

3D МОДЕЛИРОВАНИЕ В ВЕБ-ДИЗАЙНЕ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Волк А. О.

Киселевский О. С. – к.т.н., доцент

В современном мире активно развивается 3d-моделирование. Оно нашло применение в таких сферах человеческой деятельности как: архитектура, инженерия, дизайн и многое другое.

Анализ перечня самых распространенных в сфере 3d-моделирования инструментов показал, что на данный момент большая часть этих инструментов используются как классические приложения для рабочего стола [1]. Данная тенденция обусловлена тем фактом, что подобные инструменты требуют большой производительности от пользовательской рабочей станции и поэтому выполнены с использованием платформозависимого стека технологий, а также сами приложения достаточно велики для того чтобы быть портативными.

Тем не менее, последние разработки в сфере инструментария 3d-моделирования показали, что производительность компьютеров достаточно выросла для того чтобы уйти от платформозависимых и высокопроизводительных технологий (за исключением некоторых особенно ресурсоемких процессов) в пользу интерпретируемых языков (например, python для Blender[2], Godot[3]). Более того, сама инфраструктура веб-браузеров достаточно обогатилась чтобы поддерживать собственные (т.н. Hosted-приложения) веб-приложения и управлять их ресурсами. API браузеров позволяет, где нужно, реализовать низкоуровневый доступ к ресурсам компьютера (asm-js[4]) а также эффективно использовать аппаратное ускорение для даже самых тривиальных задач рендеринга графики.

Удобство использования веб-браузера в качестве платформы для реализации приложения состоит в том, что работа приложения не зависит от особенностей платформы, на которой оно выполняется (за исключением частных особенностей различных интерпретаторов JavaScript), что позволяет переиспользовать код приложения на разных операционных системах (Будь то MS Windows, любые из дистрибутивов Linux, Google Web OS, OS X, и даже ОС мобильных устройств, например Android) реализовав минимальный набор polyfill функций. Еще один критерий, добавляющий удобства подобным web-based системам состоит в том, что очень многие приложения (Редакторы документов, чаты, медиаплееры) адаптируются к такой модели работы, что упростит взаимодействие с пользователем - процесс его работы с приложением уже будет ему знаком по аналогии с другими приложениями. Также использование веб-браузера позволяет упростить процесс обновления приложения, так как оно находится в централизованной локации и доступ к нему управляется сервером, который контролируется разработчиками.

Подобную практику уже адаптировали такие крупные компании, как Google (в их продуктах Docs, Sheets, Play и прочих), Microsoft (Серия приложений Office 365, Dynamics CRM, SharePoint Online, Azure), Adobe (Creative Cloud) и некоторые из компаний среднего звена также начинают переносить свои продукты на веб-платформы (SelfCAD, Clara.io). Основной их недостаток состоит в том, что пока что они достаточно примитивны и не позволяют пользователю получить законченную профессионально выглядящую картинку, и используются лишь для редактирования трехмерной геометрии.

Для реализации высокопроизводительной системы для работы с 3D графикой предлагается использовать модель организации приложения, указанную на рисунке 1. Та часть приложения, которая отвечает за ключевой функционал обработки и рендеринга графики, должна быть реализована в виде Hosted приложения для браузера для того чтобы облегчить процесс его загрузки и переносимость.

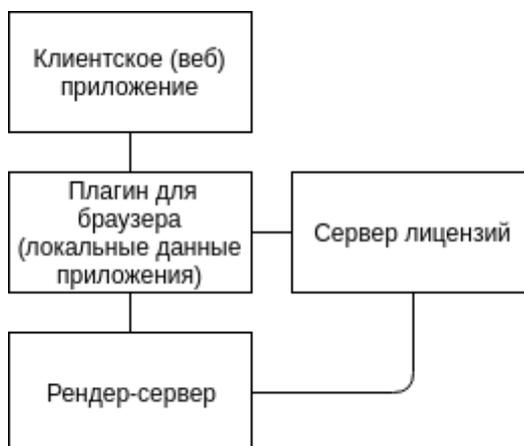


Рис. 1 - Диаграмма взаимодействия компонентов системы

Клиентская часть системы - обычный веб-сайт, предоставляющий пользовательский интерфейс для работы с функционалом, реализованным внутри плагина для браузера. Наиболее ресурсоемкий функционал (рендеринг законченных сцен) переносится на сервер и выполняется асинхронно в процессе работы с приложением.

Для реализации пользовательского интерфейса можно использовать любой из клиентских фреймворков, предназначенных для разработки одностраничных веб-приложений, таких как ReactJs, Angular, Vue, Elm, MarkoJs. Для отображения трехмерной графики можно использовать библиотеки WebGL или three.js. Ключевой функционал плагина должен быть более производительным, поэтому для его реализации целесообразно использовать более низкоуровневые инструменты, такие как asm.js. В то время как сервер может быть реализован на любом из ныне используемых стеков серверных технологий.

К плюсам такого подхода можно отнести:

- Относительную переносимость решения
- Усложнение несанкционированного доступа к приложению
- Потенциальную прогрессивную модель монетизации
- Возможность постепенной разработки и централизованной доставки обновлений
- Облачное резервное копирование работы пользователя в автоматическом режиме

Однако решение не лишено ряда недостатков:

- Приложение доступно лишь через веб, то есть требует постоянный доступ к интернету
- Затраты на поддержку серверов приложения
- Несколько более низкая производительность по сравнению с классическим подходом

В заключение можно отметить перспективность данного направления в свете тенденций развития информационных технологий и сферы 3D моделирования в целом. Рынок приложений для 3D моделирования пока еще не насыщен продуктами, позволяющими портативный доступ к подобному инструментарию, а децентрализованность такой системы позволит реализовать инновативный способ монетизации (оплата серверного времени на рендеринг, единая лицензия для всех платформ).

Список использованных источников:

1. 3D Design/3D Modeling Software Tools for Model Makers <https://www.g2crowd.com/categories/3d-modeling>
2. GitHub - Исходный код пакета 3D графики Blender <https://github.com/dfelinto/blender>
3. GitHub - Исходный код игрового движка Godot Engine <https://github.com/godotengine/godot>
4. Черновик спецификации ASM.js - <http://asmjs.org/spec/latest/>