

28
2014

FPS

**Интервью
с создателем OSL**

Рисуем фрактал на D

Обзор альтернативных ОС

Кому нужен Linux?

**Культовые игры:
The Elder Scrolls V**

**Как создавались
старые игры**

**Каково быть гиком
в тюрьме?**

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





FPS

№28

FPS - бесплатный, свободно распространяемый электронный журнал, посвященный различным видам цифрового творчества. Тематика FPS охватывает разработку игр и игровых движков, обработку изображений и звука, уроки по трехмерному моделированию и ретуши фотографий, а также свежие новости из мира цифровых технологий.

FPS - это журнал для программистов, художников, моделлеров, линуксоидов, энтузиастов движения СПО, хакеров и просто творческих людей, ищущих и умеющих находить в этом мире качественную «пищу для ума».

Мы ратуем за свободу слова и свободный обмен информацией - журнал распространяется на условиях лицензии Creative Commons (CC-BY-NC-SA). Мы рады любому сотрудничеству, приветствуем любые новые идеи и приглашаем в наш авторский коллектив всех желающих!

Журнал издается в Казани с января 2008 г. и в данный момент выходит раз в два-три месяца.

© 2008-2014 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**
Корректор: **Наталья Чумакова**

По вопросам сотрудничества обращайтесь по адресу:
gecko0307@gmail.com

- **Blender**
 - Новости
 - Интервью с Ларри Гритцем
 - Обзор дополнений. Выпуск 7
- **GIMP**
 - Новости
- **Кодинг**
 - Язык D: новости
 - Рисуем фрактал на D
 - Обзор альтернативных ОС
- **Кому нужен Linux?**
 - Правда и вымыслы
- **Linux-гейминг**
 - Игровые новости из мира Linux
- **Культовые игры**
 - The Elder Scrolls V: Skyrim
- **Angry Birds в России**
 - Сердитые птички против сердитых родителей
- **Хакеры без страха и упрека**
 - Как создавались игры 80-х и 90-х
- **Каково быть гиком в тюрьме?**
 - Исповедь Анонимуса



Blender

Новости

В новый год – с новыми проектами и идеями! **Blender Foundation** анонсировали новый, 5-й по счету открытый фильм – проект под кодовым названием **Gooseberry**. Планируется съемка полнометражной анимационной картины, для коллективной работы над которой приглашаются талантливые художники и программисты со всего мира. Проект будет охватывать десять CG-студий в разных точках земного шара: в общей сложности, над **Gooseberry** будут работать 50-80 человек.

При этом у фильма уже есть стартовое финансирование: фонд EU Media согласился поддержать инициативу в рамках своей программы «Open Production Environments». На два года съемок (2014-2015) будут выделены около 440 тысяч евро; деньги, как обычно, пойдут на развитие Blender, а также на технические расходы по поддержанию сети участников этого проекта.

В рамках **Gooseberry** планируется уделить внимание инструментам по управлению контентом и «виртуальными студиями» для обеспечения удаленной совместной работы, развитию открытых облачных технологий, интеграции различных открытых инструментов (GIMP, Krita, Inkscape, Ardour и т.д.) в единый производственный конвейер, улучшению качества инструментов самого Blender, включая анимацию, симуляцию физики, систем частиц, волос, меха, поддержку сетей распределенного рендеринга и многое другое.

Из известных людей, привлеченных к работе над **Gooseberry**, можно отметить Брехта ван Ломмеля, автора рендер-движка Cycles, и нашего соотечественника Сергея Шарыбина, который прославился разработкой системы отслеживания движений.

Узнать свежие подробности по проекту можно на официальном сайте: <http://gooseberry.blender.org>

Согласно намеченному плану, разработчики **Blender** недавно перевели репозиторий исходников пакета с SVN на Git. На самом деле, работа над многочисленными ветками и экспериментальными версиями Blender давно уже велась на основе Git – данный переезд лишь завершает этот большой этап. В ближайшее время планируется также создать зеркала основного репозитория на Github и Gitorious.

Вместе с этим, обновилась и инфраструктура разработки программы. Багтрекер, аудит кода и веб-интерфейс репозитория переехали на портал <http://developer.blender.org>, построенный на движке Phabricator. Там же вы найдете нестабильные "ночные сборки" Blender для Windows, Linux и Mac OS X.





Радует, что сообщество Blender не забывает и «старых героев». Некоторые из них уже стали культовыми: так, мексиканец Даниэль Ордонез назвал свою новорожденную дочь... Синтел – в честь героини знаменитой блендеровской короткометражки "Sintel", отважной девушки-воительницы. Само имя, конечно, красивое – остается только надеяться, что девочка не повторит драматическую судьбу своей тезки...

Есть свои поклонники и у толстого кролика Бака: Янус Кристенсен потратил целых 3 года на то, чтобы отрендерить «Big Buck Bunny» в формате 4K HD со стереоскопией и 60 кадрами в секунду. Для этого использовалась рендер-ферма BURP (<http://burp.renderfarming.net/>). Скачать фильм можно на <http://bbb3d.renderfarming.net/>.

Отметим, что изначально «Big Buck Bunny» был отрендерен в стандартном HD – теперь фильмом можно наслаждаться и на кинопроекторах высокого класса.

А в конце прошлого года состоялась онлайн-премьера второй части полюбившихся всем приключений ламы по имени Питти – короткометражки Пабло Васкеза «Caminandes: Gran Dillama». Напомним, первая часть фильма удостоилась награды на анимационном фестивале SIGGRAPH 2013. На своем пути через прерии Питти постоянно сталкивается с разного рода препятствиями – удастся ли настойчивой ламе в этот раз найти выход из ситуации?..

Подробнее – на сайте проекта: <http://www.caminandes.com>

Режиссер картины – 3D-художник из Патагонии Пабло Васкез, на протяжении 10 лет работающий над полнометражными и короткими фильмами, играми и рекламными роликами. Музыкальную дорожку к «Caminandes» записал известный в мире свободного кино немецкий композитор Ян Моргенштерн, работавший также над «Elephants Dream», «Big Buck Bunny» и «Sintel».

От киноанонсов – к анонсам книжным: как раз к праздничным дням вышла книга «The Art of Blender» от сайта CGCookie.com – альбом-компиляция лучших работ, созданных при помощи пакета. Информацию по заказу книги ищите в интернет-магазине Blender:

http://www.blender3d.org/e-shop/product_info_n.php?products_id=164



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!



Интервью: Ларри Гритц, создатель Open Shading Language

Не так давно сайт <http://blenderdiplom.com> взял интервью у Ларри Гритца - ведущего разработчика Open Shading Language (OSL) в Sony Pictures Imageworks.

Напомним, OSL - это мощный язык описания шейдеров, используемый в рендер-движках, за последние годы ставший де-факто стандартом в области компьютерной графики наряду с GLSL и RSL.



В интервью Гритц раскрыл особенности и преимущества языка, а также детали его разработки в качестве открытого проекта.

Какие изменения привнес OSL в рабочую среду Sony Pictures Imageworks?

До появления OSL, все наши шейдеры представляли собой плагины к Arnold (собственный рендер-движок SPI - прим. ред.), написанные на C. Программисты шейдеров тратили все больше и больше времени на работу с низкоуровневыми деталями: все-таки C - не самый подходящий язык для написания шейдеров, в нем легко «выстрелить себе в ногу». Стало довольно сложно достичь нужного нам качества материалов и освещения, и стало ясно, что нам нужен другой уровень абстракции.



OSL, в сущности, имеет три главных преимущества. Первое - синтаксис OSL был спроектирован специально для описания шейдеров, и это сделало язык более удобным для данной задачи, по сравнению с C. Второе - абстрактный API, который не ставит программиста перед необходимостью разбираться во внутренних особенностях рендер-движка. В совокупности это позволяет программисту сосредоточиться на описании материала, а не на технических деталях реализации. Наконец, значительным концептуальным шагом вперед является парадигма OSL - описание не цветовых пикселей, а замыканий, без зависимости от вида и особенностей сэмпинга. Это позволяет совместить программируемость и физическую корректность моделей освещения.

Что побудило сделать OSL свободным ПО?

Проприетарные инструменты имеют свои преимущества, но у них есть и серьезный недостаток: для работы с несвободной внутренней программой или форматом невозможно нанять специалиста, как и получить стопроцентную совместимость со сторонними инструментами. Выгоды от закрытого кода не сравнимы с преимуществами открытой экосистемы, которая распространялась бы широко за пределами SPI. Мы предвидели, что сторонние программы постепенно «обрастут» поддержкой OSL, и, таким образом, теперь мы можем нанимать сотрудников, уже знакомых с этой технологией.

Кроме того, для разработчиков ПО, имена которых обычно не упоминаются в титрах фильмов, очень почетно работать над проектами, которые имеют широкое признание за пределами студии. Наконец, знать, что множество людей видят вашу работу - это лучшая мотивация для того, чтобы писать качественный код. Мы уверены, что открытость идет проекту только на пользу.

Получаете ли вы патчи «со стороны»?

Да. На самом деле, еще до того, как OSL был написан, мы показали ранние черновики другим заинтересованным студиям, и получили много отзывов. Если же говорить о вкладе в виде кода, то главные разработчики, конечно, в SPI, но мы также получаем патчи от сообщества. Особенно сейчас, когда OSL интегрирован в Blender/Cycles, V-Ray и Autodesk Beast, мы регулярно получаем вклад от авторов этих пакетов. Это багфиксы, оптимизации и различные изменения, помогающие портировать код на другие платформы - например, под Windows, которую мы изначально не собирались поддерживать.

Код OSL был открыт в 2010 году - с тех пор прошло уже немало времени, но Cycles все еще является единственным свободным рендер-движком, поддерживающим OSL. В чем же причина? Почему разработчики движков тянут с этим?

Несмотря на то, что код был открыт в 2010-м, только в 2012 году вышли первые фильмы SPI с использованием OSL - это был важный этап, которого все ждали, перед тем, как всерьез заняться поддержкой OSL. Так что времени, на самом деле, прошло не так много. Кроме того, помимо Cycles, поддержку OSL уже включили такие уважаемые пакеты, как V-Ray и Autodesk Beast, а это что-нибудь да значит.

Переключиться на OSL с существующего движка сложно по следующим причинам. Во-первых, движок выполнения шейдеров - это слишком крупная часть кодовой базы, чтобы можно было так просто заменить ее сторонней библиотекой, которую вы, по сути, полностью не контролируете. Во-вторых, при таком серьезном обновлении придется пересмотреть систему освещения и сэмпинга. Наконец, в-третьих, продукты с большой потребительской базой вообще достаточно инертны, когда дело касается серьезного изменения функциональности.

```
1 #include "stdosl.h"
2
3 shader pattern(
4     point Vec = P,
5     float Intensity = 1.0,
6     float Wave = 1.0,
7     float Amplitude = 1.0,
8     output float Fac = 1.0)
9 {
10
11     point Pos = transform("object", Vec)
12     float t = Pos[1] + 0.5;
13     t *= Pos[1] - 0.5;
14     t *= Pos[0] + 0.5;
15     t *= Pos[0] - 0.5;
16     t *= (Pos[0] + Pos[1]) + sin(Pos[1] * Wave) * Amplitude;
17
18     Fac = abs(Intensity/t);
19
20 }
```

Синтаксис языка напоминает RenderMan Shading Language. Можно ли сказать, что это было сделано специально, чтобы облегчить портирование существующих шейдеров?

В RSL многое сделано хорошо, и мы хотели взять за основу плюсы этого языка, добавив при этом необходимые улучшения. Мы не слишком заботились о портируемости существующих шейдеров, но действительно хотели создать нечто, что было бы знакомо программистам. RSL, в свою очередь, по тем же причинам похож на C - как и все остальные широко используемые языки описания шейдеров (GLSL, Cg, Mantra и т.д.) Нет больших преимуществ в том, чтобы начинать с синтаксиса, совершенно незнакомого пользователям.

Оригинал:
<http://blenderdiplom.com>



Обзор дополнений Blender

Выпуск 7

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 или создания игрового контента.

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: http://gplus.to/fps_community

Archimesh

В FPS №22 '13 мы писали о генераторе окон Window Generator. Аналогичный инструмент Archimesh предоставляет еще большую функциональность, позволяя генерировать сразу целые комнаты, квартиры и дома – с дверными и оконными проемами, колоннами, лестницами и кровлей. Последние версии Archimesh даже позволяют создавать некоторые стандартные предметы мебели и интерьера: книжные шкафы, этажерки, жалюзи и т.д. Все генерируемые объекты создаются с автоматической разверткой UV-координат. Аддон также генерирует материалы Cycles.

Дополнение идеально подходит для «конвейерного» дизайна, когда нужно быстро получить базовый шаблон интерьера для последующей индивидуальной обработки. Его, кстати, многие используют в паре с Window Generator – для получения более качественных окон.

Автор: Антонио Васкес

<https://github.com/Antonioya/blender/tree/master/archimesh>

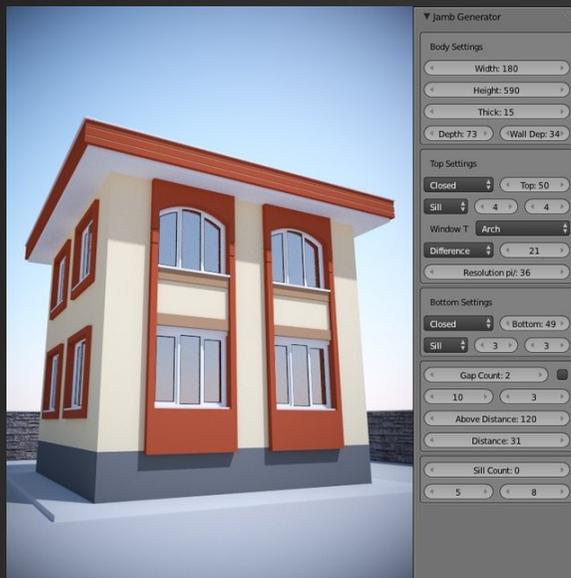


Jamb Generator

Генератор наличников и оконных проемов от создателя Window Generator. В отличие от WG, позволяет редактировать параметры объекта в любое время, а не только при создании! Идеальный инструмент для использования совместно с Window Generator и Archimesh.

Автор: Сеит Али Япракчи

https://dl.dropboxusercontent.com/u/23340323/Jamb_Generator.py

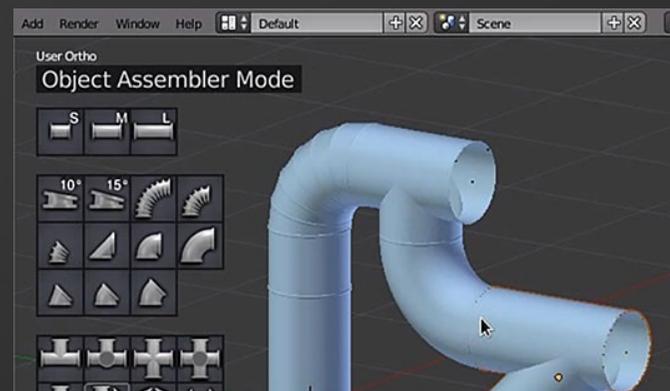


Object Assembler

Этот аддон реализует в Blender библиотеку префабов (prefab) – готовых стандартных объектов, из которых можно быстро собирать новые сцены. Объекты могут иметь материал, скелет, силовые поля и другие поддерживаемые в Blender свойства. Object Assembler также позволяет автоматически выравнивать их, что хорошо проиллюстрировано на примере коммерческой библиотеки с комплектом труб.

Автор: Николаус Моравитц

<http://www.emptygraphics.com/scripts/details/1/>



Floor Generator

Генератор паркетных досок. Обычно паркет делается путем наложения на плоскость пола соответствующей повторяющейся текстуры, но в таком случае невозможно добиться мелких дефектов, изменений цвета или каких-либо других деталей, уникальных для определенной доски или участка пола. Данный скрипт решает эту проблему путем генерации геометрии паркета: каждая доска представлена отдельным объектом, на который вы можете наложить любую текстуру.

Автор: Мишель Андерс

<https://raw.github.com/varkenvarken/blenderaddons/master/planks.py>

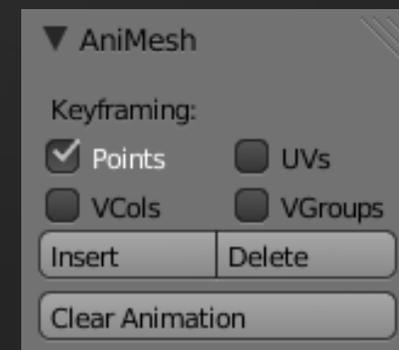


AnimAll

Полезное дополнение, позволяющее анимировать все виды данных модели – вершины, группы вершин, UV-координаты, а также кривые и сплайны. AnimAll входит в состав стандартных аддонов Blender, но отключен по умолчанию.

Автор: Даниэль Салазар

<http://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Animation/AnimAll>



GIMP: НОВОСТИ

GIMP 2.8.10

Представлен корректирующий выпуск GIMP 2.8.10. В новой версии внесено 11 изменений, в основном связанных с устранением ошибок. Одновременно выпущена новая версия руководства по GIMP, в котором устранены выявленные недоработки.

Из значимых изменений можно отметить

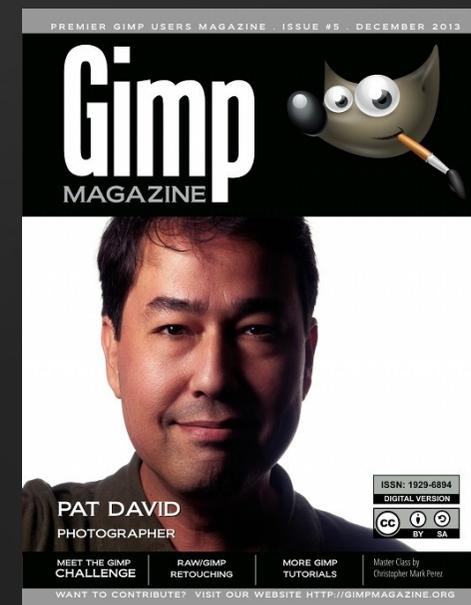
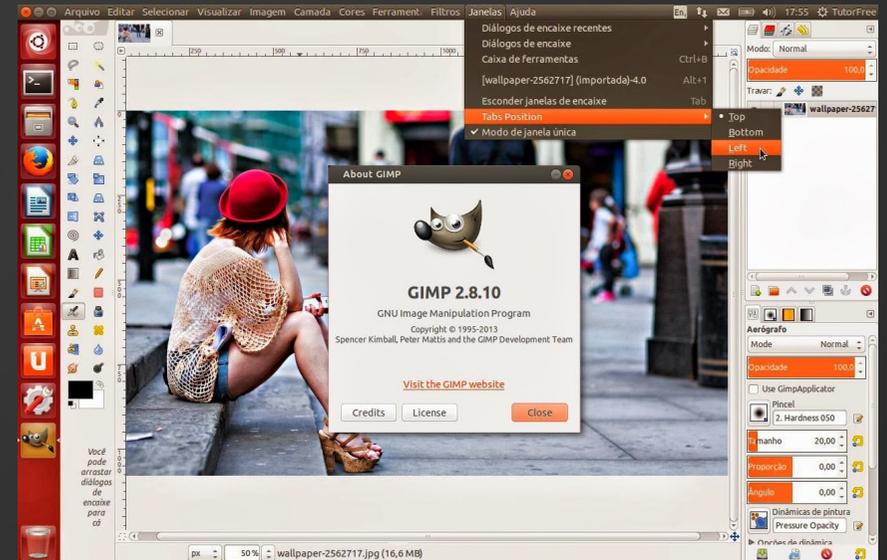
- Решение проблем при запуске в окружении Mac OS X 10.9 (Mavericks)
- Устранение ошибок компиляции в NetBSD
- Внесение незначительных корректировок, упрощающих сохранение и экспорт: в диалог завершения работы добавлен индикатор экспорта, переименованы метки в меню экспорта для соответствия меню «Сохранить» и «Сохранить как».

GIMP Magazine #5

Состоялся выход пятого номера замечательного англоязычного журнала GIMP Magazine. Журнал, как всегда, радует интересным материалом, красочными иллюстрациями и качественным дизайном. В данном выпуске вы найдете интервью с фотографом Патриком Дэвидом, статью от уже знакомого нам Кристофера Марка Переза, урок по ретушированию портретных фотографий, обзор программы TuxPaint, галерею художественных работ и многое другое.

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 распространяется по лицензии CC-BY-SA.

Подробности на сайте проекта:
<http://gimpmagazine.org>





Язык D

Новости «с Марса» свежие релизы и обновления

DMD 2.065

Вышла новая версия референсного компилятора DMD 2.065. Релиз носит, в основном, исправляющий характер - крупных изменений и нововведений в языке нет. Добавлен новый спецификатор доступа `const inout`, добавлена поддержка срезов для встроенных в язык кортежей, улучшена взаимная совместимость различных модулей в Phobos - в частности, функции из `std.algorithm` теперь можно использовать в качестве предикатов в других функциях.

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

LDC 0.13.0-alpha1

Состоялся выход альфа-версии LDC 0.13.0 - компилятора D, использующего LLVM для генерации машинного кода. Релиз основан на фронтенде D2 2.064.2 и бэкенде LLVM 3.1-3.4. Релиз включает официальную поддержку ARM/Android (добавлен `version`-идентификатор «Android»), хотя и без поддержки Phobos на этой платформе. Также компилятор теперь способен предупреждать о неявных обращениях к рантайму, что полезно при написании кода под ARM, где вы не можете полагаться на Phobos/рантайм.

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

dlib 0.2.4

Коллекция библиотек `dlib` обновилась до версии 0.2.4. Релиз носит исправляющий характер, улучшена поддержка 64-битных архитектур, добавлена поддержка пакетных модулей в DMD 2.064.

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

Dgame 0.3.2

Фреймворк для создания игр Dgame обновился до версии 0.3.2. Dgame - это набор инструментов для разработки 2D-игр, основанный на SDL и OpenGL. Дизайн фреймворка продолжает идеи таких проектов, как Pygame для Python и SFML для C++.

<http://dgame-dev.de/>
<https://github.com/Dgame/Dgame>

DSFML

Анонсирован DSFML - биндинг к популярной мультимедийной библиотеке SFML (Simple and Fast Multimedia Library). Представляет собой объектно-ориентированный аналог SDL, содержит средства взаимодействия с оконной системой, вывода графики и звука, работы с сетью.

<https://github.com/Jebbs/DSFML>
<http://www.sfml-dev.org/>

shplib.d

Анонсирован биндинг к библиотеке Shapefile для работы с одноименным популярным векторным форматом хранения географических данных. «Шейп-файл» позволяет хранить следующие различные геометрические объекты: точки, линии, полигоны и др.

<https://github.com/craig-dillabaugh/shplib.d>

Cerealed

Появилась альтернатива таким решениям, как Orange и std.serialization - библиотека двоичной сериализации данных Cerealed. Кроме простых типов данных, она поддерживает строки, простые и ассоциативные массивы, а также структуры.

<https://github.com/atilaneves/cerealed>

Vibe.d 0.7.18

Веб-фреймворк Vibe.d обновился до версии 0.7.18. Этот релиз обеспечивает совместимость с DMD 2.064, включает улучшения сетевой подсистемы, HTTP-сервера, системы сериализации.

<http://vibed.org>

Трассировка стека для D

backtrace-d - библиотека, реализующая обратную трассировку стека в D под Linux. Это отладочное средство, которое выводит последовательность вызовов функций и позволяет установить, в какой функции произошла неполадка.

<https://github.com/yazd/backtrace-d>

DStep

DStep - это транслятор заголовочных файлов C и Objective-C в D для автоматической генерации биндингов. Для анализа кода программа использует Clang. Есть версии для Mac OS X, Linux и FreeBSD, исходники распространяются по лицензии Boost.

<https://github.com/jacob-carlborg/dstep>

Поддержка D в ACE

В веб-редакторе кода ACE появилась поддержка D. Реализована подсветка синтаксиса, поддерживаются UDA и встроенный ассемблер.

<http://ace.c9.io>

DUB 0.9.21

Менеджер пакетов DUB обновился до версии 0.9.21. Из наиболее значимых нововведений этого и ряда предыдущих минорных релизов следует отметить поддержку кэширования при сборке и улучшение генератора проектов VisualD. Также в DUB теперь есть встроенный локальный HTTP-сервер для выдачи DDOC-документации напрямую в браузер, без явного создания HTML-файлов. Также проделано несколько важных улучшений в синтаксисе и семантике JSON-файлов конфигурации.

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

DVM 0.4.2

DVM (D Version Manager) - система для автоматической установки и управления различными версиями компиляторов D - обновилась до версии 0.4.2. Проект переписан на D2, появилась поддержка 64-битных платформ, исправлены баги.

<https://github.com/jacob-carlborg/dvm>

Digger

Вопреки названию, этот проект не имеет никакого отношения к культовой игре Digger эпохи DOS :) На самом деле, это утилита для сборки определенных ревизий DMD из репозитория.

<https://github.com/CyberShadow/Digger>

DCD 0.2.0

Сервер автозаполнения кода DCD обновился до версии 0.2.0. Это, в основном, исправляющий релиз: улучшена производительность, снижено потребление памяти, исправлено 28 багов.

Напомним, DCD - это демон автодополнения, совместимый практически со всеми IDE и текстовыми редакторами, где есть поддержка плагинов или скриптов. На данный момент DCD работает с Textadept, Kate/KDevelop, Vim, Emacs и Zeus.

<https://github.com/Hackerpilot/DCD>

DDT 0.9.0

Вышла новая версия DDT 0.9.0 под кодовым названием «Debugging is Magic». Главным новшеством релиза стала поддержка отладки при помощи GDB. Кроме того, исправлено несколько багов.

Напомним, DDT (D Development Tools) - это плагин поддержки D для среды разработки Eclipse. Проект распространяется по лицензии Eclipse Public License.

<http://code.google.com/p/ddt/>
<http://www.eclipse.org/>

Mono-D 1.5.2

Mono-D - плагин для среды MonoDevelop/Хamarin Studio, предоставляющий поддержку языка D в этой IDE - обновился до версии 1.5.2. Этот и ряд предыдущих релизов обеспечивают поддержку DMD 2.064, включают улучшения парсера, рефакторинг автодополнения кода и многие другие исправления.

<http://mono-d.alexanderbothe.com>

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/cook>
<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>



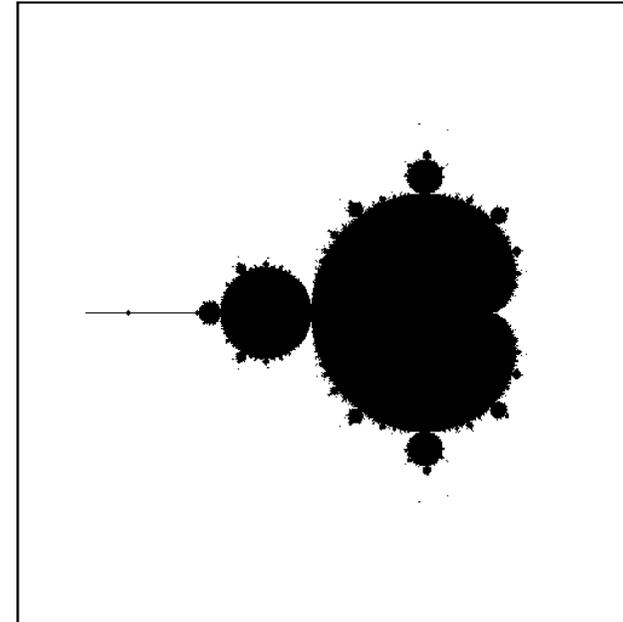
Рисуем фрактал на D

Фрактал – это геометрическая фигура, обладающая свойством самоподобия, то есть, составленная из нескольких частей, каждая из которых подобна всей фигуре целиком. В математике фракталами называют множества, имеющие дробную метрическую размерность. К примеру, размерность линии – 1, плоскости – 2, объемного пространства – 3. У фрактала же значение размерности может быть между 1 и 2 или между 2 и 3. К примеру, фрактальная размерность скомканного бумажного шарика приблизительно равна 2,5.

Для самоподобных множеств размерность может быть вычислена при помощи следующей формулы. Если множество разбивается на n частей, подобных исходному множеству с коэффициентами r_1, r_2, \dots, r_n , то его размерность S является решением уравнения $r_1^S + r_2^S + \dots + r_n^S = 1$.

Один из самых популярных фрактальных объектов – множество Мандельброта (названо по имени французского математика Бенуа Мандельброта, который ввел понятие «фрактал»). Определение множества Мандельброта звучит следующим образом. Это множество точек z на комплексной плоскости, для которых итерационная последовательность $z_{n+1} = z_n^2 + c$ при $z_0 = 0$ не уходит в бесконечность.

На человеческом языке это звучит так: берем точку на плоскости. Эта точка у нас будет комплексным числом, имеющим мнимую и действительную части, соответствующие двумерным координатам точки. Возводим это число в квадрат и прибавляем к результату исходную точку. Снова возводим полученное в квадрат и опять прибавляем исходное число. И так повторяем много раз. Смысл в следующем: исходная точка принадлежит множеству Мандельброта в том случае, если, при бесконечно большом количестве итераций этого алгоритма, результат по модулю не обращается в бесконечность.



Теперь вопрос в том, за сколько итераций мы можем в этом убедиться – ведь не повторять же алгоритм бесконечное количество раз :)

Доказано, что всякое комплексное число, модуль которого больше 2, не принадлежит множеству Мандельброта. Следовательно, мы повторяем вышеописанный алгоритм, пока модуль нашего результата меньше, чем 2. При реализации алгоритма необходимо отметить, что количество итераций нужно ограничивать – ведь если точка принадлежит множеству, то количество итераций для нее будет равно бесконечности, и мы получим бесконечный цикл.

Код к статье написан на языке D, с использованием библиотеки `dlib` (<http://github.com/gecko0307/dlib>).

```

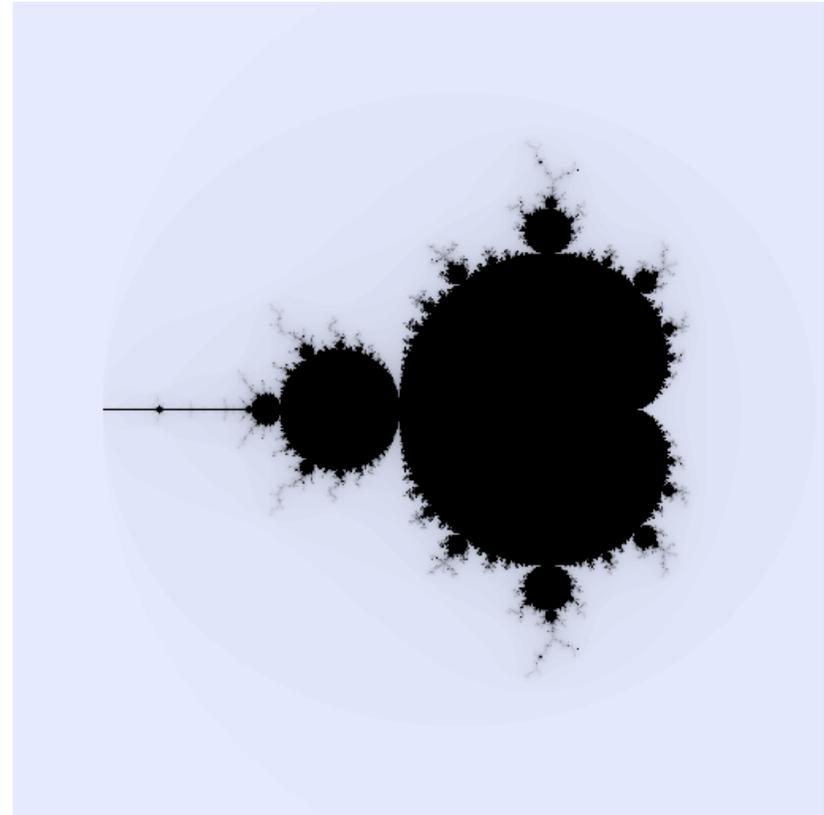
import dlib.image;
import dlib.math.complex;

uint mandelbrot(Complex!float c, uint N)
{
    uint i = 0;
    Complex!float z;

    for (z = c; abs(z) < 2.0f && i < N; i++)
        z = z * z + c;

    return i;
}

```



Функция `mandelbrot` возвращает количество итераций, за которое результат обращается или не обращается в бесконечность. Если оно равно `N`, то точка принадлежит множеству (если на каком-то этапе, меньшем `N`, элемент последовательности по модулю стал больше 2, то точка множеству не принадлежит).

Теперь алгоритм построения множества становится очевиден: мы создаем пустое изображение, проходимся по всем пикселям и проверяем, попадают ли соответствующие им точки в множество. Если попадают, рисуем черные пиксели, если нет – белые. При этом необходимо масштабировать координаты точки, чтобы полученный рисунок имел подходящий растровый размер.

```

void main()
{
    auto img = image(512, 512);

    uint maxIter = 100;

    auto origin = Complex!float(356, 256);
    float scale = 150;

    foreach(x; 0..img.width)
    foreach(y; 0..img.height)
    {
        auto point = Complex!float(x, y);
        uint i = mandelbrot((point - origin)
            / scale, maxIter);
        float v = maxIter - i;
        img[x, y] = Color4f(v, v, v);
    }

    img.save("mandelbrot.png");
}

```

Таким образом, можно получить черно-белое изображение множества, которое и было получено Мандельбротом. Чтобы сделать его цветным, можно, например, каждую точку не из множества красить в цвет, соответствующий номеру итерации, на котором ее последовательность вышла за пределы двойки.

```

void main()
{
    auto img = image(512, 512);

    uint maxIter = 100;

    auto color = Color4f(0.9f, 0.92f, 1.0f);

    auto origin = Complex!float(356, 256);
    float scale = 150;

    foreach(x; 0..img.width)
    foreach(y; 0..img.height)
    {
        auto point = Complex!float(x, y);
        uint i = mandelbrot((point - origin)
            / scale, maxIter);
        float v = cast(float)(maxIter - i)
            / maxIter;
        img[x, y] = Color4f(color * v);
    }

    img.save("mandelbrot.png");
}

```

Тимур Гафаров

Обзор альтернативных ОС

Согласно данным NetApplications на февраль 2014 года, самой популярной операционной системой на десктопах является Windows 7 (47,53%). За ней идут непотопляемая Windows XP (28,53%), Windows 8 (10,68%), Mac OS X (7,68%), Windows Vista (2,1%) и, наконец, Linux (1,48%).

Совсем иная ситуация в секторе мобильных устройств: здесь уже давно лидирует Android (79%), с громадным отрывом ему уступают iOS (14,2%), Windows Phone (3,3%) и BlackBerry (2,7%).

На рынке серверов неизменно лидируют ниски: различные разновидности Linux, BSD и др. (67,4%). На втором месте – серверная редакция Windows (32,6%). Среди крупных суперкомпьютеров и мейнфреймов доля Linux и вовсе приближается к 99%.

Интересна ситуация на рынке игровых консолей: Windows и Linux занимают примерно одинаковую долю (29,6%). При этом доля Linux относится к PlayStation 3, а Windows – естественно, к Xbox. Остальные 40,9% приходятся на эксклюзивное ПО от производителей. Впрочем, с ростом популярности SteamOS от Valve весы должны качнуться в сторону Linux.

Но все это – лишь сухая статистика, которая ничего не говорит нам о десятках альтернативных операционок, которые живут своей жизнью и успешно развиваются в тени своих более крупных собратьев. Давайте познакомимся с ними поближе!

GNU/Hurd

Ричард Столлман не устает твердить о том, что систему Linux следует именовать GNU/Linux, так как основная часть операционной системы на базе Linux – это системное окружение GNU плюс ядро Linux. Причем сам Столлман считает Linux неким временным решением, пока не будет разработано собственное ядро GNU – с теми самыми блекджеком и сами-знаете-кем :) Система на базе этого ядра – GNU/Hurd – уже существует и даже прекрасно работает на архитектурах x86, однако уже много лет находится в состоянии, далеком от стабильной версии.

В отличие от Linux, Hurd имеет микроядерную архитектуру. Оно представляет собой набор небольших автономных программ – серверов и программ пользователя, которые обмениваются между собой сообщениями. Основа Hurd – микроядро GNU Mach – обеспечивает только взаимодействие этих программ между собой. Особый класс серверов – трансляторы – обеспечивают возможность работы с файловыми системами, сетевыми протоколами, контролируют доступ к файлам.

Существует версия Debian на основе GNU/Hurd – регулярно выпускаются новые сборки и CD-образы этого дистрибутива. Также в 2010 году появился новый дистрибутив на базе GNU/Hurd и Archlinux – Arch Hurd.



Haiku

За этим поэтичным названием скрывается проект, созданный поклонниками легендарной операционной системы BeOS, популярной в начале 90-х. После того, как разработка BeOS была официально прекращена, энтузиасты решили возродить проект в виде новой совместимой с ней ОС, написанной полностью с нуля. Изначально она называлась OpenBeOS, потом OBOS, и, в конце концов, Haiku.

Имя проекта было выбрано в честь сообщений об ошибках, выдаваемых браузером NetPositive: они представляли собой нерифмованные японские трехстишия – хайку.

Haiku воплощает в себе основные идеи BeOS. Это модульная система, архитектурно решенная как гибридное ядро: микроядерная архитектура, способная динамически подгружать необходимые модули. Сетевой стек работает в режиме ядра, но, в целях двоичной совместимости с BeOS R5, имеет также интерфейс в пользовательской области памяти. Одной из отличительных особенностей системы является архитектура трансляторов – системных интерпретаторов файловых форматов, например, JPEG.

В настоящее время под Haiku портировано множество свободных программ. Под нее есть даже Qt, что позволяет писать для Haiku современные бизнес-приложения.



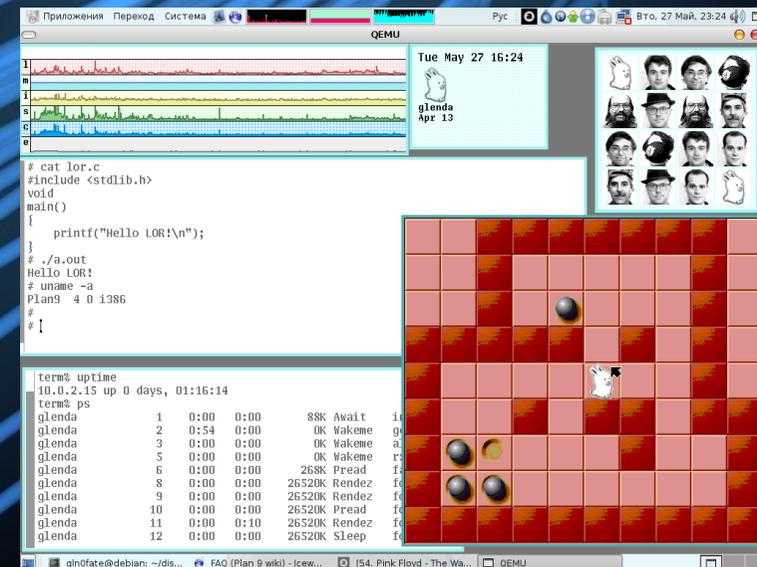
Plan 9

Эту систему разработали в Bell Labs лично отцы-основатели C и UNIX: Кен Томпсон, Деннис Ритчи и Брайан Керниган. Однако это вовсе не очередная UNIX-система, как можно было ожидать: проект направлен на переосмысление некоторых концепций и преодоление принципиальных недостатков дизайна UNIX.

Plan 9 основан на парадигме UNIX о том, что «все является файлом». Все ресурсы в системе представлены как файлы и доступны в иерархической файловой системе. Причем локальные и удаленные ресурсы не различаются. Все устройства – это тоже файлы. Программы могут осуществлять ввод-вывод текста через устройство /dev/cons, вывод графики через /dev/draw, получать события мыши, читая устройство /dev/mouse и т.д. Можно получить доступ к FTP и HTTP-серверам посредством чтения файлов – например, /net/http.ru.wikipedia.org

В отличие от UNIX с его пресловутым X11, в Plan 9 графическое окружение напрямую интегрировано в систему. Лежащая в его основе модель взята из системы Oberon Никлауса Вирта.

Еще одна особенность: в Plan 9 не существует супер-пользователя root, как нет и SUID-программ. Все административные задачи разрешено проводить только напрямую человеком-администратором из консоли.



Inferno

Фактически, продолжатель идей Plan 9. Inferno обладает межплатформенной переносимостью и может выполняться как пользовательское приложение или как независимая операционная система.

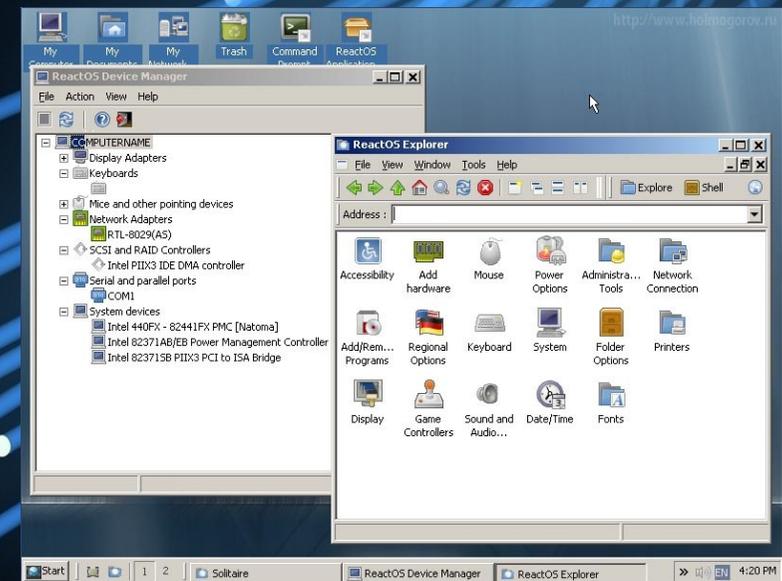
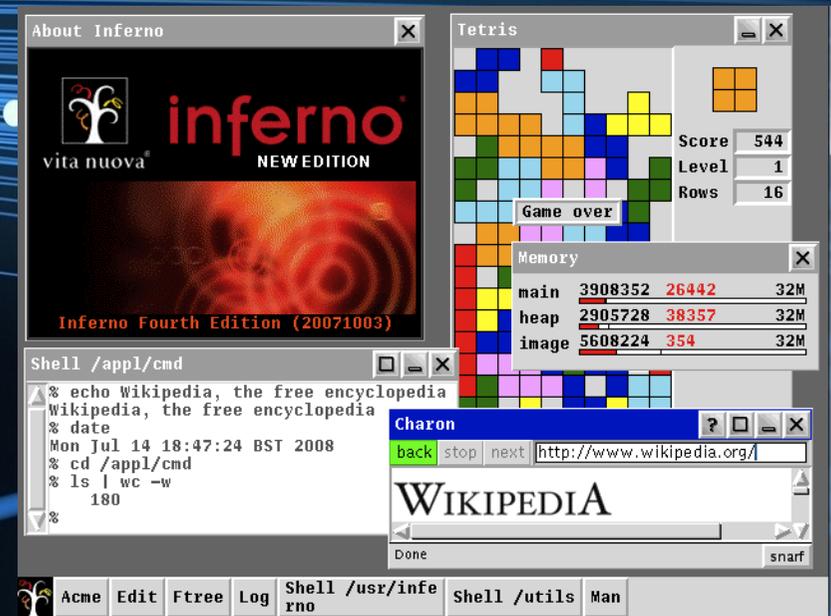
Как и Plan 9, эта система предлагает полную прозрачность ресурсов и данных, применяя систему именованного пространства. Ресурсы представляются как файлы, для управления которыми применяется один стандартный коммуникационный протокол - 9P, который получил здесь название Styx.

Приложения в среде Inferno пишутся на языке Limbo, который является модульным параллельным языком программирования с C-подобным синтаксисом.

Кстати, названия «Styx», «Inferno», «Limbo» и т.д., а также название компании-разработчика - «Vita Nuova» - выбраны под влиянием «Божественной комедии» Данте, которую читал Роб Пайк во время работы над Inferno в Bell Labs.

ReactOS

Свободная реализация MS Windows. Ядро системы уже стабильно и реализует большую часть API, система позволяет запускать простые Windows-приложения: как поставляемые с Windows (Блокнот, Regedit, cmd), так и сторонние (например, WinRAR, AbiWord, OpenOffice.org). Поддерживаются даже некоторые игры: например, в режиме программного рендеринга прекрасно работают Quake, Quake II, Unreal Tournament и Deus Ex. Ведется работа над улучшением поддержки звуковых карт и USB. Есть поддержка файловой системы Ext2 и системного загрузчика GRUB, что позволяет ReactOS работать на одном компьютере с Linux.



Genode

Это универсальная операционная система, совместимая со многими различными ядрами, в том числе Linux, Fiasco, OKL4, Codezero и NOVA. Для Genode осуществлено портирование различных компонентов Linux и BSD, обеспечена поддержка Gallium3D, Qt4 и WebKit, реализована возможность организации гибридных программных окружений.

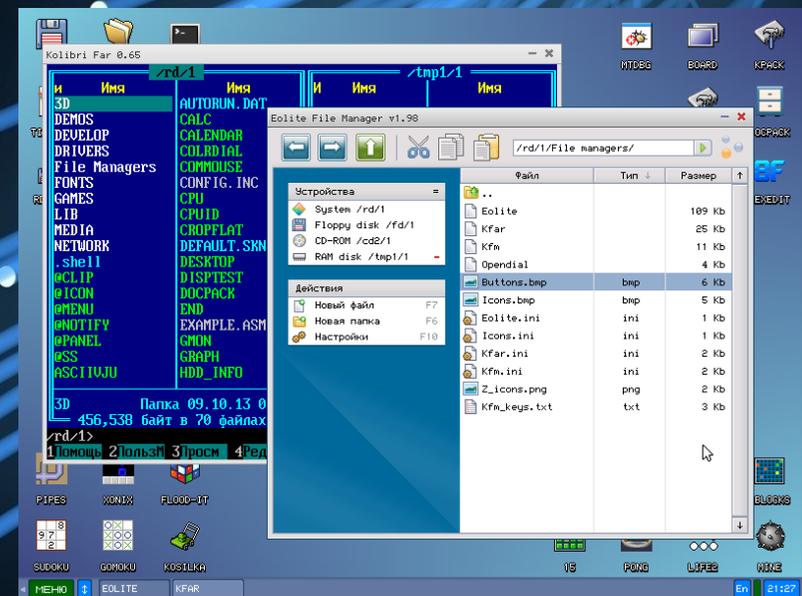
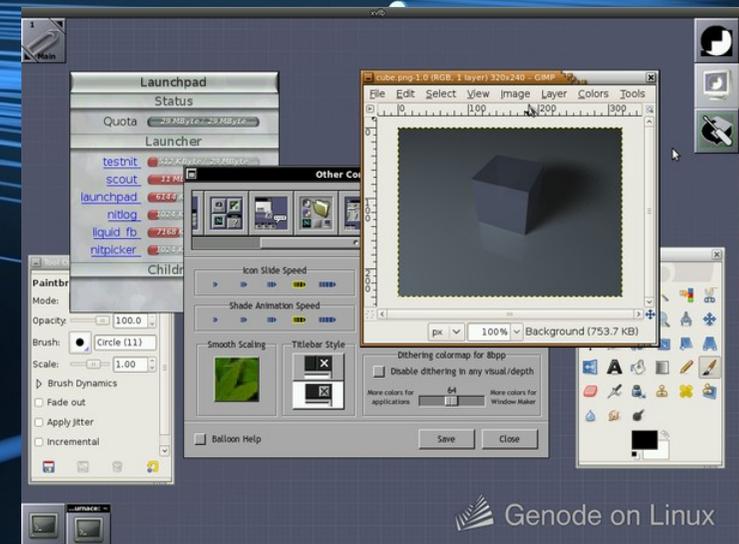
Fiasco

Fiasco – это микроядро нового поколения, поддерживающее современные архитектуры процессоров (все версии ARM, а также 32 и 64-битные варианты x86). Ядро масштабируется от малых встраиваемых систем до суперкомпьютеров. Fiasco используется, например, в системе osFree – проекте по созданию свободного клона IBM OS/2. Из-за названия Fiasco в среде русскоязычных разработчиков системы закрепилось неофициальное название «Проект Беда» – по аналогии с известным мультфильмом и книгой о приключениях капитана Врунгеля.

Kolibri

Минималистичная ОС, полностью написанная на ассемблере. Главная особенность Kolibri – предельно малый размер: основной дистрибутив занимает 1,44 Мб и помещается на одной дискете! При этом в системе есть полноценный GUI на основе VESA с Рабочим столом и оконной системой, среда разработки с текстовым редактором и макроассемблером, включена поддержка файловых систем FAT12, FAT16, FAT32, NTFS (в экспериментальном состоянии – также Ext2, Ext3, Ext4), поддержка звуковых кодеков AC'97 и Intel HD Audio для большинства популярных чипсетов, поддержка USB, TCP/IP-стека и некоторых сетевых карт. Впечатляет? Еще бы!

Также под Kolibri портированы сторонние программы – эмуляторы DOS, NES, ZX Spectrum, игры Quake и Doom. Для запуска системы достаточно всего 8 мегабайт оперативной памяти...





Кому нужен Linux?

Правда и вымыслы

Лет 10 назад начался «бум Linux» на десктопах: с появлением дружественных к пользователю дистрибутивов вроде Ubuntu и Mandriva, не требующих глубоких познаний в архитектуре Unix для установки и эксплуатации, систему начали осваивать не только программисты, сисадмины и хакеры, но и все, кому не жаль свободного времени. Появилось множество сайтов с доступной рядовому пользователю документацией «на человеческом языке», множество простых руководств и готовых рецептов. О Linux стали чаще писать в компьютерных журналах и блогах: эпоха Web 2.0 породила настоящую Linux-субкультуру с собственными мемами, анекдотами и прочим фольклором.

Сегодня, в эру высокоскоростного Интернета, это самая доступная ОС из всех существующих: достаточно скачать любой дистрибутив из сотен вариантов на любой вкус, записать на болванку и загрузиться. И его даже необязательно устанавливать навсегда – режим LiveCD позволяет запустить рабочую ОС прямо с диска, что превращает Linux в идеальное средство аварийного восстановления данных на рухнувшей системе.



И, само собой, на сегодняшний день под Linux гораздо больше софта, нежели 10 лет назад. К сожалению, скорость развития системы сейчас сильно опережает реакцию интернет-сообщества: до сих пор о Linux ходит много нелепых мифов, все еще встречаются статьи десятилетней давности с устаревшей информацией. Многим кажется, что система еще не созрела для «серьезного использования». Я неслучайно взял это в кавычки: Linux вполне серьезно используется многими крупными западными компаниями и госорганами: на ней построены сервера, суперкомпьютеры, мощные инженерские рабочие станции, рендер-фермы, различные терминалы и развлекательные платформы. И только российским офисным ракообразным из «шарашкиных контор» не хватает MS Office, из чего они делают свои «авторитетные выводы», засоряя русскоязычную Сеть бестолковыми мнениями и разного рода кривотолками. Возьму не себя смелость опровергнуть некоторые из них

Заблуждение 1. Linux бесплатен, а все бесплатное по определению не может быть качественным

Бесплатность у большинства ассоциируется с тем самым сыром в мышеловке. В случае с проприетарным софтом, естественно, бесплатность – это всего лишь один из вариантов рекламы: вы зачастую можете пользоваться некой функциональностью бесплатно, но вам при этом навязывают коммерческие услуги, «расширенные версии», «премиум-аккаунты» и прочее. Ключевое слово здесь – «можете пользоваться». Проприетарное ПО всегда принадлежит его владельцам, а вы покупаете (либо получаете бесплатно) лишь право на использование – и то со множеством ограничений.

В мире Linux бесплатность – лишь вершина айсберга. Нужно, в первую очередь, помнить о том, что Linux относится к свободному ПО, которое не имеет владельца в имущественном понимании этого слова. Получая копию свободной программы, вы получаете не только право на ее использование, но также право на изучение, модифицирование и передачу другим лицам. И, естественно, никто не заставляет брать за это деньги – в большинстве случаев пользователь платит только за носитель и доставку, а не за само ПО. Вы можете получить дистрибутив бесплатно, а можете купить – есть много платных Linux-систем, цены на которые включают техподдержку, например, Red Hat Enterprise Linux.

Система существует по принципу общечеловеческого достояния, как, например, наука: вы платите за учебники, но сами научные знания не продаются и не покупаются – вы можете получить их бесплатно. И, как и в научных исследованиях, в разработку Linux вкладываются крупные суммы денег: это делают компании, напрямую заинтересованные в улучшении этой системы, которую они используют для внутренних нужд или на основе которой выпускают коммерческие продукты.

Linux появился не по «щучьему велению», а в результате многомиллионных вложений и долгих лет работы профессионалов, и его бесплатность для конечного индивидуального пользователя – это не приманка корпораций, и не альтруизм энтузиастов, а вполне ожидаемый итог всего проекта в целом. Linux существует и развивается для всех и благодаря всем.

Заблуждение 2. Под Linux нет Photoshop и 3ds Max, поэтому он не годится для профессионального дизайна и художественной работы

Это заблуждение не имеет под собой никаких других оснований, кроме того, что так называемые «профессиональные дизайнеры» привыкли пользоваться Photoshop и 3ds Max. Самое смешное в том, что крупные киностудии и дизайнерские лаборатории покупают рабочие станции на том же Linux и пользуются внутренними программами, написанным нанятыми программистами: все VFX, которыми изобилуют нынешние блокбастеры, создавались при помощи софта, который вы никогда не увидите в магазинах. А «фотошопы» и «максы» лицензируют, в основном, рядовые студии и мелкие типографии, не обладающие достаточным бюджетом для создания собственной программной инфраструктуры. Решения на основе Windows – это, по сути, ширпотреб, призванный покрывать нужды большинства, а в большой индустрии ширпотребу не доверяют.

Кстати, в последнее время наметилась тенденция к открытости профессиональных решений: Pixar и другие гиганты открывают один внутренний инструмент за другим. И это не просто щедрость, а способ ведения бизнеса: открытые программы проще развивать и улучшать. Blender, который сегодня можно смело ставить в один ряд с 3ds Max, родился именно как внутренний инструмент в коммерческой анимационной студии.

Заблуждение 3. Linux – сырой, так как на нем не работает мое оборудование

Такое мнение часто высказывается перешедшими с Windows или Mac OS. Такие пользователи привыкли к готовым аппаратно-программным решениям – для них «железо» и софт неотделимо связаны. И это вполне объяснимо: производители компьютеров предустанавливают на свою продукцию лицензионную Windows со всеми нужными драйверами и специализированным ПО, а у Apple и давно операционная система своя собственная. В таких замечательных условиях любому может показаться, что весь софт мира должен быть совместим с его «железом». Но это не более чем потребительская иллюзия: Linux просто выводит пользователя из его привычного мирка, заботливо построенного корпорациями. Если бы те же корпорации взяли в качестве встроенного решения Linux, ситуация была бы в точности такая же.

На самом деле, мир компьютеров «с обратной стороны дисплея» гораздо более сложен, хаотичен и жесток: ведь если вы не никому ничего не заплатили, то никто вам ничего и не должен. В том числе – заботиться о поддержке вашего оборудования. Хотите поддержки – платите, иначе быть попросту не может. В этом отношении Linux «сырой» в той же мере, как и вся капиталистическая система – но ведь коммунизма вам никто и не обещал!

Подводя итог: Linux – это не то же самое, что Windows, это сущность другого уровня. Если коротко: Windows – это «покупай и используй», а Linux – «бери бесплатно, нанимай программистов, доделывай, как тебе надо, и используй». Есть те, кому надо первое, а есть те, кому надо второе. И бывает много ситуаций, в которых второе более оправданно с точки зрения бизнеса. Потребительское отношение к Linux неприменимо, зато вполне применимо отношение производственное.



Игровые новости из мира Linux

Состоялся выход первой публичной бета-версии операционной системы SteamOS от Valve. В качестве основы SteamOS использована пакетная база Debian GNU/Linux 7.1. По мнению Valve, дистрибутив Debian является оптимальным вариантом для создания собственных продуктов, адаптированных для решения специализированных задач.

Напомним, SteamOS – это операционная система на основе Linux, созданная специально для развлекательных устройств, таких как игровые консоли и смарт-телевизоры. Valve использует ее в консоли Steam Machine и предлагает сторонним производителям в качестве бесплатно лицензируемого встроенного решения для своих продуктов. Обычные пользователи также могут установить SteamOS на персональные компьютеры. В ОС встроен клиент сервиса Steam, что, вероятно, делает данный дистрибутив Linux самым дружественным к геймерам!

Для загрузки доступны два установочных образа: SteamOSInstaller.zip (960 Мб) с классическим инсталлятором от проекта Debian и SYSRESTORE.zip (2.4 Гб) с уже развернутым архивом системы, пригодным для прямого копирования на USB-носитель для загрузки с него.

В системе задействовано ядро Linux 3.10 с патчами для обеспечения работы в режиме реального времени. Используются стандартные для Linux компоненты графического стека (Mesa, X.Org, X Server) и проприетарные драйверы NVIDIA. При этом в настоящее время работа возможна только на системах с видеокартами NVIDIA – из коробки GPU от Intel и AMD пока, к сожалению, не поддерживаются.

Системное рабочее окружение основано на GNOME 3.4. В качестве графического режима по умолчанию задействован Big Picture, интегрированный в клиент Steam. В дистрибутиве используется обычный клиент Steam для Linux, не отличающийся от варианта, поставляемого для Ubuntu.



В состав входит типичный для многих дистрибутивов набор консольных утилит и компонентов – в том числе wget, vim, perl, python, openssh, usbutils, util-linux, coreutils. В качестве web-браузера поставляется Iceweasel (форк Firefox). Дополнительные программы можно установить, подключив штатные репозитории Debian.

Также SteamOS снабжена системой автоматической установки обновлений. Обновления загружаются с использованием APT из специально подготовленного репозитория.

Что особо важно: пользователю не запрещается получение root-доступа. Для этого необходимо перейти в системное графическое окружение, в терминале установить пароль (по умолчанию вход под обычным пользователем производится без пароля) и выполнить sudo.

Также радует, что все компоненты операционки, за исключением клиента Steam и проприетарных драйверов, доступны в виде исходных кодов.

Системные требования: современный процессор с архитектурой x86_64, 4 Гб памяти, жесткий диск 500 Гб, видеокарта NVIDIA, материнская плата с поддержкой UEFI.

<http://store.steampowered.com/steamos>



А в мире мобильных устройств по-прежнему царит Android: эта ОС начинает уверенно завоевывать рынок портативных консолей. Не так давно компания NVIDIA объявила о начале продаж консоли Shield, работающей на Android и построенной на базе процессора Tegra 4.

Консоль оснащена 5-дюймовым сенсорным экраном 1280x720 с поддержкой мультитач. Дополнительно поддерживается подключение внешнего дисплея или телевизора через порт HDMI. Заряда аккумулятора хватает на 5-10 часов при выполнении игровых приложений или на 24 часа при просмотре видео.

Для установки предлагается более ста игр, специально оптимизированных для Shield, которые распространяются через каталоги Google Play и NVIDIA TegraZone. Из Google Play также можно установить любые приложения, доступные для Android - в Shield используется обычное системное окружение Android.

Отдельный интерес вызывает функция PC Streaming, превращающая приставку в беспроводной игровой контроллер, позволяющий играть в игры, выполняемые на стационарном компьютере.

Стоимость приставки составляет \$299.

Благодаря поддержке Valve и NVIDIA, развитие графического стека Linux сейчас идет быстрее, чем когда-либо. Так, в Mesa – свободной реализации OpenGL – достигнута совместимость с OpenGL 3.2. Первым MESA-драйвером с поддержкой OpenGL 3.2, стал драйвер для GPU фирмы Intel. Реализации OpenGL 3.2 мешала неполная поддержка геометрических шейдеров и языка описания шейдеров GLSL 1.50, которая теперь доведена до полноценного вида.

Поддержка распространяется на чипы Ivy Bridge и более поздние – старые чипы типа Sandy Bridge пока предоставляют только поддержку OpenGL 3.1, так как для них реализация геометрических шейдеров готова только частично. То же самое касается и драйверов Radeon и Nouveau.

Проблема отставания свободных GPU-драйверов от проприетарных наталкивает многих разработчиков на мысль о полностью открытом графическом ускорителе. И эта идея уже недалеко от реализации: компания Silicon Spectrum объявила о намерении разработать полностью свободный графический процессор, поддерживающий механизмы 2D и 3D-ускорения и совместимый с OpenGL и Direct3D. Все схемы, дизайн платы, испытательный стенд и полную реализацию на языке Verilog планируется опубликовать под лицензией LGPL v3.

Функциональность GPU будет зависеть от суммы, которую удастся собрать на Kickstarter. Если будет собрано 200 тысяч долларов, в разряд открытых будет сразу переведена уже разработанная плата с полноценной поддержкой 2D-ускорения, PCI-интерфейсом и CRT-контроллером для вывода через DVI/HDMI. Если размер собранных средств составит 400 тысяч долларов, то плата будет расширена поддержкой базового 3D-ускорителя, совместимого с OpenGL и Direct 3D.



Культовые игры: The Elder Scrolls V

The Elder Scrolls V: Skyrim была выпущена в 2011 году Bethesda Softworks как пятая часть легендарной серии игр The Elder Scrolls. Спустя почти 3 года, игра не сдает своих позиций – она является одной из любимых RPG у геймеров во всем мире. Что же такого захватывающего в мире Skyrim?

Skyrim – это огромный мир, неограниченная свобода действий и множество квестов. Можете проходить их, стать во главе Коллегии Магов, быть главарем Гильдии Воров и даже оборотнем – а можете оставаться в одном из маленьких городков провинции и просто попивать эль в таверне. Мир очень большой, с необычными местами и интересными персонажами...

В Skyrim есть все: и болота, над которыми лежит густой пеленой туман, и горные вершины, покрытые снегом, и степи, на которых кого только не встретишь... Отдельно хочется отметить плавную перемену погоды, когда постепенно вместо дождя и туч над землями Скайрима появляется солнце. А точнее не солнце, а Магнус. Это не звезда, как мы привыкли, а... отверстие в «небесной тверди» – в этой вселенной все по-другому.

Планета, на которой происходит действие игр серии, называется Нирн. Она окружена черной пустотой Обливиона, за толщей которого находится Этериус – «верхний мир» богов. Звезды, которые видны с Нирна, это отверстия в Обливионе: чем больше звезда, тем больше разрыв. Магнус, соответственно – самый большой из таких разрывов, сквозь который льется божественный свет Этериуса. Вокруг Нирна также вращаются две луны – Массер и Секунда – и восемь планет, на которых живут Принцы Даэдра, могущественные сверхъестественные существа, которых обычно причисляют к богам.



Холодный Скайрим заселяют норды – они являются коренным народом этих земель. Скайрим – провинция великой Империи, которую населяют множество рас. Но не все его жители согласны с законами Империи: после смерти верховного короля Скайрима начинается гражданская война между Имперцами и Братьями Бури, которые сражаются за независимость Скайрима. Их главный девиз: «Скайрим для нордов!..»

В начале игры вам предстоит выбрать расу своего персонажа. Каждая из них обладает своими уникальными способностями и навыками. Здесь выбор велик: норды, каджиты (люди-кошки), орки, несколько видов эльфов и другие. В зависимости от выбранной расы, к вам меняется отношение. Например, при выборе эльфа, стражник к вам обратится со словами: «Что ты забыл здесь, эльф?» Их высказывания меняются в зависимости от игровой ситуации – если вы подхватили какую-то болезнь, они заметят, что ваше состояние неважное.

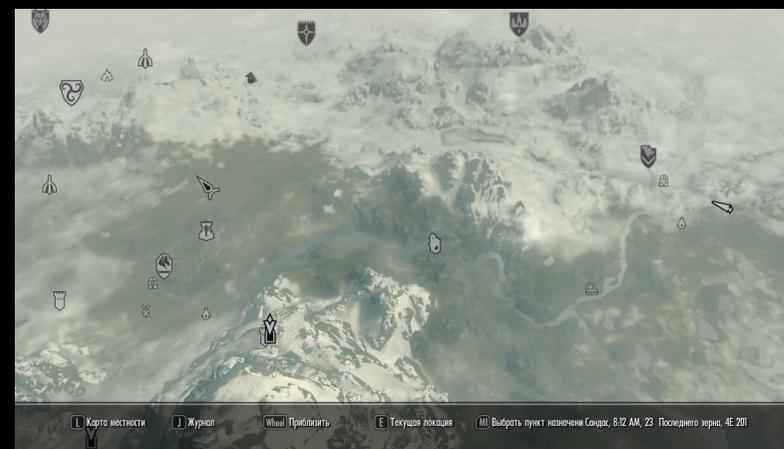
Сюжетная линия развивается с самого начала. Вас вместе с другими мятежниками везут на казнь в крепость Хелген, где уже поджидают палачи. Однако во время казни на Хелген нападает дракон, который, можно сказать, спасает вас от гибели в самую последнюю минуту. Вам удастся бежать, и тут предлагается выбор: идти с Ралофом, членом Братьев Бури, или имперцем Хадваром. Это воистину судьбоносное решение, от которого зависит, на чьей стороне вы будете в начавшейся гражданской войне. Да, в Скариме – море политических интриг!..



Потом оказывается, что вы являетесь Драконорожденным или Довакином – единственным, кто может убить дракона, поглотив его душу. Вас отправляют к старцам Седобородым, которые обучат вас искусству Крика – способности использовать в бою особые заклинания на языке драконов...

В современном мире TES 5: Skyrim стал уже легендой, в нее играют много раз – чего только стоят созданные в огромном количестве плагины и моды. Этой игрой живут, а не просто играют. В ней можно провести множество часов – причем каждую минуту вас ждет какое-то новое открытие и приключение. Я прохожу в нее уже во второй раз и старюсь придерживаться другого стиля игры, других ориентиров – благо, выбор здесь огромен. В Skyrim можно играть вечно!..

По слухам, шестую часть игровой серии следует ожидать только к 2017 году. Остается только коротать время за любившимся всем Skyrim и ждать нового чуда от разработчиков...



Интересные факты

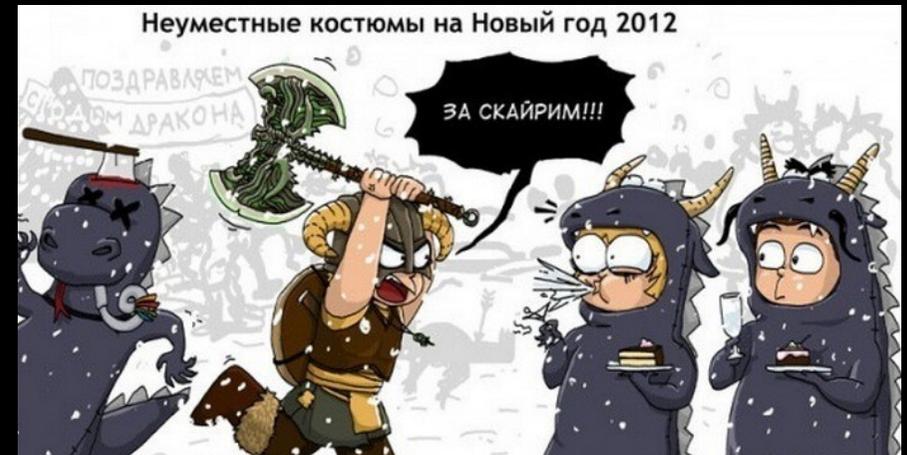
- Игра была признана самой покупаемой в истории сервиса Steam. За первые 48 часов с момента начала продаж *Skyrim* разошлась тиражом в 3,5 миллиона копий и поставила рекорд по одновременному количеству игроков.



- В одном из учебных заведений Канады занимаются... географией Скайрима. А в одном американском университете для студентов кафедры английского языка станет обязательным курс под названием «Мир скандинавского фэнтези: древние северные саги и *Skyrim*». Как поясняет декан факультета, этот курс познакомит учащихся с северной мифологией через понимание культуры игровой вселенной *The Elder Scrolls*: студенты будут читать скандинавские саги и играть в *Skyrim*. Везет же некоторым!..



- Разработчики пообещали награду тем родителям, у которых 11 ноября (в день выхода *TES 5*), родится ребенок. Новорожденного нужно было назвать Довакином – в честь главного героя игры. Победителями конкурса стали американцы Меган и Эрик Келлермайеры – они смогли документально подтвердить, что 11 ноября у них родился сын, которого назвали Довакином Томом Келлермайером. Теперь родителей ждет награда – пожизненная подписка на игры Bethesda!



- Опытные разработчики, знающие все тонкости и секреты игры, проходят ее очень быстро. Джефф Браун сумел пройти основной квест за 2 часа, 16 минут и 30 секунд, а Сэм Бернштейн – за 2 часа, 16 минут и 10 секунд. Не менее интересен и тот факт, что оба участника проходили игру Высшими Эльфами, которые к концу игры имели уровни 1 и 4 соответственно.

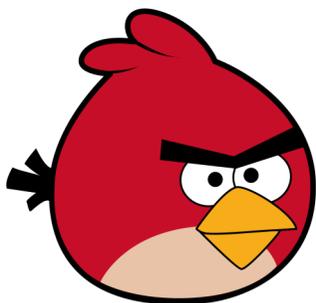
Наталья Чумакова

Angry Birds в Казани

Сердитые птички против сердитых родителей

В октябре прошлого года в Казани (городе, где находится редакция нашего журнала) побывал создатель всемирно известной игровой франшизы Angry Birds – Петер Вестербака. В 2011 году журнал «Time» включил его в число 100 самых влиятельных людей планеты. На встрече, которая состоялась в казанской ратуше, мэр города обсудил с совладельцем Rovio Entertainment возможность создания в городе парка развлечений Angry Birds!

Напомним, выпущенная Rovio Entertainment в 2009 году игра-головоломка Angry Birds за несколько месяцев завоевала мировой рынок и в настоящее время является самым скачиваемым приложением для мобильных платформ. После успеха Angry Birds компания стала быстро развиваться в индустрии развлечений и издательского дела. Так, наряду с созданием мультсериала по мотивам игры Angry Birds и открытием сети фирменных магазинов, одним из новых продуктов компании являются тематические парки развлечений «Angry Birds Activity Park». Суть концепции заключается в том, что персонажи, полюбившиеся детям в электронной игре, делают для них более увлекательными подвижные игры в реальном мире.



Первый парк площадью более 8 тысяч кв. метров был открыт в 2012 году в Финляндии. Позднее появились парки в Великобритании и Китае – на данный момент их насчитывается свыше двух десятков. В России есть лишь небольшая игровая площадка во Внуково, однако полномасштабного парка пока нет нигде.

Делегация Rovio осмотрела разные площадки в Казани, где можно было бы создать парк. Среди них – участок недалеко от парка Победы, а также рядом с Дворцом единоборств «Ак Барс» и комплексом «Ривьера».

Казалось бы – отличная идея. В рамках масштабной модернизации города, начавшейся с подготовки к Универсиаде 2013 года, такой парк сделает Казань еще привлекательнее в глазах туристов. Однако с этим, как ни странно, согласны далеко не все жители города. Некоторые родители-активисты считают, что детям с помощью подобных развлечений... «навязываются не свойственные российской культуре западные ценности».

Наталья Реснянская, мама троих детей 15, 8 и 2-х лет, генеральный директор сети детских садов «Егоза»: «У меня консервативное мнение. Эта перекидка западных стандартов, западных героев мне, как маме и руководителю детских учреждений, не нравится. У нас есть своя культура, свои герои, которых можно ничуть не хуже воплотить в парках. Баба Яга, Кощей, Василиса... Наши «Волк и семеро козлят» могут стать теми же самыми персонажами, как эти Angry Birds. Да, мои дети периодически смотрят, западают на какие-то вещи с яркими наклейками. Но я считаю, что нужно продвигать наших героев, которые нам ментально ближе. Потому что с приходом этих героев западных приходит и другая культура, которая нам не свойственна. А детям нужно передавать то, что создавалось веками нашими предками».

Яна Булыгина, мама двух девочек-школьниц, психолог, руководитель кабинета психологической помощи «Точка опоры»: «Вопрос в том, почему продвигаются чужие культурные ценности, когда наша родная культура даст сто очков вперед любой западной культуре? Да, я, конечно, понимаю все эти экономические обоснования, про инвесторов и про рентабельность. Но почему власти и бизнес считают, что парк развлечений в русском или татарском национальном стиле на основе наших народных сказок окажется менее привлекательным и прибыльным? С моей точки зрения, и горожане с детьми, и уж тем более туристы с большим интересом будут ходить именно в традиционный для данной местности парк развлечений. А парк развлечений по мотивам компьютерной игры – и довольно примитивной игры! – кажется мне, каким-то обезличенным, каким-то шаблонным что ли, туда ни разработчики, ни строители, ни бизнесмены душу не вкладывают. Меня такая ситуация с подменой культурных ценностей пугает потому, что дети начинают воспринимать чужую культуру как свою, она им становится ближе, чем родная».

О какой «ментальной близости» можно говорить, если речь идет о детях младшего поколения, выросшего уже далеко не на советских мультфильмах – для меня лично загадка. Кроме того, персонажи Angry Birds – не герои фольклора и мифологии, к культуре какой-либо конкретной страны они отношения не имеют, поэтому говорить о «навязывании западных стандартов» здесь тоже вряд ли уместно.



Наши ретивые общественники, как обычно, видят «подмену культурных ценностей» абсолютно во всех идеях и начинаниях, если эти идеи исходят не от таких же квасных патриотов, как и они сами. Все западное у них по определению «бездушно» и «бездуховно», а все русское – ценно и высоконравственно. Даже через 20 с лишним лет после развала СССР в нашем менталитете сильна мысль о «загнивающем Западе»: в то время, как Европа из оплота империализма давно превратилась в процветающее постиндустриальное либеральное общество, мы все еще «идем своим путем», всеми силами противясь разуму и прогрессу. Но куда ведет этот путь?..

В отсутствие главенствующей общественной идеи, понятной всем, люди хватаются за иллюзорные ценности. Коммунизм мы отменили, науку мы не понимаем, либеральные социально-философские концепции нам чужды. Но зато у нас есть «вековые традиции» и «истинно русская духовность»!

При этом сторонники «нашего, родного» забывают, что настоящие русские сказки имеют мало общего с теми, что знакомы им с детства – в оригинале в них царят жестокость, варварство и средневековые ужасы. Необходимо осознавать, что добрые и жизнерадостные советские мультфильмы и книги, на которых выросло теперешнее взрослое поколение – не что иное, как результат мощной цензуры и идеологической переработки. Если показать этим «патриотам» настоящую русскую народную культуру, они первыми выступят против того, чтобы воспитывать на ней детей!

Нельзя, впрочем, не согласиться, что современная массовая культура способствует развитию потребительского мышления и мало стимулирует творческое начало. Но в таком случае глупо надеяться на популярные компьютерные игры и другие коммерческие развлекательные продукты как на средство воспитания – ведь они создаются только ради извлечения прибыли. То же касается телевидения, поп-музыки, кассовых кинофильмов и много чего еще.



Но, в конечном счете, время расставит все по своим местам: что достойно называться культурными ценностями, а что нет – будут решать наши внуки, а не мы. Современники считали блюз, джаз и рок-н-ролл «дьявольскими песнопениями», боролись с ними как с каким-то злом, а теперь это серьезные общепризнанные музыкальные жанры, занимающие почетное место в мировой культуре. Во все времена новаторы в области культуры и искусства сталкивались с закоснелым консерватизмом и инертностью общественности. Можно, конечно, мерить все совковой линейкой и продолжать строить кондовые «ЦПКИО им. Горького» с деревянными статуями крокодила Гены, но не лучше ли поощрять креативность, открыться новым идеям и новой эстетике?

Нам, взрослым, самим не нравится, когда нам кто-то диктует, как одеваться, что смотреть, что говорить и как думать. Так почему же мы становимся «диктаторами вкуса» по отношению к детям? Вчерашние хиппи, панки, рок-н-рольщики и стилиаги, которым не давала проходу советская система, сами превратились в занудливых старперов, готовых запретить все, что для них непонятно и неприемлемо.

И есть ли смысл бояться массовых увлечений? Мода приходит и уходит, а высокая культура остается – но она непрерывно обогащается. Ведь глупо беспокоиться, что современные арт-объекты вытеснят из музеев шедевры античности. Но кто знает – может быть, через двести лет они будут стоять рядом!

Процессу культурного обогащения нельзя мешать только из страха перед новыми веяниями, иначе мы рискуем вовсе остановиться в своем развитии. А вклад в культуру сейчас вносят не только художники, композиторы и писатели – в наш век развитых коммуникационных возможностей культуру делают все люди, и участвовать в этом процессе может любой желающий. Именно на этой идее и нужно воспитывать детей, а не на бабушкиных сказках и «деревянных игрушках, прибитых к полу».

То, что у нас в детстве не было компьютеров и Интернета, не значит, что в этом есть какая-то «сермяжная правда», даже если нам кажется, что мы при этом были здоровее и умнее. Раньше, как известно, и трава была зеленее, и небо голубее. Напомню, что у наших прадедов не было, ко всему прочему, еще и электричества и водопровода, а более далекие предки и вовсе были вынуждены ютиться в пещерах. Нельзя ставить этот примитивизм себе в заслугу: добровольным аскетом в советские годы никто не был, все точно так же охотились за различными новинками и модными вещами.

Вместо того, чтобы отбирать у детей электронные игрушки, лучше расскажите им, как они устроены. Не нравится, что ребенок просиживает часами за игрой – так пусть игровая культура станет для него хобби: пусть изучает азы программирования, учится работать в графических редакторах, пусть пробует сочинять музыку, пусть придумывает свои собственные вселенные. Это громадный полигон для выявления профессиональных способностей: пусть он и не станет вторым Джоном Кармаком, но, может быть, найдет себя в науке и технике, живописи, дизайне, музыке, кинематографе, анимации, архитектуре, литературе, журналистике или экономике. И то же самое касается буквально всех областей массовой культуры – любое «бессмысленное» развлечение можно легко направить в нужное русло и превратить в полезную деятельность!



Но, к сожалению, большинство родителей на такую педагогическую революцию отнюдь не способны. Мы до сих пор воспитываем в детях бесполезную внутреннюю дисциплину, мешающую им мыслить свободно и широко. Предметы в школах преподаются настолько скучно и шаблонно, что науки и искусства не вызывают у учеников другой реакции, кроме как «Зачем мне это надо?..»

На этот вопрос учителя отвечают самым неудачным ответом, какой только можно придумать: «Это нужно, чтобы быть образованным человеком». Извините, но человек, считающий математику каррой божьей, вряд ли может считаться образованным, сколько бы знаний в него не впихивали!



Если бы вместо этого школьникам говорили, что наука – это очень интересно, что она используется во всех областях жизни, что в ней еще много неизведанного, что чуть ли не каждый день совершаются новые открытия, и что первооткрывателем может стать любой, кто захочет заниматься этим с душой и энтузиазмом – это звучало бы куда более вдохновляюще, не находите?

Поэтому, пока мы внушаем детям, что Истина только в учебниках, музеях и библиотеках, а в Интернете – сплошная ложь, мы ни к чему не придем. Творческий энтузиазм, которого нашей стране сейчас так не хватает, не вырастет на бесплодной почве этой устаревшей системы ценностей...

Тимур Гафаров

Хакеры без страха и упрека

Как создавались игры 80-х и 90-х

В наши дни перед разработчиками ПО не стоит никаких ограничений (исключая, разумеется, только границы их собственных знаний и умений) – компьютеры за последние 20 лет стали настолько мощны, что в большинстве случаев уже не нужно скрупулезно оптимизировать размер и производительность программ. Нынешний программист – это уже не хакер, ежедневно проявляющий чудеса ловкости и хитроумия, а просто специалист, знающий свой ЯП и предметную область. Большого не требуется: операционные системы, компиляторы, отладчики и рантаймы автоматически делают за человека львиную долю сложной работы.

Но так было не всегда. К хорошему привыкаешь быстро – и, к сожалению, многие приемы и хитрости, распространенные в ту эпоху, когда «640 килобайт хватало всем», сегодня уже практически забыты. И мы задаемся вопросом: как разработчики прошлых поколений умудрялись уместить полноценные игры со сложной графикой и музыкой в считанные килобайты?

Для примера: объем памяти картриджа Atari 2600 составлял 4 кб – сейчас в такой размер трудно уместить даже картинку сколько-нибудь приемлемого качества, не говоря уже о целой игре!

Начнем хотя бы с того, что сам подход к программированию игровой системы значительно отличался от теперешнего. Не было ни операционных систем, ни промежуточных средств типа OpenGL – все функции ОС и драйверов оборудования брала на себя сама игра. Программисты работали, как говорится, «на голом железе»: напрямую обращались к памяти и видеопроцессору. Каждое устройство имело свой определенный участок в памяти, и взаимодействие с устройством осуществлялось путем записи/чтения данных в этом участке. Компьютеры и сейчас работают таким же образом, просто современные операционные системы скрывают эти низкоуровневые детали под удобными интерфейсами. Программирование микроконтроллеров в нынешних встраиваемых устройствах, кстати, во многом похоже на то, что было в мейнстриме лет 30 назад.



Предтечами современных игровых движков были коллекции кода общего назначения, которые накапливались в игровых студиях. Зачастую этот код был общим у нескольких студий одного и того же издателя. Но о лицензировании стороннего middleware, конечно, никакой речи не шло.

Графический контент создавали кто как мог. В 80-х был популярным графический пакет DeluxePaint, изначально написанный для внутреннего использования программистами Electronic Arts для Commodore Amiga 1000, а затем портированный под разные другие платформы, включая DOS и Apple II. Редактор вскоре стал для Amiga тем же, чем сейчас является Photoshop для Mac и PC. Он, кстати, использовался в Lucasarts для создания знаменитого Monkey Island и ряда других игр. Последняя версия DeluxePaint (5.2) вышла в 1994 году.

Проблема оптимизации расходования памяти стояла перед разработчиками вплоть до конца 90-х. Популярная в те годы платформа PlayStation была оснащена всего 2 мегабайтами оперативной памяти. **Дэйв Багетт**, который в Naughty Dog был одним из программистов, работавших над легендарным Crash Bandicoot, вспоминает:

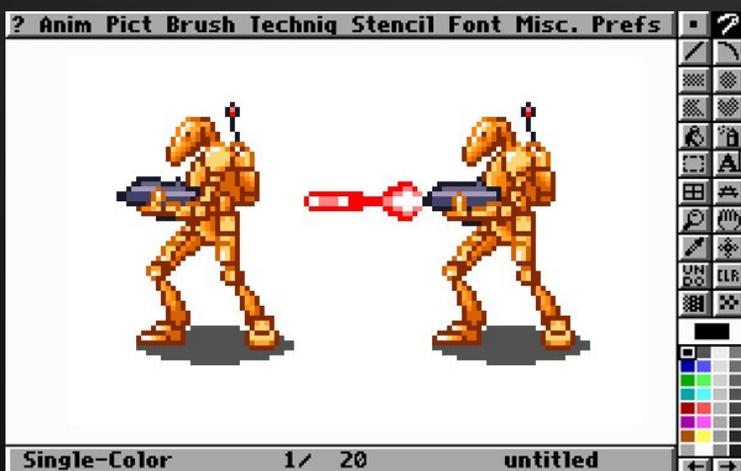
«Нам приходилось прибегать к сумасшедшим трюкам, чтобы уместить игру в эти рамки. Уровни занимали до 10 Мб, и данные должны были подгружаться динамически, без каких-либо заметных глазу «тормозов». Энди (Эндрю Гэвин, главный программист Naughty Dog – прим. ред.) написал удивительно эффективную систему подкачки, которая подгружала данные пакетами по 64 кб, по мере того, как Креш перемещался по уровню.

Это был просто шедевр: система охватывала управление памятью как на высоком уровне, так и на низком уровне DMA-команд процессора. Энди даже контролировал физическое расположение байтов на CD-ROM – так, что даже на скорости 300 кб/с PlayStation успевала загрузить нужную секцию уровня до того, как Креш доберется до нее.

А я написал упаковщик, который сжимал данные – звуки, графику, Lisp-код, управляющий поведением персонажей – в эти самые пакеты по 64 кб. Кстати, эта задача – идеальная упаковка данных произвольного размера в фиксированные контейнеры – является NP-полной, и поэтому она неразрешима за полиномиальное (то есть, разумное) время.

Мой упаковщик использовал ряд полиномиальных алгоритмов (Best Fit, First Fit и др.) для того, чтобы найти самый оптимальный вариант упаковки – включая стохастический поиск вроде градиентного спуска, который используется в алгоритме имитации отжига (simulated annealing). У меня было несколько разных стратегий упаковки, и я просто выбирал ту, которая давала лучший результат.

При этом часто уровни укладывались в максимально возможное количество пакетов (21) благодаря чисто случайному успеху стохастического упаковщика. Это значило, что, если приходилось изменить управляющий код для какой-нибудь черепахи, то приходилось искать стратегию упаковки заново. И так – всякий раз, когда художникам вздумается что-нибудь поменять. Поди объясни им это!..



Но, пожалуй, самой интересной и самой сложной частью работы было умещение кода на C/ассемблере. Буквально за несколько дней до выхода в печать мы занимались тем, что делали случайные перестановки конструкций, дающие одну и ту же семантику, но имеющие разный синтаксис – чтобы заставить компилятор выдавать код на 200, 150, 50 и 8 килобайт меньше. Например, в случае с циклом «for (i=0; i<x; i++)» – что будет, если переписать это в виде цикла «while», используя уже существующую переменную, которую ранее использовали для чего-то другого? И это уже после того, как мы исчерпали все обычные трюки – такие, как задействование под данные двух младших байтов у указателей (это работало, потому что адреса у процессора MIPS R3000 имели размер 4 байта).

В конечном счете, Crash умещался в памяти PlayStation с запасом в 4 байта. Да, всего 4 байта из 2097152. Вот это были времена!..»



Интересный факт: для игр Nintendo64 существовало строгое требование по времени загрузки уровня – около 5 секунд. Чтобы уместиться в этот отрезок, приходилось идти на различные ухищрения. Например, перед экраном загрузки появлялся экран предпросмотра уровня, на котором могла быть показана карта с отмеченными важными местами. Для начала загрузки пользователь должен был нажать на кнопку. Но собственно загрузка данных начиналась в момент появления этого экрана, поэтому, чисто психологически, казалось, что загрузка длится меньше.

Уважаемые читатели!

Наш журнал регулярно выходит на протяжении почти 6 лет – с января 2008 года. Все эти годы он оставался бесплатным изданием, предлагая публике эксклюзивный контент с минимумом рекламы. Мы всегда работали на совесть – не ради денег, а на благо наших читателей. «FPS» был и остается проектом энтузиастов и полностью независимым изданием – мы не защищаем интересы корпораций или политиков, мы пишем о том, что считаем нужным и важным. Мы стоим за свободу слова и творчества, за обмен информацией и знаниями: все материалы журнала можно беспрепятственно копировать, распространять и использовать в любых производных работах.

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

Наш WMR-кошелек: **R120156543694**

Адреса Bitcoin:

16PSGbj5foeqMN8isdoyiKvWYGM9V5idFk
16XaSt1U5eXWG7EAKuEMpFE6M6fPia5o4F
1PdNHTL5njsZJGHyNJ4c5xPW3eApEoB9pQ
1MFauzBqewUN8MWPY7DqGeLiJayxCqGBCL
1MSPN7TXuTPbuGjT1AQxex3ECjRgPhpFEA
1NB1xXoyJ71beEPpmmKioWMaon2uCigCeQ

Адрес PayPal:

gecko0307@gmail.com

Яндекс.Деньги:

410012052560079

Заранее благодарны!
Редакция

Каково быть гиком в тюрьме?

Исповедь Анонимуса

Я хакер, который пять лет назад отсидел половину от девятимесячного тюремного срока. Я побывал в двух тюрьмах, но провел большую часть срока во второй, с менее строгим режимом. Полученный опыт полностью изменил меня – но в положительную сторону.

Во-первых, мне помогло мое образование: я легче ориентировался в некоторых аспектах тюремной жизни – например, в правовой документации и в спорах с охраной. Мои навыки починки сломанной электроники быстро стали известны и помогли мне чувствовать себя в безопасности, потому что давали мне защиту. Благодаря им я чувствовал себя важным в этом обществе.

Все началось с того, что ко мне подошел один из заключенных и спросил, что я знаю о ремонте мобильных телефонов (в тюрьмах Великобритании у многих есть мобильники – их достают «подзаборной контрабандой»). Этот человек был очень важным лицом – у него была своя «банда», и люди часто приходили отдать ему деньги. Я сказал, что знаю достаточно о телефонах и спросил, чего он хочет. Он ответил, что кто-то ему задолжал, но не смог заплатить. Тогда он забрал у того парня телефон. Но аппарат был защищен PIN-кодом, и он не мог его разблокировать.

Это был старый Samsung, хорошо знакомая мне модель – когда-то у меня самого была такая. Я знал, что этот аппарат не накладывает ограничений на число попыток ввода PIN-кода. Я сказал: «Да, я знаю пару трюков. Но мне нужны мои инструменты, поэтому я заберу телефон на одну ночь».

Что самое смешное, у меня не было никаких инструментов. Мне оставили телефон, и я всю ночь перебирал все возможные 10000 комбинаций из четырех цифр. К счастью, код оказался в районе 2-й тысячи.

На следующий день тот парень вернулся и был очень удивлен, что я подобрал код. Он рассказал всем вокруг, что я «технический гуру», и чтобы люди приходили со своими проблемами ко мне. В обмен на мою услугу, он организовал доставку PlayStation 2 в мою камеру на две недели и доступ к телефону в любое время.

С тех пор люди приносили мне сломанную электронику, и я удалялся на весь вечер – со стороны казалось, что я делаю что-то сложное. Это значительно повысило качество моей жизни.

Во-вторых, этот опыт показал мне, как живут люди, которым повезло меньше, чем мне – и как страшна тюремная система для большинства людей. В тюрьме было много людей с психическими заболеваниями. Здесь они не получали никакой поддержки. Некоторые из них были убиты в тюрьме другими заключенными или персоналом, потому что окружающие боялись, когда у них случались припадки.

Еще одна группа – наркозависимые. «Авторитеты» эксплуатировали их, добывая наркотики контрабандным путем и создавая целую армию из наркоманов, которые были готовы выполнить любой их приказ ради очередной дозы. Это была очень взрывоопасная ситуация: практически все случаи насилия были связаны либо с долгами, либо с наркотиками.

Кроме того, пребывание в тюрьме навсегда изменило мое отношение к жизни. Я происходил из среднего класса, у меня было хорошее образование, но я увлекся взломом и был осужден за попытку подделки водительских прав – потеряв, таким образом, множество преимуществ, которые мне дали судьба, общество и семья. Другие заключенные вовсе не имели таких преимуществ: большинство из них не имели образования, росли среди нищеты, наркотиков и насилия – все это и привело их в тюрьму. Я оказался там, главным образом, из-за своей неуместной любознательности, а другие – из-за настолько невыносимых условий жизни вне тюрьмы, что преступление было для них единственным способом выжить. ..

И я сам себе опротивел. Освободившись из тюрьмы, я стал изучать программирование, работал фрилансером, защитил кандидатскую и теперь надеюсь использовать свои навыки так, чтобы улучшить жизнь других людей. Если случается, что мне лень что-то делать, я вспоминаю тюрьму, и отвращение к себе появляется вновь – я начинаю работать с удвоенной энергией.

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://qplus.to/fpsmag>