

Интерактивное отображение зон помещений на трёхмерной модели здания в веб-приложении

Андриенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема
Студент

Пащенко Екатерина Александровна

Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация

В работе рассматривается реализация веб-приложения для интерактивного отображения зон помещений на трёхмерной модели здания. В основе решения лежит использование заранее подготовленных зональных мешей, встроенных в 3D-модель, которые автоматически выявляются и преобразуются в визуальные оверлеи. Каждая зона отображается в виде полупрозрачного цветного объекта с текстовой подписью и может быть выбрана пользователем для получения дополнительной информации. Взаимодействие с зонами реализовано с применением механизма трассировки лучей, обеспечивающего определение выбранного помещения и отображение соответствующих данных в боковой информационной панели.

Ключевые слова: трёхмерная визуализация; веб-приложение; Three.js; интерактивные зоны; помещения; 3D-модель здания; Raycaster; CSS2DRenderer; пользовательский интерфейс.

Interactive display of indoor areas on a three-dimensional model of a building in a web application

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University
Student

Paschenko Ekaterina Alexandrovna

Sholom-Aleichem Priamursky State University
Student

Abstract

The paper considers the implementation of a web application for interactive display of indoor areas on a three-dimensional model of a building. The solution is based on the use of pre-prepared zonal meshes embedded in the 3D model, which are automatically detected and converted into visual overlays. Each zone is displayed as a translucent colored object with a text caption and can be selected by the user to

receive additional information. Interaction with the zones is implemented using the ray tracing mechanism, which ensures the identification of the selected room and the display of relevant data in the sidebar.

Keywords: three-dimensional visualization; web application; Three.js; interactive zones; rooms; 3D model of the building; Raycaster; CSS2DRenderer; user interface.

1 Введение

1.1 Актуальность

Интерактивные трёхмерные модели зданий находят широкое применение в навигационных, информационных и презентационных веб-системах. Особую значимость имеют решения, позволяющие не только просматривать модель, но и получать контекстную информацию о конкретных помещениях. Использование зон, встроенных в геометрию модели, позволяет автоматизировать процесс привязки данных к объектам сцены и повысить наглядность представления пространственной информации. Веб-реализация подобных механизмов делает такие системы доступными для широкой аудитории без установки специализированного программного обеспечения.

1.2 Обзор исследований

А.П. Митяева и Е.Н. Дроздова анализируют основные принципы создания трёхмерной графики в браузере на примере библиотеки Three.js, уделяя внимание структуре сцены и базовым средствам визуализации [1]. Д.И.Зуева рассматривает показатели производительности рендеринга трёхмерной графики с использованием WebGL и Three.js, выявляя ключевые факторы, влияющие на скорость отображения сцены [2]. А.Н. Вильданов и А.И. Исмагилов описывают возможности браузерного трёхмерного моделирования с применением Three.js, подчёркивая доступность и интерактивность веб-ориентированных 3D-решений [3]. В другой работе Д.И.Зуева исследует способы оптимизации производительности рендеринга 3D-графики в среде WebGL Three.js, акцентируя внимание на снижении вычислительной нагрузки и повышении стабильности отображения [4]. Mohan Balakrishnan Ja. и Manjunath Chavan S. рассматривают применение предварительной визуализации в процессе проектирования веб-анимаций на Three.js, отмечая её значение для планирования структуры и поведения трёхмерных сцен [5].

1.3 Цель исследования

Цель исследования – разработать и описать веб-механизм интерактивного отображения и выбора зон помещений на трёхмерной модели здания с последующим выводом информации о выбранной зоне.

2 Материалы и методы

В качестве основы использовано веб-приложение на базе библиотеки Three.js для рендеринга трёхмерной сцены в браузере. Зоны помещений

формируются на основе заранее подготовленных мешей `zone_*`, встроенных в модель здания.

3 Результаты и обсуждения

Для реализации интерактивных зон помещений используется предварительная подготовка 3D-модели здания. В модели каждому помещению соответствует отдельный меш, имя которого начинается с префикса `zone_` (например, `zone_Лекторий`, `zone_Гараж`). Такой подход позволяет закладывать семантику помещений непосредственно на этапе моделирования и в дальнейшем автоматически извлекать необходимые элементы из структуры сцены без ручной разметки в коде (рис. 1).

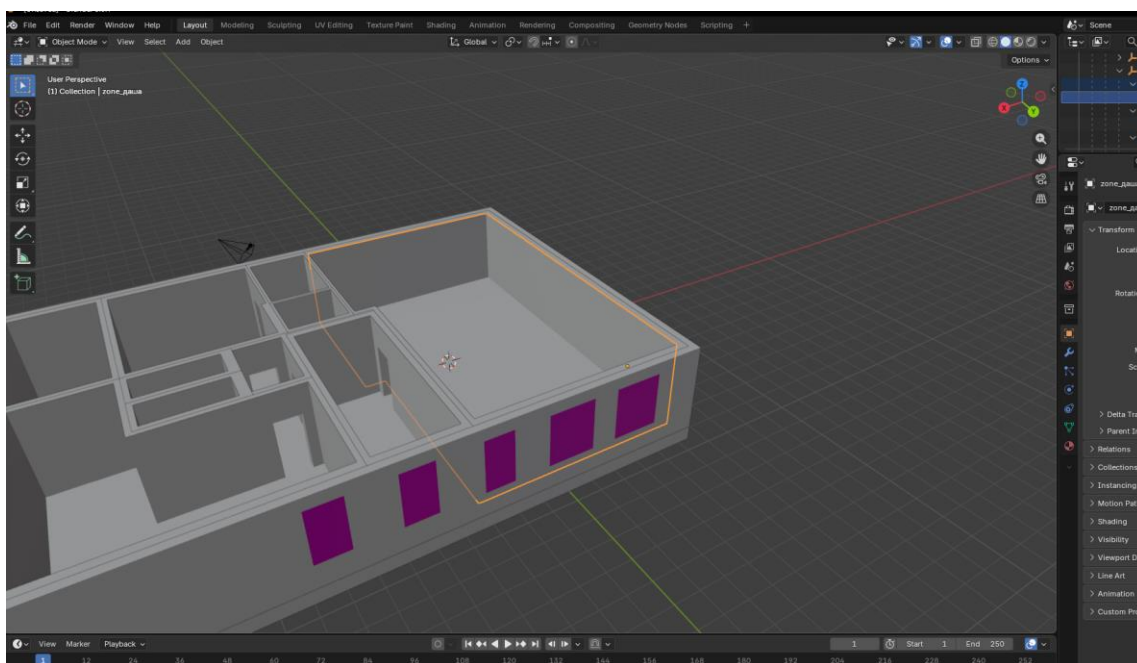


Рис. 1. Подготовка зон помещений в 3D-модели

После загрузки модели в сцену выполняется обход всех вложенных объектов. В процессе обхода выявляются меши, имена которых соответствуют заданному шаблону `zone_*`. Обнаруженные меши добавляются в список зон, а их исходная геометрия скрывается от пользователя. Это позволяет использовать исходные меши исключительно как источник геометрии и логической привязки, не отображая их напрямую в интерфейсе (рис. 2).

```
126 function colorForRoom(name, floorKey) {
127   const hash = hashName(name);
128   const hueOffset = floorKey === 'floor2' ? 0.15 : 0;
129   const hue = ((hash % 360) / 360 + hueOffset) % 1;
130   const color = new THREE.Color();
131   color.setHSL(hue, 0.55, 0.55);
132   return color;
133 }
134
135 function isZoneMesh(name) {
136   return typeof name === 'string' && name.toLowerCase().startsWith('zone');
137 }
138
139 function zoneDisplayName(name) {
140   if (!name) return '';
141   return name.replace(/^zone[_\s-]*\s/i, '');
142 }
143
144 const EDGE_KEY_PRECISION = 1000;
145 const FLOOR_EDGE_EPSILON = 0.02;
146 const DEBUG_ZONES = true;
147
148 function pointKey(x, z) {
149   const ix = Math.round(x * EDGE_KEY_PRECISION);
150   const iz = Math.round(z * EDGE_KEY_PRECISION);
151   return `${ix},${iz}`;
152 }
153
154 function polygonArea(points) {
155   let area = 0;
156   for (let i = 0; i < points.length; i += 1) {
157     const a = points[i];
```

Рис. 2. Поиск зональных мешей в сцене

Для каждой найденной зоны на основе её геометрии создаётся новый полупрозрачный меш-оверлей. Геометрия зоны клонируется и преобразуется в мировые координаты, после чего применяется материал с заданной прозрачностью и цветом. Цвет подбирается в зависимости от помещения и этажа, что обеспечивает визуальное различие зон. Созданные оверлеи добавляются в сцену и используются как основной визуальный элемент взаимодействия с пользователем (рис. 3).

```

107     const zoneMaterialBase = {
108         transparent: true,
109         opacity: 0.3,
110         roughness: 0.4,
111         metalness: 0.1,
112         depthWrite: false,
113         polygonOffset: true,
114         polygonOffsetFactor: -1,
115         polygonOffsetUnits: -4,
116     };
117
118     function hashName(value) {
119         let hash = 0;
120         for (let i = 0; i < value.length; i += 1) {
121             hash = (hash * 31 + value.charCodeAt(i)) >>> 0;
122         }
123         return hash;
124     }
125
126     function colorForRoom(name, floorKey) {
127         const hash = hashName(name);
128         const hueOffset = floorKey === 'floor2' ? 0.15 : 0;
129         const hue = ((hash % 360) / 360 + hueOffset) % 1;
130         const color = new THREE.Color();
131         color.setHSL(hue, 0.55, 0.55);
132         return color;
133     }
134
135     function isZoneMesh(name) {
136         return typeof name === 'string' && name.toLowerCase().startsWith('zone_');
137     }
138     0
139     function zoneDisplayName(name) {
140         if (!name) return '';

```

Рис. 3. Создание цветного оверлея зоны

Для повышения информативности каждой зоне добавляется текстовая подпись с названием помещения. Подписи реализованы с использованием CSS2DRenderer, что позволяет отображать HTML-элементы поверх 3D-сцены без влияния на рендеринг геометрии. Положение подписи вычисляется по границам зоны и размещается над её центром, обеспечивая читаемость при различных углах обзора (рис. 4).



Рис. 4. Отображение подписей зон

Для обработки пользовательских кликов используется механизм трассировки лучей (Raycaster). При нажатии мышью вычисляется луч, пересекающий сцену, и определяется, попадает ли он в один из активных оверлеев зон. Если пересечение найдено, из пользовательских данных объекта

извлекается информация о выбранной зоне, которая далее используется для отображения сведений в интерфейсе.

После определения выбранной зоны открывается боковая информационная панель. В панели отображаются название помещения и этаж, а также текстовое описание зоны. Содержимое панели формируется динамически на основе данных выбранного объекта, что позволяет централизованно управлять выводимой информацией и в дальнейшем расширять её состав (рис. 5).

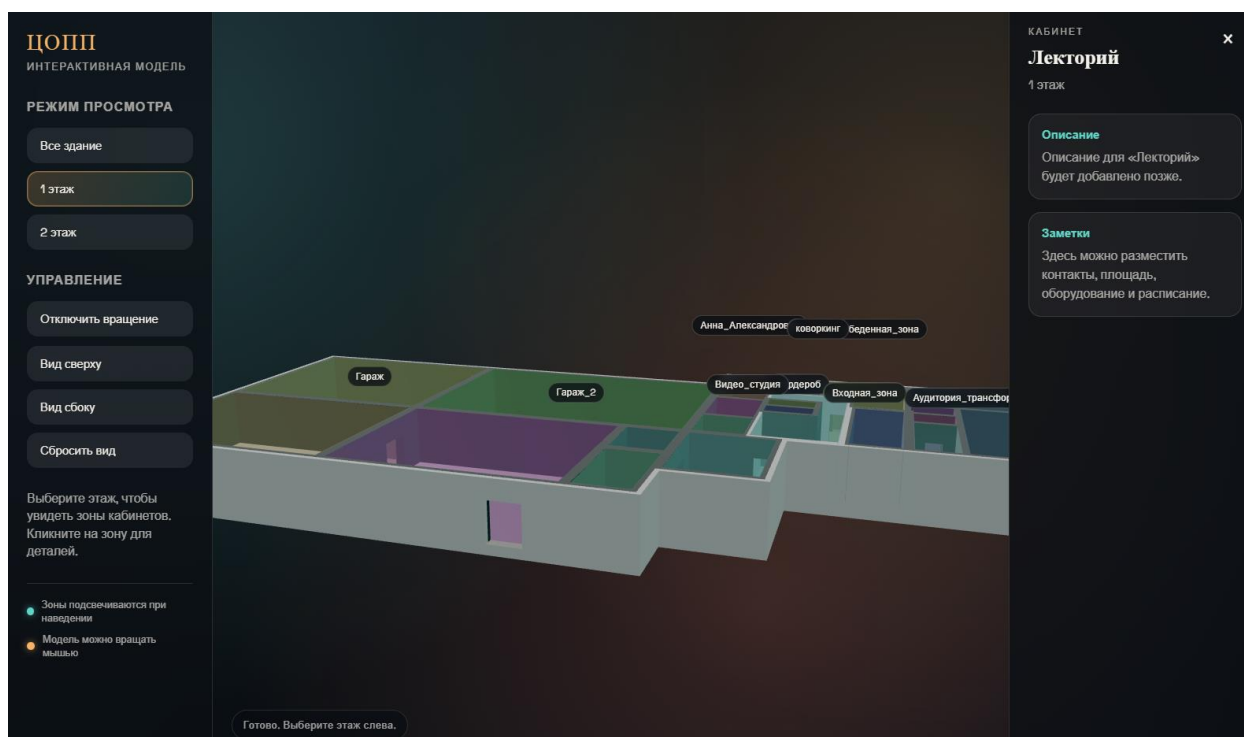


Рис. 5. Вид сайдбара

В результате пользователь получает интерактивную 3D-модель здания с возможностью визуального выбора помещений и просмотра связанной информации. Использование зональных мешей и оверлеев обеспечивает точную привязку данных к геометрии модели и формирует удобный механизм взаимодействия с архитектурным объектом в веб-среде (рис. 6).

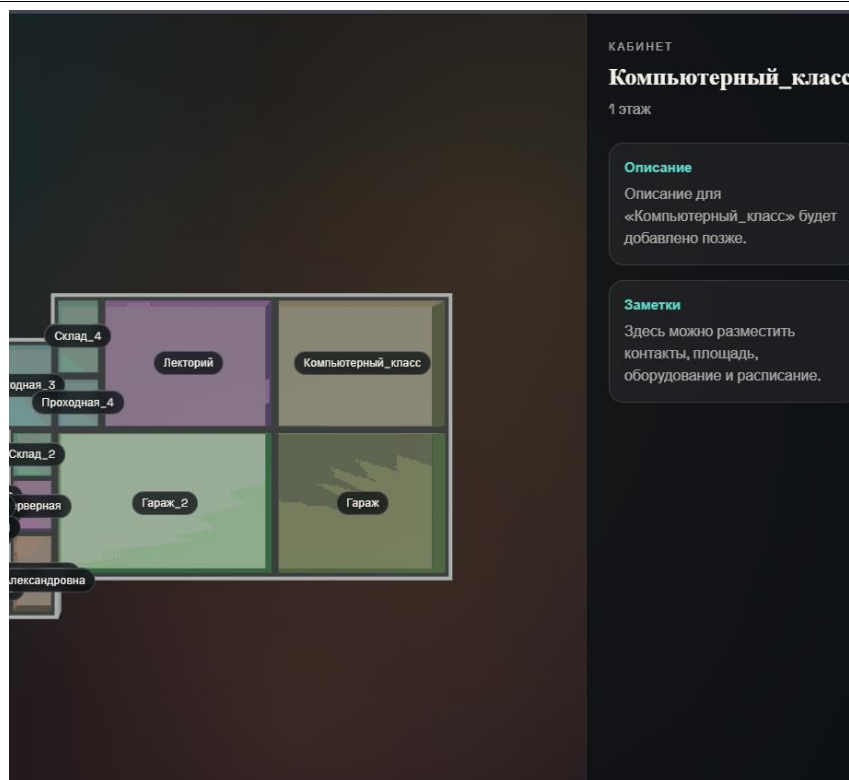


Рис. 6. Результат работы системы

4 Выводы

В ходе работы разработан механизм интерактивного отображения зон помещений на трёхмерной модели здания в веб-приложении. Реализовано автоматическое выявление зональных мешей, их преобразование в полупрозрачные визуальные оверлеи и привязка текстовых подписей. Для взаимодействия с пользователем применён механизм трассировки лучей, обеспечивающий корректное определение выбранного помещения и вывод связанной информации в боковой панели. Разработанный подход повышает наглядность трёхмерной визуализации и демонстрирует возможность эффективного использования браузерных технологий для создания интерактивных навигационных и информационных систем на основе 3D-моделей зданий.

Библиографический список

1. Митяева А.П., Дроздова Е.Н. Анализ принципов создания 3D графики на примере библиотеки Three.js // Вестник молодых ученых Санкт-Петербургского государственного университета технологии и дизайна. 2021. № 3. С. 42–47.
2. Зуева Д.И. Исследование показателей производительности рендеринга трехмерной графики WebGL Three.js // Научно-технические инновации и веб-технологии. 2023. № 1. С. 64–69.
3. Вильданов А.Н., Исмагилов А.И. Трехмерное моделирование в браузере с помощью Three.js // Дневник науки. 2023. № 6 (78).

4. Зуева Д.И. Исследование способов оптимизации производительности рендеринга 3D-графики WebGL Three.js // Научно-технические инновации и веб-технологии. 2023. № 2. С. 27–32.
5. Mohan Balakrishnan Ja., Manjunath Chavan S. Programmers' perspectives on the use of previsualization in the design process of Three.js web animation // American Journal of Art and Design. 2022. Т. 7. № 1. С. 1.