

А. Е. ЗАГОРСКИЙ, М. В. МАЛАЩЕНКО, В. В. ШЕПЕЛЕВИЧ
МГПУ им. И.П. Шамякина (г. Мозырь, Беларусь)

ПОСТРОЕНИЕ ИНТЕРАКТИВНЫХ МОДЕЛЕЙ В HTML5

В настоящее время стандарт языка HTML5 активно внедряется многими веб-разработчиками. С использованием этого популярного языка гипертекстовой разметки страниц разворачиваются крупные веб-порталы, коммерческие и развлекательные сайты (см., например, статистику использования HTML5 [1]). Новый стандарт включил в язык html большое количество удобных инструментов. Появилась возможность добавлять на страницу видео и аудиозаписи, размещать на холсте интерактивные рисунки, использовать возможности геолокации и взаимодействия с пользователем посредством форм и локальных хранилищ данных [2].

Указанная выше интерактивность HTML5 представляет большой интерес для образовательных задач на различных ступенях обучения. Создавая наглядные модели изучаемых явлений и предметов с возможностью взаимодействия с пользователем и изменения параметров исследуемой системы, можно значительно повысить качество усвоения материала с меньшими затратами времени. В этом смысле стандарт HTML5 является не только передовым решением с точки зрения технологии создания моделей, но и предоставляет разработчикам невиданные ранее инструменты по созданию интерактивных и виртуальных моделей. Ещё одним плюсом при использовании HTML5 для моделирования является его общедоступность: любой современный браузер понимает данный стандарт и корректно с ним работает.

Рассмотрим основные приёмы создания интерактивных моделей в HTML5 с использованием холста (canvas). Структура html документа в нашем случае будет предельно проста:

```
<!DOCTYPE HTML>
<html>
  <head>
    <script src="model.js"></script>
  </head>
  <body>
    <canvas id="example"></canvas>
    <script>init()</script>
  </body>
</html>
```

Заголовок документа содержит строку с подключаемым модулем `model.js`. В тело документа добавлен холст с идентификатором `example` и вызов метода `init()` из модуля `model.js`. Свойства холста и методы взаимодействия с пользователем реализованы на языке JavaScript в модуле `model.js`, рассмотренном ниже.

В начале модуля заданы классы `round` и `rect`, экземпляры этих классов используются для создания фона холста и точки на экране, с которой будет взаимодействовать пользователь.

```
function round(x, y, r) // класс, задающий круг
{
  this.x = x; // координата x
  this.y = y; // координата y
  this.r = r; // радиус
  this.draw = function draw(color, globalAlpha)
  {
    context.globalAlpha = globalAlpha;
    context.fillStyle = color;
    context.beginPath();
    context.arc(this.x, this.y, this.r, 0, Math.PI * 2, true);
    context.fill();
  };
}
function rect(x, y, width, height) // класс, задающий прямоугольник
{
  this.x = x; // координата x
  this.y = y; // координата y
  this.width = width; // ширина
  this.height = height; // высота
  this.draw = function(color, globalAlpha)
  {
    context.globalAlpha = globalAlpha;
    context.fillStyle = color;
    context.fillRect(this.x, this.y, this.width, this.height);
  };
}
```

Следующим шагом будет описание метода, реагирующего на возникновение события `mousedown` (пользователь щёлкнул по холсту). Результатом работы метода будет перемещение точки в позицию, заданную курсором мыши.

```
function mousedown(evt) {
  var mouseX = evt.pageX - canvas.offsetLeft;
  var mouseY = evt.pageY - canvas.offsetTop;
  point.x = mouseX;
  point.y = mouseY;
}
```

Элементы холста необходимо отобразить, это задача метода `draw`.

```
function draw()
{
  background.draw("#ffa", 0.5); // закрашиваем фон
  point.draw("#00f", 0.5); // рисуем точку
}
```

Заключительным блоком в программе будет метод `init`, именно он вызывается из `html` документа. Метод `init` создает экземпляры описанных ранее классов фона и точки, устанавливает связь между объектами и холстом и затем отображает полученную интерактивную модель, реагирующую на действия пользователя.

```
function init()
{
  background = new rect(0, 0, 800, 400);
  point = new round(400, 285, 10);
  canvas = document.getElementById("example");
  canvas.width = background.width;
  canvas.height = background.height;
  context = canvas.getContext("2d");
  canvas.onmousedown = mousedown;
  setInterval(draw, 1000 / 12); // перерисовываем 12 раз в секунду
}
```

Используя приведенный выше подход несложно получить гораздо более сложные модели, в том числе с интерактивными изображениями, взаимодействующими объектами, учетом физических закономерностей.

ЛИТЕРАТУРА

1. Websites using HTML5 DocType [Электронный ресурс] / <http://www.trends.builtwith.com>. – Режим доступа: <http://trends.builtwith.com/docinfo/HTML5-DocType>. – Дата доступа: 03.02.2014.
2. HTML5 – введение в новый стандарт [Электронный ресурс] / <http://www.adnotes.ru>. – Режим доступа: <http://adnotes.ru/page/html5-vvedenie-v-novyy-standart-chast-1>. – Дата доступа: 03.02.2014.

МГТУ им. И.П.Шамякина