

Применение qr-кодов для передачи геометрических моделей

Е.А. Курушин¹

¹ НГТУ им. Р.Е. Алексеева, ул. Минина, 24, г. Нижний Новгород, 603155, Россия

Аннотация

Quick Response code, или код быстрого реагирования, кодирует текстовую информацию, которая может быть задана в разных форматах, таких как ссылка, контактная информация, данные для проведения платежа и др. Такие коды широко используются в современном мире, а средством их распознавания чаще всего становится обычный смартфон. 3D модели в свою очередь требуют специализированного ПО для прочтения и взаимодействия с ними. В результате проведения данной работы геометрическая модель передается в информационную систему посредством qr-кода, который представляет собой ссылку на этот ресурс. Текстовое описание геометрической модели в формате файла obj преобразовано в формат x-www-form-urlencoded и передается GET-параметром. При загрузке страницы по QR-коду в клиентских сценариях происходит обработка данных и последующая визуализация с помощью технологии WebGL с использованием библиотеки three.js. Применение такого подхода позволяет передать геометрическую модель для её визуализации и взаимодействия с ней без специализированного ПО, требуя при этом только наличие браузера, предустановленного в каждой ОС.

Ключевые слова

Геометрическая модель, qr-код, obj, передача данных, WebGL, three.js.

Using qr Codes to Transfer Geometric Models

Е.А. Kurushin¹

¹ Nizhny Novgorod State Technical University n.a. R.E. Alekseev, Minina str., 24, Nizhny Novgorod, 603155, Russia

Abstract

Quick Response code encodes text information that can be specified in different formats, such as a link, contact information, payment details, etc. Such codes are widely used in the modern world, and the means of recognizing them is often a common smartphone. Such codes are widely used in the modern world, and the means of recognizing them is most often an ordinary smartphone. 3D models in turn require specialized software to read and interact with them. As a result of this work, the geometric model is transferred to the information system by means of qr-code, which is a link to this resource. The text description of the geometrical model in the obj file format is converted to the x-www-form-urlencoded format and is passed by GET-parameter. When loading the page by QR code in client scenarios data processing and subsequent visualization with the help of WebGL technology with the use of three.js library takes place. Applying this approach allows to transfer a geometric model for its visualization and interaction with it without specialized software, requiring only a browser pre-installed in every OS.

Keywords

Geometric model, qr code, obj, data transfer, WebGL, three.js.

ГрафиКон 2023: 33-я Международная конференция по компьютерной графике и машинному зрению, 19-21 сентября 2023 г., Институт проблем управления им. В.А. Трапезникова Российской академии наук, г. Москва, Россия

EMAIL: eu.kurushin@yandex.ru (Е.А. Курушин)

ORCID: 0000-0002-7320-9492 (Е.А. Курушин)



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Введение

QR-код (сокращение от Quick Response code) – это двухмерный штрих-код, который разработан японской компанией Denso Wave в 1994 году. Этот код предназначен для быстрого и эффективного чтения информации, содержащейся в нем. История qr-кода началась с потребности в усовершенствовании стандартных одномерных штрих-кодов, которые могли хранить ограниченное количество информации. Первоначально qr-код был использован для облегчения отслеживания автомобилей в процессе производства на заводах. Позже он стал популярным в Японии для маркировки товаров и предоставления информации клиентам [1].

QR-код применяется для передачи данных, а в некоторых случаях и графической информации. Они устойчиво вошли в жизнь современного человека, так как используются повсеместно: при оплате, в процессе обучения, маркетинге и других сферах [2]. Отдельно стоит отметить маркетинговую сферу, где в рекламных продуктах добавляется QR-код, ведущий на интернет ресурс. Это может быть сайт предприятия, страница о продукте, о специалисте и т.д.

Целью данной работы является рассмотрение возможности применения qr-кода для передачи геометрической модели. Особенность исследования заключается в том, что сама по себе геометрическая модель представляет мало интереса, если не используется специализированное ПО для её просмотра или работы с ней.

Ранее рассматривалось применение технологии WebGL в рекламно-выставочной деятельности, где qr-код представлял собой ссылку на специально созданную страницу о продукте. По сути это интерактивная веб-версия листовки, где вместо статичного изображения использовалась геометрическая модель, с которой можно взаимодействовать путем вращения, масштабирования и перетаскивания.

Такой подход подразумевает заранее подготовленные материалы: настроенный сервер, страница и сама модель. Когда пользователь запрашивает определенную страницу, то после её возвращения подгружаются также используемые на ней ресурсы: таблицы стилей, изображения, скрипты и, в данном случае, геометрическая модель.

2. Особенности передачи

Как упоминалось ранее – qr-код применяется в разных сферах. Здесь стоит отметить следующий факт: если передается специализированный набор данных в определенном формате, то приложение должно правильно их обработать. Так, например, работает qr-код при оплате покупок. Разработчик задает определенный шаблон [3], а при сканировании кода приложение обрабатывает зашифрованные в нем данные заданным образом и выполняет определенный набор действий. Это означает, что для осуществления возможности передачи модели приложение должно знать, как её обработать и какие операции выполнять дальше.

Стандартное приложение «Камера» (название может отличаться в разных ОС и версиях) в современных смартфонах умеет обрабатывать qr-код и показывать закодированную информацию. Это широко используется для указания гиперссылки на Интернет-ресурс.

Для каждой страницы можно передать GET-параметры, дописываемые после адреса страницы [4]. Получается, что GET-параметры являются частью адреса страницы. Из этих вводных возникает потребность уточнить наличие возможности передачи геометрических моделей в качестве get-параметра, являющегося частью url, которая в свою очередь будет закодирована в qr-код. Это позволит обработать геометрическую модель без применения специализированного разработанного ПО – необходимые инструменты будут переданы в клиентский браузер, в котором произойдет визуализация модели.

3. Подготовка модели

Для исследования возможности передачи геометрической модели выбран простой куб, т.к. он не содержит сложной геометрии. Для этого в программе Blender создан новый файл с кубом в центре сцены по умолчанию, который экспортирован в формате obj. Стоит отметить

содержимое данного файла (рисунок 1). Этот файл описывает набор всех точек, определяющих модель, после чего следует перечисление нормалей, а затем координаты текстуры, излишние в рассматриваемом случае, т.к. для модели не назначено ни материалов, ни, тем более, текстур. Данный формат также записывает группы сглаживания. Основным моментом в то же время является описание поверхностей. Видно, что при описании модели используются слешы, пробелы и переносы строк, поэтому содержимое файла необходимо привести к формату x-www-form-urlencoded, используемому по умолчанию при отправке html форм [5], потому что url содержит переделённые требования. Таким образом описание модели приобретает вид, представленный на рисунке 2.

```

1  o Cube
2  v -1.000000 -1.000000 1.000000
3  v -1.000000 1.000000 1.000000
4  v -1.000000 -1.000000 -1.000000
5  v -1.000000 1.000000 -1.000000
6  v 1.000000 -1.000000 1.000000
7  v 1.000000 1.000000 1.000000
8  v 1.000000 -1.000000 -1.000000
9  v 1.000000 1.000000 -1.000000
10 vn -1.0000 -0.0000 -0.0000
11 vn -0.0000 -0.0000 -1.0000
12 vn 1.0000 -0.0000 -0.0000
13 vn -0.0000 -0.0000 1.0000
14 vn -0.0000 -1.0000 -0.0000
15 vn -0.0000 1.0000 -0.0000
16 vt 0.375000 0.000000
17 vt 0.375000 1.000000
18 vt 0.125000 0.750000
19 vt 0.625000 0.000000
20 vt 0.625000 1.000000
21 vt 0.875000 0.750000
22 vt 0.375000 0.250000
23 vt 0.125000 0.500000
24 vt 0.625000 0.250000
25 vt 0.875000 0.500000
26 vt 0.375000 0.750000
27 vt 0.625000 0.750000
28 vt 0.375000 0.500000
29 vt 0.625000 0.500000
30 s 0
31 f 1/1/1 2/4/1 4/9/1 3/7/1
32 f 3/7/2 4/9/2 8/14/2 7/13/2
33 f 7/13/3 8/14/3 6/12/3 5/11/3
34 f 5/11/4 6/12/4 2/5/4 1/2/4
35 f 3/8/5 7/13/5 5/11/5 1/3/5
36 f 8/14/6 4/10/6 2/6/6 6/12/6

```

Рисунок 1 – Описание куба в формате obj

```

o+Cube%0Av+-1.000000+-1.000000+1.000000%0Av+-1.000000+1.000000+1.000000%0Av+-
1.000000+-1.000000+-1.000000%0Av+-1.000000+1.000000+-1.000000%0Av+1.000000+-
1.000000+1.000000%0Av+1.000000+1.000000+1.000000%0Av+1.000000+-1.000000+-1.000000%0Av
+1.000000+1.000000+-1.000000%0Avn+-1.0000+-0.0000+-0.0000%0Avn+-0.0000+-0.0000+-
1.0000%0Avn+1.0000+-0.0000+-0.0000%0Avn+-0.0000+-0.0000+1.0000%0Avn+-0.0000+-1.0000+-
0.0000%0Avn+-0.0000+1.0000+-0.0000%0Avt+0.375000+0.000000%0Avt+0.375000+1.000000%0Avt
+0.125000+0.750000%0Avt+0.625000+0.000000%0Avt+0.625000+1.000000%0Avt
+0.875000+0.750000%0Avt+0.375000+0.250000%0Avt+0.125000+0.500000%0Avt
+0.625000+0.250000%0Avt+0.875000+0.500000%0Avt+0.375000+0.750000%0Avt
+0.625000+0.750000%0Avt+0.375000+0.500000%0Avt+0.625000+0.500000%0As+0%0Af
+1%2F1%2F1+2%2F4%2F1+4%2F9%2F1+3%2F7%2F1%0Af
+3%2F7%2F2+4%2F9%2F2+8%2F14%2F2+7%2F13%2F2%0Af
+7%2F13%2F3+8%2F14%2F3+6%2F12%2F3+5%2F11%2F3%0Af
+5%2F11%2F4+6%2F12%2F4+2%2F5%2F4+1%2F2%2F4%0Af
+3%2F8%2F5+7%2F13%2F5+5%2F11%2F5+1%2F3%2F5%0Af
+8%2F14%2F6+4%2F10%2F6+2%2F6%2F6+6%2F12%2F6%0A

```

Рисунок 2 - Описание модели в формате x-www-form-urlencoded, открытое в приложении «Блокнот» с включенной опцией «Перенос по словам»

4. Принимаемая сторона

Конечная точка передачи данных - простая веб-страница, структура которой представляет из себя по большей части элемент canvas (рисунок 3) ввиду акцентирования внимания именно на модели. На холсте отрисовывается переданная модель посредством технологии WebGL. Для более простого взаимодействия с API используется библиотека three.js.

```

index.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6 </head>
7 <body>
8   <canvas id="object"></canvas>
9   <script src="js/three.min.js"></script>
10  <script src="js/OBJLoader.js"></script>
11  <script src="js/OrbitControls.js"></script>
12  <script src="js/script.js"></script>
13 </body>
14 </html>

```

Рисунок 3 - Структура страницы

Помимо самой библиотеки используются также её вспомогательные модули, подключаемые отдельно: OBJLoader для загрузки и обработки текстового описания модели в формате Wavefront OBJ File и OrbitControls, используемый для осуществления интерактивности в части вращения, масштабирования и перетаскивания.

Функционал обработки описания модели представлен на рисунке 4. В строках 5-22 создаются и настраиваются элементы 3D пространства, без которых увидеть переданную модель не получится. Здесь происходит создание рендера, сцены, камеры, и источника освещения. Функция animate используется для отрисовки содержимого сцены, а также результатов взаимодействия пользователя с «моделью» (вращение, перемещение, масштабирование).

```

1 document.addEventListener("DOMContentLoaded", function(){
2   renderWidth = window.innerWidth
3   renderHeight = window.innerHeight
4
5   const renderer = new THREE.WebGLRenderer({canvas: object})
6   renderer.setSize( renderWidth, renderHeight)
7   renderer.setClearColor(0x101010)
8
9   const scene = new THREE.Scene()
10
11  const camera = new THREE.PerspectiveCamera( 45, renderWidth / renderHeight, 1, 1000 )
12  camera.position.set(8, 5, 8)
13
14  const controls = new THREE.OrbitControls( camera, renderer.domElement )
15
16  const light = new THREE.PointLight( 0xffffff, 3, 100 );
17  light.position.set( 8, 5, 8 );
18
19  let model = get_request_data('model') //описание модели в формате application/x-www-form-urlencoded
20  model = xWWWFormDecode(model) //расшифровка описания модели
21
22  const loader = new THREE.OBJLoader(); //загрузчик моделей в формате OBJ
23  var cube = loader.parse(model) //создание объекта на основе описания модели
24  scene.add(cube) //добавление модели на сцену
25
26  function animate() {
27    requestAnimationFrame( animate )
28    controls.update()
29    renderer.render( scene, camera )
30  }
31  animate()
32 }
33
34 //Расшифровка данных в формате "application/x-www-form-urlencoded"
35 function xWWWFormDecode(s) {
36   return decodeURIComponent((s + '').replace(/\+/g, ' '));
37 }
38
39 //получение get-параметра из строки запроса
40 function get_request_data(parameter){
41   let params = (new URL(document.location)).searchParams;
42   return params.get(parameter)
43 }

```

Рисунок 4 - Содержимое файла script.js с комментариями

Функции `get_request_data(parameter)` и `xWWWFormDecode(s)` используются для получения описания модели из `get` параметра и конвертирования из формата `x-www-form-urlencoded` в формат, понимаемый форматом файла `.obj`, соответственно, где `parameter` – название GET-параметра, а `s` – модель, описанная в формате стандартного кодирования URL. После этого используется метод `parse` конструктора `OBJLoader` для парсинга описания модели и создания 3d модели на основе полученных данных и добавления её на сцену.

Полученный qr-код и результат передачи представлены на рисунке 5.

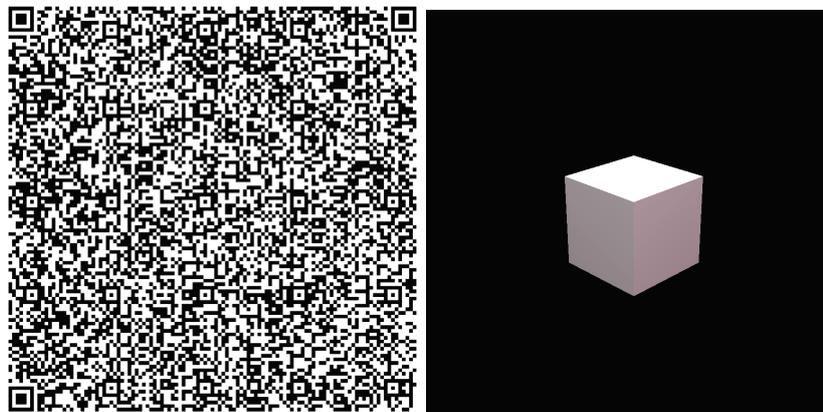


Рисунок 5 – qr-код (слева) и результат передачи (справа)

5. Заключение

Проделанная работа отражает возможность применения qr-кода для передачи геометрических моделей с учетом следующих оговорок. Во-первых, максимальная длина гиперссылки зависит от браузера и от типа и настроек сервера, который обрабатывает запрос. Так, ссылки длиной до 2000 символов корректно будут работать во всех браузерах и устройствах [6], а максимальная длина алфавитно-цифровой кодировки кода быстрого реагирования составляет 4296, что накладывает дополнительные ограничения [2]. Во-вторых, передана модель простого куба, состоявшего из 8 точек и 6 граней, а не сложного объекта.

С другой стороны, применение такого подхода избавляет от дополнительных обращений к серверу, которые происходят после получения страницы и подгружающих дополнительные данные, такие как шрифты, таблицы стилей, клиентские сценарии и т.д.

Темой для дальнейших исследований может послужить сжатие без потерь геометрических моделей, а также анализ структуры форматов файлов описания геометрических моделей и объем получаемого файла.

6. Список источников

- [1] Голенков С.А. QR-код. История создания и применения // .Юный ученый. – 2023. – 33(66). – С. 93-95.
- [2] Понамарев Р.А. -коды. Создание и применение // Моя профессиональная карьера. – 2023. – №46. – С. 164-171.
- [3] ГОСТ Р 56042-2014. Двумерные символы штрихового кода для осуществления платежей физических лиц. – Введ. 2014-09-01. – М.: Стандартинформ, 2020. – 27 с.
- [4] URI — сложно о простом (Часть 1) // Хабр URL: <https://habr.com/ru/articles/232385/> (дата обращения: 21.06.2023).
- [5] POST // MDN Web Docs URL: <https://developer.mozilla.org/ru/docs/Web/HTTP/Methods/POST> (дата обращения: 30.06.2023).
- [6] Какой максимальной длины может быть URL? // GitHub URL: https://github.com/dreikanter/paradigm.ru/blob/master/posts/2007-12-19_url-max-length.md (дата обращения: 02.07.2023).