

# СПЕЦ

03|64|МАРТ 2006  
ЕЖЕМЕСЯЧНЫЙ  
ТЕМАТИЧЕСКИЙ  
КОМПЬЮТЕРНЫЙ  
ЖУРНАЛ



GAME

LINE

ЖАНРОВОЕ ЖАРКОЕ **6** НЕУПРАВЛЯЕМЫЙ DIRECTX **8**  
ТАЯГА К РЕВОЛЮЦИЯМ **12** ФИЗИКА В ИГРАХ **20**  
ОПТИМИЗАЦИЯ ИГРОВОЙ ГРАФИКИ **24**  
ЗВУКОВОЕ НАСИЛИЕ **30** СКРИПТОВЫЕ ЯЗЫКИ **36**  
РЕЦЕПТЫ LUA! **42** ПРОБЛЕМЫ ПЕРЕНОСИМОСТИ **46**  
ШАГ В ПРОШЛОЕ **56** СЕТЕВЫЕ ИГРЫ **62** ЗАЩИТА ИГР ОТ ВЗЛОМА **66**  
СТРОИМ ДВУХМЕРНОЕ СЧАСТЬЕ **76** САГА О 3D-ДВИЖКАХ **80**  
КАК ЗАКАЛЯЛСЯ FLASH **90** ИНТЕРВЬЮ СО STEP CREATIVE GROUP **96**



# как закалялся **flash**

## СОЗДАНИЕ ОНЛАЙН-ИГРЫ THE OFFICE SPACE

FLASH-ИГРЫ МЕГАПОПУЛЯРНЫ В НАШЕ ВРЕМЯ. РЕДКИЙ ОФИСНЫЙ СОТРУДНИК ХОТЬ РАЗ В ЖИЗНИ НЕ ПРОЖИГАЛ КАЗЕННОЕ ВРЕМЯ И ФИРМЕННЫЙ ТРАФИК В ПОДОБНОМ ВРЕМЯ ПРЕПРОВОЖДЕНИИ. В ЭТОЙ СТАТЬЕ МЫ РАССКАЖЕМ И ПОКАЖЕМ, КАК СОЗДАВАЛАСЬ ОДНА ИЗ ТАКИХ ИГР | СЕРГЕЙ ПОДБЕРЕСКИЙ

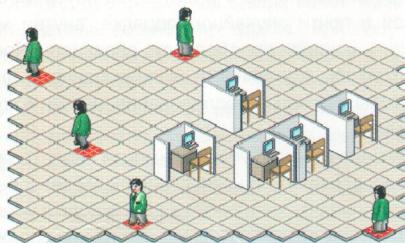
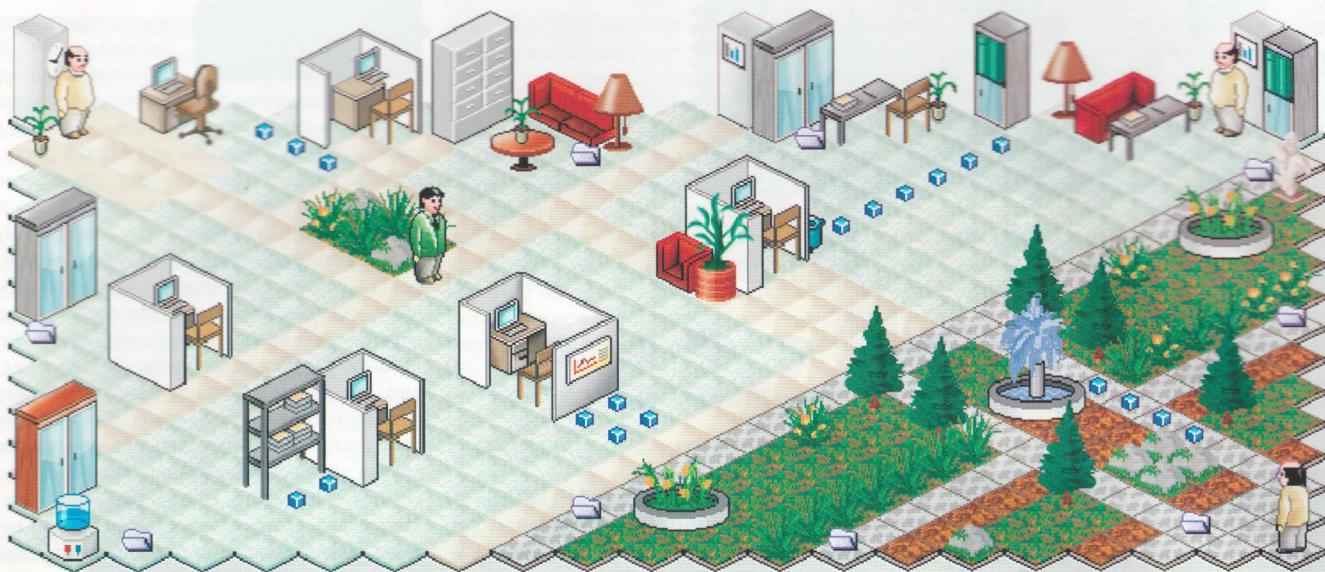


### The Office Space

ИДЕЯ, РУКОВОДСТВО ПРОЕКТОМ, КОДИНГ И ДИЗАЙН — СЕРГЕЙ ПОДБЕРЕСКИЙ  
АНИМАТОР — АНДРЕЙ МАЦКО, ТЕКСТЫ — ERIN VAN SICKLE, ЗВУК — АЛЕКСАНДР ГЛАДКИЙ

THE OFFICE SPACE, НЕБОЛЬШАЯ ИГРА, СОЗДАННАЯ С ПОМОЩЬЮ MACROMEDIA FLASH В 2003 ГОДУ, ДОВОЛЬНО БЫСТРО ЗАВОЕВАЛА ПОПУЛЯРНОСТЬ. ХОТЯ ЯДРОМ ИГРЫ ЯВЛЯЕТСЯ СЛОЖНЫЙ ИЗОМЕТРИЧЕСКИЙ ДВИЖОК (ENGINE), THE OFFICE SPACE ДОВОЛЬНО ПРОСТА И НЕ РАССЧИТАНА НА ДОЛГИЙ ИГРОВОЙ ПРОЦЕСС. ИГРА БЫЛА УДОСТОЕНА НЕСКОЛЬКИХ ИНТЕРНЕТ-ПРИЗОВ И СТАЛА СЕРЕБРЯНЫМ ПРИЗЕРОМ КОНКУРСА, ОРГАНИЗОВАННОГО ЖУРНАЛОМ I.D.MAGAZINE В 2004 ГОДУ, В КАТЕГОРИИ ИНТЕРАКТИВНОГО ДИЗАЙНА.

ИГРА ДОСТУПНА НА [WWW.THEOWORLDS.COM/GAMES/OS](http://WWW.THEOWORLDS.COM/GAMES/OS)



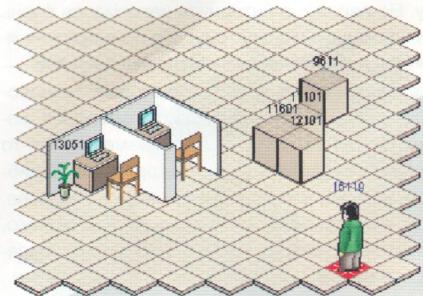
## 2 начало. разработка «движка» и определение задач

Идея с уровнями/картами органично сочеталась с офисной обстановкой, поскольку нарисовать офис легко: мебель, кулеры, секретарши и прочие атрибуты :). Я поискал техническое решение и решил остановиться на изометрической проекции (вид сбоку-сверху) — выглядит не так плохо, кроме того, к тому времени появились аналогичные проекты в интернете, а значит, технически моя идея была реализуема. Итак, я сразу сел писать код actionscript (AS1 на Flash 5), чтобы создать базу движка и со временем построить проект на его основе. Я начал с простой карты, разделенной на квадраты (ромбы из нашего вида). Планировалось расположить все элементы по границам этих квадратов. Так они легче запоминаются в массиве (двумерный массив с ячейкой для каждого квадрата). Привязка к квадратам, а не к координатам очень облегчает просчет взаимодействий персонажей и предметов, упрощает определение z-порядка элементов (вопрос о том, какие объекты «ближе»), нужного для организации правильного перекрывания предметов.

Как создавался персонаж? Было решено сделать его с тремя фазами анимации: стоит на месте, правая нога поднята, левая нога поднята. Последние две фазы чередуются в то время, пока персонаж движется (самая элементарная анимация ходьбы :)). Герой/персонаж двигается в четырех направлениях, два из них зеркальные (можно просто перевернуть картинку, для этого персонаж рисуется симметричным).

Итого — шесть разных картинок, то есть минимум, хотя если персонаж захочет скакать на одной ноге, можно обойтись и двумя фазами.

Однако трех фазказалось слишком мало, движения персонажа оказались черезчур резкими, так что пришлось добавить еще две фазы (итого десять картинок для каждого персонажа). Кстати, несмотря на то, что увеличение количества картинок с фазами обеспечивает достойную плавность и красоту, оно же приводит к увеличению времени работы аниматора и размера конечного файла. Так что тут, как и во многом в жизни :), нужно искать золотую середину. Кстати, профессиональные аниматоры знают много трюков по достижению достойной плавности движения с упрощенной анимацией. Мы же двигались методом проб и ошибок. Андрей Мацко занимался анимацией персонажей, и уже когда наш первый персонаж делал свои первые шаги, мы поняли, что анимация будет сложнее, чем ожидалось. Работы над персонажами начались рано, так как мне не хотелось



**1 начало: идея** Какую игру создать? Сначала для нас этот вопрос не казалось важным, но призадумавшись, я понял, что на самом деле не все так просто и нужно определиться. В первую очередь — с желаниями :). Мои желания были просты: создать небольшую игру, в которую можно поиграть во время обеденного перерыва, предварительно не читая длинные инструкции. Чтобы ограничить игровое время, было решено сделать игру из конечного числа уровней. На случай если понадобится расширить ее, мы запланировали наличие редактора карт. Так стали выявляться первые технические детали, а конечную идею нам подсказал фильм Office Space. Кто смотрел, тот поймет — комедия о самой обычной скуче и офисной рутине. В каждой компании есть какие-то особенности (они и описаны в фильме), иногда бывает полезно посмеяться над ними. Вот так на почве юмора и родился проект.



### Создание редактора карт/уровней

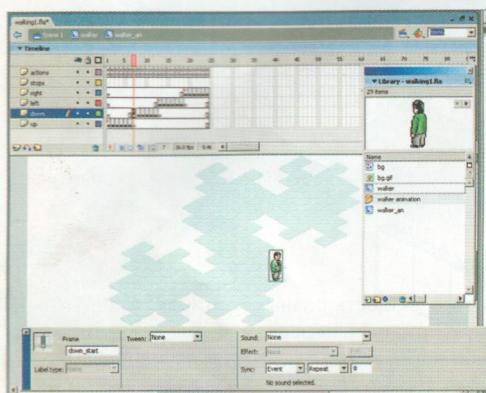
Аниматор сел рисовать начальников («толстый и лысый менеджер...»),

я начал изменять алгоритм их движения со случайного на более осмысленный — «ловить работника» (то есть началась разработка простого AI для «монстров»). Для подобных целей обычно используется алгоритм поиска кратчайшего пути, и есть множество вариантов его реализации. Я решил ограничиться упрощенным решением: определять, в каком направлении находится цель, и по нему посылать движение. Этот алгоритм «прокручивался» при каждом новом цикле анимации. Можно было бы определить весь путь до цели, записать его, а дальше просто двигаться по намеченному пути. В этом случае нужно было бы прокручивать алгоритм только один раз для конкретного случая, но в нашей ситуации цель не стоит на одном и том же месте, то есть все время двигается, что свело бы эффективность решения на нет. Нам нужно постоянно пересчитывать алгоритм, поэтому более простое решение было выбрано прежде всего по соображениям проблем скорости. После доработки алгоритма, особенно в области обхода препятствий («мебели»), оказалось, что убегать от «монстров» очень сложно. Соответственно, я ограничил «зону видимости» менеджера до определенного радиуса, за которым он двигался в почти случайному порядке, внутри же радиуса — начинал преследовать свою цель. Кое-какие факторы случайности были добавлены для того, чтобы сделать преследование менее предсказуемым.

Величина «радиуса видимости монстров» напрямую влияет на сложность игры. Можно было бы увеличивать ее с переходом на новые уровни (карты), тем самым постепенно повышая сложность игры. Гораздо круче — прибавить нового «монстра» с более высоким уровнем интеллекта (то есть с большим «радиусом видимости»). Этих новых «монстров» можно было бы включать в карты на более сложных уровнях, добавляя новые элементы и новый интерес. Так появился «начальник отдела» — опасный тип в черном костюме и темных очках, убежать от которого довольно сложно.

Итого. Сложность игры можно варьировать, меняя количество папок, которые нужно собрать, количество «монстров» и их типы, а также сам рельеф карты. Факторов вполне достаточно. А главное — общий, но настраиваемый алгоритм AI для всех монстров.

Далее в игру были добавлены дополнительные детали. Появился подсчет очков и бонусы. Были рождены на свет «жизни» — бонусы, которые добавляют общее допустимое количество попыток закончить игру. Также мы придумали «магический бонус», который замораживает монстров на какое-то время. Когда были введены все эти довольно распространенные моменты, игра приобрела свой окончательный вид.



### Первые шаги (персонажа) во Flash'e.

долго пользоваться временной заменой (куб подходящего размера) — нужно было определиться с размерами, скоростью анимации ходьбы и т.д.

В нашем проекте картинки анимации персонажа организованы по кадрам и слоям. Кроме простого кода передвижения, было добавлено пару строк для реагирования на клавиатуру. Еще несколько десятков строк — и персонаж начал двигаться, отвечая на команды. Хотя к тому моменту он еще не попадал точно в квадраты, выходил за карту и плохо реагировал на резкую смену направления, было приятно видеть первые визуальные результаты.

Я не буду описывать технические детали создания изометрического движка и сейчас объясню почему. Во-первых, в интернете накопилось очень много информации на эту тему. Для примера [www.kirupa.com/developer/actionscript/isometric\\_transforms.htm](http://www.kirupa.com/developer/actionscript/isometric_transforms.htm) — очень хорошая серия уроков. Во-вторых, технологии не стоят на месте, actionscript изменился, изменился и сам Flash. Кое-что я бы изменил, что-то поменял бы совсем. Не стоит повторять мои ошибки, пусть каждый делает как ему легче и понятнее. Главное — чтобы работало хорошо. Кроме того, этот текст не об actionscript'e, а о процессе создания игры в целом.

### 3 развитие проекта. создание игры

После нескольких экспериментов я решил перейти к организации самой игры. У нас было время обдумывать общую структуру и то, как все части будут взаимодействовать друг с другом, чтобы потом не пришлось менять что-то в корне (хотя абсолютной гарантии, конечно, никто нам не давал).

На этом этапе я создал два основных actionscript-класса: класс персонажа и класс карты. И вот настало время работать над взаимодействием этих классов, вскоре анимация была налажена: персонаж двигался ровно по квадратам, в границах карты и хорошо реагировал на команды. Итак, мы уже создали персонажей. Но что они могут без мебели? :) Тут оказалось чуть больше работы. Нужно было расставить мебель в правильном порядке перекрывания (z-order), что оказалось довольно просто, поэтому была использована структура массивов. Правда, мебель у нас занимала больше одного квадрата, а точнее, до двух квадратов ширины/длины.

Для каждого предмета мебели пришлось определять занимаемую площадь и точку регистрации на карте. Оказалось, что кое-какие конфигурации мебели не перекрываются правильно, от таких пришлось отказаться. Все эти новые детали обсуждались с аниматором, который как раз параллельно рисовал оставшуюся графику.

С мебелью было покончено. Мы взялись за добавление остальных персонажей, которые, в отличие от мебели, двигаются по карте, а значит, изменяют z-order динамически.

Вот уже персонажи ходят по карте, случайным образом меняя направление при столкновении и обходя препятствия. Тут сами собой определились правила игры: «убегать от других персонажей-монстров...» и, скажем, собирать какие-то предметы. В терминах нашего офиса — «убегать от начальника» (добра от него не дождаться) и «собирать папочки» (такая вот рутинная работа). Кроме того, можно кинуть на карту какие-то бонусные предметы для дополнительных очков.

## 4 создание редактора карт/уровней

Когда детали игры были доработаны, пришла пора создать визуальные редакторы карт, причем в достаточном количестве, а карты стали довольно сложными (на этом этапе они включали три массива, содержащие данные о форме карты, местоположении мебели и персонажей).

Началась работа над редактором карт — это та часть, которую не видят игроки, но которая облегчает генерирование новых карт. На разработку редактора мы потратили времени больше, чем собственно на игру. На этом же этапе была изменена структура movie-клипов. Вся графика, использованная в игре, была перенесена в общий файл-библиотеку. Movie-клипы из самой игры и из редактора карт привязаны к этой одной внешней библиотеке. Теперь при изменении/добавлении графики мы модифицировали что-либо только в одном месте.

В редактор также были добавлены полезные функции, облегчающие быстрое создание карт, например «закрашивание» пола определенными квадратами.

Сами карты/уровни хранятся в отдельных внешних файлах. Есть несколько решений насчет того, в каком формате хранить данные такого типа: 1) как пары переменных/значений, 2) в XML-формате, 3) в форме массивов. Также можно хранить данные во внешних файлах или получать их из базы данных. Я решил использовать массивы во внешних SWF-файлах. Почему? Объяснение простое: хотя массивы содержат больше лишней информации и не столь компактны, как пары переменных/значений, работать с ними легче и быстрее. Кроме того, SWF-формат применяет свое сжатие, что обеспечивает небольшой размер файлов — около 2 Кб для карт The Office Space.

Сохраняя карты в отдельных SWF-файлах, можно подгружать их отдельно по мере перехода с уровня на уровень, благодаря чему мы сможем использовать неограниченное количество карт не увеличивая размер самой игры. Единственный недостаток — необходимость перекомпилировать карты в SWF при изменениях. Я также не использовал базу данных для записи-чтения карт и ограничивался простым copy/paste кода массивов, который генерировал на выходе редактор карт (так как Flash не может записывать файлы на диск самостоятельно). Нужно не забыть сказать несколько слов об интерфейсе нашего редактора: он был сделан для чисто «внутреннего» использования, не было особых причин делать его слишком user-friendly.

**5 графика для игры** Что касается графики в целом, изначально мы решили придерживаться пиксельного стиля, то есть имитировать старые компьютерные игры. Во многом на наше решение повлиял вопрос скорости. Векторная графикаурская и не дает преимущества в размере при высокой детализации, а в нашей компании многие до сих пор работают на старых компьютерах.

Для игры было создано всего три разных персонажа, около 15-ти разных текстур пола, около 70-ти предметов, таких как мебель, плюс бонусы, включая несколько анимированных. Комбинируя эти элементы, можно было создавать довольно много разнообразных карт, а поскольку каждая картинка экспорттировалась в GIF-формат с наименьшим возможным количеством цветов, вся игровая графика занимает меньше ста килобайт.

**6 сюжет игры** Когда настало время создавать карты для всей игры, естественно, возник вопрос: «Сколько же карт нужно сделать?» Я стал

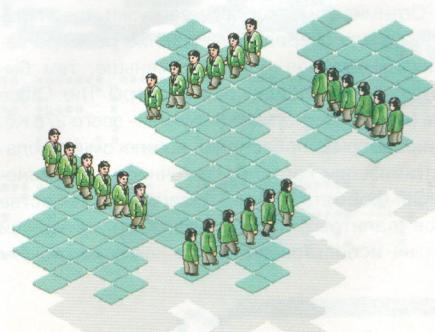
тестировать разные карты, прикидывая среднее время для прохождения одной карты, чтобы определить минимальное игровое время. Параллельно началась работа над интерфейсом: добавились подгрузчики, окна сообщений «Вы стали победителем!», «Вы проиграли!»...

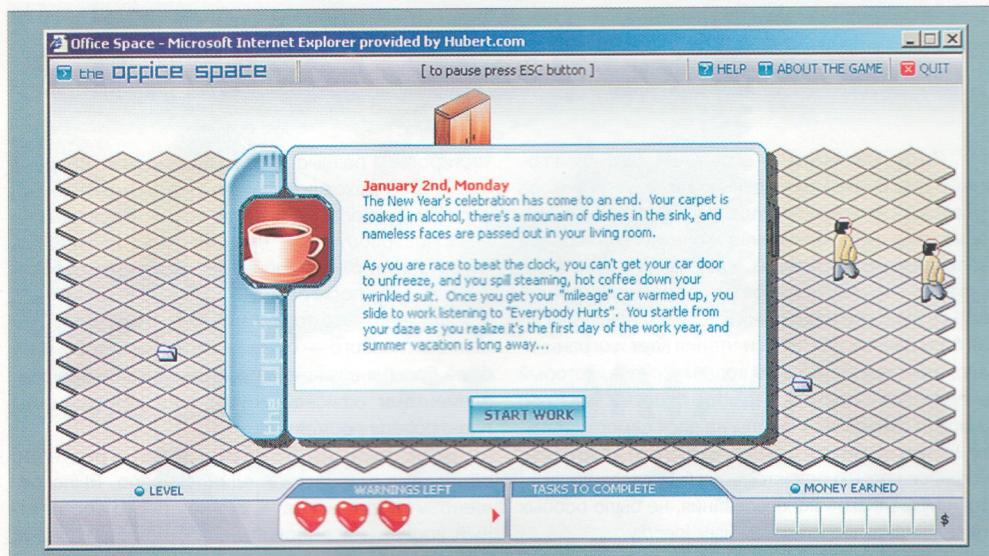
«Как-то чего-то не хватает», — подумал я. Нехватало главного — сюжета. Хотелось, чтобы игра была больше связана с идеей рутиной и с прочими элементами офисной жизни, над которыми мы планировали посмеяться.

Так родился простой сюжет игры, и он объединил все части в единое целое. Было решено, что одна карта — это не просто абстрактный уровень, а рабочий день в офисе. Каждый новый уровень — это новый день офисной жизни. Игра начинается второго января, то есть сразу после новогоднего праздника. Главный герой — невзрачный клерк, которому уже давно надоела его работа и который просыпается с мыслью о горе немытой посуды, с усталостью после буйного веселья. С самого утра у него ничего не вяжется: он проспал звонок будильника, дверь машины примерзла, горячий кофе опрокидывается... Грустя, невзрачный клерк вспоминает, что начался всего лишь первый день долгого рабочего года, а каникулы так далеки... Цель игры (то есть цель жизни персонажа) — дождаться каникул, до далекого лета, точнее, до четвертого июля (чтобы добиться символичности, мы выбрали День независимости).

И вот рождается девиз игры: «дождаться каникул», пробиваясь через ежедневную рутину и избегая неприятностей с начальством (просто избегая его вообще). Каждый рабочий день начинается с короткого текста, который сообщает что-то об очередном дне из жизни нашего героя (даже если их никто не читает, они намного лучше обычного «следующий уровень номер такой-то»).

Серия коротких текстов, объединенных общей идеей (по количеству уровней), была создана Erin Van Sickle. Каждый день мы узнаем все больше о нашем главном герое и становимся свидетелями обычной жизни неудачника. Он пробует бросить курить, пытается похудеть, смотрит плохие фильмы в одиночку, от скучи меряет длину своего стола, предвкушает уикенды и справляется любые праздники (часто воображаемые)... И ждет, ждет, когда наконец настанет время каникул.





Идея с рабочими днями и неделями также была совмещена с перезагрузкой игры: если игрок теряет все жизни, он возвращается не в самое начало, а в первый рабочий день после зарплаты (то есть каждый второй понедельник месяца).

Именно сюжет связал, наконец, все части воедино, придал смысл игре и добавил в нее юмора. Кстати, жюри I.D. Magazine особо отметил этот момент в своих отзывах, так что, видимо, наша идея достойна и хороша, а подход, выбранный нами, стоит перенять, если ты собираешься реализовать собственный проект :).

**7 интерфейс игры** Довольно часто небольшие онлайн-игры, аналогичные нашей, имеют недостаток — непривлекательный интерфейс. Наверное, после всех сотен и тысяч строк кода, исправления багов и оптимизации алгоритмов, эта часть перестает казаться важной. Главное — чтобы сама игра хорошо работала. Тем не менее игроки очень чувствительны к подобным деталям. В финальных стадиях проекта мы много работали над интерфейсом, стараясь улучшить его общий вид (правда, в игре не так много элементов интерфейса, и многие из них повторяются). Больше времени было удалено оптимизации самой графики и поиску решений, которые не слишком сильно увеличили бы размер игры.

**8 оптимизация** Финальная стадия разработки игры (параллельно с последними тестами) — это оптимизация. В случае с flash-играми у нас есть два основных типа оптимизации — оптимизация времени загрузки и оптимизация скорости работы (CPU нагрузки).

Оптимизация скорости загрузки предполагает в первую очередь уменьшение размера загружаемых файлов. Чем они меньше, тем быстрее гружаются. Общий размер игры The Office Space (в том числе музыка и звуки) — всего 278 Кб. Такой размер был обеспечен во многом благодаря оптимизации картинок, а точнее — применению GIF-формата с минимизацией количества цветов палитры. Полупрозрачность пикселей почти не использовалась, так как в противном

случае пришлось бы применять PNG, который (даже если сжат в самом Flash'e) занимал бы больше места (к примеру, пришлось отказаться от легкого отражения предметов от пола, хотя смотрелось бы довольно красиво). Количество графики также пришлось ограничить. Где возможно — одна картинка применялась много раз. Самый простой пример — одна общая картина для всех кнопок, с надписями в отдельном слое (как текст или как отдельные картинки). Другие примеры — зеркальное отражение одной и той же картинки, разделение картинок на части, если присутствуют какие-то общие элементы (к примеру, некоторые рабочие места на карте созданы из разных «кусков» рабочих мест и собраны вместе как Lego-модель).

Однако задача не сводится к оптимизации размера файлов. Есть много других способов сократить время ожидания игрока (или создать такую иллюзию). Загрузчик — самый распространенный способ. Расставив загрузчики в правильных местах, можно если не сократить время ожидания игрока, то хотя бы распределить это время по разным этапам игры. Например, предлагаю грузить все карты в самом начале игры и больше не применять никаких загрузок, но тогда начальное ожидание увеличивается. Так зачем это нужно? Грузить все карты с самого начала не имеет смысла, так как есть определенный шанс, что они и не понадобятся (не все игроки пройдут игру полностью сразу). Однако не следует злоупотреблять загрузчиками, если их много, — удобство в таком случае не повысится. В игре The Office Space каждая карта грузится отдельно перед своим уровнем. Так как карты занимают до 2 Кб, время загрузки очень мало. Другой способ — загрузка карт «блоками» (к примеру, можно было бы грузить по пять карт — одну рабочую неделю), что уменьшает количество загрузчиков и обеспечивает некоторый выигрыш в скорости (за счет одного запроса вместо нескольких).

Организация файлов также играет важную роль. Можно оставлять во внешних файлах какие-

то данные и подгружать их по мере необходимости (скажем, файл с музыкой).

Оптимизация CPU-загрузки решает проблему замедления игры на «слабых» компьютерах (понижение FPS), поскольку Flash довольно требователен к CPU-ресурсам, и это особенно важно для графической прорисовки (Flash не применяет графические оптимизаторы видеокарт).

The Office Space использует целую серию решений, обеспечивающих стабильную скорость даже на очень старых компьютерах. Прежде всего укажу на то, что игра работает на 8 FPS (восемь кадров в секунду), большинство компьютеров способно виться в такой медленный темп. Несмотря на то, что эта частота кадров слишком низкая (12 FPS рекомендуется как самое низкое значение) и анимация перестает быть гладкой, для The Office Space это решение оказалось вполне органичным, поскольку лишь дополняло стиль старых компьютерных игр. Кроме того, The Office Space работает в низком качестве, без antialiasing, что никак не влияет на качество картинок, так как все они пиксельные, а шрифты не применяют antialiasing.

Плюс ко всему, я избегал полупрозрачных movie-клипов (\_alpha), векторных градиентов и масок — это очень ресурсоемкие операции. Для имитации полупрозрачной тени (скажем, для диалоговых окон) вместо полупрозрачного фона я применял непрозрачную картинку с «дырками» размером в пиксель, расположеннымными в шахматном порядке, — как в интерфейсах старых операционных систем.

Конечно же, как правило, незачем жертвовать столькими возможностями ради оптимизации. Компьютеры становятся все мощнее, а интернет — быстрее. На самом деле в моем стремлении к оптимизации примешался спортивный интерес :). Я разрабатывал The Office Space в основном как эксперимент: «А насколько я смогу все «сжать и ускорить»?» Тем не менее никогда не стоит забывать об оптимизации. Будет очень жаль, если кто-то так и не оценит твою работу только из-за того, что пользователь не прождал лишние пару секунд...

**9 а что дальше?** The Office Space — довольно простая игра. Эта особенность становится очевиднее сейчас, когда прошло несколько лет после выпуска. Я стремился лишь рассказать о всех фазах разработки одной игры. Даже в такой простой игре, как The Office Space, их оказалось больше, чем можно представить навскидку. Я уверен, что любой, кто изучает Flash, сможет создать аналогичную игру за короткий срок. Главное — найти собственную оригинальную идею, часто такие лежат на поверхности. Используй свои знания в технологиях, чтобы имплементировать и развить идею, а не наоборот. Однако, естественно, универсального рецепта не существует ☐

## Дизайн интерфейса игры