

УДК 004.942

ЭЛЕКТРОННЫЙ МУЗЕЙ ДЛЯ УЧЕБНОГО ЗАВЕДЕНИЯ

В. С. Ленч, А. Я. Аноприенко, С. В. Иваница

Донецкий национальный технический университет,
кафедра компьютерной инженерии

Описывается оптимальный подход к разработке визуальной составляющей сайта электронного музея, применение современных средств и правил создания web-страниц.

На сегодняшний день широко стала проблема переноса информации в электронный формат, так чтобы к ней можно было обратиться в любой момент. Для этой задачи лучше всего подходит технология WEB, так как доступ в Интернет на сегодняшний день не предоставляет никаких трудностей, и, почти, каждое современное устройство имеет WEB-браузер с возможностью обмена информацией по протоколу HTTP.

Страница для музея должна обладать рядом отличий и особенностей от других страниц. Доступ к информации должен осуществляться быстро и интуитивно понятно, поиск информации не должен создавать трудностей, подача информации о том или ином событии должна быть емкой и понятной. При всем этом пользователю должно быть комфортно, пользуясь элементами управления страницы.

Большинство сайтов-музеев не используют современных средств WEB-разработки. Это в основном набор статических страниц, на которые пользователю приходится переходить отдельно (например, <http://www.kmrm.com.ua>, <http://grinworld.org>, <http://museum.kpi.ua>). Такой подход значительно затрудняет поиск информации и загружает работу сервера, так как необходимо обрабатывать каждый запрос отдельно. Намного проще было бы выводить краткую информацию о каждом событии, и дать пользователю удобные средства перемещения по этой информации и удобный подробный просмотр.

Отдельную роль необходимо отвести динамической составляющей страницы. Под динамической составляющей понимаются анимационные эффекты при переходах на странице, загрузка информации с сервера без перезагрузки страницы. Раньше такие средства предоставляла технология Flash. Эта технология не является приемлемой для WEB-разработки, так как требует установки

отдельного программного обеспечения, сильно загружает систему и требует большого Интернет-трафика. Из-за изобилия устройств в современном мире (смартфоны, электронные книжки, планшетные ПК) от технологии Flash в сфере WEB-разработки стали отказываться. Вместо Flash более приемлемо стало использовать средства JavaScript, HTML5 и CSS3 для создания динамических WEB-страниц. На сегодняшний день технологии HTML5 и CSS3 предоставляют широкий набор для создания анимационных эффектов (SVG, canvas, свойства CSS transition и animation). Также анимация средствами HTML5 и CSS3 поддерживает аппаратное ускорение, что значительно повышает производительность страницы. Использование этих средств гораздо эффективней еще и тем, что код на JavaScript и CSS можно сжимать, в отличие от скомпилированных Flash приложений, что позволяет повысить скорость загрузки. Помимо динамической части, должна быть выбрана приятная цветовая палитра.

25 июня 2014 года компания Google предложила спецификацию для разработки пользовательского интерфейса для устройств Android, которая называется Material Design. Спецификация включает в себя правила создания анимационных эффектов, правила правильной подачи теней объектов интерфейса, правила выбора цветов и пиктограмм. Эта спецификация широко применяется при разработке и WEB-приложений. В современных CMS системах и WEB-фреймворках элементы управления переделаны по этой спецификации (Joomla, Angular, Bootstrap).

Также, так как для выхода в интернет используются различные устройства, которые отличаются размерами экранов и различными WEB-браузерами, то сайт необходимо разрабатывать так, чтобы он работал на любом экране и в любом браузере. Раньше для решения этой проблемы создавался отдельный WEB-домен сайта для мобильной версии сайта. Но этот подход не является эффективным. Поэтому в CSS3 появилась возможность определять размер экрана и применять соответствующий экрану стиль. Для этого применяются media-запросы. Но если создавать для каждого экрана отдельный стиль, то css-файл будет иметь слишком большой объем, что будет тормозить работу системы, поэтому обычно выбирается несколько промежуточных размера экранов, и используются относительные единицы, при установке размеров элементов. Например, шаблон, используемый во Фреймворке Bootstrap3, выглядит следующим образом:

```
@media (max-width:367px) {...}
@media (min-width:768px) {...}
@media (min-width:992px) {...}
@media (min-width:1200px) {...}
```

Для, разрабатываемой, страницы музея, используется линейка дат. Каждый элемент в линейке соответствует году, в который произошло то или иное событие. Если в этот год произошло несколько событий, то при нажатии на элемент выводятся элементы, соответствующие месяцам событий. Визуальное отображение линейки приведено на рисунке 1.



Рисунок 1 – Линейка дат

Все элементы в HTML коде имеют идентичный вид, что позволяет добавлять их динамически. Разметка линейки выглядит следующим образом:

```
<div id="rule">
    <div class="rule-element">1990</div>
</div>
```

Элементы месяцев добавляются динамически в javascript коде. Каждый элемент представлен в виде объекта класса MonthElement, конструктор которого принимает на вход родительский элемент (элемент года в линейке), CSS класс элемента месяца (.month-element) и атрибут, соответствующий месяцу. В конструкторе, помимо сохранения передаваемых параметров, указываются размеры элемента относительно размера родительского элемента и позиционирование в центре родительского элемента. Также в классе имеется три метода:

1) add – добавляет элемент в DOM (DocumentObjectModel) страницы, устанавливает угол наклона выхода элемента кратный 30° относительно предыдущего и вычисляется координата элемента в конце анимации, т.е. относительно наклона элемента:

$$y = centerY - \sin(\alpha) * r, x = centerX + \cos(\alpha) * r,$$

где r – радиус элемента; $centerY$, $centerX$ – начальная центральная координата; α – угол наклона.

2) Start – метод запуска анимации выхода всех элементов. Вызывается на первом элементе, после добавления всех элементов на странице. После вывода первого элемента

рекурсивно выбирается соседний элемент в структуре DOM, и на нем запускается такая же анимация. Таким образом, все элементы календарных месяцев отображаются на экране.

3) End – метод, аналогичный Start, но только убирает элементы со страницы и удаляет их.

При наведении курсора мыши на элементы линейки, информационная лента прокручивается на соответствующее событие. Каждый элемент информационной ленты представляет собой краткое описание события и кнопку «Подробнее», при нажатии на которую появляется модальное окно с подробным описанием события.

Навигация по информационной ленте осуществляется не только с помощью линейки, но и кнопок в виде стрелок по бокам ленты и колесика мыши. Такое отображение информации значительно экономит трафик и время на поиск информации, так как не нужно переходить от страницы к странице, а все происходит в одном месте. На рисунке 2 показан элемент информационной ленты в кратком и подробном виде.

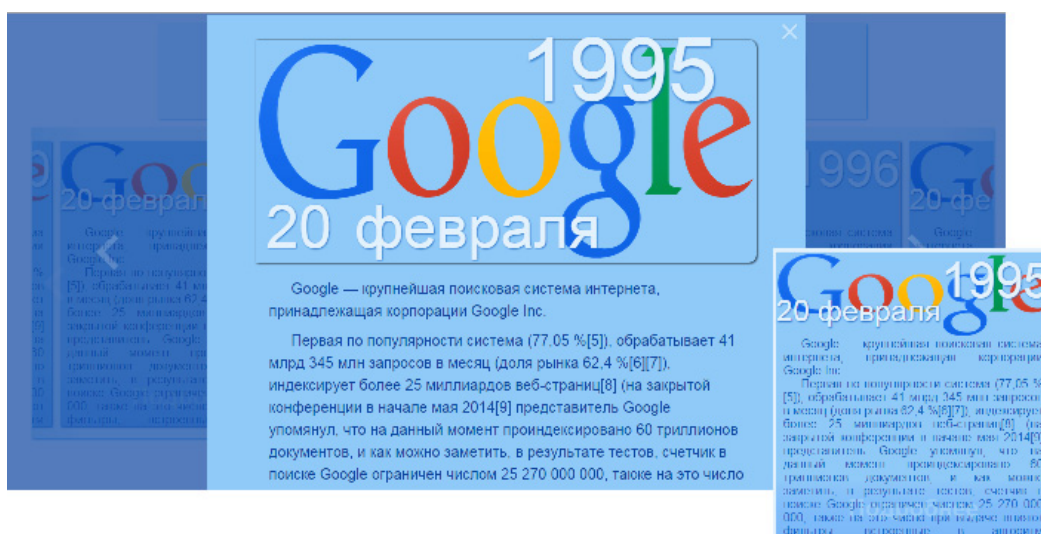


Рисунок 2 – Краткое и подробное отображение информации в ленте

Все элементы информационной ленты, как и элементы линейки, имеют идентичный вид, что позволяет систематизировать добавление информации на страницу. Шаблон HTML разметки информационного элемента выглядит следующим образом:

```
<div class="info-element" year="<год>"
month="<месяц>">
    
    <h2 class="info-year"><год></h2>
```

```
<h3 class="info-day"><день></h3>
<h3 class="info-month"><месяц></h3>
<div class="info-annotation">
  <информация>
</div>
<div class="info-detail">Подробнее</div>
</div>
```

Здесь элементы `<год>`, `<день>`, `<месяц>` следует заменить на значения даты события, а `<информация>` – на html разметку информации о событии. Атрибуты `year` и `month` в родительском элементе являются определяющими для поиска элементов с помощью линейки дат.

Каждое действие на странице сопровождается анимационным эффектом. Анимационные эффекты имеют функции появления или исчезновения элемента со страницы, описанные в спецификации по MaterialDesign (рис. 3).

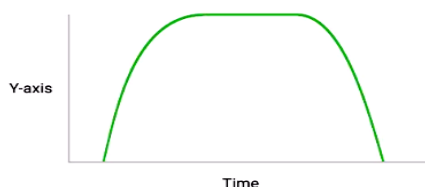


Рисунок 3 – Функция анимационного появления/исчезновения элемента

Из рис. 3 следует, что функция появления должна быть симметрична функции исчезновения, т. е. элемент должен резко появляться и замедляться к окончанию анимации, а при исчезновении должен плавно уходить и постепенно ускоряться. В данной работе используются функции $y = x^3$ и $y = x^4$. Цветовая палитра была выбрана также из спецификации по MaterialDesign.

Вывод

Таким образом, был разработан эффективный дизайн для сайта музея, который имеет удобный интерфейс и удобные средства поиска информации, при этом, не прибегая программному обеспечению сторонних разработчиков.

Список источников

1. Адаптивный и мобильный дизайн с MediaQueriesCSS3 — [Электронный ресурс] — <http://habrahabr.ru/post/119127>.
2. CSS media queries – Web developer guide — [Электронный ресурс] — https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries.
3. Спецификация MaterialDesign — [Электронный ресурс] — <http://www.google.com/design/spec/material-design/introduction.html>.