

24
2013

FPS

Проект «Лондон»
Интервью с режиссером

Трехмерная печать
Тренд года

Физический движок
своими руками

История игр

Демосцена

+ многое другое!



FPS

№24

FPS – бесплатный, свободно распространяемый электронный журнал, посвященный разработке компьютерных игр и сопутствующей тематике.

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-три месяца.

© 2008-2013 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: **Тимур Гафаров**
Обложка: **Алия Тягичева**

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

- **Авторы «FPS» – кто они?**
 - :: О тех, кто делает журнал для вас
- **Blender**
 - :: Новости
 - :: Проект «Лондон». Интервью с режиссером
 - :: Тренд года – 3D-печать
- **GIMP**
 - :: Новости
- **Язык D**
 - :: Новости
 - :: Физический движок своими руками. Часть I
- **Новости игровой индустрии**
 - :: Armikrog
 - :: Incredipede для Linux – бесплатно!
 - :: SDL 2.0
- **История компьютерных игр**
 - :: От Pong до PlayStation 4
- **Демосцена**
 - :: Алгоритм как произведение искусства...



Авторы «FPS» – кто они?

Краткая информация о тех, кто делает журнал для вас

Наш журнал существует уже 5 с половиной лет, но все это время его немногочисленный авторский состав оставался, так сказать, «в тени» – публичной фигурой всегда был только основатель и бессменный главный редактор **Геско**. Поэтому мы посчитали актуальным рассказать читателям подробнее о тех, кто так или иначе способствует созданию «FPS».



Тимур Гафаров

Геско

Должность: главный редактор, художник

Возраст: 23 года

Страна, город: Россия, Казань

Образование, место работы: закончил КХУ им. Н. И. Фешина по специальности «Живопись». В настоящее время учусь в казанском филиале МГАХИ (Суриковский институт). Работаю в казанской газете «Отражение» и альманахе «Пришелец»

Интересы: живопись, журналистика, философия, эзотерика

Умения и навыки: разработка игровых движков, 3D-графика, моделирование в Blender, компьютерная верстка

«У меня с детства была мечта издавать журнал. К сожалению, мир печатных СМИ сейчас переживает не лучшие времена – а Интернет позволил энтузиастам, подобным мне, опробовать свои силы в этой области, ничем не рискуя. Но «FPS» – это не просто информационное издание, это мой личный авторский уголок, место, где я могу высказывать свои мысли в удобной мне форме, делиться своими знаниями и взглядом на мир».



Фархад Гафаров

DarkGameWizard

Должность: корреспондент, глава рубрики игровых обзоров

Возраст: 18 лет

Страна, город: Россия, Казань

Образование: среднее

Умения и навыки: видеомонтаж, звукозапись, разработка 2D-игр.



Наталья Чумакова

Doriana

Должность: корреспондент, переводчик

Возраст: 20 лет

Страна, город: Россия, Казань

Образование: КФУ, факультет вычислительной математики и кибернетики

Интересы: книги, сериалы, музыка, всякая чепуха.)

Умения и навыки: математика, программирование, разработка веб-приложений в среде Ruby on Rails

«Будущий программист, либо математик, либо еще кто-то... Не знаю, что в итоге получится. Интересуюсь теорией игр и линейным программированием. До журнала "FPS" у меня не было особо каких-то публикаций, каких-то важных творческих успехов. Рада, что я этому изданию пригодилась и надеюсь, что наше сотрудничество продолжится. Желаю "FPS" долгих лет жизни, много-много интересного материала и благодарных читателей».



Александр Рыжов

FatPir

Должность: корреспондент

Возраст: 20 лет

Страна, город: Россия, Санкт-Петербург

Образование: СПбГУ, строительный факультет

Интересы: ММА, бокс, турники

Умения и навыки: программирование на Flash, веб-дизайн, разработка игр

Проекты: flash-игра **Jazz Matches**

«Пожалуй, статья в «FPS» осталась единственным творческим опытом. Недавно снова решил заниматься разработкой игр, планирую вскоре запустить свою первую игру на Android».



Леон Завальский

ZavLeon

Должность: корреспондент, глава рубрики «Фотолаборатория»

Возраст: 19 лет

Страна, город: Россия, Калининград

Образование: среднее, занимаюсь фрилансом

Интересы: фотография, обработка изображений, рисование

Умения и навыки: фотографирование, обработка изображений, дизайн, полиграфия

«О журнале впервые узнал на сайте <http://xtremezd.narod.ru>, мне понравилась его разносторонность, в том числе уроки по графическим редакторам. Я фотограф-любитель, использую GIMP для обработки снимков, недавно начал публиковать свои уроки в «FPS». Надеюсь, они кому-нибудь помогут».



Алия Тятигачева

fkbz

Должность: художник, автор обложек

Возраст: 21 год

Страна, город: Россия, Стерлитамак

Образование, место работы: окончила колледж искусств, работаю в мастерской рекламы

Интересы: традиционное рисование, CG и многое другое

Умения и навыки: дизайн интерьеров, логотипов, 3D-графика, моделирование в 3ds Max



Сергей Куманев

revfytd

Должность: автор мобильной версии журнала

Возраст: 29 лет

Страна, город: Украина, Днепропетровск

Образование, место работы: в 2007 году окончил УГХТУ

Умения и навыки: программирование на Java/Android, разработка игр, создание мобильных версий печатных и интернет-изданий, информационных сайтов и т. д.

Проекты: сервис электронных СМИ «**Мобильный киоск**»



Blender

Новости

Номера версий ветки 2.6x подходят к концу – и это значит, что нужно готовиться к очередному большому шагу в развитии Blender, каким в свое время стала 2.5x. До сих пор все релизы Blender были достаточно хорошо совместимы между собой, но будет ли так и дальше? Недавно Тон Розендаль озвучил стратегии относительно обозримого будущего программы.

Версии 2.68 и 2.69 сохраняют обратную совместимость и будут, в основном, сфокусированы на улучшение стабильности. Все потенциально нестабильные или нарушающие совместимость нововведения будут введены только в ветке 2.7. Разработчики не будут заботиться о том, чтобы файлы, созданные в 2.7x, могли быть прочтены в 2.6x или более старых версиях. Однако обратная совместимость является критической необходимостью для многих пользователей, поэтому ее нарушение будет происходить только в случае больших и важных улучшений.

Изменения также могут коснуться графического интерфейса, наименований команд и опций, темы оформления и дефолтных горячих клавиш. Также вполне возможно, что репозиторий Blender перейдет на использование Git, подобно многим другим современным OpenSource-проектам.

Параллельно с разработкой Blender 2.7x, будут выходить обновления и багфиксы для 2.69, чтобы миграция на новую ветку происходила как можно менее болезненно.

Возможные нововведения в 2.7x:

- Переход на OpenGL 2.1 для отрисовки интерфейса;
- Рефакторинг внутренних механизмов для улучшенной поддержки многопоточности;
- Менеджер ресурсов, новые инструменты для управления связями;
- Новый видеоредактор.

Ветка 2.8 будет предполагать наиболее серьезные изменения в программе. Работа над ней будет вестись одновременно с 2.7.

Возможные нововведения в 2.8x:

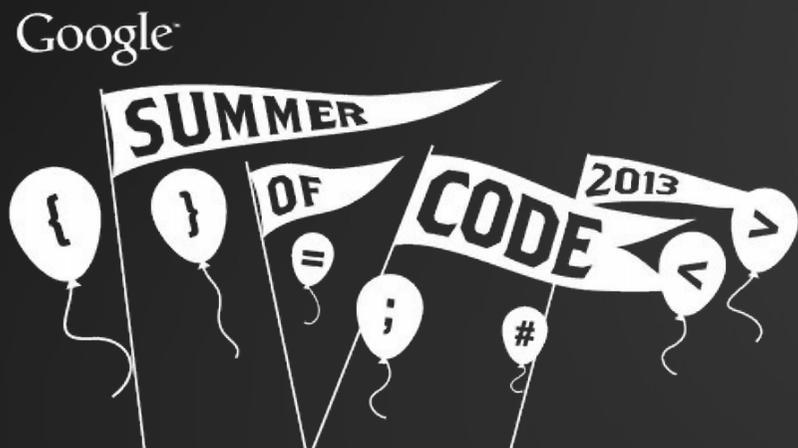
- Переход на OpenGL 3.x (под вопросом);
- Унифицированная физика, более тесная интеграция Bullet;
- Узлы для системы частиц;
- Поддержка узлов для некоторых других компонентов Blender (к примеру, модификаторов и ограничений);
- Полный рефакторинг игрового движка – улучшение Python API, игровой физики, обеспечение поддержки внешних движков. Приближение BGE по возможностям к таким продуктам, как Unity и UDK.

Касательно ветки 3.x пока мало что известно – можно только сказать, что в ней подвергнется рефакторингу само ядро Blender, написанное еще в 90-х и содержащее системы управления сценой, объектами и данными (система DNA и т.д.).

И буквально пара слов о ближайших планах. Согласно текущему расписанию, выпуск 2.68 запланирован на начало июля. Ветка «multiview» с улучшенной подготовкой стереоскопического рендеринга туда не попадет, но переход на OpenEXR 2.0 все же состоится.

Кстати, в ближайшее время произойдет серьезное обновление официального сайта Blender (<http://blender.org>) – планируется переход на новый движок, в качестве которого был выбран Wordpress.

В этом году Blender традиционно участвует в Google Summer of Code. В конце мая были озвучены 15 проектов GSoC – среди них можно отметить улучшение режима видеомонтажа (VSE), реализацию скетчинга, улучшение инструментов рисования, создание LOD-системы в игровом движке, улучшение системы отслеживания движений, симуляции твердых тел и многое другое.



Напомним, GSoC – это инициативная программа Google, в рамках которой ежегодно проводится отбор открытых проектов, в которых могут принять участие студенты. Проекты должны предложить будущим участникам задания по реализации новой или улучшению существующей функциональности. Победителям выплачиваются денежные гранты. В 2007 году в программе участвовало 900 студентов из 90 стран мира, которые внесли вклад в разработку 130 различных OpenSource-проектов.

Стали известны кое-какие подробности о планах участия Blender Foundation на предстоящей конференции SIGGRAPH 2013, которая пройдет с 21 по 25 июля в Анахайме (Калифорния, США). Тон Розендаль представит отчет о проделанной за год работе над Blender и расскажет о планах проекта на будущее, разработчики и художники со всего мира проведут демонстрации своих достижений.

Совместно с Pixar, Dreamworks, Sony Imageworks и ILM, представители Blender Foundation обсудят нынешнее положение OpenSource в индустрии трехмерной компьютерной графики. Подробности об этом событии, а также детальный обзор всех интересных новинок и разработок, представленных на SIGGRAPH 2013, ждите в следующем номере журнала.

Разработчик Жан-Франсуа Галлан представил бета-версию Molecular – реализацию мультифизики для системы частиц Blender, во многом напоминающую известный движок Lagoa Multiphysics. Мультифизика – это объединение математических моделей нескольких разных физико-химических механизмов в одной системе. Такой подход позволяет с высокой степенью реализма симулировать поведение и взаимодействие сыпучих и вязких веществ, несжимаемых жидкостей, разного рода разрушаемых материалов и т. д.



Molecular предоставляет возможности для взаимодействия частиц между собой и соединения их в связные структуры – наподобие того, как в реальном мире множество молекул образуют материю. Для получения максимального быстродействия, а также поддержки многопоточности, Molecular собирается при помощи Cython – статического компилятора Python, выдающего оптимизированный код на C. Благодаря этому движок работает в 30 раз быстрее, чем на обычном Python.

Доступны бинарные сборки Molecular для Windows, Linux и Mac OS X. Есть 32- и 64-битные версии. Исходники проекта размещены на GitHub.

<http://pyroevil.com/molecular-script-download/>

<https://github.com/Pyroevil/Blender-Molecular-Script>



В начале этого года состоялась премьера фильма «Проект «Лондон». Это полнометражный фантастический фильм режиссера Яна Хьюберта, известного нам по «Стальным слезам». Картину можно назвать настоящим прорывом в сфере независимого кино: это чуть ли не первый безбюджетный боевик голливудского качества, от начала до конца созданный силами энтузиастов при помощи свободных инструментов – полностью за счет пожертвований!

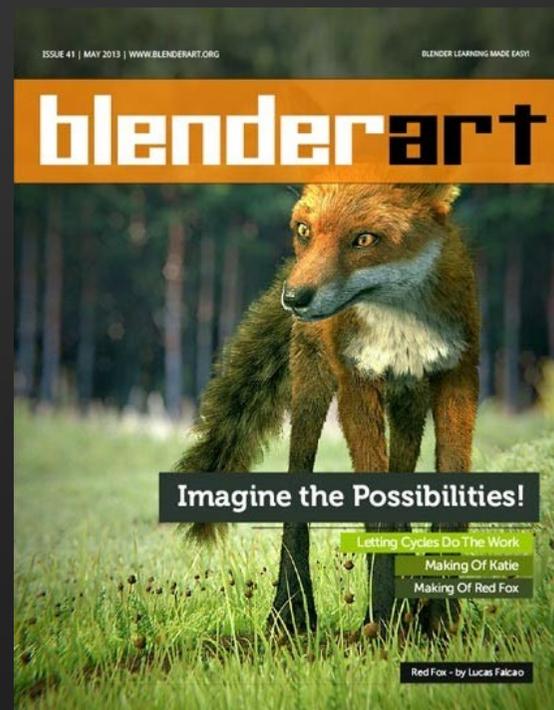
Руководство проекта – так называемый Триумvirат – расположен в Сиэтле (США). В этом же городе начинается повествование фильма.

Небраска Хиггинз живет обычной мирной жизнью, работая механиком в Контрольной Башне Объединенного командования – глобальной полиции и вооруженных сил Земли, совместно образованной людьми и расой налардианцев. «Добрые» пришельцы передали людям свои технологии, но взамен установили единое мировое правительство. Грозному Объединенному командованию противостоит лишь группа революционеров, к которым примыкает и Небраска – после того, как люди ОК убили его отца...

Фильм можно посмотреть на Vimeo за \$3.99, либо заказать на DVD или BlueRay.

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

В конце мая вышел 41 номер замечательного журнала «BlenderArt» – самого популярного интернет-издания, посвященного Blender. В этом выпуске читателей ждет много полезной информации: уроки, статьи жара «making-of», галерея работ и другие материалы.



Команда сайта **Blender Cookie** анонсировала проект по созданию «Blender Art Book» – альбома с лучшими работами, созданными при помощи этого пакета. В отборе работ могут принять участие все желающие. Проект отчасти благотворительный – 5% дохода от продаж книги пойдут в Blender Foundation.

На популярном сервисе вопросов и ответов StackExchange.com появился раздел, посвященный Blender – теперь находить интересующую вас информацию по тонкостям работы с пакетом станет еще удобнее.

<http://blender.stackexchange.com/>

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!



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

Спецпредложение!

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

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

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

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

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

Заявки на рекламу принимаются на почтовый ящик редакции: gecko0307@gmail.com (просьба в качестве темы указывать «Сотрудничество с FPS», а не просто «Реклама», так как письмо может отсеять спам-фильтр).

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

PROJECT LONDON



Интервью с режиссером
Яном Хьюбертом

Как появилась идея снять такой фильм?

Летом 2006 года я окончил школу и обдумывал планы на будущее. Вместе с отцом мы назначили встречу Филу МакКою, медиа-продюсеру компании Fortune 500.

Мы обсудили с ним карьеру в кино, подумали – а почему бы мне, вместо колледжа, просто не снять фильм? Фил сказал: «Если ты напишешь сценарий, в который я поверю, я смогу его спродюсировать». Затем последовало еще несколько встреч. Я переделывал сценарий чуть ли не десять раз и только потом представил его Филу. Он предложил несколько важных изменений, я переписал сценарий еще пару раз, и мы, наконец, получили в итоге нечто дельное!

Через пару месяцев сумасшедшего препродакшена (у нас даже не было полного актерского состава за неделю до съемок!) мы были готовы снимать. Съемки проходили с субботы по среду каждую неделю в течение августа 2007 года – так что люди с полным рабочим днем могли хоть изредка появляться на основной работе.

Довольно напряженный график...

Я скажу больше: это был самый напряженный период в моей жизни. Я выполнял роль сценариста и режиссера, не имея никакого опыта работы с профессиональными актерами. Каждый день на площадке находились 20-30 человек, и успех этого фильма – бесспорная заслуга 250 отличных ребят, которые все вместе шли к единой цели.

По современным голливудским стандартам, 650 визуальных эффектов для боевика – не такая уж беспрецедентная цифра. Куда необычнее, если речь при этом идет о фильме... без бюджета!

Мы имеем в виду «Проект «Лондон». Пост-продакшн к нему полностью был произведен добровольцами, сотрудничающими через Интернет – а спецэффекты были отрендерены не в каком-нибудь Autodesk Maya, Softimage или Goudini, а в свободном и бесплатном Blender. Решение использовать СПО, конечно, было вполне разумным для инди-проекта, но одновременно представляло определенный риск – до сегодняшнего дня Blender еще не был никем задействован в полнометражном кино: команда под руководством Яна Хьюберта, фактически, стала в этом первопроходцем. Не так давно сайт Jawbone.tv взял у него интервью, которое мы решили перевести и опубликовать на страницах «FPS».

Начнем с названия. Почему фильм называется «Проект «Лондон», если вы находитесь в США?

Верно подмечено! Почему так вышло – тайна, затерянная в веках... Я придумал это название несколько лет назад, специально выбрав нечто абстрактное – чтобы потом подобрать более подходящее название к картине. Но мы этого так и не сделали.



Сколько людей работало вместе с вами?

В течение нескольких лет численность команды по VFX варьировалась от 12 до 2 человек – все это были добровольцы, поэтому их вовлеченность в проект зависела от наличия свободного времени.

Это значит, что вам пришлось как-то их мотивировать – ведь они тратили на фильм свое свободное время, не получая вознаграждения. Насколько это было трудно?

Это была задача не из легких! Фактически, всего двое из команды по VFX работали над фильмом от начала до конца. Всего через нас прошло человек двадцать. У большинства из них были основные места работы, семьи – и нам пришлось подстраиваться под их расписание. Большая часть нашей работы основывалась на заданиях, которые мы распределяли между членами команды в зависимости от их доступности и умений. В целом, это сработало.

Так как происходило финансирование съемок?

«Проект «Лондон», в целом, – безбюджетный фильм. У нас не было инвесторов, и все то небольшое, что мы потратили, ушло на еду, костюмы, реквизит и другие мелочи.

Насколько был сложен пост-продакшн?

В общей сложности, было применено около 650 спецэффектов – гораздо больше, чем мы планировали во время съемок.

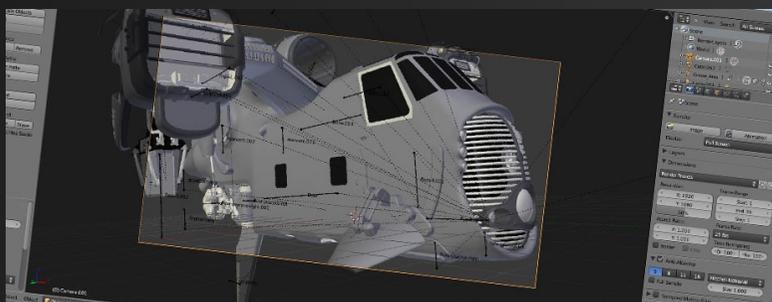
Удивляет тот факт, что вы не использовали «голливудский» софт. Почему вы выбрали Blender в качестве основного пакета для анимации?

Я познакомился с Blender несколько лет назад – через друга, который использовал его для создания видеоигры. Бесплатность Blender притягивает к нему, в основном, энтузиастов, изучающих CG для собственного интереса, а не для работы в киноиндустрии. Вокруг программы образовалось большое сообщество талантливых людей, любящих экспериментировать – именно оно и помогло нам справиться с пост-продакшном. Мы получили помощь от тех, кто просто хотел продемонстрировать свои умения, одновременно работая над интересным проектом.



Насколько нам известно, это крупнейший VFX-проект, когда-либо созданный при помощи Blender. Справилась ли программа со своими задачами?

Blender справился на отлично. Фильм был снят в HD, и, в среднем, время рендеринга одного кадра заняло 5 минут (для голливудской студии были бы приемлемы даже несколько часов, в зависимости от сложности работы – *прим. ред.*) Только недавно я начал осознавать всю мощь, которой обладает Blender.



На таком большом проекте не все работает «из коробки», поэтому голливудские студии нанимают программистов для улучшения и адаптации программного обеспечения. Сложно ли было найти разработчиков для этой задачи?

Открытая природа Blender дала нам высокую степень гибкости, когда появилась необходимость модифицировать код. Назову один случай: однажды я попросил Дольфа Винвлье (член совета сертификации Blender – *прим. ред.*) сделать возможность выделить все объекты на сцене без текстур – и буквально через 20 минут он выслал мне файл. Развитое сообщество значительно упрощает разработку в Blender – и мы весьма благодарны ему за это.

Оригинал интервью:
<http://www.jawbone.tv>

Blender

НАСТОЛЬНАЯ КНИГА

«Blender. Настольная книга» – это проект от журнала «FPS» по созданию полноценного русскоязычного электронного руководства по основам работы в Blender 2.6. 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 активно ведется подготовка текста книги.

К работе над книгой 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.



Тренд года: 3D-печать

Ажиотаж вокруг 3D-печати стремительно набирает обороты. Обама инвестирует в развитие этой технологии \$30 млн, «Боинг» печатает самолеты, а канадские инженеры – трехколесные гибридные автомобили. Медики создают протезы, искусственные органы и анатомические пособия, архитекторы и дизайнеры – оригинальные скульптуры и здания, модельеры и ювелиры – одежду, обувь, украшения и аксессуары. Если раньше эта технология была доступна лишь инженерам в их лабораториях, то в последние годы стало появляться множество фирм, предлагающих услуги трехмерной печати всем желающим, по самым демократичным ценам. Более того, в продаже появились 3D-принтеры, которыми можно пользоваться в домашних условиях. По всей видимости, мы стоим на пороге очередной индустриально-цифровой революции...

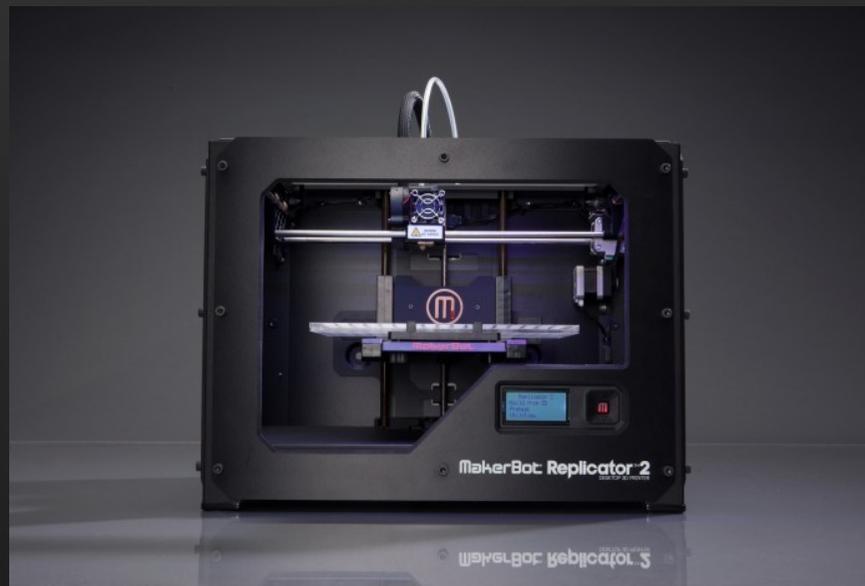
3D-печать осуществляется разными способами и с использованием различных материалов, но в основе любого из них лежит принцип послойного создания (выращивания) твердого объекта. При этом применяются две технологии:

- Лазерная. Ультрафиолетовый лазер постепенно, пиксель за пикселем, засвечивает жидкий фотополимер (или сплавляет порошок), который сразу затвердевает и превращается в достаточно прочный пластик;
- Струйная. Раздаточная головка выдавливает на охлаждаемую платформу-основу капли разогретого термопластика. Капли быстро застывают и слипаются друг с другом, формируя слои будущего объекта.

Самый простой способ изготовления моделей для трехмерной печати – 3D-сканирование. Недорогим 3D-сканером, в сущности, является игровой контроллер Kinect, который разработали в Microsoft для консоли Xbox 360.

А некоторые методы 3D-сканирования даже не требуют специального устройства – достаточно программного средства, которое воссоздаст объемную модель на основе фотографий реального предмета. Например, мобильное приложение 123D Catch, созданное в Autodesk, позволяет скомбинировать в трехмерную модель серию снимков, сделанных с помощью обычного смартфона.

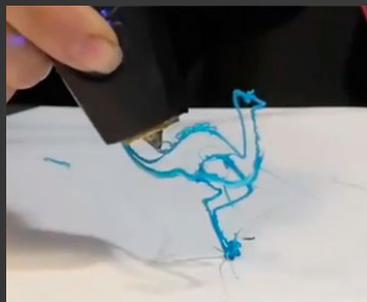
Недавно нью-йоркская компания MakerBot, чьи недорогие «настольные» 3D-принтеры пользуются успехом по всему миру, продемонстрировала на конференции SXSW прототип недорогого 3D-сканера под названием Digitizer. С его помощью можно отсканировать существующий предмет, получить трехмерную модель и тут же отпечатать ее на 3D-принтере MakerBot. Возможные применения очевидны – таким образом, к примеру, можно чинить сломавшиеся устройства. Сломалась деталь – скопируй целую и поставь ее взамен!



Свои собственные модели для 3D-принтеров можно создавать в современных CAD-системах. В Интернете уже накапливаются целые архивы пригодных для распечатки объектов – как отсканированных, так и созданных вручную. Кстати, в последней версии Blender были реализованы инструменты для подготовки данных к трехмерной печати.

При этом размер производимых объектов не имеет никакого значения. «Большие» принтеры будут собирать целые жилые многоквартирные дома, а нанороботы – объекты микроскопического масштаба (например, интегральные микросхемы). Ученые университета Южной Калифорнии уже разработали систему Contour Crafting, способную за 20 часов напечатать двухэтажный жилой дом.

Это огромный принтер, похожий на башенный кран с компьютерно-программным управлением, не требующим участия человека. Он способен самостоятельно выполнить до 90% всех строительных работ, людям здесь останется лишь внутренняя отделка помещений. В 2014-15 годах США планирует наладить серийное производство подобных «строительных принтеров».



Ручкой zDoodler можно рисовать трехмерные объекты... прямо в воздухе. Пластик застывает практически мгновенно, что позволяет рисовать без опоры, на весу. Кроме того, можно рисовать на любой поверхности – рисунки легко отделяются от бумаги, а к пластику достаточно прочно «привариваются». Одного метра исходного материала хватает больше чем на 10 метров готовой линии. Сама ручка весит меньше 200 граммов, есть две скорости «печати» – для быстрого заполнения больших участков и для тонких работ. Нетрудно представить, какие перспективы это устройство открывает для художников – размывается грань между живописью и скульптурой!..

До недавнего времени считались фантастикой 3D-принтеры, способные воспроизводить детали собственной конструкции – то есть, печатать сами себя. Сейчас разработка такой машины ведется английскими конструкторами университета Бата в рамках проекта RepRap. На данный момент принтер уже производит более половины собственных деталей. Проект представляет собой открытую разработку – вся информация о конструкции распространяется по условиям лицензии GNU GPL.

Стоит отметить, что трехмерная печать – чуть ли не самый экологичный способ производства. Продукты 3D-печати можно перерабатывать и использовать вторично – уже разработан вид трехмерного принтера, который работает исключительно на бумажном сырье. Фактически, это наиболее эффективный способ утилизации уже отслуживших свое время вещей, сделанных из синтетических материалов – резины, каучука, различных видов пластмасс.

Впрочем, инженеры не ограничиваются одной только пластмассой. Thiel Foundation анонсировал технологию 3D-печати... мяса. «Искусственное мясо» может стать гуманным источником животных белков для тех, кто против насилия над животными.

А NASA выделило грант на создание 3D-принтера, способного «печатать» пищу – он готовит еду из ингредиентов, которые в порошковой форме хранятся в сменных картриджах. Смешав компоненты в разных пропорциях и добавив воду или масло, можно получить разные блюда. Срок годности заправленного картриджа составляет до тридцати лет. Предполагается, что принтер будет обеспечивать питанием участников длительных космических экспедиций. Кроме того, устройство, теоретически, может помочь справиться с проблемой нехватки пищи на Земле.

Разработки в этой области имеют громадное социально-политическое значение. Все модели и цифровые прототипы будут свободно распространяться и коллективно модифицироваться через Интернет – подобно уже «обкатанной» модели OpenSource, делая «цифровую утопию» возможной и технологически вполне осязаемой. Люди сами будут создавать нужные им вещи, превратятся в автономных частных производителей, не зависящих от крупных заводов и фабрик. Уже сейчас можно утверждать, что это вызовет коренные изменения в экономике – монополии и централизованное серийное производство, со временем, уйдут в небытие.



Впрочем, как и все новое, данная технология вызывает восторг не у всех. Государства всерьез обеспокоены возможностью использования ее во вред – скажем, для создания оружия. А такие разработки уже имеются. К примеру, американский оружейник-энтузиаст Коди Уилсон спроектировал пистолет The Liberator, который можно напечатать на 3D-принтере в домашних условиях. Выполненный из пластмассы, он не обнаруживается металлодетекторами, что вкупе с возможностью домашнего производства оружия всеми желающими вызвало недовольство американских властей. Госдепартамент США потребовал, чтобы оружейник убрал из Интернета трехмерные модели пистолета, пригодные для печати, а сенаторы предложили пересмотреть Акт о необнаруживаемом огнестрельном оружии 1988 года. Власти считают, что доступность технологии производства The Liberator может привести к новой волне терактов.

Проект печатаемого на 3D-принтере оружия стартовал в США в начале августа 2012 года. Его основателем стал юрист Коди Уилсон, вместе с несколькими друзьями приступивший к разработке простого в производстве стрелкового оружия. Проект получил название Wiki Weapon. Его основной идеей является создание трехмерных моделей огнестрельного оружия, которые мог бы скачать из Интернета любой желающий. При этом данные распространялись бы на свободной основе, чтобы каждый мог вносить в них собственные доработки и улучшения. Группа разработчиков во главе с Уилсоном назвала себя Defense Distributed.

The Liberator состоит из 16 частей, 15 из которых – пластмассовые: рукоять, спусковой крючок, две пружины, корпус, открытый курок, крепление курка, ствол, шайба, три крепежа для корпуса, крепеж для курка, крепеж для рукояти и возвратная пружина спускового крючка. Металлический в оружии только боек, накалывающий капсюль патрона, – это обычный гвоздь с большой шляпкой. Печать пистолета осуществлялась на 3D-принтере SST производства американской компании Stratasys Dimension, купленном энтузиастом-оружейником за \$8000.

Оружие печатается из стандартного АБС-пластика (из которого изготавливаются бамперы автомобилей, пылесосы, мебель, спортивный инвентарь). Распечатать The Liberator можно, в принципе, на любом принтере, работающем с АБС-пластиком – от MakerBot Replicator до RepRap.



Надо сказать, что Уилсон имел лицензию на разработку оружия, и ему не были предъявлены обвинения в незаконном создании пистолета. Дело касалось только чертежей и моделей в публичном доступе. Сам оружейник считает, что они были размещены законно: они не попадают под действие международного законодательства о продаже оружия, поскольку продавать The Liberator никто не собирался.

Разрабатывая пистолет, американец заявил, что его распространение через Интернет позволит разрушить намерение властей ввести полный контроль за оборотом оружия. Кроме того это позволит размыть грань между законодательством о контроле за оборотом оружия и информационной цензурой в Интернете – ведь чиновники, фактически, хотят запретить свободный обмен виртуальными 3D-моделями, которые сами по себе никому не наносят вреда!

А это уже серьезный повод задуматься. Кто знает, до какого абсурда могут довести соблюдение подобных законов и поправок ретивые правоохранительные органы? Не повяжут ли завтра всех, кто моделирует, скажем, оружие для шутеров от первого лица и выкладывает свои работы в Интернет?..

Тимур ГАФАРОВ





GIMP: Новости

Вышел GIMP 2.8.6

Не так давно произошло очередное минорное обновление стабильной версии GIMP. Этот релиз повышает работоспособность и устраняет ряд ошибок и недочетов прошлой версии. В частности, исправлено сохранение URI с доработкой сохранения/экспорта сжатых файлов, увеличен максимальный размер кистей и текстур до 1024x1024, добавлена проверка сохранности данных при изменении их названий, улучшен пользовательский интерфейс, усовершенствована работа с библиотеками и плагинами. Также обновлены локализации.



Необходимо сказать, что GIMP, несколько уступая в «раскрученности» Blender и развиваясь заметно медленнее одного, на данный момент стоит на пороге значительной эволюции. GIMP 2.8 переехал на платформу GEGL уже более чем на 90%. 16- и 32-битные каналы, столь ожидаемые сообществом профессиональных типографов, полностью поддерживаются в GEGL и планируются в GIMP 2.10. Обработка изображений с точностью 16 и 32 разряда на цветовой канал – int или float по выбору пользователя – уже реализована в нестабильной версии пакета.

Кстати, в рамках участия GIMP в GSoC 2013, стартовал проект по улучшению поддержки OpenCL для GEGL-операций. Кроме того, разрабатывается новый инструмент деформации – N-Point Deformation (NPD), аналог Puppet Warp в Photoshop CS6 и Free Transform в Krita. Он позволяет изгибать, масштабировать и перемещать элементы изображения, сохраняя их объем и целостность, а также не меняя фон – почти как в трехмерном редакторе!

В числе других новых проектов GSoC можно отметить улучшение поддержки формата PSD, а также унификацию всех механизмов для работы с выделениями в более общий инструмент.

А недавно была реализована востребованная очень многими возможность: вращение холста. Оно активизируется нажатием Shift+СКМ. Можно также вращать холст 15-градусными «шагами» при помощи комбинации Ctrl+Shift+СКМ. Поворот холста целиком будет полезно при работе на мобильных устройствах, а также для удобного редактирования изображений в «нестандартных» ракурсах – например, перевернутых портретов.



Язык D: Новости «с Марса»

свежие релизы и обновления

DMD 2.063

Состоялся релиз очередной минорной версии референсного компилятора DMD – 2.063. Релиз вносит, в общей сложности, 260 улучшений – в том числе изменения в самом языке, стандартной библиотеке Phobos (например, переписан модуль std.process). Исправлено множество багов. Доступны бинарные пакеты DMD для всех платформ, включая Windows, Linux, Mac OS X и FreeBSD. Напоминаем, что поддержка D1 официально прекращена.

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

LDC 0.11.0

Состоялся выход LDC 0.11.0 – компилятора D, использующего LLVM для генерации машинного кода. Релиз основан на D2 2.062, заявлена полная поддержка Linux x86/x86_64, Mac OS X 10.7+, Win32 (альфа-стадия), в ближайшем будущем ожидается также поддержка Win64. Кроме того, в тестовой стадии работает генерация кода для остальных платформ инфраструктуры LLVM – таких, как ARM и Sparc. В новой версии компилятора добавлены новые специфичные для LDC прагмы, улучшена оптимизация, исправлено множество багов. Прекращена поддержка D1.

<http://d32gngvpvl2pi1.cloudfront.net/ldc2-0.11.0-linux-x86.tar.gz>
<http://d32gngvpvl2pi1.cloudfront.net/ldc2-0.11.0-linux-x86.tar.xz>
http://d32gngvpvl2pi1.cloudfront.net/ldc2-0.11.0-linux-x86_64.tar.gz
http://d32gngvpvl2pi1.cloudfront.net/ldc2-0.11.0-linux-x86_64.tar.xz
<http://d32gngvpvl2pi1.cloudfront.net/ldc2-0.11.0-mingw-x86.7z>
http://d32gngvpvl2pi1.cloudfront.net/ldc2-0.11.0-osx-x86_64.tar.gz

GtkD 2.2

GtkD – биндинг и объектно-ориентированная обертка над функциями кроссплатформенного тулкита Gtk+ – обновился до версии 2.1. Поддержка теперь распространяется на Gtk 3.8 и GStreamer 1.0. Также обеспечена полная поддержка Win64. Кроме того, GtkD обзавелся собственным сайтом (<http://gtk.d.org>) и новым форумом (<http://forum.gtk.d.org>). Для тех, кто еще не готов отказаться от Gtk 2, официально сохраняется поддержка ветки GtkD 1.x. Исходники проекта доступны по лицензии LGPL – она позволяет проприетарным программам статически линковаться с GtkD, не требуя при этом открытия кода. Это позволяет использовать биндинг для создания коммерческих продуктов.

<http://gtk.d.org/download.html>

D + Objective-C

Якоб Карлборг объявил об обновлении созданной им двумя годами ранее связки DMD и Objective-C, которая теперь основывается на последней версии компилятора DMD. В рамках проекта d-objc представлен форк компилятора DMD и рантайма языка с возможностью работать с объектами Objective-C. В планах – перенос наработки в основную ветку DMD/druntime.

<http://michelf.ca/projects/d-objc/>

mysql-native

Анонсирован mysql-native – нативный драйвер СУБД MySQL для D, не имеющий зависимостей от официального клиента MySQL или какого-либо другого стороннего ПО. Изначально драйвер создавался для использования совместно с веб-фреймворком Vibe.d, но сейчас это практически независимый проект.

<https://github.com/rejectedsoftware/mysql-native>

templ-d

Представлен проект templ-d – язык разметки шаблонов для веб-приложений, аналогичный eRuby. Обработка шаблонов происходит полностью во время компиляции, что дает выигрыш в производительности при запуске приложения. Автор библиотеки планирует со временем превратить ее в альтернативный движок шаблонов для Vibe.d. Исходники проекта распространяются по лицензии Boost.

<https://github.com/dymk/templ-d>
<http://registry.vibed.org/packages/templ-d>

Rhodeus Script

Совет по науке и технологиям Турции (TUBITAK) представил работу, победившую на ежегодном научно-исследовательском конкурсе, проводящемся среди вузов страны. Ей оказался Rhodeus Script – серверный интерпретируемый язык, реализованный на D.

<https://github.com/Rhodeus/Script2.0>

DUB 0.9.15

Вышла новая версия DUB – менеджера пакетов для D, призванного облегчить процесс сборки программ и разрешение зависимостей. В версии 0.9.15 добавлена поддержка нескольких пакетов на директорию (это необходимо, например, в случае с Dedelect), переменной среды DUBPATH для пользовательского пути поиска пакетов, проделаны другие мелкие изменения, улучшения и багфиксы.

<http://registry.vibed.org/download>

BUB – новая система сборки

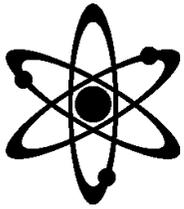
Анонсирован проект Bottom-Up-Build (BUB) по созданию системы сборки проектов, поддерживающей D, C и C++. BUB работает на Linux, ведется работа над поддержкой Windows и Mac OS X. Система поддерживает конфигурационные файлы, отслеживание зависимостей, кэширование импортов.

<https://github.com/GrahamStJack/bottom-up-build>

Обновление Glamour и gl3n

Подверглись апдейту Glamour (враппер над OpenGL) и gl3n (математическая библиотека для OpenGL). В Glamour добавлена поддержка VAO, FBO, улучшена поддержка Mac OS X, внесены другие изменения. В gl3n добавлена поддержка цветового пространства HSV, стек матриц, AABV, плоскостей, пирамид видимости (frustum). Класс векторов теперь поддерживает векторы произвольного размера. В планах на будущее – поддержка SIMD, перенос геометрического кода в gl3n.geometry, добавление новых геометрических объектов.

<https://github.com/Dav1dde/glamour>
<https://github.com/Dav1dde/gl3n>



Физический движок своими руками

Часть 1. Динамика твердого тела. Интегрирование

Казалось бы, кому в наше время может прийти в голову идея создания собственного физического движка? Это далеко не тривиальная задача, которую давно уже решили за вас разработчики Bullet, PhysX, ODE, Newton и других известных библиотек симуляции динамики твердых тел. А дело вот в чем.

Поскольку физика – это достаточно сложная предметная область, требующая специальных знаний, к этим библиотекам традиционно сложилось отношение как к «черным ящикам». Пользователь не знает и не пытается понять все хитроумные алгоритмы и тонкости внутренней архитектуры физического движка, он лишь использует его внешний API. При этом он может прекрасно разбираться в 3D-графике и линейной алгебре, но физика для него так и останется «terra incognita», если он будет полагаться на готовые решения.

Знать физику с точки зрения пользователя физического движка и с точки зрения его разработчика – это две совершенно разные вещи. Изучение физики поможет вам лучше понять то, как работает ваш любимый физический движок, перестать относиться к физике как к некой «магии», научиться использовать готовые решения более грамотно и эффективно.

К тому же, абсолютное большинство движков перегружены функциональностью и представляют собой, фактически, комбайны с собственной системой проверки столкновений, громадной математической библиотекой (векторы, матрицы, кватернионы), реализацией геометрических тел и еще Бог знает чего еще. И, скорее всего, в вашем проекте произойдет сильное дублирование функциональности – ведь ни один графический движок не обходится без собственной реализации вышеупомянутой математики и геометрии. А это, мягко говоря, не есть хорошо.

Поэтому во многих случаях использование сторонней физики как минимум неоправданно – например, если вы создаете бильярд или боулинг, то вряд ли вам понадобятся архисложный солвер, поддержка всех мыслимых типов сочленений и неконвексной геометрии...

Тем не менее, даже простой физический движок реализовать весьма нелегко. Мы решили исправить эту ситуацию и опубликовать подробное руководство, призванное раскрыть ключевые моменты создания игровой физики. Руководство разбито на несколько частей, которые шаг за шагом описывают теорию и реализацию компонентов простейшего физического движка.

Предполагается, что читатель имеет опыт работы с векторной алгеброй, матрицами и кватернионами. Само собой, необходимо хорошее знание какого-либо ЯП, желательно – из семейства фигурных скобок (C, C++, Java, C#). В качестве референсного языка используется D, но представленный код будет аналогичным для других родственных языков. Для понимания научных основ материала будет достаточно школьного курса ньютоновской механики.

Теория

То, что понимается под словом «физика» в приложениях реального времени, практически полностью относится к классической механике Ньютона – несмотря на то, что сама физика как наука за последние столетия сделала громадный шаг вперед, и ньютоновские законы уже не считаются всеобъемлющими. Например, они неприменимы для объектов со скоростями, близкими к скорости света (поведение таких объектов описывает специальная теория относительности) или для очень маленьких систем (таких, как атомы и элементарные частицы – тут действуют законы квантовой механики). Область применения ньютоновской механики – системы со скоростями много меньше скорости света и размерами, значительно превышающими размеры атомов и молекул. Законы Ньютона – это следствие СТО для предельного случая $v \ll c$.

Надо заметить, что до сих пор не создана «теория всего», которая согласовала бы теорию относительности и квантовую механику – это одна из главных нерешенных проблем современной физики.

Игровая физика, как правило, реализует раздел механики, именуемый динамикой – то есть, движение тел с учетом причин, вызывающих движение. Эти причины могут включать различные силы (например, силу тяжести), воздействие одних тел на другие (столкновения), воздействие на тела различных ограничений степеней свободы (которые иногда называют сочленениями или джоинтами).

Для описания динамики тела используются две основные физические величины: масса (m) и сила (F). В ньютоновской механике масса характеризует инертность тела (способность тела сопротивляться приложенной силе). В современной физике понятие массы имеет несколько иной смысл (в частности, в СТО масса движущегося тела – относительная величина и зависит от его скорости), но мы будем придерживаться «классического» определения. Масса – скалярная величина (выражена одним вещественным числом). По системе СИ масса измеряется в килограммах (кг).

Сила – мера интенсивности взаимодействия тел. Сила является причиной изменения скорости тела. Сила является векторной величиной, так как имеет направление (для привычного нам трехмерного пространства она выражается трехкомпонентным вектором). По системе СИ сила измеряется в ньютонах (Н).

Для того, чтобы описать движение тела, вводятся также понятия скорости (v) и ускорения (a). По системе СИ скорость измеряется в метрах в секунду (м/с), ускорение – в метрах в секунду в квадрате (м/с²). Зависимость ускорения тела от его массы и суммы приложенных к нему сил описывает второй закон Ньютона:

$$\vec{a} = \frac{\vec{F}}{m}$$

Ускорение – это не что иное, как скорость изменения скорости. Другими словами, ускорение показывает, на сколько изменится вектор скорости за интервал времени (Δt).

Формальное определение звучит так: ускорение – это первая производная скорости по времени (t):

$$\vec{v}(t) = \vec{v}_0 + \int_{t_0}^t \vec{a}(t) \Delta t$$

Для тех, кто незнаком с дифференциальным исчислением, необходимо пояснить: и скорость, и ускорение в данном случае рассматриваются как функции. Функция, определяющая скорость, нам изначально неизвестна. Однако известна скорость изменения этой функции в заданной точке (в заданный момент времени) – ускорение. Функция скорости изменения функции – это и есть производная функции. Нахождение производной называется дифференцированием, а обратный процесс (нахождение первообразной), который нас и интересует – интегрированием.

Если ускорение не меняется со временем, то интегрирование скорости осуществляется по следующей формуле:

$$\vec{v} = \vec{v}_0 + \vec{a} \Delta t$$

Данный метод интегрирования называется методом Эйлера. Аналогично интегрируется и позиция тела – координаты центра масс (x):

$$\vec{x} = \vec{x}_0 + \vec{v} \Delta t$$

Таким образом, динамическое тело может быть описано следующими минимальными характеристиками:

- *масса (скаляр)*
- *позиция (вектор)*
- *скорость (вектор)*
- *ускорение (вектор)*
- *сумма сил (вектор)*

Однако это еще не все. В абсолютном большинстве случаев, помимо линейного (поступательного) движения тел, нас интересует еще и вращательное. Концепции силы, скорости и ускорения во вращательном движении похожи на их линейные аналоги – с той разницей, что вместо массы в уравнениях вращения используется момент инерции. В строго реалистичных симуляциях это тензор (то есть, матрица 3x3) но мы в целях упрощения обойдемся скалярной величиной.

Момент инерции (I) – это мера инертности во вращательном движении тела вокруг оси. По системе СИ измеряется в килограмм-метрах в квадрате ($\text{кг}\cdot\text{м}^2$). Для сферы момент инерции равен

$$I = \frac{2}{5} mr^2$$

Зная эту величину, по второму закону Ньютона можно вычислить угловое ускорение тела (α):

$$\vec{\alpha} = \frac{\vec{M}}{I}$$

где M – момент силы (вращающая сила). В системе СИ измеряется в ньютон-метрах ($\text{Н}\cdot\text{м}$). Вращательное действие линейной силы определяется как векторное произведение силы на точку, в которой эта сила приложена:

$$\vec{M} = \vec{r} \times \vec{F}$$

Исходя из углового ускорения, интегрируется угловая скорость (ω , рад/с):

$$\vec{\omega} = \vec{\omega}_0 + \vec{\alpha} \Delta t$$

С аналогом позиции для вращательного движения – поворотом – дело обстоит сложнее. Если угловую скорость и ускорение можно описать трехкомпонентными векторами, то накапливаемый поворот из-за проблем с блокировкой оси векторами обычно не описывают. Вместо них используются матрицы 3x3, либо кватернионы. Последние более выгодны, так как требуют всего 4 вещественных значения (против 9 для матриц).

Интегрирование кватерниона поворота (q) из угловой скорости выглядит так:

$$q = \frac{1}{2} q_0 \vec{\omega} \Delta t$$

Итак, для того, чтобы учесть вращательное движение, мы добавляем к нашему описанию тела еще пять характеристик:

- **момент инерции** (скаляр)
- **поворот** (кватернион)
- **угловая скорость** (вектор)
- **угловое ускорение** (вектор)
- **сумма моментов сил** (вектор)

В любой момент времени скорость тела зависит от времени, прошедшего между двумя шагами симуляции – dt . Это время, как правило, фиксировано и должно соответствовать времени между двумя кадрами рендеринга – например, 1/60 с. Это соответствует 60 кадрам в секунду, как в системах с включенной вертикальной синхронизацией.

В кинематике (идеализированном описании движения) в качестве dt часто берется реальное время между кадрами, но динамика такой подход «не любит»: при резких скачках dt (например, вследствие подтормаживания графики), тела приобретают неестественно большие скорости и начинают пролетать сквозь препятствия.

Реализация

Представим полученное описание тела в виде класса:

```
class RigidBody
{
    float mass;
    Vector3f position;
    Vector3f velocity;
    Vector3f acceleration;
    Vector3f forceSum;

    float inertia;
    Quaternionf orientation;
    Vector3f angVelocity;
    Vector3f angAcceleration;
    Vector3f torqueSum;

    void integrate(float dt) { /* ... */ }
}
```

Интегрирование осуществляет метод integrate:

```
void integrate(float dt)
{
    acceleration = forceSum / mass;
    velocity += acceleration * dt;
    position += velocity * dt;

    angAcceleration = torqueSum / inertia;
    angVelocity += angAcceleration * dt;
    orientation += 0.5f * Quaternionf(angVelocity, 0.0f)
        * orientation * dt;
}
```

Продолжение следует...

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://code.google.com/p/cook-build-automation-tool/>

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.

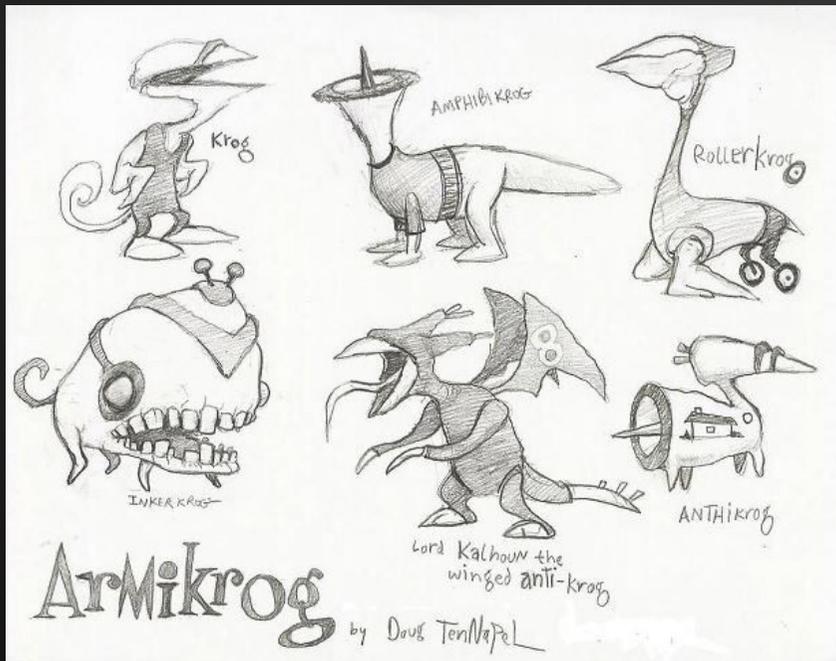
<http://code.google.com/p/dlib/>

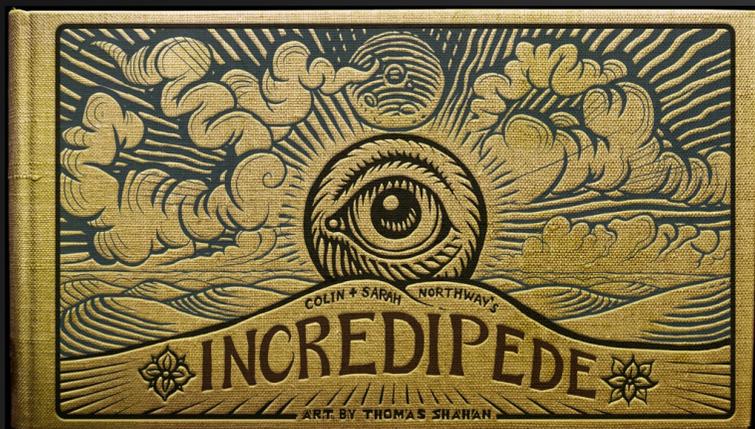
Armikrog



Разработчики культовых *The Neverhood* и *Earthworm Jim* объявили о начале работы над новым, как ожидается, шедевром – *Armikrog*. Классический пластилиновый квест возвращается! В новой игре бесстрашный пластилиновый космический исследователь по имени *Tom Naut* и его верный четвероногий друг терпят катастрофу на неприветливой планете и попадают в западню в крепости *Armikrog*. К сожалению, никакой связи с *Клейменом* и вселенной *The Neverhood* у игры не будет, так как все права на персонажей принадлежат *Electronic Arts*, а *Armikrog* – независимый проект с народным финансированием (создатели успешно запустили краудфандинг на *KickStarter*, быстро собрав необходимую сумму в \$900000).

Саундтрек для *Armikrog* будет написан *Терри Скоттом Тейлором*, композитором *The Neverhood*. Обещается полная поддержка *Linux* – как и возможность получить лишнюю DRM-привязки копию игры. Релиз намечен на июль 2014 – ждать осталось всего год! =)





Incredipede для Linux – бесплатно!

Компания Northway Games приняла решение о бесплатном распространении игры Incredipede для платформы Linux. В качестве причины такого шага называется симпатия к философии свободного ПО и сильная поддержка инди-игр со стороны пользователей Linux. Версии для Windows и Mac OS X остаются платными. При этом половину прибыли, получаемой от продажи на сайте игры (игра также поставляется через Steam), компания Northway Games решила пожертвовать на развитие двух открытых проектов: FlashDevelop и Vox2D, использованных в процессе разработки Incredipede.

Incredipede - двумерная физическая головоломка, написанная Flash и работающая в окне браузера. У вас есть существо по имени Quozzle, выглядящее, как шарик с одним глазом. Его можно заставить двигаться, «вырастив» ему конечности. Чтобы конечности двигались в нужном направлении, к ним необходимо правильно присоединить мускулы. Трансформируя существо таким способом, его можно заставить не только ходить, но и ползать, лазить, парить или прыгать. В игре 60 уровней, расположенных в трех мирах; имеется также редактор уровней. Incredipede вошла в число финалистов конкурса Independent Games Festival 2013.



SDL

Simple Directmedia Layer



SDL 2.0

Версия 2.0 библиотеки SDL (Simple DirectMedia Layer) достигла состояния кандидата в релизы. В числе наиболее интересных новшеств можно отметить собственный API для рендеринга 3D-графики с аппаратным ускорением (потенциальный аналог Direct3D?), поддержку ForceFeedback (вибрации) для джойстиков, поддержку захвата звука, аудио в формате 7.1, многооконный режим, доступ к буферу обмена, поддержку одновременного использования нескольких устройств ввода, multi-touch, поддержку многомониторных конфигураций и многое другое.

Стоит отметить, что, хотя стабильная версия SDL 2.0 еще не выпущена, ряд продуктов Valve уже пользуется данной веткой библиотеки, а ABI версии 2.0 считается стабилизированным.

Напомним, что SDL предоставляет кроссплатформенный доступ к таким средствам, как быстрый вывод 2D-графики, обработка ввода, проигрывание звука, вывод 3D-графики с использованием OpenGL и множеству иных сопутствующих операций, независимо от используемой ОС, что значительно упрощает создание мультимедийных приложений и игр. SDL можно назвать «кроссплатформенным DirectX». Библиотека выпускается под лицензией LGPLv2 и поддерживает Linux, Windows, Mac OS X, BeOS, BSD, Solaris, IRIX и QNX. Также существует ряд неподдерживаемых официально портов на иные системы – например, Android и Wayland.



История компьютерных игр

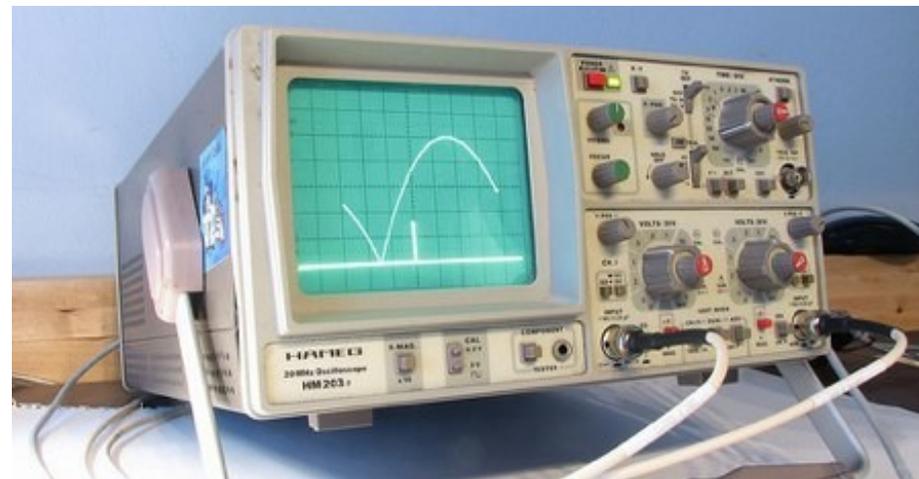
От Pong до PlayStation 4

Безусловно, компьютерные игры можно считать настоящим социокультурным феноменом второй половины XX века. И если даже не принимать во внимание тот факт, что этот феномен оказал громадное влияние на жизни людей нескольких поколений, неизвестно, каким путем бы шло развитие мировой культуры и экономики, не будь компьютерных игр. Электронные игры постепенно стали совершенно уникальным явлением на грани науки, промышленности, искусства и спорта. Поэтому история их возникновения и становления, объединившая деятельность множества людей самых разных профессий и кругов заинтересованности, является столь интересной темой, заслуживающей отдельного подробного рассказа.

Историческая хронология развития компьютерных игр традиционно разбита на несколько периодов – поколений. Это деление, будучи чисто условным, тем не менее, позволяет проследить основные качественные скачки, связанные с научно-техническим прогрессом и оказавшие серьезное влияние на эволюцию аппаратного и программного обеспечения. Всего выделяют восемь поколений длительностью от пяти до десяти лет.

Ранние игры

Указать точную дату возникновения первой компьютерной игры довольно трудно. Известно, что первые эксперименты с катодно-лучевой трубкой предпринимались еще в конце сороковых годов XX века. Но принято считать, что основоположником компьютерных развлечений является Уильям Хигинботэм, разработавший в 1958 г. игру Tennis for Two в Национальной лаборатории Брукхэйвена в Нью-Йорке. Созданная для развлечения посетителей лаборатории, Tennis for Two выводила на осциллоскоп аналогового компьютера упрощенное изображение теннисного корта, по которому прыгал мяч. Игроки управляли условными «ракетками» при помощи специальных контроллеров.



Ранние компьютерные игры создавались операторами электронно-вычислительных машин крупных научно-исследовательских учреждений. Это было скорее хобби в свободное от работы время, чем серьезная индустрия. Так, в 1961 г. группа студентов Массачусетского Института Технологий (MIT) разработала игру Spacewar! для нового по тем временам компьютера DEC PDP-1. Игроки управляли космическими кораблями, стреляющими ракетами. В центре экрана находилась «черная дыра», представляющая собой своеобразную ловушку для кораблей. Игра распространялась вместе с новыми машинами DEC, а также по тогдашним электронным сетям.

Первой игрой, способной выводить изображение на экран домашнего телевизора, стала Chase (1966 г., Ральф Бэр). В 1967 г. был изобретен первый световой пистолет и было создано несколько игр с ним.

В 1969 г. Кен Томпсон, программист AT&T, написал игру Space Travel для операционной системы MULTICS. Игра симулировала солнечную систему с несколькими планетами, на которые игрок должен был посадить космический корабль. Когда проект MULTICS был остановлен, Томпсон переписал игру на языке ФОРТРАН для операционной системы GECOS мэйнфреймового компьютера компании General Electric – GE 635. Из-за дороговизны работы с этой машиной (час машинного времени обходился в \$75), Томпсон принялся за поиск компьютера подешевле. Им стал подержанный PDP-7, на ассемблер которого Томпсон совместно с Деннисом Ритчи и портировали Space Travel. Для этого была создана легендарная операционная система UNIX.

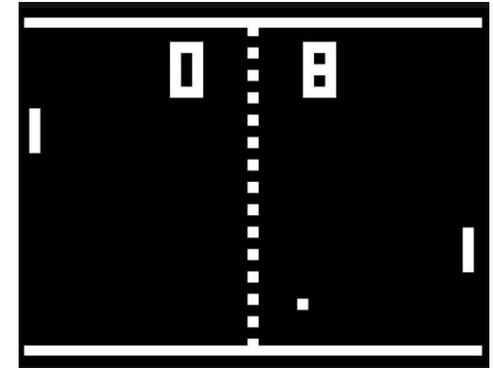


Первое поколение: 1972-1977

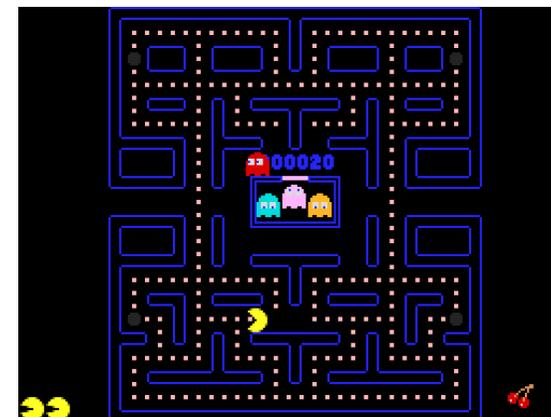
Начало первого поколения компьютерных игр связано с появлением игровых автоматов, которые управлялись компьютером и требовали денежной оплаты игрового сеанса. Такие устройства стали называть аркадами или видеоиграми. Первый такой автомат назывался Galaxy Game и был установлен в 1971 г. в Стэнфордском университете. Автомат был создан на базе PDP-11/20.

В том же году Нолэн Бушнелл и Тед Дебни создали аркадную версию Spacewar! и назвали ее Computer Space. Игру купила компания Nutting Associates, которая выпустила 1500 таких автоматов. Несмотря на то, что игра не пользовалась большой популярностью из-за чересчур длительного процесса обучения, ее выход стал знаковым событием, ознаменовавшим начало массовой продукции видеоигр.

А Бушнелл и Дебни в 1972 г. основали компанию Atari. Первый успех пришел к Atari с выходом популярнейшей видеоигры PONG, которая во многом походила на Tennis for Two. Всего было продано 19000 машин PONG.



Золотой век индустрии видеоигр наступил в 1978 г., когда компания Taito выпустила знаменитую Space Invaders. Успех Taito вдохновил множество других производителей, и в том же году Atari выпустила не менее знаменитую Asteroids. Цветные видеоигры стали популярны после выхода Pac-Man. Автоматы стали устанавливаться буквально во всех общественных местах – в ресторанах, барах, торговых и развлекательных центрах и т. д.



В то же время развивалась и индустрия «больших» ЭВМ – мейнфреймов. Создавались игры для DEC PDP-10, Hewlett-Packard HP2000, а также для системы PLATO компьютеров CDC. В период с 1971 по 1980 были выпущены Maze War и Spasim – родоначальники шутеров от первого лица, первый текстовый квест Adventure, компьютерная версия настольной ролевой игры Dungeons & Dragons. Для операционной системы BSD вышла игра Rouge. Были заложены основы сетевой игры, а также использования так называемой псевдографики – рисования картинок посредством вывода текстовых символов. Впоследствии такой вид графики получил название «ASCII-арт».

Кроме того, в середине семидесятых начали появляться первые домашние персональные компьютеры. Многие из них имели встроенные интерпретаторы популярных языков программирования. Возникли сообщества программистов-любителей – «хоббиистов». Исходные коды программ и игр стали публиковаться на страницах книг и журналов. Для таких компьютеров, как Commodore и Tandy было выпущено огромное количество программного обеспечения, распространявшегося на гибких дисках и магнитофонных кассетах.

Второе поколение: 1977-1983

Начало второго поколения связано с еще одним значительным событием: созданием ROM-картриджей. До этого дискретная логика игры была «зашита» в постоянную память игрового компьютера, поэтому на одной машине могла быть только одна встроенная игра. На смену встроенному ПЗУ пришло сменное: игровая программа записывалась на отдельные платы (картриджи), которые вставлялись в специальный слот компьютера и считывались его микропроцессором. Картриджевые компьютеры стали называться консолями.

Первым таким компьютером стал VES (Video Entertainment System), выпущенный компанией Fairchild в 1976 г. Ну, а главными консолями второго поколения стали VCS, Intellivision и ColecoVision.



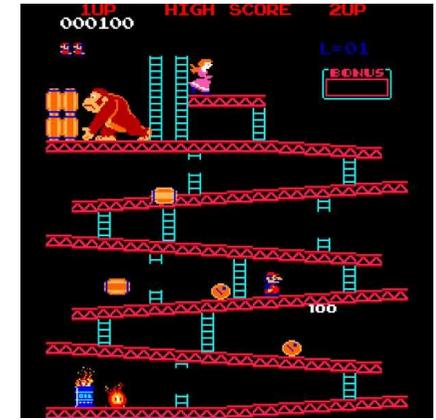
Третье поколение: 1982-1992

Восьмидесятые стали эпохой бурного развития индустрии персональных компьютеров. В 1982 г. был выпущен Commodore 64. Тем временем за океаном появился его основной конкурент – Sinclair ZX Spectrum. Этот британский 8-разрядный компьютер быстро завоевал популярность во всей Европе, став настоящим символом эпохи. Позднее его клоны распространились и в России, занимая значительную часть рынка любительских ПК вплоть до конца девяностых.



С приходом IBM PC-совместимых компьютеров интерес к играм вновь начал возрастать. Мощные для того времени звуковые чипы вроде Sound Blaster, 16-цветовая EGA-графика, а позднее – и 256-цветовой стандарт VGA стали толчком к возникновению новой игровой индустрии.

В то же время развивались и игровые консоли. Из-за обвала рынка видеоигр в 1983 г., персональные компьютеры на некоторое время вышли вперед, но все изменилось с выходом в 1985 г. 8-битной консоли Nintendo Entertainment System (NES), бесспорного хита десятилетия, завоевавшего популярность во всем мире. Игра для NES, Super Mario Brothers, сделалась де-факто настоящим символом, не только компании Nintendo, но и видеоигр в целом. Герой игры, водопроводчик Марио, стал одним из самых запоминаемых персонажей компьютерных игр.



NES доминировала на североамериканском и японском рынках в течение пяти лет. Кроме нее, свою публику также нашла Sega Master System, которая приобрела широкое распространение в Европе, Австралии и Южной Америке.

Так вышло, что именно в восьмидесятых были заложены многие современные стандарты компьютерных игр. В частности, одним из таких стандартов стал игровой контроллер-геймпад с крестовиной направления и несколькими клавишами действия. Этот стандарт используется и по сей день. Кроме того, в период до середины 90-х вышло множество игр, положивших начало культовым сериям и даже целым игровым жанрам: Dragon Quest, Final Fantasy, Legend of Zelda, Metal Gear, Dune, Alone in the Dark, Monkey Island, SimCity и т. д.



Четвертое поколение: 1989-1994

Этот период можно смело назвать эпохой расцвета индустрии компьютерных игр. Производители аппаратного и программного обеспечения, команды разработчиков, дистрибьюторы, бюджеты, фонды, средства рекламы, распространения и сбыта росли и развивались с огромной скоростью. Игровые компании начали сотрудничать с со звуко- и киностудиями. Этому росту также способствовало наступление «цифровой эры»: процессоры становились мощнее и дешевле, появились более дешевые и вместительные носители данных, оборудование обрело аппаратной поддержкой сложной графики и качественного звука, компьютерные сети стали гораздо доступнее и охватили весь мир. В этот период начали свой путь к славе гиганты современного мира КИ: Apogee (впоследствии 3D Realms), Epic Megagames (впоследствии Epic Games), id Software, Maxis, Electronic Arts. Вышли такие известные лейблы, как Myst, Wolfenstein 3D, Doom, Heroes of Might and Magic, Mortal Kombat, Street Fighter и т. д.



Начало четвертого поколения, бесспорно, связано с выходом 16-битной системы Sega Mega Drive (в Северной Америке она была выпущена под именем Sega Genesis), дебютировавшей в 1989. Nintendo ответила на это выпуском в 1991 г. Super Nintendo Entertainment System (SNES). Кроме того, в это время вышли менее известные консоли TurboGrafx-16, Neo-Geo, PC Engine.



Популярность Sega Mega Drive связана с игровой серией о похождениях сверхзвукового ежа Соника, которая составила нехилую конкуренцию играм серии Mario от Nintendo.



Пятое поколение: 1994-1999

Традиционные двумерные игры постепенно уступали место модной новинке — играм с 3D-графикой. На арену вышли 32-битные процессоры. Выход Atari Jaguar в 1993 г. окончательно определил направление развития игровой индустрии. Год спустя вышли Sega Saturn и Sony PlayStation. 3D-играм был дан зеленый свет, и бестселлеры посыпались как из рога изобилия: Crash Bandicoot, Spyro the Dragon, PaRappa the Rapper, Tekken, Resident Evil, Silent Hill, Grand Theft Auto, Final Fantasy VII, Need for Speed... Sony доминировала на рынке более пяти лет, и даже появление такой мощной системы, как Nintendo 64 не смогло поколебать ее успех. PlayStation прочно заняла свою нишу и покинула ее очень нескоро.



Одновременно с повсеместным распространением PlayStation, мощный толчок получили 3D-технологии для персональных компьютеров. Их развитие определялось в разное время компаниями 3Dfx, Silicon Graphics и Microsoft. В связи с растущим рынком видеоускорителей, вполне закономерным стало возникновение программных стандартов, позволяющих программистам писать игры, совместимые с большим количеством графических чипов. Такими стандартами стали OpenGL и DirectX. Стандарты получили поддержку крупнейших разработчиков аппаратного и программного обеспечения, и вот перед нами результаты этой ситуации: серии шутеров от первого лица Quake от idSoftware и Unreal от Epic Games, на долгие годы ставшие полноправными эталонами жанра и технологии.



Шестое поколение: 1998-2006

Начало шестого поколения было положено выходом консоли Sega Dreamcast в 1998 г., которая, впрочем, не добилась особого успеха и, в какой-то мере, стала причиной неожиданного ухода Sega с мировой арены видеоигр. Этот уход сопровождалось массовым увольнением сотрудников компании и другие скандальные события. Sony и Nintendo укрепили свои позиции выпуском соответственно PlayStation 2 и GameCube.

Вездесущая Microsoft также не преминула отхватить часть рынка и выпустила собственную игровую консоль Xbox (2001 г.). Кроме того, широкое распространение получили карманные консоли Nintendo GameBoy Color и GameBoy Advance. В это же время началось развитие игр для мобильных телефонов и карманных ПК, преимущественно на базе платформ Java, BREW, Windows Mobile и Symbian. Компания Nokia выпустила собственную мобильную игровую платформу N-Gage.

Индустрия игр для ПК также не стояла на месте. Широкое распространение недорогих ноутбуков и субноутбуков средней вычислительной мощности привело к образованию рынка казуальных игр. Низкие системные требования этих несложных, но симпатичных и увлекательных игр сделали казуальный жанр невероятно прибыльным для shagwage-дистрибьюторов, использующих для сбыта своей продукции Интернет. В качестве примеров можно назвать Luxor, Bejeweled, Diner Dash. Двумерная графика снова в моде!

Образовалась довольно широкая «пропасть» между казуальным и «хардкорным» геймингом. Все более ощутимой становится разница между мощными и дорогими конфигурациями и маломощными бюджетными решениями. Такое разделение рынка привело к резкому контрасту в жанровой направленности: если казуальные геймеры предпочитают аркады, головоломки и квесты, то «хардкорщики» стали основными потребителями ресурсоемких и технологичных экшенов вроде Quake III-IV, Doom 3, Half-Life 2, Grand Theft Auto: San Andreas, Unreal Tournament 2004, Call of Duty III и др.



Отчасти это связано с «гонкой вооружений» производителей аппаратного обеспечения. На мировую арену производителей оборудования выходят nVidia, ATI, AMD, составившие серьезную конкуренцию Intel. Производительность CPU и GPU возрастает с каждым годом, появляются качественно новые идеи, такие, как многоядерность, увеличение значения GPU, который становится способным выполнять все более широкий круг задач, ориентация на сетевые сервисы и кроссплатформенность.



Седьмое поколение: 2004-2011

Седьмое поколение открыло появление портативных консолей DS и PlayStation Portable (PSP) от Nintendo и Sony с месячным отрывом в 2004 г. В то время, как PSP обеспечивала более мощные графические и звуковые составляющие, DS предлагала геймерам совершенно новый подход в управлении. Консоль имела сразу два экрана, один из которых — сенсорный, позволяющий отдавать команды простым прикосновением пальцев.

Чуть позднее вышли домашние консоли седьмого поколения: Microsoft Xbox 360 (2005 г.), Sony PlayStation 3 и Nintendo Wii (2006 г.). Они полностью отвечали требованиям эры высоких технологий и несли на себе все последние новинки: диски формата Blu-Ray, HD-video, многоядерные процессоры, высокопроизводительные GPU. Брендowymi стали игры Grand Theft Auto IV, Gears of War, Metal Gear Solid 4, ставшие воплощением эпохи High-Definition и задавшие планку для более молодых проектов.



Что касается ПК, то ситуация с разделением рынка продолжала усугубляться. Этому способствовал выход сверхресурсоемких Crysis, Half-Life 2: Episode Two, Clive Barker's Jericho, GTA IV, а также игр на базе Unreal Engine III. Постоянно растущие системные требования, невозможность угнаться за стремительным развитием оборудования, а также условия мирового финансового кризиса «пересадили» часть игроков с мощных когда-то машин на нетбуки и неттопы, тем самым вызвав в последнее время снижение спроса на мощное аппаратное и программное обеспечение. Появилось множество независимых инди-разработчиков, энтузиастов, создающих открытые кроссплатформенные игры и игровые движки.

В связи с разочарованием в вышедшей в 2007 г. ОС Windows Vista, многие пользователи заинтересовались свободными операционными системами, в том числе – Linux, под которую в последние годы начали выпускать все больше качественных игр.

Восьмое поколение: 2011-...

Восьмое поколение игровых приставок берет свое начало после релиза портативной консоли Nintendo 3DS 27 марта 2011 г. Анонс преемников Xbox 360 и PlayStation 3 состоялся в 2013 году: Microsoft представила Xbox One, а Sony – PlayStation 4.

А несколькими годами ранее началась так называемая «сенсорная эра»: безкнопочные устройства с полностью сенсорным управлением (смартфоны, планшеты и ноутбуки-трансформеры) буквально взорвали мобильный рынок.



Косвенно это повлияло и на характер игр: сенсорные экраны и гироскопы позволили реализовать принципиально новые виды геймплея. Порождения этой эпохи – к примеру, Angry Birds или Doodle Jump – даже стали настоящим предметом масскульта.

Взросшая роль Интернета (и, в частности, Web 2.0) в жизни людей оказала влияние на игровую индустрию: современные консоли – это полноценные развлекательные медиацентры, пригодные не только для игр, но и веб-серфинга, просмотра кино в высоком разрешении, прослушивания музыки. Традиционные способы распространения игр на физических носителях (CD, DVD, BD) постепенно начали уступать место интернет-дистрибуции: настоящей «золотой жилой» для всех разработчиков (включая инди) стала система Steam от Valve, магазины приложений для Android и iOS и т. д.

Тимур ГАФАРОВ

Демосцена

Алгоритм как произведение искусства...

Не так давно увидел свет новый документальный фильм о демосцене – «The Art Of The Algorithms». Над созданием фильма около двух лет трудилась команда проекта «Moleman» – это уже второй полнометражный фильм, созданный в рамках этого проекта. Первый рассказывал о различных молодежных субкультурах, лишь отчасти затрагивая тему демосцены, второй же полностью посвящен рассказу о данном феномене. В диалогах встречаются английский, польский, финский, а так же венгерский языки, поэтому настоятельно рекомендуется смотреть вместе с субтитрами.

<http://www.youtube.com/watch?v=iRkZcTg1JWU>



Глубоко в андеграунде компьютерной культуры находится то, что известно как «демосцена» – это неформальное сообщество программистов, создающих так называемые «демы», крошечные по размерам программки, которые изначально являлись чем-то вроде «электронной визитки» с логотипом их создателя. К примеру, в эпоху любительских ПК демы часто внедрялись пиратами-крэкерами во взломанные ими коммерческие программы.

С самого начала демомейкеры уделяли много внимания аудиовизуальному содержанию своих «визиток». Это было что-то вроде соревнования – все старались удивить зрителя спецэффектами, приятной музыкой и красочной картинкой. Однако старые компьютеры были не слишком хорошо приспособлены для вывода сложной анимации в реальном времени, носители информации имели небольшую емкость, и демы приходилось уместать в ограниченный по объему машинный код. Сейчас перед программистами, естественно, уже не стоят такие жесткие ограничения, но демомейкеры соблюдают традиции – уместить музыкальный клип с трехмерной графикой в 64 килобайта до сих пор считается показателем высочайшего мастерства хакерского искусства...

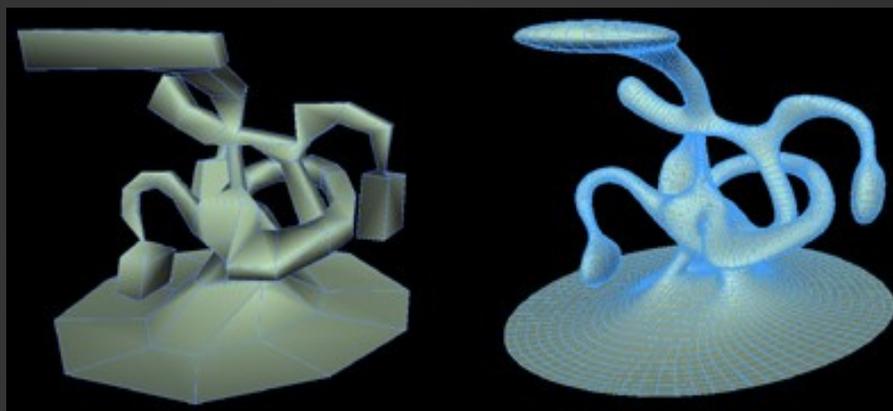
Эстетическая культура демосцены уходит корнями в «хаки отображения» (display hacks) – это простые вычислительные приемы, позволяющие рисовать на экране различные узоры и фигуры. Например, легендарный хакер Билл Госпер изобрел хак отображения под названием «smoking clover» – своеобразный психоделический «лист клевера», переливающийся всеми цветами радуги. Госпер даже пошутил, что эту программу нужно держать в секрете от правительственных органов – по причине ее «галлюциногенных свойств».

В отличие от большинства других приложений, демы обычно пишутся на языке ассемблера или на C, без использования сторонних библиотек (кроме, разумеется, системных). Деммы могут использовать возможности аппаратного обеспечения для ускорения графики – современные демомейкеры полагаются на функции OpenGL и DirectX, используют шейдеры и т. д. Если что-либо не удастся реализовать в силу ограниченности аппаратных средств, хакеры придумывают новые приемы, позволяющие эффективно обойти эти ограничения.

Многие из этих передовых методов впоследствии становятся общеупотребительными, а самих демомейкеров наперебой приглашают на работу в крупные софтверные корпорации.



Поскольку на счету каждый байт, демомейкеры не используют заранее подготовленные внешние данные (так называемые ассеты – модели, текстуры, спрайты и т. д.). Все эти данные генерируются программой «на лету» по специальным алгоритмам. Например, широко применяются простые геометрические фигуры, которые можно описать математически. Более сложные формы создаются путем построения полигональной сетки вокруг каркасов и ребер, которые, в свою очередь, также генерируются процедурно. Классические примеры – создание деревьев и растений на основе фракталов или кривых Безье, тел вращения на основе контура профиля, а также сложных органических объектов на основе метасфер. Полученные формы можно масштабировать и трансформировать, сгибать, скручивать, соединять – чтобы в итоге получить желаемую модель. Кроме того, некоторые модели хранятся в грубом низкополигональном виде – полигоны, составляющие ее геометрию, подразбиваются программой перед рендерингом в более гладкое представление.



Текстуры и другие изображения создаются путем многоуровневой фильтрации какой-либо простой исходной структуры, построенной, опять-таки, самим компьютером при помощи математических формул. Такой структурой может быть сетка, синусоидная «плазма», «шум Перлина», «диаграмма Вороного» и др.

Музыка в демах также процедурная. Все звуки (сэмплы) хранятся в сжатом виде, либо генерируются встроенным синтезатором – а затем воспроизводятся в нужное время по заданным нотам. Искусство создания таких сверхсжатых музыкальных композиций – это отдельное направление, получившее название «трекерная музыка». Программы, в которых пишется такая музыка, называются трекерами (trackers), а сами файлы-композиции – модулями (modules). В модуле хранятся как звукозаписи инструментов в импульсно-кодовой модуляции, так и партитура мелодии, включающая ноты и аудиоэффекты. Композиторы демосцены именуются «трекерщиками» или «MOD-музыкантами».

Сами алгоритмы, по которым работают демы, оптимизируются и компактифицируются до предела – если полученная в итоге программа превышает установленный максимальный объем, ее сжимают и снабжают загрузчиком, который распаковывает сжатый код перед запуском. Самые популярные из «компрессоров» для программ – это UPX и kkrunchy.

Собственно 64-килобайтовые демы называются «интро». Также изредка встречаются интро в 4 килобайта, 512 байт, 256 байт и даже в 128 байт! Несмотря на малый размер, авторам удается поместить в них интереснейшие видеоэффекты, поражающие зрителей. Впрочем, существуют и «большие» демы – размером от 4 до 15 МБ.

Практически с самого момента зарождения демосцены появились «демопати» (demoparty) – мероприятия, на которых демомейкеры встречаются, делятся опытом, устраивают фестивали и конкурсы с различными номинациями. За первые места организаторы дают всевозможные призы. Самые известные демопати – «Assembly» (Финляндия), «Breakpoint» (Германия), «Chaos Constructions» (Россия) и другие.

В России расцвет демосцены пришелся на 1996-1999 год, когда в Санкт-Петербурге и Москве проходили такие события, как «ENLIGHT», «Bytefall» и «FunTop». «Второй эшелон» демопати в 1999-2006 гг. возглавила демопати «Paradox» в Ростове-на-Дону, а также «CAFe» в Казани, «ASCII» в Ижевске, «DiHalt» в Нижнем Новгороде и «Chaos Constructions» в Санкт-Петербурге.



Изначально создание дем было делом единичных хакеров, но сейчас демомейкеры зачастую объединяются в команды – демогруппы, состоящие из программистов, художников, композиторов и режиссеров. Одна из самых известных европейских демогрупп – немецкая «Farbrausch» (от «farb gausch» – «буйство цвета»). Она прославилась благодаря своему 64-килобайтному интро конца 2000 года под названием «fr-08 .the .product».



Группа отличается крайней скрупулезностью в оптимизации размера своих работ. Яркий пример – та же «fr-08». В ней удалось реализовать полноценное 11-минутное аудиовизуальное представление, которое при полном отсутствии сжатия и оптимизации размера потребовало бы... около 2 гигабайт для хранения используемой информации!

Другая нашумевшая работа «Farbrausch» (а, точнее, ее подразделения под названием «.theprodukt») – полноценный трехмерный шутер от первого лица «.kkrieger». Размер игры составляет всего 96 килобайт – и, при этом, в ней есть один большой футуристический уровень с множеством комнат, несколько типов оружия и врагов. Группа также выпустила оригинальный специализированный инструмент для создания дем – «.werkzeug». Ей также принадлежит вышеупомянутый компрессор kkrunchy и еще ряд других программных средств.

Как уже было сказано, демо-технологии зачастую становятся «мейнстримом» в области компьютерной графики и разработки игр. Так, например, один из любимых приемов демомейкеров и других компьютерных энтузиастов – воксельная графика, использующая в качестве элемента пространства объемную точку (обычно куб), с ростом вычислительных мощностей ПК получила распространение и в «большой» игровой индустрии: достаточно вспомнить такие тайтлы, как Worms 3D, Minecraft и Outcast. Воксели также используются в Crysis. Кроме того, в разрабатываемом движке id Tech 6 компания id Software дебютирует технология Sparse Voxel Octree (Разреженное воксельное октодерево) для визуализации статических объектов уровней – ландшафта, массивных строений и т. д.



Кстати, несколько лет назад новостные ленты облетела информация о другом революционном проекте, использующем для построения игровых миров технологию, подобную воксельной графике. Это Unlimited Detail («Неограниченная детализация»), в которой трехмерная сцена хранится в виде «облака» точек или «атомов», аналогичных вокселям (длина одной точки – 1/4 мм).

В отличие от вокселей, облака точек, как и полигоны, предназначены лишь для моделирования поверхностей, но не «внутренностей» трехмерных объектов. На экран выводятся лишь те точки, которые в данный момент времени может увидеть пользователь, и размер одной точки равняется размеру одного пикселя на экране.

Задачу отсеивания невидимых точек берет на себя оригинальный поисковый алгоритм, скорость работы которого позволяет отображать на сцене неограниченное количество объектов с очень высокой детализацией без существенной потери производительности. Единственный фактор, снижающий производительность, это разрешение экрана – чем больше разрешение, тем больше точек требуется найти алгоритму, тем больше времени тратится на формирование одного кадра изображения.

Геометрия является полностью изменяемой, причем изменения будут производиться на атомарном уровне и останутся постоянными. К примеру, следы ног на песке не будут «запечатанными» в текстуру, а будут представлять собой реальное изменение геометрии уровня. Имеется возможность конвертировать полигональные модели в облака точек при помощи специальных утилит. И, наконец: в это трудно поверить, но, по заявлению разработчиков, Unlimited Detail работает исключительно на центральном процессоре и не задействует GPU в своих вычислениях!

Было бы странно, если бы здесь не было подвоха. И он имеется. Разработчики с неохотой отзываются о возможностях своей системы в области освещения (пока поддерживается всего один динамический источник света на сцену) и анимации персонажей (облака точек пока невозможно эффективно анимировать).

Кроме того, Маркус Перссон, создатель Minecraft, выступил с разоблачением всего этого «ноу-хау». По мнению Перссона, маркетинговый отдел Euclidean (разработчика Unlimited Detail) «раскручивает» уже существующие технологии, преувеличивая инновационность своих разработок в целях привлечения инвестиций.

Джон Кармак из id Software подтвердил критику Перссона, добавив, что возможность создать статичный неинтерактивный воксельный мир с имеется уже сейчас. Кармак описал свое решение проблемы с анимацией в воксельных сценах, примененное в id Tech 6: воксели применяются лишь к статичным объектам на сцене, а все движущиеся объекты (персонажи, транспорт и т. п.) выполнены при помощи классических полигонов.



Еще одна технология, «выросшая» из среды демосцены – Single Texture («Одна текстура»). Когда перед вами стоит ограничение на размер программы, один из логичных выходов – сделать так, чтобы один и тот же блок статичных данных использовался разными способами. Например, чтобы одна и та же текстура подходила к нескольким (или даже ко всем) объектам сцены.

Именно такой прием не так давно продемонстрировал левелдизайнер Тор Фрик, опубликовав дему «Sci-fi Lab», в которой показана возможность оптимизации сцены за счет использования для целой карты уровня всего одной текстуры размером 256x512 пикселей. Демонстрация, кстати, сделана на движке Unreal Development Kit. Такие эксперименты доказывают, что и в нашу эпоху терабайтных винчестеров «размер имеет значение»...



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://gplus.to/fpsmag>