

Секция  
**ПРИКЛАДНАЯ МАТЕМАТИКА**

**Визуализация трехмерных динамических  
объектов на примере кубика Рубика**

*Никифоров Сергей Владимирович*

КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ В.И. ВЕРНАДСКОГО  
ТАВРИЧЕСКАЯ АКАДЕМИЯ  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 601-П)  
e-mail: nikin1994@gmail.com

*В работе рассматривается алгоритм построения трехмерных динамических объектов на примере кубика Рубика и особенности использующейся для этого javascript-библиотеки Three.js*

Three.js — легковесная библиотека JavaScript, используемая для создания и отображения анимированной компьютерной 3D графики при разработке веб-приложений. Поддерживается подавляющим большинством современных браузеров. Three.js скрипты могут использоваться совместно с элементом HTML5 CANVAS, SVG или WebGL. Преимущества её в том, что библиотека Three.js работает во всех браузерах, которые поддерживают технологию WebGL. Также может работать с базовым интерфейсом элемента CANVAS, благодаря чему работает и на многих мобильных устройствах. К тому же она выполняет низкоуровневую работу с WebGL, что значительно упрощает и ускоряет работу программиста. В Three.js присутствует базовый порядок действий, необходимых для корректного отображения объектов: 1 – Построение сцены, 2 – Размещение камеры, 3 – Добавление объектов и событий, 4 – Рендеринг(процесс получения изображения по модели).

Поставлена задача построить кубик Рубика размером  $N \times N$ . Центр куба будет расположен в точке  $(0, 0, 0)$ . Куб состоит из 8 угловых,  $6 \cdot (N-2) \cdot (N-2)$  центральных и  $(N-2) \cdot 12$  реберных элементов. Пошаговое построение большого куба ведется путем добавления элементов через трехмерный массив. Координаты расположения для каждого последующего куба рассчитываются с помощью векторов, исходящих из центра координат и изменяющихся в соответствии с введенными данными, такими как размер одного элемента и отступ между элементами. Хранятся элементы в трехмерном массиве с соответствующими координатам расположения индексами, что упрощает нахождения нужного элемента. Элементы, которые попадают внутрь куба - отсекаются с помощью

условного оператора, анализирующего индекс элемента. Куб окрашивается в соответствии с заданной цветовой схемой.

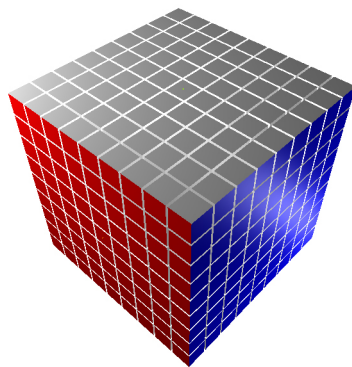


Рис. 1. Пример куба 9x9

Поворот одной из граней куба происходит путем присоединения к группе объектов соответствующего индекса в массиве через цикл, например: `[x][toObject][z]` и поворота этого объекта на 90 градусов в заданном направлении. При каждом повороте объект создается заново. Проблемой данного метода является сложность замены индекса каждого повернутого элемента в массиве. На каждой грани есть элементы, которым можно поставить в соответствие другие элементы на этой же грани. То есть, несколько соответствующих элементов при разных положениях грани могут занимать одну и ту же позицию.

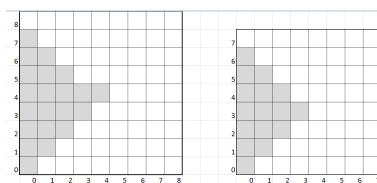


Рис. 2. Пример набора ‘уникальных’ элементов грани

На рис. 2 показан пример с четным и нечетным количеством элементов грани куба. Выделив эти элементы через условие, с помощью цикла их индексы меняются по кругу (рис. 3).

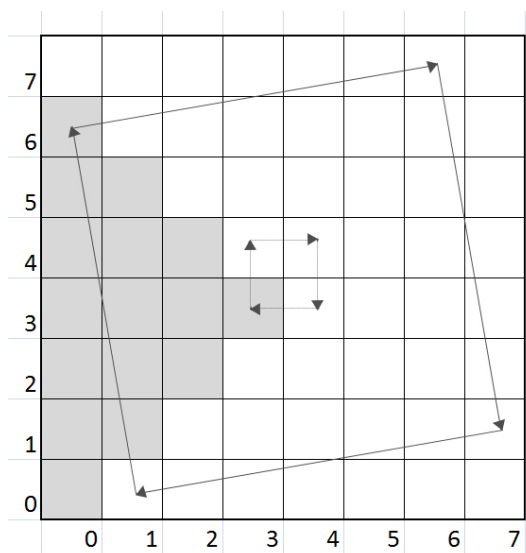


Рис. 3. Визуализация замены индексов элементов

Результатом работы является программа, способная построить куб любого размера с реализованным механизмом поворота граней. Из всего выше описанного следует, что библиотека Three.js является отличным вариантом для трехмерного веб-моделирования и может быть использована как для написания игр, так и для создания интерактивных веб-страниц. Библиотека Three.js находится на стадии разработки и постоянно дополняется и улучшается, но уже сейчас она демонстрирует широкий спектр возможностей, необходимых для написания 3D-моделей.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Свободная энциклопедия [ru.wikipedia.org/wiki/Three.js](http://ru.wikipedia.org/wiki/Three.js) Определения и основная информация о Three.js
- [2] Ricardo Cabello <http://threejs.org/> Документация о библиотеке