

УДК 004.932.2

**РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА ДЛЯ ЯЗЫКА JAVASCRIPT****Степанов С.С., Костин В.И.**

*Донецкий национальный технический университет, г. Донецк  
кафедра прикладной математики и информатики  
E-mail: Doctor07s@mail.ru*

**Аннотация:**

**Степанов С.С., Костин В.И. Разработка графического редактора для языка JavaScript.** Задачи, решаемые с помощью графики в мире Web, очень разнообразны - это и построение графиков, и создание диаграмм, создание различных нестандартных пользовательских интерфейсов и т.д. Но с ростом производительности аппаратных средств и возможностей существующих технологий растут и требования к функционалу Web-приложений. На сегодняшний день язык JavaScript дает широкие возможности для реализации поставленных задач. В основном – это графические библиотеки. Это направление стремительно развивается. С каждым днем появляются новые библиотеки с более широкими возможностями. Это говорит о том, что графика в Web становится все более и более востребованной.

**Общая постановка проблемы**

В настоящее время появилось некоторое количество графических библиотек, написанных на языке JavaScript. Каждая из них имеет свои сильные стороны и недостатки. Изначально все графические библиотеки можно разделить на следующие категории:

- библиотеки для анимации;
- библиотеки для рисования;
- библиотеки для рисования диаграмм;
- комплексные библиотеки.

К числу библиотек для анимации относятся:

- библиотека \$fx() для анимации html элементов;
- Sylvester — JavaScript Library;
- Facebook Animation Library.

Библиотеки для рисования:

- Библиотека для простого рисования с помощью jQuery;
- jsDraw2D - 2D графическая библиотека.

Библиотеки для рисования диаграмм:

- [Flot](#);
- [PlotKit](#);
- [Emprise JavaScript Charts](#).

Комплексные библиотеки:

- Processing.js библиотека;
- Raphael—JavaScript библиотека для работы с векторной графикой.

Библиотека \$fx() – это небольшая библиотека для анимации элементов html. Она позволяет изменять любые свойства CSS и проста для конфигурирования: можно комбинировать эффекты, группировать их и запускать параллельно и в произвольном порядке. На основе просто анимации можно организовывать более сложные эффекты.

Sylvester – это JavaScript библиотека для работы с векторами и матрицами без необходимости записывать множественные циклы и стопки массивов. Она включает классы для моделирования векторов и матриц в любых величинах и для моделирования большого количества линий и проекций в 3-хмерном пространстве. Она позволяет писать объектно-ориентированный читабельный код.

[Flot](#) – библиотека, построена на jQuery и разработана в [IOLA](#). Легкая в использовании, она имеет несколько дополнительных функций, таких как: увеличение при помощи мыши (кликните и тащите нужную область) и возможность определения координат клика. Достоинство библиотеки: кроссбраузерность. Работает в следующих браузерах: Internet Explorer 6/7/8 (IE8 только в версии для разработчиков), Firefox 2.x+, Safari 3.0+, Opera 9.5+ и Konqueror 4.x+.

Processing.js - это открытый язык программирования для изображений и анимации, а также для их взаимодействия в Интернете без использования Flash и Java апплетов. Processing.js использует Javascript для рисования форм и манипулирования изображениями в элементах [HTML](#). Код достаточно легкий и простой для изучения, поэтому это хороший инструмент для визуализации данных, создания интерфейсов пользователей и разработки интернет-игр.

Raphael – это маленькая библиотека JavaScript, которая упрощает работу с векторной графикой в вебе. Она позволяет создавать графики, диаграммы, схемы, а так же работает с изображениями. Raphael использует SVG W3C рекомендации и VML для построения графики. Это значит, что любой графический объект – это DOM объект. Задача библиотеки – обеспечить возможность простого и кроссбраузерного векторного рисования.

Для реализации анимации на сайте среди описанных библиотек можно использовать библиотеку *\$fx()*, которая работает с css свойствами объектов, и потому обладает кроссбраузерностью, а так же бесплатна. По такому же принципу работают и *Facebook Animation Library*, но вся документация для *Facebook Animation Library* на английском языке и в ней сложно разобраться.

Для рисования в JavaScript можно использовать библиотеку jsDraw2D, так как набор основных функций для рисования в основном одинаков у всех библиотек. К тому же она так же работает во всех браузерах и достаточно проста.

Среди библиотек для рисования графиков очень выделяется [Emprise JavaScript Charts](#), но мы не можем воспользоваться ею с силу того, что эта библиотека платная.

Отдельно можно выделить комплексные библиотеки, поскольку они позволяют как рисовать примитивы, так и рисовать графики, управлять клавиатурой, работать с мышкой, создавать анимацию, дописывать свои методы и функции, т.е. являются более гибкими и функциональными по сравнению с отдельно взятыми библиотеками для анимации и рисования примитивов. Но они очень сложны и содержат излишнее количество функций, что делают их более тяжелыми.

Поэтому была выбрана библиотека jsDraw2D, поскольку она обладает следующими достоинствами:

- библиотека дает широкие функциональные возможности для рисования;
- является бесплатной.

js Draw2D – это библиотека, написанная на JavaScript, для рисования 2D-графики на Web-страницах внутри браузера без использования SVG и VML. Все пользователи могут воспользоваться библиотекой, чтобы добавить графики функций рисования в свои Web-приложения или Web-сайты.

js Draw2D – библиотека с открытым кодом, свободно распространяется по лицензии LGPL. Дополнительного программного обеспечения для запуска не требует.

Особенности:

- С помощью *jsDraw2D* можно нарисовать такие фигуры, как кубическая кривая Безье, общая кривая Безье, открытой кривой, проходящей через заданные точки, и замкнутой кривой.
- Можно нарисовать все основные фигуры, как линия, прямоугольник, полилинии, полигон, окружность, эллипс и дуги. Также можно нарисовать текст или изображение в указанном месте.
- Можно рисовать закрашенные прямоугольники, многоугольники, окружности, эллипсы, дуги и замкнутые кривые.
- По умолчанию начальные координаты (0,0) документа или элемента Div находятся в верхнем левом углу, но с *jsDraw2D* можно установить начальные координаты для рисования в любое место / точку на документе или Div-элемент.
- *jsDraw2D* provide two options for co-ordinate system; "default" and "cartecian". In "default" system, on x axis, values above origin are positive and values below origin are negative while in "cartecian" system it is opposite. *jsDraw2D* предоставляет два варианта системы координат: "по умолчанию" и "декартовы". В системе "по умолчанию" по оси x выше находятся отрицательные значения, а положительные – ниже, а в "декартовых" координатах - наоборот

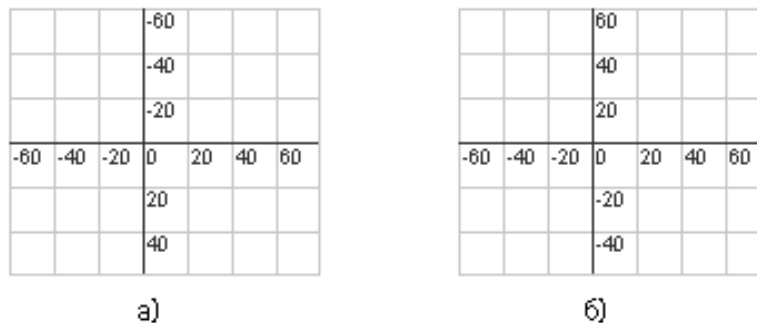


Рис. 1. Координаты в *jsDraw2D*: а) по умолчанию, б) декартовы

- Еще одной важной особенностью является то, что можно задать масштаб рисунка и вместо использования 1px за единицу, можно задать собственные значения шкалы.

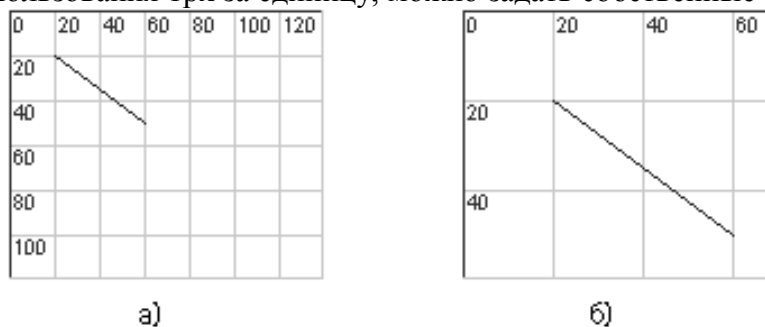


Рис. 2. Масштабирование в *jsDraw2D*: а) без масштабирования, б) с масштабированием

- Как можно видеть на диаграмме выше, можно опционально показывать сетку / оси или диапазон по X и Y оси.
- Поддерживает большинство современных браузеров. Но результаты могут быть немного разными в разных браузерах. Проверено на IE 7, Mozilla Firefox 3, Google Chrome, Opera 9.64 и Safari 4 с Windows XP.
- Библиотека включает *jsColor* класс, который обеспечивает различные цветовые функциональности например, конвертирование RGB в шестнадцатеричную и наоборот, получить ярких или темных теней данного цвета.

- Библиотека разработана с использованием объектно-ориентированного подхода. Это позволяет разработчикам иметь много преимуществ объектно-ориентированного дизайна.
- Основным недостатком библиотеки является отсутствие возможности сохранения рисунка.

### **Выводы**

В результате анализа существующих программных продуктов и их возможностей были сформулированы и реализованы следующие технические требования к разрабатываемому графическому редактору:

- организация системы в виде сервиса, работающего на площадке Web-хостинга;
- кроссбраузерная реализация системы.

Функциональные возможности:

- рисование простых фигур (линия, круг, эллипс, прямоугольник);
- рисование сложных фигур (многоугольник, кривая, замкнутая кривая, кривая Базье);
- параметры рисования (ширина линии, цвет);
- наличие подсказок;
- сохранение рисунка.

Особенностью возможности сохранения рисунка в виде кода в файл с расширением txt является то, что библиотека jsDraw2D написана на языке JavaScript, который, как известно, не «работает» с файлами.

В настоящее время реализован графический интерфейс, который позволяет без особых трудностей использовать возможности графической библиотеки jsDraw2D.

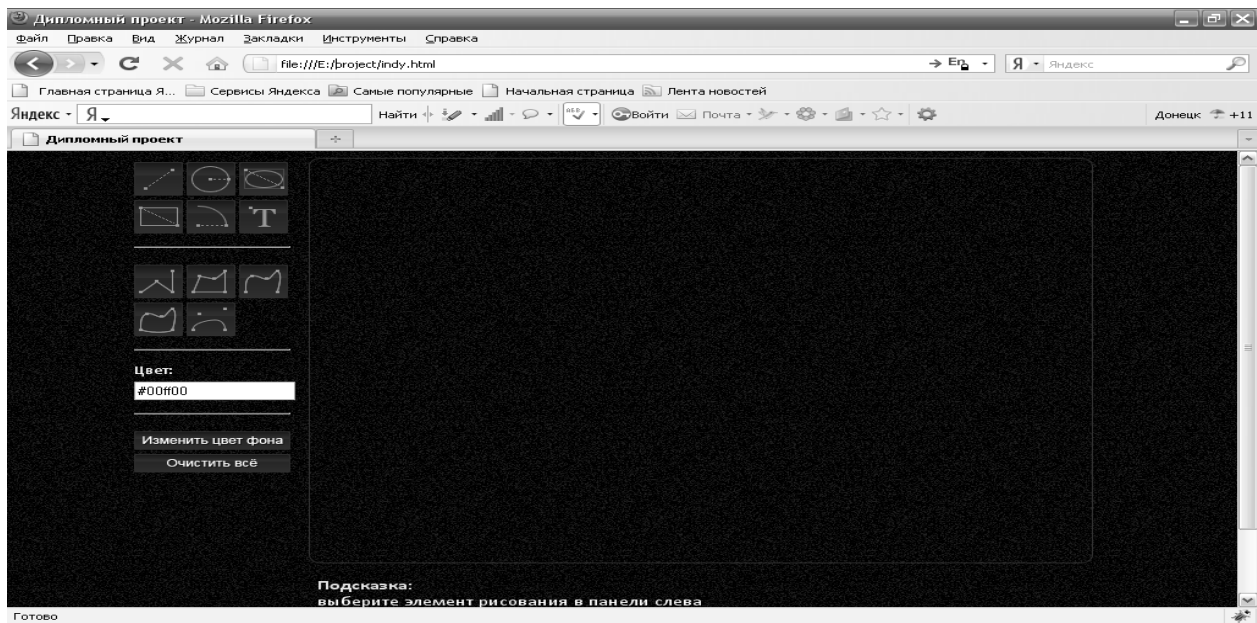


Рис. 3. Графический интерфейс

Создана группа кнопок для рисования простых фигур:

- а) рисование линии;
- б) рисование круга;
- в) рисование эллипса;
- г) рисование прямоугольника;
- д) рисование дуги.

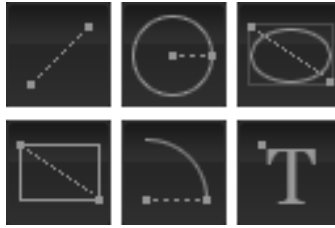


Рис. 4. Кнопки для рисования простых фигур

Так же была создана группа кнопок для рисования сложных фигур:

- а) рисование ломаной линии;
- б) рисование многоугольника;
- в) рисование кривой;
- г) рисование замкнутой кривой;
- д) рисование кривой Безье.

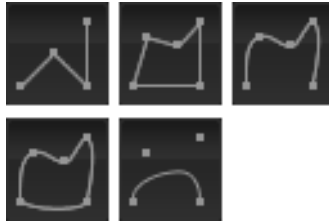


Рис. 5. Кнопки рисования сложных фигур

Кроме того, есть ещё группа задания параметров рисования:

- а) ширина линии,
- б) цвет. Задаёт цвет контура фигуры, а так же цвет заливки фигуры, если она заполненная. А так же задаёт цвет фона при его изменении.
- в) выбор рисования заполненной фигуры или залитой,
- г) начальный угол. Используется для рисования дуги и сектора. Изменяется в градусах.
- д) конечный угол. Используется для рисования дуги и сектора. Изменяется в градусах.
- е) степень кривизны. Используется для рисования кривых.

Ширина: <input type="text" value="1"/> Цвет: <input type="text" value="#1e291e"/> Кривизна: <input type="text" value="0"/>	<b>Шрифт:</b> Название: <input type="text" value="Arial"/> Размер: <input type="text" value="12"/>	<b>В градусах:</b> Начальный угол: <input type="text" value="0"/> Конечный угол: <input type="text" value="45"/>
<input type="checkbox"/> <b>заполненный</b>		

Рис. 6. Изменяемые параметры

### Список литературы

1. Аллен Вайк JavaScript. Энциклопедия пользователя, Киев, - 2001. - 480 С.
2. Дэвид Флэнаган JavaScript Подробное руководство, Санкт-Петербург, - 2008. - 984 С.
3. JavaScript Graphics Library [Electronic resource] / Интернет-ресурс. - Режим доступа : www/ URL: [http:// jsdraw2d.jsfiction.com](http://jsdraw2d.jsfiction.com). - Загл. с экрана.