

УДК 004.42+004.925+004.94:544.1

ТРЕХМЕРНАЯ БРАУЗЕРНАЯ ВИЗУАЛИЗАЦИЯ МОЛЕКУЛ С ПОМОЩЬЮ ТЕХНОЛОГИИ WEBGL

Вильданов А.Н., Шафеева Е.П.

ФГБОУ ВПО «Башкирский государственный университет», Нефтекамский филиал, г. Нефтекамск, e-mail: alvild@mail.ru

Разработан AJAX-загрузчик для отображения трехмерных моделей молекул различных химических веществ в браузере с помощью новейшей технологии WebGL. Приложение берет информацию из готовых файлов программы Tinker с расширением .xyz, в которых хранятся данные о трехмерных координатах атомов и о связях между ними, и строит по ним трехмерную модель молекулы на WebGL. Разработанный алгоритм демонстрируется на молекулах этанола, бензола, пропана, и т.д., и универсален для всех файлов с расширением .xyz. Для облегчения работы с WebGL используется библиотека Three.js. Приведены основные моменты развертывания приложения. Особенностью разработанного приложения (как и WebGL-приложений вообще) является возможность просмотра готовых моделей непосредственно в браузере, без установки дополнительного программного обеспечения. Результаты работы можно использовать, например, для демонстрации строения отдельных молекул на уроках химии, и для построения webgl-приложений.

Ключевые слова: WebGL, Three.js, AJAX, JavaScript, 3D технология, визуализация, молекула, Tinker.

THREE-DIMENSIONAL BROWSER VISUALIZATION OF MOLECULES BY MEANS OF THE WEBGL TECHNOLOGY

Vildanov A.N., Shafeeva E.P.

Neftkamsk branch of Bashkir State University, Neftkamsk, Republic of Bashkortostan, Russia, e-mail: alvild@mail.ru

The AJAX loader was developed to display three-dimensional models of various chemical molecules in a browser by means of the latest WebGL technology. The application works with the programme Tinker, takes the information from its files with the extension .xyz, in which three-dimensional atomic coordinates' data is stored, and by means of them constructs a three-dimensional model of a molecule with WebGL. The developed algorithm is demonstrated on molecules of ethanol, benzene, propane etc., it is universal for all kinds of files in this format. Using the library Three.js makes the work with Three.js easier. One of the WebGL-applications' features is the possibility of viewing ready models directly in a browser without installation of any additional software. The results of the work one can use, for example, to demonstrate a structure of molecules during chemistry lessons, and to create WebGL-applications.

Keywords: WebGL, Three.js, AJAX, JavaScript, 3D technology, rendering, molecule, Tinker.

С развитием **Web** стал неизбежно нуждаться в **3D** возможностях. Сначала появился объект **HTML5 Canvas**. Он предназначен для создания растрового двухмерного изображения при помощи языка JavaScript, и используется для построения графики, простой анимации и игр в браузерах.

Ясно, что графика в виде **3D** (трехмерная) выглядит эффектнее и реалистичнее, чем традиционная двухмерная графика. Кроме того, она позволяет лучше представить объект в том виде, в котором он может быть в реальности, рассмотреть его со всех сторон. Все это позволило трехмерной графике найти широкое применение, и в настоящий момент **3D** графика стала необходимым инструментом в работе большого числа приложений. Но, до недавнего времени, возможности **3D** графики (в браузерах) сильно ограничивались возможностями процессора.

Высокопроизводительную графику можно было создать, только преодолев табу использования в браузерах возможностей видеоадаптера. В начале 2009 года **Mozilla** и **Khronos** организовали рабочую группу **WebGL**.

WebGL (Web-based Graphics Library) – программная библиотека, предназначенная для создания интерактивной трехмерной графики в веб-браузерах с использованием ресурсов графического процессора [1].

За счёт использования низкоуровневых средств поддержки **OpenGL** часть кода на **WebGL** может выполняться непосредственно на видеокартах, что дает выигрыш по быстродействию.

Благодаря технологии **WebGL** можно обрабатывать и показывать реальные **3D** модели непосредственно на страницах сайта. При помощи мыши можно поворачивать, перемещать и масштабировать модель, чтобы рассмотреть все ракурсы и интересные детали. Не нужно устанавливать дополнительных приложений, достаточно открыть в браузере нужную ссылку и любоваться результатом [1, 3].

Границы применимости **WebGL** неизменно расширяются, и эта тенденция будет только возрастать. Данная работа посвящена использованию возможностей **WebGL** для визуализации трехмерных моделей химических молекул различных химических веществ.

1. Описание приложения

Разработанное приложение выводит трехмерные модели химических молекул различных химических веществ. Приложение написано с помощью скриптового языка JavaScript, языка разметки HTML и каскадной таблицы стилей css, и представляет собой набор нескольких обычных текстовых файлов.

Рабочее поле программы представляет собой окно с моделью молекулы, которая медленно вращается (рисунок 1). С помощью мышки можно управлять положением молекулы, рассматривать ее с любых сторон.

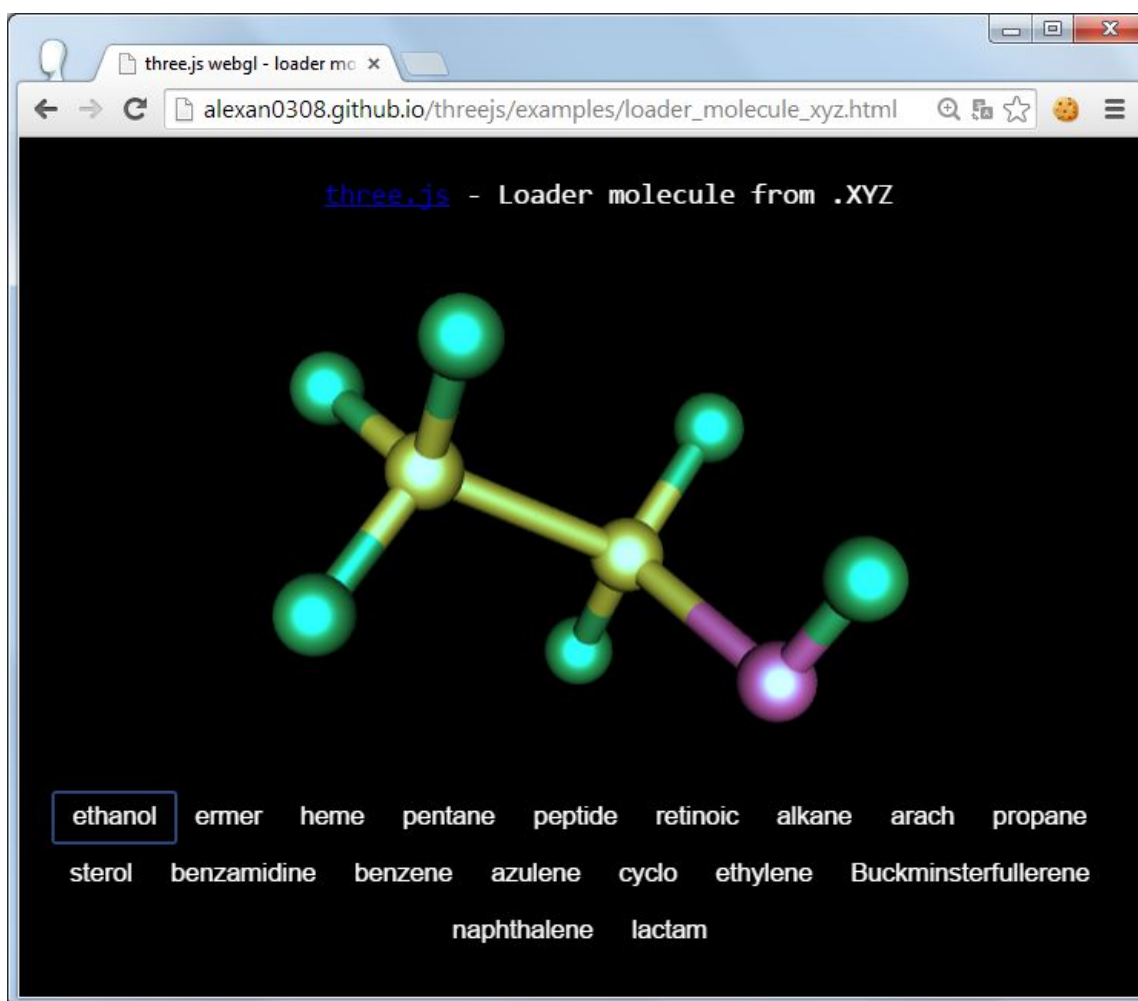


Рисунок 1 – Рабочее окно приложения. Молекула этанола

Для изображения молекул используются готовые файлы программы **Tinker**. Эта программа хранит координаты молекул в текстовых файлах с расширением `.xyz`. Эти файлы устроены следующим образом. Первая строка файла содержит целое число, равное количеству атомов в молекулярной системе, и N строк, с разделёнными пробелами или табуляцией именами атомов и их координатами (в Ангстремах). Например, файл `ethanol.xyz` для координат атомов молекулы этанола выглядит следующим образом:

```

9      Ethanol
1  C   -0.231579  -0.350841  -0.037475  1  2  4  5  6
2  C   0.229441   0.373160   1.224850  1  1  3  7  8
3  O   0.868228  -0.551628   2.114423  6  2  9
4  H   0.619613  -0.833754  -0.565710  5  1
5  H  -0.709445   0.352087  -0.754607  5  1
6  H  -0.976393  -1.144198   0.191635  5  1
7  H  -0.628785   0.860022   1.736350  5  2
8  H   0.952253   1.174538   0.962081  5  2
9  H   0.204846  -1.119563   2.483509  21  3

```

Рисунок 2 – Содержание файла `.xyz` для этанола

Здесь также содержатся данные о связи атома с другими атомами (седьмая и дальнейшие цифры). Например, первый атом нужно соединить со 2, 4, 5 и 6 атомами.

Разработанное приложение строит молекулу по этим данным. Для загрузки файла .xyz используется технология Ajax.

Внизу на специальной панели приложения располагаются кнопки для выбора просмотра моделей других химических элементов. Например, можно выбрать молекулу фуллерена (рисунок 3), или молекулу гема (рисунок 4), и т.д.

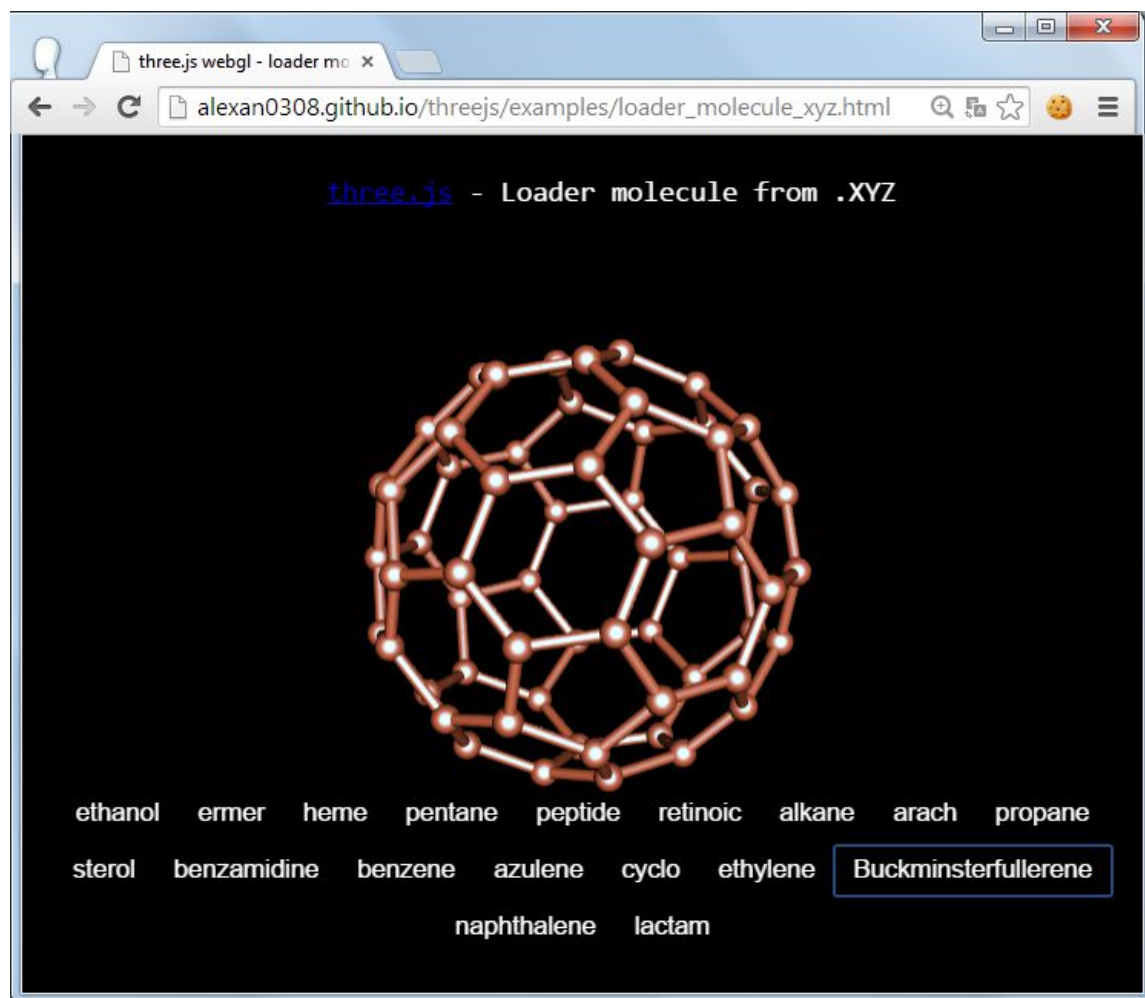


Рисунок 3 – Рабочее окно приложения. Молекула фуллерена

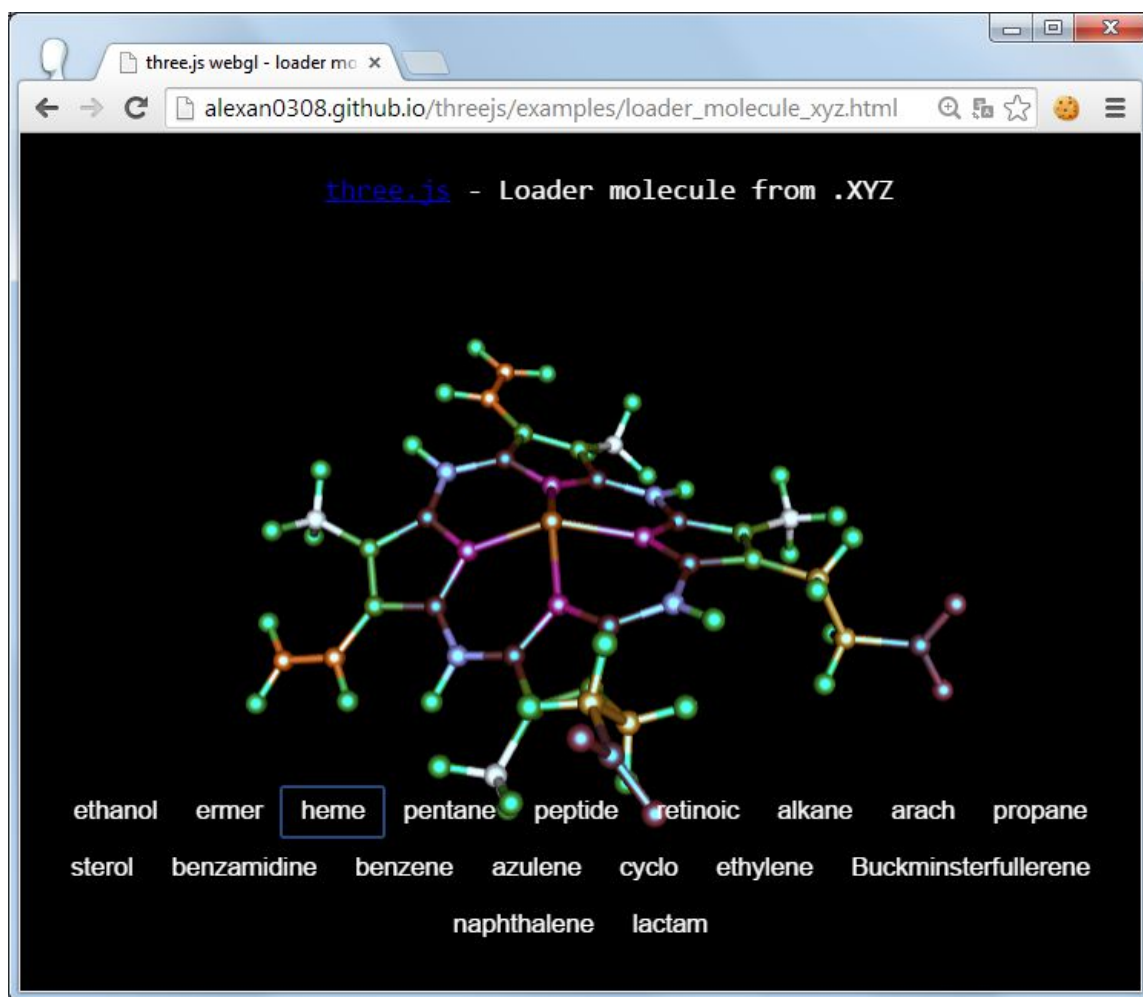


Рисунок 4 – Рабочее окно приложения. Молекула гема

2. Основные этапы разработки и алгоритмы приложения

Для облегчения разработки приложений на **WebGL** разработано множество библиотек. В этой работе выбор пал на библиотеку **Three.js**.

Three.js – это библиотека **JavaScript**, содержащая набор готовых классов для создания и отображения интерактивной компьютерной 3D графики в **WebGL** [1].

Библиотека **Three.js** облегчает работу с **WebGL**. При использовании **Three.js** отпадает необходимость в написании шейдерных процедур (но эта возможность остается), и появляется возможность оперировать с более привычными и удобными понятиями сцены, света и камеры, объектами и их материалами.

Для использования библиотеки нужно скачать с сайта **threejs.org** файл **three.min.js** [5] (собственно, сама библиотека, содержащая минимальный набор функций для работы с **WebGL**), и подключить к вашему проекту.

Работу с 3D графикой **WebGL** с помощью **Three.js** можно условно разбить на следующие этапы [1]:

- добавление сцены;
- добавление камеры;
- добавление света (освещения);
- добавление графических объектов на сцену;
- создание объекта визуализации;
- рендеринг (визуализация);
- анимация (движение объектов, их взаимодействие).

Эти этапы более-менее стандартизированы в библиотеке **Three.js**, перейдем непосредственно к реализации основной части приложения.

Для загрузки файла .xyz с координатами молекул используется технология Ajax. Основой использования Ajax-запроса является объект XMLHttpRequest [2]. В самом простом виде метод загрузки на языке JavaScript может выглядеть так:

```
file_get_contents = function( url ) {  
    var req = new XMLHttpRequest();  
    req.open( "GET", url, false );  
    req.send( null );  
    return req.responseText;  
}
```

В результате получим строку со всем содержимым данного файла. Конечно, работать напрямую с такими «бесструктурными» данными неудобно. После получения текста файла нужно его обработать, перевести в массив данные о количестве вершин, связях, и т.д. Превратим полученный текст в двумерный массив данных **arr** с помощью метода **parseXYZ**, так что, например, после парсинга верно равенство (сравните с первой строкой файла на рисунке 1):

```
arr[ 0 ] = [ 1, 'C', -0.231579, -0.350841, -0.037475, 1, 2, 4, 5, 6 ]
```

В таком массиве 3, 4 и 5 элементы содержат координаты атомов. Молекула строится как трехмерный объект, в который потом будем добавлять атомы (в виде обычных сфер):

```
molecule = new THREE.Object3D();
```

Для изображения модели молекул достаточно использовать два вида примитивов – сферы и цилиндры. В Three.js для них имеются готовые методы SphereGeometry и CylinderGeometry.

Осталось вывести связи. К сожалению, в **Three.js** нет возможности управлять толщиной обычных линий, поэтому для соединения двух точек цилиндром используется функция cylinderMesh [5]:

```
function cylinderMesh( pointX, pointY ) {
    var direction = new THREE.Vector3().subVectors( pointY, pointX );
    var arrow = new THREE.ArrowHelper(
        direction.clone().normalize(), pointX, direction.length() );
    var edgeGeometry =
        new THREE.CylinderGeometry( 0.1, 0.1, direction.length(), 36, 4 );
    var edgeMesh = new THREE.Mesh( edgeGeometry,
        new THREE.MeshBasicMaterial( { color: 0x0000ff } ) );
    edgeMesh.position.copy( new THREE.Vector3().addVectors(
        pointX, direction.multiplyScalar(0.5) ) );
    edgeMesh.setRotationFromEuler( arrow.rotation );
    return edgeMesh;
}
```

Осталось добавить вращение молекулы:

```
function render() {
    molecule.rotation.y = molecule.rotation.y + 0.007;
    light.position = camera.position;
    controls.update();
    renderer.render(scene, camera);
}
```

Заключение

В качестве технического средства, используемого при написании программы, выступил ноутбук DELL Inspiron 5720, с двухъядерным процессором Intel Core i5 3210M @

2.50GHz, с оперативной памятью 4 Гб Single-Channel DDR3, с дискретной видеокартой GeForce GT 630M (128 бит), видеопамятью 1024 Мб.

Использовались установленные на ноутбуке операционная система Windows 7 Home Basic (64 bit), редактор Notepad++, браузер Google Chrome, Git-клиент для GitHub msysgit.

При написании приложения использовалась бесплатная библиотека **Three.js** - библиотека JavaScript, содержащая набор готовых классов для создания и отображения интерактивной компьютерной 3D графики в WebGL.

Для работы с программой необходимо иметь персональный компьютер типа IBM PC Pentium IV с операционной системой семейства MS Windows или Linux, и оперативной памятью от 128 Мб, современный браузер (Google Chrome и т.п.).

Результаты разработки опубликованы на **GitHub** и доступны по адресу http://alexan-0308.github.io/threejs/examples/#loader_molecule_xyz. Там же можно ознакомиться и с исходниками приложения.

Список литературы

1. Вильданов А.Н. 3D-моделирование на WebGL с помощью библиотеки Three.js: учебное пособие. – Уфа : РИЦ БашГУ, 2014. – 114 с. – ISBN 987-5-7477-3560-6.
2. Кантор И. Центральный Javascript-ресурс. Учебник с примерами скриптов. Форум. Книги и многое другое [Электронный ресурс] – Режим доступа: <http://javascript.ru>, свободный (дата обращения: 01.09.14).
3. Фреймворки WebGL - Three.js [Электронный ресурс]. – Режим доступа: <http://ru.tmssoftstudio.com/file/page/webgl-frameworks/three-api-ru/three.html>, свободный (дата обращения: 30.10.14).
4. Mr. Doob. Three.js – JavaScript 3D library [Электронный ресурс]. – Режим доступа: <http://threejs.org>, свободный (дата обращения: 30.10.14).
5. Stemkoski Lee. Three.js – examples [Электронный ресурс]. – Режим доступа: <http://stemkoski.github.io/Three.js>, свободный (дата обращения: 01.08.14).

Рецензенты:

Хабибуллин И.Т., д.ф.-м.н., профессор, ведущий научный сотрудник Института математики с вычислительным центром Российской академии наук, г. Уфа;

Спивак С.И., д.ф.-м.н., профессор, заведующий кафедрой математического моделирования ФГБОУ ВПО БашГУ, г. Уфа.