УЧЕБНАЯ МАСТЕРСКАЯ

Романовский Иосиф Владимирович,
Мойсеюк Владимир Григорьевич

АЛЬБОМ «САНКТ-ПЕТЕРБУРГ» В ФОРМАТЕ HTML: МАРШРУТНЫЕ ОГНИ ТРАМВАЯ

Эта статья описывает работу, в которой мне помогала студентка математико-механического факультета СПбГУ Н. Лепескина.

И. Романовский

1. СИГНАЛЬНЫЕ ОГНИ ТРАМВАЕВ

Когда в Петербурге, а затем Ленинграде, развивалось трамвайное движение, была придумана очень удобная и красивая система маркировки маршрутов: на каждом трамвайном поезде спереди и сзади горели цветные огоньки. Эти огни называются *маршрутными*.

Они очень удобны в осенние и зимние сумерки, когда и ждать-то на улице неприятно, а при тумане и моросящем дожде номера трамвая не распознать... – яркие огоньки шестерки, или сорокового, или тройки весело сигналят издалека.

Используются фонари пяти цветов – синий, красный, зеленый, белый, желтый. Легко подсчитать, что это дает нам 25 возможностей. Так как маршрутов в городе много больше, то некоторые имеют одинаковые комбинации цветов. Комбинации стараются сохранять, но, по-моему, они все-таки изредка изменяются. Расцветки для нашей страницы взяты из Интернета.

Хотелось дать возможность посмотреть, как выглядит трамвай любого маршрута со своими огнями, просто выбрав в таблице нужный номер.

2. ПРОЕКТИРОВАНИЕ СТРАНИЦЫ

Мы решили взять какой-нибудь типичный городской пейзаж, сфотографировать идущий трамвай, взять этот снимок за основу, а при выборе конкретного маршрута вклеивать в снимок небольшую заплатку. Все необходимые заплатки приготовила Н. Лепескина. Картинки были все одного размера 17×76 пикселей. Вначале это были 65 маленьких графических файлов, затем мы соединили их в одну полоску высотой в 1105 пикселей.

На рис. 1 вы можете увидеть получившуюся страницу. Сфотографирован трамвай 20-го маршрута, но в нужное место уже вставлена заплатка, соответствующая маршруту 1. Справа внизу, после уже знакомого нам текста, расположена таблица выбранных для показа 65 номеров (маршрутом 90 мы пожертвовали).

3. ПРИНЦИПИАЛЬНАЯ СХЕМА СТРАНИЦЫ

Посмотрим, из каких частей составлен html-текст нашей страницы (см. листинг 1).

Рис. 1

Листинг 1.

```
<html>
обычный заголовок
<script language="javascript">
  текст процедуры
</script>
</head>
<body scroll=no><table width=900><tr><td>
<center><h1 class=sec>Маршрутные огни трамвая</h1>
<table class=norm width=900 height=600 align="center">
<tr><td class=norm>
<img src="JPG/tra_tra_1.jpg" hspace="10">
<img  name="zapl" style=". . . . . src="JPG/tra_tra_1_t.jpg">
<td class=norm>У петербургского трамвая ...
  текстовое заполнение страницы
<p>Вы можете посмотреть, как выглядит трамвай на этой сумеречной
картинке, кликнув в таблице тот номер маршрута, который вас заинтересовал.<br><br>

<table border="1" bordercolor="#b0b0b0" width="400" >
 <tr align="center">
  <td><span id="1" onClick="change(1)" style="cursor: pointer;">1</span></td>
   . . . . . .
  <td><span id="1" onClick="change(13)" style="cursor: pointer;">13</span></td>
 </tr>
 . . . . . .
 <tr align="center">
  <td><span id="1" onClick="change(53)" style="cursor: pointer;">53</span></td>
   . . . . . .
  <td><span id="1" onClick="change(65)" style="cursor: pointer;">65</span></td>
 </tr>
</table>
<q class=sm>В разработке этой страницы активное участие принимала студентка
математико-механического факультета СПбГУ Н. Лепескина (2007).</q>
</td></tr>
</table></center>
</td></tr></table></body>
</html>
```

Обратите внимание на совершенно одинаковые заполнения клеточек таблицы: с каждой клеткой связан ее номер, и в ответ на кликание этого номера вызывается процедура **change(1)** с номером клетки (и маршрута) в качестве параметра.

Эта процедура написана на языке **javascript**. Она изменяет параметры изображения **zapl**, которое у нас описано не полностью.

4. ЗАПЛАТКА

Полное описание заплатки выглядит первоначально так

```
<img name="zapl"
 style="position:absolute;
     top:290px; left:225px;
     clip: rect(0, 76, 17, 0);
     overflow: hidden; "
 src="JPG/tra_tra_l_t.jpg">
```

Этим описанием определяется имя изображения для заплатки, ее начальное изготовление и размещение в основной картинке. Параметры **position:absolute; top:290px; left:225px;** задают положение заплатки, параметр **clip: rect(0, 76, 17, 0);** определяет ее расположение в сводном файле заплаток, параметр **overflow: hidden;** защищает нас от неприятностей, возникающих из-за того, что очень высокая картинка заплаток вылезает за пределы фотографии с трамваем (мы берем из файла заплаток только маленький кусочек и ставим его правильно, но размеры картинок будут сравниваться полностью, если мы не изменим правил сравнения).

...мы берем из файла заплаток только маленький кусочек и ставим его правильно...

Все эти параметры помещены в строковом параметре **style**, который и будет изменяться в процедуре **change**.

5. ПРОЦЕДУРА

Пора обратиться к описанию процедуры

```
function change (v) {
 var top_z = (v-1)*17;
 var bottom_z = v*17;
 var s = "rect(" + String(top_z)
  + ",76," + String(bottom_z) + ",0)"
 zapl.style.clip = s;
 zapl.style.top = 290 - top_z;
}
```

Все очень просто: вычисляем верхнюю и нижнюю границы требуемой заплатки. По ним формируем описание параметров прямоугольника (системная функция **String** переводит числовой параметр в строку) и правильное положение сводного файла заплаток относительно снимка, при котором выбранная заплатка попадает на нужное место.

Затем (как изящно!) в объекте по имени **zapl** внутри параметра **style** простым присваиванием изменяются нужные атрибуты.

6. ТЕГ <SPAN>

Сейчас все большую популярность приобретают сайты с многослойной структурой. В них элементы страницы накладываются друг на друга слоями – каждый следующий кодируемый элемент содержится в слое, который располагается поверх предыдущих элементов. Как правило, это разделение на слои не очевидно, и его не требуется знать, так как элементы страницы обычно не перекрываются. Однако, когда элементы явно позиционируются на странице, они могут перекрывать-

ся, делая разбиение на слои очевидным.

Как вы могли догадаться, это и происходит, когда вы нажимаете на ссылку с номером маршрута. На статической картинке появляется ещё один слой, в котором содержится наша заплатка.

Возникает вопрос: «Неужели нам каждый раз необходимо переопределять заново каждый последующий слой?» Нет. Имеется тег ****, который позволяет выделить часть информации внутри других тегов и установить для нее свой стиль. То есть нам достаточно один раз установить место для заплатки, и все последующие заплатки будут появляться в том же месте и в том же слое. При этом, благодаря встроенным событиям, мы можем производить изменения стиля уже созданной статической страницы, не загружая ее заново.

Здесь тег **** используется с атрибутами **id="1"** – уникальный номер стиля; **onClick="change(1)"** встроенный обработчик события, который при нажатии вызывает функцию **change; style="cursor: pointer;"** описывает стиль курсора (указатель). Стоит также отметить, что в последнем атрибуте **style** можно описывать стиль всего, что мы обычно описываем в CSS. Можно было бы для сокращения текста отдельно задать параметры ****, сократив в 65 местах длину вызова.

Романовский Иосиф Владимирович, доктор физико-математических наук, профессор СПбГУ,

Мойсеюк Владимир Григорьевич, студент 4 курса математико-механического факультета СПбГУ.