Lua, включающий средства вывода графики и воспроизведения звука, работы с устройствами ввода, физику и многое другое.

<https://love2d.org>



Последние месяцы оказались необычайно богаты на события из мира свободных игр под Linux – подряд вышло несколько значимых релизов. В первую очередь, это восемнадцатая альфа-версия свободной исторической стратегии 0 A.D., выпущенная под кодовым названием Rhododactylos.

В новой версии осуществлен переход на скриптовый движок SpiderMonkey V31, кодовая база игры переведена на C++11, введен в строй новый сервер, улучшен искусственный интеллект, добавлен новый режим игры «странник» и новые здания Селевкидов, исправлено множество багов.



Напомним, игра является стратегией реального времени с исторической тематикой – она охватывает цивилизации, существовавшие примерно с 500 года до н.э. до 500-го года н.э. Игра отличается неплохой графикой и поддерживает многопользовательский режим на заранее смоделированных, либо динамически создаваемых картах. Исходники 0 A.D. доступны по лицензии GPL.

<http://play0ad.com>

Любители классических пошаговых стратегий должны оценить Freeciv, которая недавно обновилась до версии 2.5. Это свободная игра, разрабатываемая по образу и подобию серии Civilization. Так, в новую версию включен набор правил, реализующий некоторые возможности из Civilization III, сохранив при этом систему сражений как в Civilization II. Представлена новая клиентская часть на Qt, позволившая начать формирование сборок для Windows. Улучшен движок искусственного интеллекта, добавлено 16 новых наций.

<https://play.freeciv.org>



Проект OpenMW недавно обновился до версии 0.35.0. Напомним, OpenMW – это свободная реализация игры The Elder Scrolls 3: Morrowind, добавляющая новые технические возможности и исправляющая старые ошибки. Для работы порт требует оригинальные ресурсы Morrowind. Также в рамках OpenMW разрабатывается собственный редактор OpenMW Construction Set, предназначенный для изменения игрового мира и создания новых квестов.

Одно из наиболее важных изменений нового релиза – возможность продолжать игру, начатую в оригинальном Morrowind. Кроме того, разработчики заявили, что этот релиз – последний на основе OGRE. В числе причин упоминаются кардинальные и неоднозначные изменения в OGRE 2.1, нарушающие обратную совместимость: например, переусложненная система материалов, проблемы с менеджером ресурсов, а также отсутствие поддержки OpenGL 1.x (разработчики OpenMW не собираются отказываться от поддержки старого оборудования, на которое был рассчитан оригинальный Morrowind).

В качестве нового графического движка был выбран OpenSceneGraph (OSG), который поддерживает все необходимые проекту функции. Как следствие, прекратится поддержка DirectX (OSG поддерживает только OpenGL), но зато игра будет работать под управлением Android (при этом Поддержка Windows, Linux и Mac OS X полностью сохранится).

<https://openmw.org>

Свободный 3D-шутер Red Eclipse обновился до версии 1.5. Релиз отмечен как наиболее крупный за всю историю существования проекта - в нем затронуты практически все области игрового процесса: улучшен баланс оружия, обновлен интерфейс пользователя, реализовано большое число новых карт и артефактов, представлен механизм автоматической установки обновлений.



Напомним, Red Eclipse – это кроссплатформенный FPS, построенный на основе движка Cube 2 и продолжающий развитие заброшенного проекта Blood Frontier. Игра поддерживает одиночный и многопользовательский режимы, геймплей базируется на идеях Cube: дезматчи на больших аренах, которые можно редактировать в реальном времени.

<http://redeclipse.net>



Вышла новая версия OpenClonk – многопользовательского 2D-платформера с открытым миром. Это игра, в которой вы управляете так называемыми клонками – маленькими, но остроумными и ловкими человекоподобными существами. В OpenClonk 6 реализована новая система освещения, добавлены новые игровые сценарии и объекты.

<http://www.openclonk.org>

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>

VULKAN

Новый графический API на смену OpenGL

На протяжении 20 лет единым промышленным стандартом на вывод 3D-графики в реальном времени безраздельно был OpenGL (Direct3D в расчет не берем как специфичное решение для игр под Windows). OpenGL позволил создавать программы, использующие аппаратное ускорение трехмерной графики, не только инженерам в крупных корпорациях под мощные рабочие станции, но и практически любому энтузиасту под любую платформу – от карманных компьютеров до ПК самостоятельной сборки, работающих на Linux. Дух кроссплатформенности в области компьютерной графики появился и успешно живет именно благодаря OpenGL.

Но, при всех его достоинствах, дизайн OpenGL начинает устаревать. Этот стандарт создавался в те времена, когда компьютеры были в тысячи раз слабее – никто не занимался трассировкой лучей на GPU, не вычислял на нем физику и т.д.

Чтобы получить на компьютере красивую анимацию, работающую с приличной скоростью, приходилось идти на самые сложные ухищрения, пользоваться различными хаками, которые к тому же были специфичны для каждой платформы. OpenGL создавался как раз для того, чтобы упростить создание типовых 3D-приложений: с ним программисты теряли возможность писать напрямую в видеопамять, но зато получали абсолютную переносимость программ – и они, естественно, выбрали второе.

Теперь ситуация несколько изменилась: с ростом вычислительных мощностей GPU вновь возникла потребность «приблизиться к железу», чтобы получать на компьютере качественно новую картинку. Простое рисование затекстуренных полигонов сейчас уже мало кого удовлетворяет: всем хочется больше шейдеров и массивной параллельности, хочется больше доступа к видеопамяти и меньше оверхеда в видеодрайверах.

Даже последние версии OpenGL недостаточно соответствуют этим требованиям, поэтому настало время для нового API.

Все началось с интерфейса Mantle, представленного в 2013 году компанией AMD (мы писали о нем в «FPS» №27 '13). Проект был анонсирован как современная альтернатива OpenGL и Direct3D – по словам разработчиков, Mantle отичает простота, переносимость, а также высокая эффективность работы с GPU. Было заявлено, что Mantle предоставляет программистам возможность прямого обращения к памяти GPU.

С тех пор прошло два года, но в Сети так и не появилось подробной информации о Mantle. Официальный SDK до сих пор недоступен для «простых смертных», публичная спецификация не вышла. Отдельные энтузиасты, используя крупинки информации, пытались написать свой собственный «Mantle-драйвер», и даже добились в этом определенных успехов, но дальше экспериментов дело не пошло, Mantle так и остался проприетарной секретной технологией, специфичной для карт AMD.

Летом 2014 года, когда ажиотаж вокруг Mantle несколько поутих, со стороны Khronos Group поступила не менее сенсационная новость. После выпуска OpenGL 4.5 консорциум приступил к работе над спецификацией нового поколения под кодовым названием OpenGL NG.

Было заявлено, что этот API не будет обратно совместимым с предыдущими версиями OpenGL. Обратной совместимостью придется пожертвовать ради революционных возможностей: разработчики приложений получат полный низкоуровневый доступ к функциональности современных графических процессоров!

И вот, в марте 2015 года был представлен пресс-релиз нового графического API под окончательным названием Vulkan. Задача стандарта – устранение основных недостатков OpenGL, таких как переусложненность драйверов и неоптимальные механизмы обработки ошибок.

Одной из основных проблем OpenGL является изначальная ориентация на последовательный поток команд в одноядерных системах, что мешает использованию всего потенциала современных многоядерных GPU. Vulkan подразумевает более активную прямую работу с видеопроцессором, минимизируя накладные расходы, вносимые драйвером, и предоставляя встроенную поддержку многопоточной обработки команд.

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 охватывает все доступные платформы и предоставляет единый API для настольных, мобильных и web-систем.

Интересна в Vulkan и реализация шейдеров. Стандарт предусматривает вынос генерации команд GPU на сторону приложения – в противовес компиляции шейдерных программ на стороне драйвера, как это происходит в OpenGL – а также использование предкомпилированного промежуточного представления кода для выполнения на стороне GPU.

В качестве такого представления предложен «универсальный GPU-ассемблер» – переносимый низкоуровневый язык SPIR-V (Standard Portable Intermediate Representation), который может применяться как для графики, так и для параллельных вычислений.

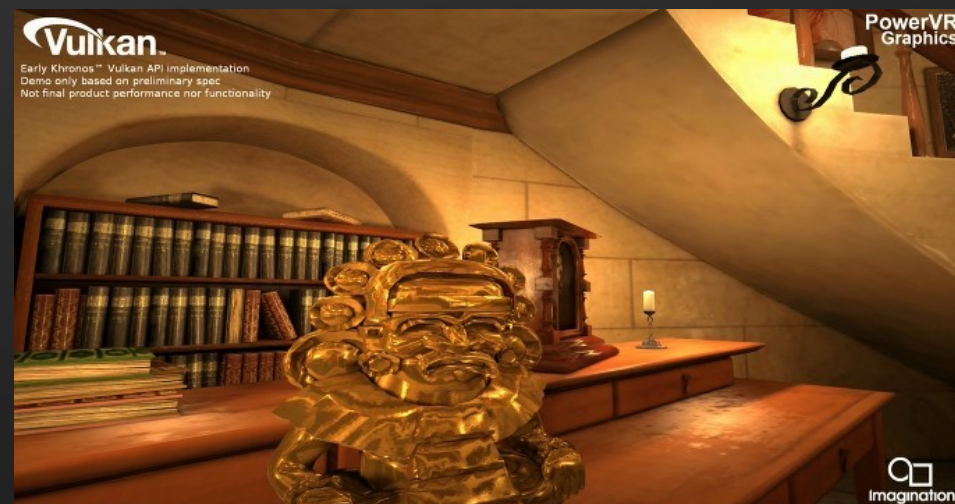
Для SPIR-V можно будет создавать компиляторы различных высокоуровневых языков – сначала будут реализованы трансляторы GLSL и OpenCL C в SPIR-V, а затем не исключено появление совершенно новых GPU-языков, заточенных под те или иные нужды.

Кстати, такие уже существуют, в их числе – язык обработки изображений **Halide** и концепция GPU-ориентированного эзотерического языка **Checkout**. Чтобы упростить эту задачу, в будущем планируется создать инструменты для преобразования биткода LLVM в SPIR-V и наоборот.

Начальная версия спецификации Vulkan и первые рабочие реализации нового API ожидаются уже в течение этого года.

А что же Mantle? Кажется, что AMD совершенно потеряли интерес к своему проекту: на сайте компании указано, что Mantle SDK доступен для партнеров, но всем остальным рекомендуется обратить внимание на DirectX 12 и Vulkan.

Тимур Гафаров



От Монополии до Warhammer

Магия настольных игр

Если начать листать страницы истории игр как таковых, то часто игра означала какое-то корпоративное занятие – команда против команды, двое на двою, один на один... Игра – это соперничество: то, что заставляет кровь бурлить, что заставляет идти на всякие ухищрения, проявлять свои лучшие качества и развиваться!

Сейчас нашим соперником в играх все чаще выступает компьютер, искусственный интеллект. Но, несмотря на все его знания и умения, компьютер – слабый игрок, потому что он не обладает и толикой той хитрости, которая есть у человека.

Уже давно существует виртуальный футбол, шашки, шахматы, различные карточные игры – да и многие другие, более сложные по своей структуре и правилам игры переходят в цифровой формат: к примеру, существует множество компьютерных вариантов знаменитой «Монополии». В этой связи любопытно, что многие современные компьютерные игры появились именно благодаря играм настольным. Например, легендарная стратегия **Civilization** когда-то была известна только как настолка – в 1980 году была издана настольная игра Civilization, ставшая основой для успешной одноименной серии.

Игра представляла собой схватку двух правителей, которым нужно было привести свою нацию к процветанию. В компьютерной версии вашим противником становится ИИ, но азарт и чувство соперничества из-за этого не пропадает.



В данной игре соперник настроен на выигрыш и может быть так же коварен и алчен, как и обычный человек. Сюжеты многих других пошаговых стратегий также были взяты из настольных игр.

Отдельного упоминания заслуживает культовая **Warhammer 40000**, созданная в 1987 году на основе фэнтезийного ролевого сеттинга Warhammer Fantasy. Она включает в себя всю галактику Млечный Путь 41-го тысячелетия (отсюда и название) со множеством звезд, планет и космических станций, охваченную бесконечной войной между различными сторонами.



В игре проводятся фантастические сражения между целыми армиями миниатюрных фигурок воинов и боевой техники. Фигурки склеиваются и раскрашиваются игроками самостоятельно, после чего вводятся в игру по определенным правилам. Перед началом партии игроки выбирают сценарий сражения, который может представлять собой как простую схватку между армиями до полного уничтожения одной из сторон, так и сложный набор регулируемых правилами заданий. В процессе сражения игроки поочередно передвигают фигурки воинов по карте, причем фактическое расстояние между ними играет роль в расчете успеха того или иного действия, определяемого по специальным таблицам и броскам кубиков.



По мотивам Warhammer 40000 была создана огромная серия компьютерных пошаговых стратегий, игры в которой продолжают выходить по сей день. Так, в конце этого года должна выйти **Warhammer 40000: Eternal Crusade** – онлайн-версия знаменитого «Вархаммера», которая вызвала большой интерес не только среди фанатов этой вселенной, но и всех любителей ММО в целом.

Но самое интересное в том, что сейчас все чаще наблюдается обратное явление: компьютерные игры переносят на бумагу! Так, серия **Warcraft** стала настолько популярна, что разработчикам пришла идея перенести свое детище в настольный вариант, тем более почва была благодатная – пошаговые стратегии можно легко реализовать как настолки. Настольная Warcraft стала новым культом у поклонников компьютерной серии.



Клонировали в настолку и другую легендарную стратегию – **Starcraft**, но по сюжету и подаче эти две игры очень похожи: обе представляют собой военную стратегию, цель которой – выжить и победить любой ценой. Играть в такие игры можно компаниями до 6 человек. Нельзя также не упомянуть про настольную игру **Doom**, созданную на основе знаменитого шутера. Тут игроки делятся на два лагеря: кто-то играет за морских пехотинцев, кто-то – за монстров. Впрочем, игра получила неоднозначные отзывы от поклонников: все-таки переносить трехмерный шутер в настольную плоскость – это неблагодарное и сложное занятие.

Российский рынок в плане настольных игр тоже начинает активно развиваться: относительно недавно увидела свет игра **World of Tanks: Rush** по онлайн-игре World of Tanks. И опять – тот же дух соперничества: имея у себя армию из танков, нужно победить любой ценой...

Не только компьютерные игры становятся основой настолок – некоторые литературные произведения так взбудоражили умы людей, что предприимчивые дизайнеры решили создать по их сюжетам полноценные настольные игры.

Например, существует огромное количество игр по вселенной ужасов Лавкрафта, и все получили хорошие оценки от игроков. Несколько игр есть и по Плоскому миру Терри Пратчетта (вечная ему память) – такая яркая и многообразная вселенная просто не могла не лечь в основу настольной игры. Игры по мотивам «Игры Престолов» тоже заняли достойное место в нише современных настолок.



Почему же наше поколение все чаще обращается к играм настольным, когда существует такой огромный выбор компьютерных? Попробуем в этом разобраться.

Настольные игры являются чем-то вроде социальных консолидаторов. Все-таки компьютерная игра – это занятие одинокое: даже если в ней участвуют несколько человек по сети, непосредственно за компьютером вы сидите одни. Настольная же игра собирает друзей, членов семьи за одним столом – а это в наше время случается все реже. Живое эмоциональное общение во время совместного занятия способно благотворно влиять на психологическое состояние каждого игрока. Кроме того, настольная игра способна развить воображение, потому что ее легко расширить и доработать: придумать свои правила, новых персонажей, новые карты... В конце концов, можно просто взять и придумать свою настолку, начертив на бумаге игровое поле – это гораздо проще, чем создать компьютерную игру.

Также немаловажно, что настольные игры не требуют электричества и цифровой техники – нужно только воображение. Но ощущения от настольной игры зачастую могут быть несколько не менее яркими, чем от компьютерной. Так что почаще отрывайтесь от экрана, собирайтесь компанией и играйте в настолки!

Наталья Чумакова



От редактора:

Прочитав эту статью, я не мог не вспомнить эпизод из своей «приставочной» юности. Не имя компьютера, я все же горел желанием создавать свои игры, и единственный путь к этому лежал через настолки – я попытался перенести компьютерную механику на бумагу.

В этом отношении меня вдохновили популярные в 90-е годы книги-квесты – такие, например, как «Кощеева цепь» Шувалова. В частности, я взял из нее концепцию характеристик персонажа и механику боя, совместив их с традиционным игровым полем, на котором игрок перемещает свою фигуру с помощью кубика, как в «Монополии». Когда игрок попадает на клетку с врагом, начинается бой: необходимо бросить кубик, и в зависимости от результата вы либо наносите удар, либо сами теряете энергию. Обычно числа 1, 2, 3 означали атаку врага, а 4, 5, 6 – ваш удар. Можно было выбирать оружие, которым вы хотите атаковать – каждое из них обладало своими характеристиками мощности. На такую механику прекрасно легли идеи экшн-игр с их показателями жизни и боеприпаса, поэтому и тематика моих самодельных настолок была соответствующей – космические войны, сражения с пришельцами и роботами.

Был, впрочем, интерес и к экономическим играм: так, я разработал эдакий научно-фантастический вариант «Монополии», где вместо недвижимости были планеты и звездные системы, а игроки перевозили на кораблях грузы, чтобы торговать ими. В результате получилась очень удачная и по-своему захватывающая вещь, но, к сожалению, игра не сохранилась, и точные ее правила были безвозвратно утрачены...

Тимур Гафаров

«Конструируя чувства...»

Техно-симфония Виктора Аргонова

В начале 2015 года увидела свет бета-версия «Пересекая черту» – техно-симфонии известного в околокомпьютерных кругах электронного композитора Виктора Аргонова, лидера групп «Комплексные числа» и «Виктор Аргонов Project». В связи с этим событием, у меня возникло желание порассуждать на тему электронной музыки в целом – и, конечно, поделиться своим собственным восприятием творчества «Комплексных чисел».

Эстетическая электронная музыка в России почему-то плохо приживается. Влияние Spase, Жана-Мишеля Жарра и экспериментов немецких краут-рокеров обошло нашу сцену стороной, дав нам только одно по-настоящему интересное явление – незаслуженно забытую советско-латвийскую группу «Зодиак», пик популярности которой пришелся на эпоху повального увлечения научной фантастикой и космосом.

Другая знаковая группа того же времени – «Технология», наш ответ Depeche Mode: на ней, по сути, начался и закончился весь русский синти-поп.

90-е, как это ни печально, не приняли идеи «эстетики будущего», потребовав от исполнителей максимальной простоты и доступности. Рейв-движение 90-х не породило сколько-нибудь ценных музыкальных изобретений, выродившись в «клубняк» и примитивные ритмы для экстази-вечеринок.

И сегодня, с легкой руки диджеев, любая электронная музыка воспринимается массами как дискотечный ширпотреб. Композиторы, способные на что-то большее, нежели аранжировка попсовых шлягеров, давно работают на Западе и для западного слушателя – назвать творчество тех же Tesla Boy или PPK «русской электронной музыкой» как-то язык не поворачивается...

При этом тяга к машинному звучанию в рядах нашего рок-бомонда определенно есть: драм-машину использовали еще сами «Кино», синтезированный звук широко применяли мэтры русского арт-рока 80-х и 90-х – «Пикник», «Агата Кристи», «Оберманекен» и др. Эдмунд Шклярский, кстати, известен как изобретатель собственных электроинструментов, а в конце 90-х он записал чисто электронный альбом «Пить электричество».

Эти эксперименты, в конечном счете, повлияли на становление в России в начале «нулевых» собственной дарквейв-сцены (Otto Dix, Unreal, The Matrixx и др.). Впрочем, она неразрывно связана с готической субкультурой и другими темными направлениями постмодернизма, что ставит собственно электронный звук в дарквейве на второй план.

Может ли эта ситуация измениться к лучшему? Судя по всему – да, принимая во внимание развитие Интернета и цифровых средств дистрибуции. За рубежом электронных композиторов сейчас великое множество: пышным цветом цветут исполнители разных сортов дабстепа и чиптюна. Этому во многом способствовало появление качественных цифровых синтезаторов и виртуальных студий «для масс» – по сути, электронную музыку записывать сейчас может любой, был бы музыкальный слух и установленный Fruity Loops.

Зная, что в компьютерной графике и играх сейчас тоже наблюдается возрождение идей 80-х, становится неудивительно, что композиторы все чаще обращаются к технически устаревшему, казалось бы, синтезированному звуку времен 8-битных систем. Роботы, космические путешествия, технологии будущего – вот тематика новой электронной музыки, родившейся на стыке геймерства, научной фантастики и аниме...

В нашей стране имеющийся вакуум в этом жанре заполнила владивостокская группа «Комплексные числа», которую можно по праву назвать «певцом 8-битной эпохи». Ее часто связывают с имиджбордовской субкультурой, но появилась она еще в середине 90-х, задолго до появления имиджборд, когда ранние российские интернет-гики и анимешники «первой волны» начали организовывать клубы и пробовать заниматься околокомпьютерным творчеством.

Виктор Аргонов, тогда еще студент Дальневосточного госуниверситета, собрал музыкальную группу, чтобы объединить «в одном флаконе» то, что по странной причине объединять не принято – доступную мелодичность и сложную интеллектуальную тематику. В каждой строчке – прыжок через миллиарды лет: картины далекого будущего, «парадокс близнецов», проблемы разума и эволюции, тепловая смерть Вселенной... Это сочетание научной фантастики, электронных аранжировок и чистого женского вокала оказалось столь удачным, что «КС» приобрели в определенных кругах статус по-настоящему культовой группы.

Но, к сожалению, большая часть интернет-аудитории оказалась неспособной воспринять эту музыку в принципе. Напомним, что это было начало «нулевых», эпоха ожесточенной борьбы между роком и попсой – «КС» появились в самое неудобное для своего творчества время и, по сути, оказались зажаты между двух огней. Любая новая музыка тогда просто не могла быть принята как самостоятельное явление: либо ты «с нами», либо «против нас».

Музыкальные снобы в ярости: как группа с таким «попсовым» саундом смеет замахиваться на интеллектуальность и философию? И есть ли тут философия вообще?

Гуманитарный ум, отполированный до блеска постмодернистской литературой и артхаусными фильмами, не смог переварить рассуждения об общечеловеческом разуме и путях развития цивилизации – им, как оказалось, все еще подавай проблемы времен Ницше, «достоевщину», Бога, существование которого надо опровергать или доказывать. И все это – в эпоху Интернета и спутниковой связи!..

Пошлость суждений этих малохольных критиков видна невооруженным глазом: музыка у них почему-то до сих пор должна быть «живой», а тексты – о любви, депрессии и прочих психосоматических феноменах. Возникает вопрос: почему мы должны брать с собой в космическую эру весь этот устаревший романтизм, пафосные театральные условности, «признанную критиками» бессмыслицу, порожденную алкогольно-наркотическими эксцессами? Если мы все делаем при помощи компьютера, то почему на нем нельзя записывать чисто компьютерную музыку, объявив ее мейнстримом будущего? Создать, наконец, новую эстетику с большой буквы?

Невольно вспоминаются Стругацкие: пока «гадки лебеди» XXI века пытаются строить новый мир, корифеи от культурной жизни заняты братанием со своими «небрityми, истеричными, вечно пьяными» кумирами...

Причем, это можно понимать в самом буквальном смысле: весь московский бомонд в восторге от Сергея Пахомова с его алкогольно-шизофреническим абсурдизмом, от поэзии, чуть менее чем полностью состоящей из матерной лексики. Но для «Комплексных чисел» в творческом мире у них места нет...

Впрочем, разброс мнений о музыке «КС» имеет резко полярный характер. Если она вам нравится, то нравится однозначно и безусловно – и наоборот. Встречаются и такие слушатели, которые начали знакомство со всей электронной музыкой именно с «КС» – мол, раньше не слушал, а сейчас «зацепило»... И это логично: согласитесь, вряд ли кто-то начинает слушать электронику с академических мэтров вроде Михаила Чекалина или Яниса Ксенакиса. А попытка поделить все в искусстве на «высокое» и «низкое» заранее обречена на провал – то, что считается «низким» сегодня, завтра станет уже классикой: чего стоит в этом отношении один только джаз!

В 2001-2007 годах Аргонов, собрав новую команду вокалистов, записывает свою первую «техно-оперу». Как будто подливая масла в огонь холиваров, он берет темой очередного футуристического «если» альтернативную историю нашей страны – действие происходит в 2032 году в СССР. Опера так и называется: «2032. Легенда о несбывшемся грядущем».

И вновь – тонкий смысл текстов, необычайно умные вопросы, поставленные героями, ускользнули от понимания большинства слушателей. Животная ненависть к «совку», прочно укоренившаяся в сознании нашей творческой интеллигенции, не дала ей воспринять «2032» отвлеченно, с позиции людей завтрашнего дня, а не сквозь призму собственных политических взглядов. Вновь звучит заезженное «коммунизм – это утопия». Может, и утопия – но что же, на этом «закрыли тему»?..

Они готовы ходить на голливудский ширпотреб, выкладывая за это любые деньги, чтобы потом сетовать в блогах на отсутствие в современной кинопродукции смысла и души. Но при этом бесплатная музыка, создаваемая энтузиастами, распространяющаяся по CC-BY-NC и наполненная высоким смыслом до последней строчки, вызывает у них отторжение.

Именно это – желание показаться высокоразвитой личностью, не затрудняя себя изучением науки и философии – отлично характеризует наших интеллектуальных снобов. Они потребляют, чтобы презирать потребление – в этом вся их жалкая и нелепая сущность!

Героев Аргонова в подобной вопиющей противоречивости не обвинишь – любой здравомыслящий человек не станет отрицать, что производство и потребление продуктов является основой нашего существования. Другое дело – как устроить мир так, чтобы продуктов хватало на всех? Можно ли сделать это при помощи машинного расчета? Существует ли единая «формула счастья» на всех и «единый подход» ко всему? Именно эти проблемы поднимает «2032»: если вас интересует политическая футурология, то эта опера – для вас.

Кстати, «2032» является первым в мире крупным музыкальным произведением с полностью открытыми исходными материалами – MIDI- и Cubase-партитурами, вокальными записями и инструментальными сэмплами. Сам Аргонов в интервью неоднократно признавался, что не против того, чтобы появлялись ремиксы и аранжировки его произведений.

Вторым шедевром «Аргонов Project» стала опера «Русалочка» (2013): оригинальная научно-фантастическая трактовка классической сказки, где Русалочка – это представитель высокоразвитой цивилизации, существующей параллельно с человечеством в глубинах океана. Существо с внешностью юной девушки – на самом деле совершенно чуждая нам форма жизни, не имеющая ничего общего с клеточными организмами. Живущие сотни лет, практически неуязвимые и намного превосшедшие человечество в морально-этическом плане подводные обитатели предостерегают героиню от того, чтобы вмешиваться в жизнь людей, но она все-таки выходит на сушу и сталкивается с нашим жестоким и противоречивым миром – непонятным, но почему-то притягательным для нее...

«Русалочка» получилась несколько более сложной, чем «2032» – и концептуально, и технически: появились чисто шумовые «кинематографические» сцены без музыки, стало меньше элементов диско 80-х и больше – транса, электро и даже хардкора. Но приятный женский вокал никуда не делся: главную героиню озвучивает все та же полюбившаяся всем Ariel.

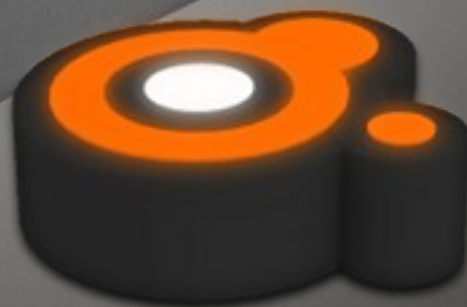
Третье крупное произведение Аргонова – техно-симфония «Пересекая черту» (2015) – содержит всего одну вокальную композицию, уже успевшую стать для поклонников «КС» хитом. Но по выразительности инструментальная часть в вокальном сопровождении особо и не нуждается, успешно создавая в восприятии слушателя картины далекого будущего – в условиях технологической сингулярности изменится не только образ жизни человека, но и сама его сущность: перейдя «черту», человек станет принципиально новым существом...

Тимур Гафаров

***P.S.** Когда эта статья уже была дописана, а номер журнала находился на этапе верстки, Аргонов, прислушавшись к отзывам некоторых поклонников, решил сменить название симфонии: теперь это «Переосмысляя прогресс» – звучит, на мой вкус, менее лирично, но зато лучше раскрывает суть произведения.*

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>