Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт математики и фундаментальной информатики |
| институт |
| Базовая кафедра математического анализа и дифференциальных уравнений |
| кафедра |

**КУРСОВАЯ РАБОТА**

|  |
| --- |
| Игра «Сапёр» |
| тема работы |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель |  | | |  | канд. физ-мат. наук | | | И.В. Фроленков | |
|  | подпись, дата | | |  | | | | | |
| Студент | ИМ11-01Б |  |  | | |  |  | | М.С. Снетков |
|  | группа |  | номер зачетной книжки | | |  | подпись | |  |

Красноярск 2013

Оглавление

[1. Постановка задачи 2](#_Toc360191415)

[2. Применяемые технологии 3](#_Toc360191416)

[2.1. HTML 3](#_Toc360191417)

[2.2. CSS 4](#_Toc360191418)

[2.3. JavaScript 5](#_Toc360191419)

[2.4. Библиотека jQuery 6](#_Toc360191420)

[2.5. Библиотека jQuery UI 7](#_Toc360191421)

[3. Описание реализации 8](#_Toc360191422)

[3.1. Общий вид 8](#_Toc360191423)

[3.2. Структура файлов 10](#_Toc360191424)

[3.3. Реализация скрипта 13](#_Toc360191425)

[3.3.1. Виджет настроек (my.field\_grid) 14](#_Toc360191426)

[3.3.2. Виджет индикатора (my.indicator) 15](#_Toc360191427)

[3.3.3. Виджет начала новой игры (my.smile) 16](#_Toc360191428)

[3.3.4. Виджет игры (my.sapper) 17](#_Toc360191429)

[4. Список используемой литературы 19](#_Toc360191430)

1. Постановка задачи

Написать скрипт, повторяющий основные элементы игры «Сапёр».

1. Применяемые технологии
   1. HTML

HTML (от англ. HyperText Markup Language — «язык разметки гипертекста») — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986—1991 годах в стенах Европейского Центра ядерных исследований в Женеве (Швейцария). С помощью HTML можно легко создать относительно простой, но красиво оформленный документ.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег <TABLE> предназначен для создания в документах таблиц, но, часто используется и для оформления размещения элементов на странице. С течением времени основная идея платформа независимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении.

* 1. CSS

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

* 1. JavaScript

JavaScript — прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

* 1. Библиотека jQuery

jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

Точно так же, как CSS отделяет визуализацию от структуры HTML, jQuery отделяет поведение от структуры HTML. Например, вместо прямого указания на обработчик события нажатия кнопки управление передаётся jQuery, которая идентифицирует кнопки и затем преобразовывает его в обработчик события клика. Такое разделение поведения и структуры также называется принципом ненавязчивого JavaScript.

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в jQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, бо́льшая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована.

* 1. Библиотека jQuery UI

jQuery UI — библиотека JavaScript с открытым исходным кодом для создания насыщенного пользовательского интерфейса в веб-приложениях, часть проекта jQuery. Построена поверх главной библиотеки jQuery и предоставляет разработчику упрощенный доступ к её функциям взаимодействия, анимации и эффектов, а также набор виджетов.

1. Описание реализации

Исходный код курсовой работы находится по адресу <http://github.com/ufoproger/game--sapper/>.

* 1. Общий вид

Игра представляет собой виджет jQuery UI, который может быть встроен в любой блочный элемент страницы. При встраивании блочный элемент делится на два блока: игра и настройки (по-умолчанию скрыты).

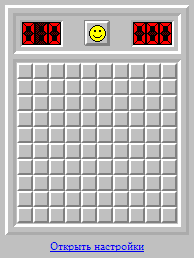


Рисунок 1. Первоначальный вид (только блок игры)

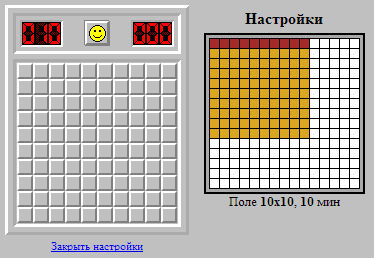


Рисунок 2. Вид после открытия панели настроек

На рис. 1 и рис. 2 изображена игра до и после открытия панели настроек.

Блок игры представляет собой панель информации, игровое поле и ссылку на открытие/закрытие настроек. Панель информации содержит два виджета индикатора. Левый индикатор отображает количество неотмеченных мин на игровом поле, а правый – количество секунд, прошедших с момента начала игры. Значение последнего индикатора является результатом, если удалось выиграть. Виджет со смайликом отображает текущее состояние игры. По нажатию на него начинает новую игру. Игровое поле полностью аналогично полю в игре «Сапёр» в поставке операционной системы Windows XP, управление так же аналогично (за исключением возможности умного открытия соседних клеток). Панель настроек представляет собой виджет настроек и информационной строки, которая отображает количество мин, а так же размеры игрового поля. Оранжевый квадрат, начинающийся в левом верхнем углу, представляет собой размеры поля (численное значение равно количеству единичных квадратов, умещающихся в соответствующей грани). Коричневые квадраты – количество мин, находящихся в игре. Количество мин имеет минимальный и максимальный пределы, меньше (или больше) которого задать количество нельзя. На рис. 3 предоставлен размер поля 13 на 13 клеток и минимальное количество мин для этого размера. На рис. 4 поле того же размера, но количество мин максимально.

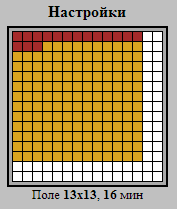


Рисунок 3. Минимальное количество мин для поля размера 13 на 13

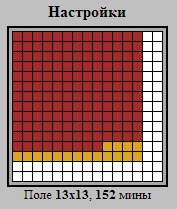


Рисунок 4. Максимальное количество мин для поля размера 13 на 13

* 1. Структура файлов

Для уменьшения нагрузки на веб-сервер во время открытия страницы с игрой, изображения были разбиты по тематическим файлам, тем самым, к примеру, несколько изображений состояния виджета новой игры уместились в один файл (рис. 5). Данная технология хранения называется спрайтами. Данная технология используется и для хранения состояний игрового поля (есть мина или нет, количество мин вокруг, флаг и вопрос), цифр индикатора – рис. 6 и рис. 7 соответственно.

C:\Users\УКП\Downloads\game--sapper-master\game--sapper-master\smiles.png

Рисунок 5. Спрайт состояний виджета новой игры

C:\Users\УКП\Downloads\game--sapper-master\game--sapper-master\field_items.png

Рисунок 6. Спрайт состояний клеток игрового поля

C:\Users\УКП\Downloads\game--sapper-master\game--sapper-master\digits.png

Рисунок 7. Спрайт значений виджета индикатора

Так же составляющей игры является файл таблицы каскадных стилей, который содержит все стили.

В главном файле скрипта описаны все виджеты, а так же вспомогательные функции. Данный файл зависит от библиотек jQuery и jQuery UI, иначе при их отсутствии появятся ошибка об неописанных функциях и переменных и игра не сможет запуститься.

На рис. 9 и рис. 12 изображены проигрыш и выигрыш соответственно. На рис. 8, рис. 10, рис. 11 показаны различные игровые ситуации.

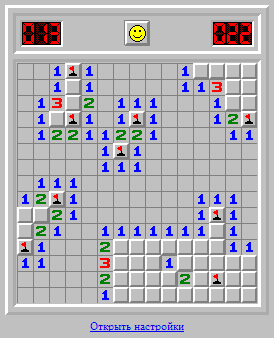


Рисунок 8. Пример игрового процесса

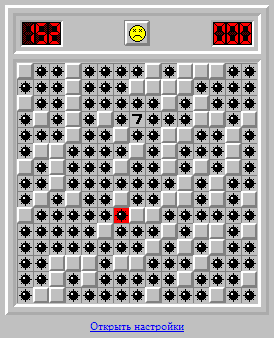


Рисунок 9. Проигрыш на втором ходу

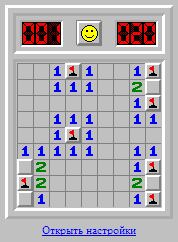


Рисунок 10. Пример успешной партии

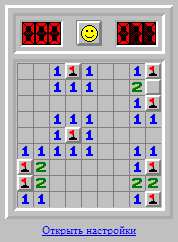


Рисунок 11. Продолжение примера успешной партии

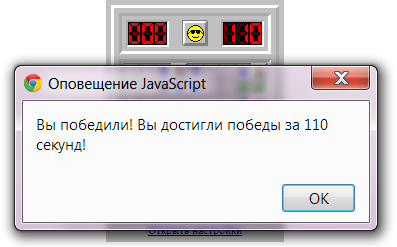


Рисунок 12. Окончание примера успешной партии

* 1. Реализация скрипта

Скрипт состоит из четырёх частей:

1. Виджет настроек (my.field\_grid),
2. Виджет индикатора (my.indicator),
3. Виджет начала новой игры (my.smile),
4. Виджет самой игры (my.sapper).
   * 1. Виджет настроек (my.field\_grid)

Представляет собой таблицу, в которой ячейки являются квадратами и могут иметь три цвета: белый, оранжевый и коричневый. Имеет конструктор (установка первоначальных значений, навешивание обработчиков событий, встраивание виджета в указанный блок), метод рендера (отрисовка виджета), а так же один вспомогательный приватный метод. Не имеет публичных методов, но при событиях указаниях нового размера игрового поля и количестве мин порождает собственное событие (field\_gridmousemove).

* + 1. Виджет индикатора (my.indicator)

Является блоком, содержащим в себе рамку и три графических изображения внутри. Имеет конструктор, метод рендера и два публичных метода: инкремент и получение/установка значения. Особенностью виджета является ограниченное количество знакомест и возможность отображать только пустое место, минус и арабские цифры как на рис. 7. В методе рендера для вывода число обрезается до трёх последних цифр (в случае положительность) и до двух (в случае отрицательности). Отсчет количества секунд, прошедших с начала игры осуществляется навешиванием таймера с интервалом в одну секунду.

* + 1. Виджет начала новой игры (my.smile)

Представляет собой графическое изображение, которое изменяется в зависимости от ситуации. Состояния представлены на рис. 5. При нажатии появляется эффект «вдавленности», при клике начинается новая игра с параметрами, указанными на панели настроек. Имеет конструктор, метод рендера, один вспомогательный приватный метод и несколько публичных методов для установки соответствующего состояния. Порождает событие (smileclick).

* + 1. Виджет игры (my.sapper)

Представляет собой сложный механизм. Состоит из нескольких блоков, которые имеют в себе виджеты и не только. Основной задачей является реакция на события от элементов игры. Содержит в себе множество публичных и приватных методов. Параллельно с реакцией на события отвечает за создание игрового поля и его обработку (каждая клетка игрового поля не является виджет, а блоков с собственным номером, по которому можно вычислить положение этой клетки в общей таблице поля). Отличительной особенностью игрового поля является то, что информация о расставленных минах хранится не в тегах клеток, а в массиве, поэтому возможность нечестной игры значительно уменьшается и требует не только поверхностных знаний об HTML.

Имеет конструктор, который создает игру и присваивает первоначальные значения внутренним переменным игры. Так же в конструкторе вызывается метод создания новой игры, чтобы сгенерировать клетки игрового поля. Примечательно, что мины расставляются по полю не во время создания новой игры, а после первого клика, чтобы исключить возможность проиграть на первом же ходу. В конструкторе навешиваются обработчики событий от различных элементов, таких как виджет новой игры, виджет настроек, ссылка на раскрытие панели настроек.

В методе начала новой игры навешиваются обработчики событий на клетки игрового поля (нажатие левой и правой кнопкой мыши, появление мыши в области, а так же её исчезновение). При первом клике по клетке с момента новой игры генерируется случайным образом набор расположения мин, а так же запускается таймер отсчета секунд с момента игры.

При клике по игровой клетке происходит сложная цепочка вызовов и проверок. Сначала проверяется, является ли клетка открытой или временной (флаг, вопрос) и какая кнопка мыши была нажата. Далее, если клетка временная, меняется её состояние на следующее (неоткрытая клетка, флаг, вопрос), а так же изменяется соответствующее значение индикатора неоткрытых мин. Если клетка является неоткрытой, то происходит проверка на наличие мины в этой клетки. Если мины нет, то игра продолжается и начинает выполняться волновой алгоритм обхода соседних клеток (реализация на основе очереди). Если количество свободных клеток вокруг равно 8, то открываются соседние клетки, иначе показывается число мин вокруг данной клетки. Если все мины обезврежены, то партия игры считается выигранной.

Если во время игры пользователь выиграл или проиграл, обработчики событий с игровых клеток удаляются, таймер останавливается, виджет новой игры получает соответствующий положению пользователя статус.

1. Список используемой литературы
2. The jQuery Foundation (2013), jQuery API Documentation. Web: <http://api.jquery.com/>.
3. The jQuery Foundation (2013), jQuery UI API Documentation. Web: <http://api.jqueryui.com/>.
4. Влад Мержевич (2002 – 2013), htmlbook – Для тех, кто делает сайты. Web: <http://htmlbook.ru>.
5. Wikimedia Foundation, Inc. (2013), jQuery – Википедия. Web: <http://ru.wikipedia.org/wiki/JQuery>