

THREE.JS: ОТКРОЙТЕ МИР 3D-ГРАФИКИ В ВАШЕМ БРАУЗЕРЕ

Ёвженко Юрий Дмитриевич

студент, Гомельский Государственный Технический университет имени П.О. Сухого,
Республика Беларусь, г. Гомель

Крепостной Александр Владиславович

студент, Гомельский Государственный Технический университет имени П.О. Сухого,
Республика Беларусь, г. Гомель

Three.js представляет собой мощную JavaScript-библиотеку, существенно упрощающую работу с WebGL [1]. В отличие от низкоуровневого WebGL API, требующего глубоких знаний графического конвейера, Three.js предлагает удобные абстракции, делая 3D-графику доступной для широкого круга разработчиков. Библиотека стала стандартом для веб-приложений с 3D-графикой благодаря нескольким ключевым преимуществам. Она значительно снижает порог вхождения по сравнению с чистым WebGL, сохраняя при этом гибкость и производительность. Кроссплатформенная природа решений на Three.js обеспечивает их работу во всех современных браузерах на различных устройствах.

Three.js в игровой разработке. Хотя Three.js не позиционируется как полноценный игровой движок как Unity или Unreal Engine (в нем отсутствуют встроенный визуальный редактор уровней, продвинутые системы физики "из коробки" или сложные инструменты для разработки искусственного интеллекта), он предоставляет исключительно мощный и гибкий графический фундамент для создания браузерных игр [2]. Для многих разработчиков, особенно тех, кто уже знаком с JavaScript и веб-экосистемой, Three.js становится привлекательным выбором.

Почему Three.js подходит для разработки игр?

1. Мгновенная доступность для игроков: это, пожалуй, одно из самых значительных преимуществ. Игры, созданные на Three.js, запускаются непосредственно в браузере.
2. Легковесность и производительность для определенного класса игр: для казуальных игр, аркад, головоломок, инди-проектов производительности, которую обеспечивает Three.js через WebGL, более чем достаточно.
3. Тесная интеграция с веб-технологиями: Three.js, будучи JavaScript-библиотекой, идеально вписывается в экосистему веб-разработки. Это означает легкую интеграцию с HTML для создания пользовательского интерфейса, CSS для его стилизации.
4. Полный контроль и гибкость: разработчик имеет полный контроль над циклом рендеринга и всей игровой логикой. Это позволяет создавать уникальные игровые механики и визуальные стили.

Представьте себе, например, создание простого 3D-платформера. С помощью Three.js вы могли бы смоделировать уровни из кубов и плоскостей, создать анимированного персонажа, настроить камеру, следующую за ним, и реализовать логику прыжков и перемещения. Three.js предоставит все инструменты для визуализации этой сцены, в то время как остальная игровая логика будет написана на JavaScript.



Рисунок 1. Пример созданной игры

Three.js как инструмент визуализации и вспомогательных интерфейсов. Мощь и гибкость Three.js находят применение далеко не только в сфере развлечений. Библиотека стала незаменимым инструментом для создания сложных интерактивных 3D-визуализаций данных, конфигураторов продуктов, обучающих симуляторов и разнообразных вспомогательных элементов отображения, где трехмерное представление способно значительно улучшить понимание и взаимодействие пользователя с информацией.

Одним из наиболее ярких и практически значимых примеров является разработка редактора планограмм – специализированного программного обеспечения для ритейла, которое позволяет визуализировать и оптимизировать расположение товаров на полках и стеллажах в торговых точках.

Классические планограммы часто представляют собой плоские схемы или таблицы, которые не всегда дают полное представление о том, как товары будут выглядеть в реальности, как они будут взаимодействовать друг с другом по высоте, глубине и ширине, и насколько эффективно используется торговое пространство. Использование Three.js позволяет создать полноценный интерактивный 3D-редактор. В таком редакторе:

1. Стеллажи и товары как трехмерные объекты, соответствующие реальным размерам и конфигурациям.
2. Интерактивное размещение: мерчендайзер может в режиме реального времени «брать» виртуальные товары и размещать их на полках, вращать, изменять количество фейсингов (видимых единиц товара), группировать.
3. Аналитика и валидация: редактор может подсвечивать ошибки (например, если товар не помещается на полку по габаритам, или если нарушены правила соседства), рассчитывать эффективность использования полочного пространства, визуализировать "горячие" и "холодные" зоны на основе данных о продажах.

Преимущества 3D-подхода в планограммировании:

1. Наглядность: трехмерное представление гораздо интуитивнее и позволяет лучше оценить визуальную привлекательность выкладки.
2. Точность: возможность работать с реальными размерами и видеть результат в 3D

минимизирует ошибки при фактической выкладке в магазине.

3. Улучшенный пользовательский опыт: прямое манипулирование 3D-объектами более естественно для человека, чем работа с абстрактными таблицами.

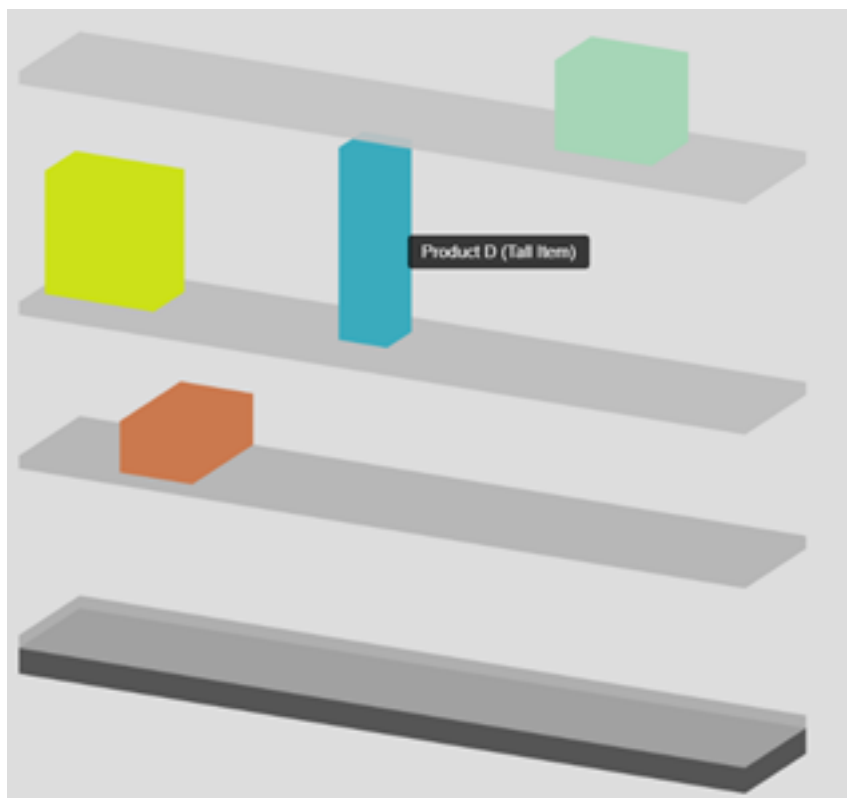


Рисунок 2. Пример редактора планограмм

Заключение. Three.js – это гораздо больше, чем просто библиотека. Это мощный инструмент в веб-разработке. Он демократизировал доступ к созданию трехмерного контента, предоставив разработчикам по всему миру инструменты для воплощения самых смелых и технически сложных идей. От захватывающих браузерных игр, не требующих установки, до высокоточных профессиональных инструментов визуализации и редактирования данных – Three.js доказывает свою универсальность и незаменимость.

Список литературы:

1. Введение в 3D: основы Three.js [электронный ресурс] // Habr. – Режим доступа: <https://habr.com/ru/articles/494810/>
2. Документация Three.js [электронный ресурс]. – Режим доступа: <https://threejs.org/docs/index.html#manual/en/introduction>