**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

**(Университет ИТМО)**

**Факультет** Программной инженерии и компьютерной техники (ФПИ и КТ)

**Образовательная программа:** Системное и прикладное программное обеспечение

**Направление подготовки (специальность)**: 09.03.04, Программная инженерия

О Т Ч Е Т

*о учебной, ознакомительной практике*

Тема задания: *Оптимизация и редизайн среды просмотра mtlx-формата в 3D в браузере*

Обучающийся: *Кульбако Артемий Юрьевич, P34112*

Руководитель практики от профильной организации: *Кравченко Андрей Сергеевич, Senior Programmer в ООО "Люксофт Профешнл"*

Руководитель практики от университета: *Маркина Татьяна Анатольевна, доцент*

Дата: 18.02.2022

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc96196382)

[1. ВВЕДЕНИЕ 3](#_Toc96196383)

[2. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 5](#_Toc96196384)

[2.1 Инструктаж обучающегося 5](#_Toc96196385)

[2.2 Изучение теоретической базы рендера трёхмерных изображений 6](#_Toc96196386)

[2.3 Оптимизация рендер-пайплайна среды просмотра 7](#_Toc96196387)

[2.4 Создание нового UI среды просмотра 9](#_Toc96196388)

[2.5 Получение обратной связи по проделанной работе 10](#_Toc96196389)

[2.6 Оформлением отчётности 10](#_Toc96196390)

[3. ЗАКЛЮЧЕНИЕ 11](#_Toc96196391)

# 1. ВВЕДЕНИЕ

Учебная практика – важнейший этап образовательного процесса, направленный на проверку и закрепление компетенций обучающегося, полученных в процессе академического обучение, перед прохождением производственной, технологической практики.

Целью учебной практики является ознакомление обучающегося с рабочими процессами согласно своему направлению подготовки в рамках организации, осуществляющей практику посредством выполнения индивидуального задания (*Таблица 1*).

Таблица 1 - Индивидуальное задание

|  |  |  |
| --- | --- | --- |
| **Порядковый № этапа** | **Наименование этапа** | **Задание этапа** |
| 1 | Инструктаж обучающегося | Необходимо ознакомиться с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка (включает следующие документы):   * Кодекс делового поведения DXC * Политика этического и делового поведения DXC * Политика DXC об ограничении сделок с ценными бумагами компании на основе инсайдерской информации * Принципы пользования социальными сетями * Антикоррупционная политика * Политика по конфликту интересов * Положение о конфиденциальной информации * Положение о защите и обработке персональных данных * Правила внутреннего трудового распорядка * Положение о ежемесячном премировании * Положение об оплате труда работников * Положение о премировании * Должностная инструкция   Также необходимо пройти вступительный онлайн-курс о информационной безопасности внутри компании. |
| 2 | Изучение теоретической базы рендера трёхмерных изображений | Изучение основных алгоритмов и процессов рендера трехмерных изображений и терминов, используемых в 3D-утилитах: сцены, геометрия, меши, постобработка, шейдеры и т.д. Подойдёт любой софт для трёхмерного моделирования (3ds Max, Blender, Maya). Изучить возможность реализации того или иного аспекта методами библиотек Three.js (URL: <https://threejs.org>) и WebGL (URL: <https://www.khronos.org/webgl>), а также актуальную версию MaterialX (URL: <https://www.materialx.org>) - открытого стандарта материалов для 3D-утилит. |
| 3 | Оптимизация рендер-пайплайна среды просмотра | Цель этапа - провести оптимизацию MaterialX 3D-viewer (далее для краткости будет использоваться термин viewer) – ПО, предоставляющее возможность просмотра файлов формата mtlx в веб-браузере. Пример работы viewer-а можно увидеть на сайте matlib.gpuopen (URL: <https://matlib.gpuopen.com>), открыв любой из материалов и нажав кнопку `3D` в правом верхнем углу галереи. Используемые технологии: Vuetify (URL: <https://vuetifyjs.com>) для построения интерфейса и Three.js для рендера, язык программирования – JavaScript (URL: <https://www.javascript.com>). Замерить потребление ресурсов с помощью средств мониторинга производительности браузера или операционной системы, найти утечку памяти, возникающую при смене модели и/или окружения, и устранить её. Уменьшить время загрузки viewer-а (стартовое и при изменении параметров), улучшить работу на слабом оборудовании, при этом сохранив или улучшив качество получаемой картинки. |
| 4 | Создание нового UI среды просмотра | Цель этапа - создать новый пользовательский интерфейс viewer-а на основе требований, выдвинутых командами QA и 3D-artist. Главное требование - сделать его легче и нагляднее, путём перехода от текстового представления параметров к графическому. Новый интерфейс должен соответствовать стилю сайта и общему стилю продуктов AMD. |
| 5 | Получение обратной связи по проделанной работе | Создать pull-request-ы, принять участие в дискуссии code-review с остальными членами команды. Внести правки и задеплоить новую версию на тестовый сервер. Получить обратную связь от команд QA (контроль качества) и 3D-artist (художники по 3D) о проделанной работе, будущих улучшениях и развитии проекта. |
| 6 | Оформлением отчётности | Оформить необходимые документы и отчёты о процессе и результатах прохождения учебной практики. Сформировать задание на технологическую практику. |

# 2. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

## 2.1 Инструктаж обучающегося

В ходе выполнения данного этапа я внимательно изучил все необходимые документы, прослушал инструктаж на рабочем месте, познакомился с коллегами и руководителем практики. После знакомства, руководитель практики провёл презентацию о рабочем проекте и выдал доступ к системе управления проектом Atlassian JIRA (*Рисунок 1*), где были заведены задачи для выполнения на моё имя, а также к репозиториям проекта в системе контроля версий и непрерывного развёртывания Gitlab (*Рисунок 2*).

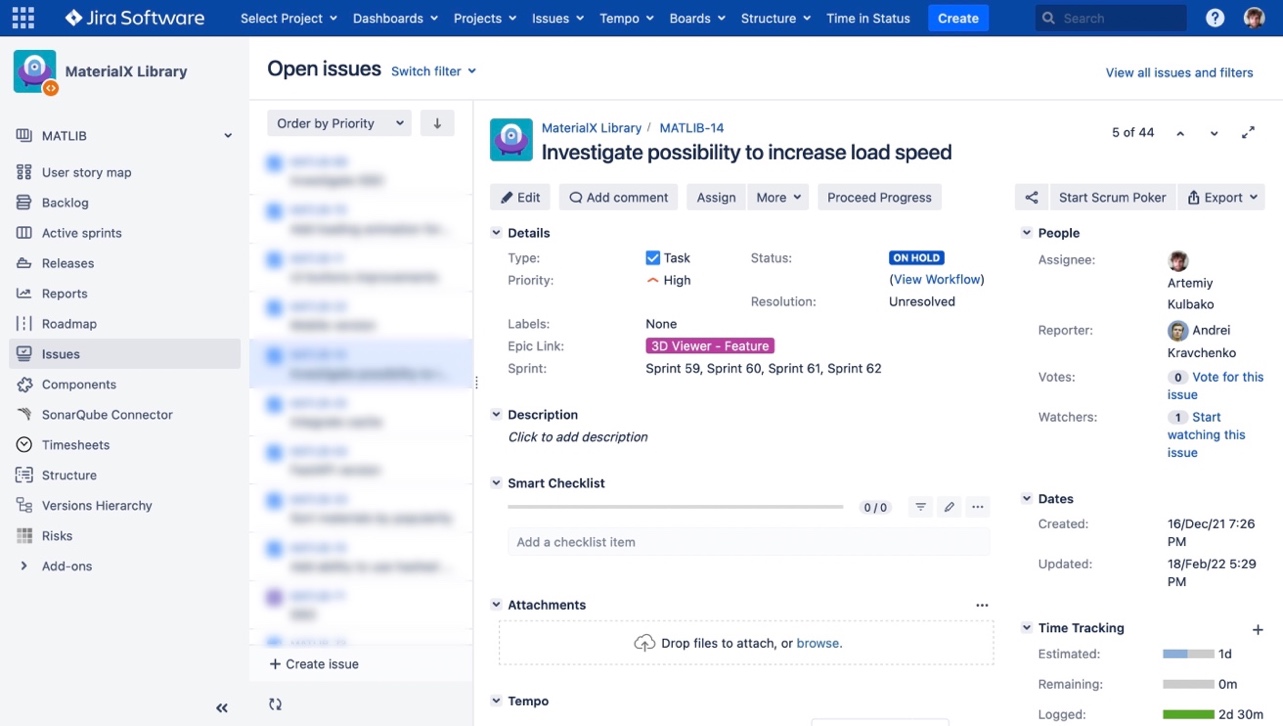


Рисунок 1 - Пример задачи в Jira

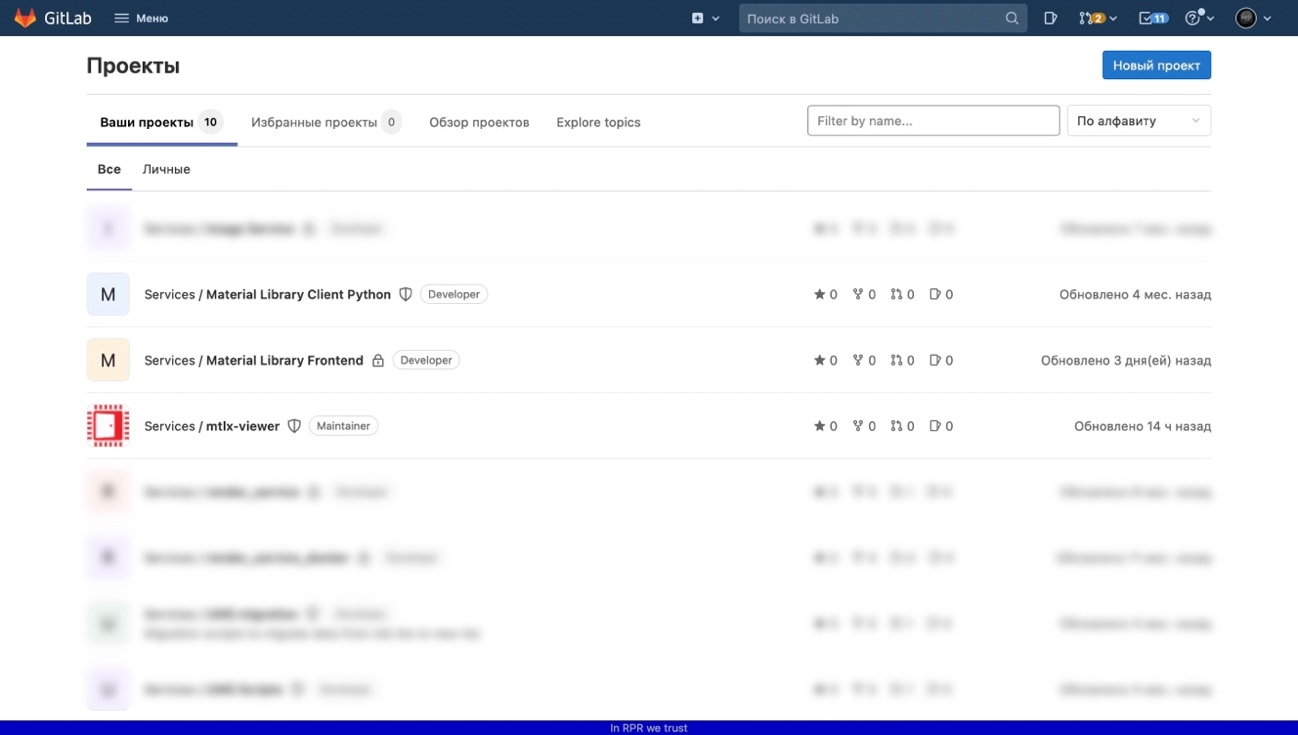


Рисунок 2 - Репозитории проекта в Gitlab

Я ознакомился с интерфейсами систем и их базовой функциональностью, необходимой для выполнения индивидуального задания: принятие задач к исполнению, логирование времени, отправка коммитов, просмотр обсуждений к pull-request-ам. Далее мне был предоставлен доступ в корпоративные средства связи: Microsoft Teams, Skype и Slack, используемые для проведения ежедневных митингов, общения с командой 3D-artist и командой QA соответственно.

Результатом выполнения этапа стали навыки работы с вышеописанными программами, что позволило мне интегрироваться в процессы разработки в команде.

## 2.2 Изучение теоретической базы рендера трёхмерных изображений

Следующим этапом моего задания стало изучение теоретической базы рендера трёхмерных изображений. Я изучил первые главы книги «Программирование компьютерной графики. Современный OpenGL» (Борисков А. В. Программирование компьютерной графики. Современный OpenGL: ДМК Пресс, 2019. 373 c.): в них были описаны операции с векторами, позволяющие трансформировать двухмерные проекции фигур в трёхмерные, алгоритм шэйдинга – создание теней, а также реализации этих операций на языке C++.

Изучать термины трёхмерного моделирования я решил на примере библиотеки Three.js – она имеет очень подробную документацию с примерами по каждой теме (URL: <https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>), а также применяется в проекте, над которым я работаю в рамках индивидуального задания. Я познакомился с такими терминами как сцена, меш, геометрия, материал, шейдер, камера, направленный и рассеянный свет, научился загружать 3D-модели в форматах glb и obj, создавать простые сцены и взаимодействовать с ними. Программный код примеров предоставлен в *Приложении 1 – Примеры\_Threejs*. Пример одной из сделанных мною сцен - загруженная из файла сфера и рендер её полигональной сетки (*Рисунок 3*).

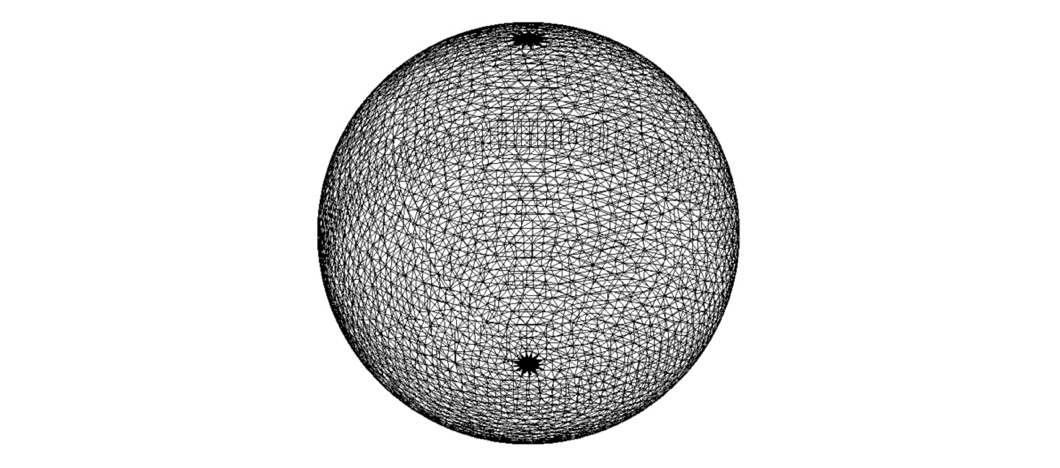


Рисунок 3 - Пример рендера полигональной сетки сферы

## 2.3 Оптимизация рендер-пайплайна среды просмотра

Cледующие несколько дней заняло изучение уже существующей кодовой базы и после я приступил к заданию. Используя знания, полученные на прошлом этапе, я предположил, что утечка памяти возникает из-за отсутствия механизма очистки ресурсов. Дело в том, что проект недавно перенесли с «чистого» JavaScript на реактивный фреймворк Vue (URL: <https://vuejs.org>), что поменяло жизненный цикл веб-страницы с рендером. Я провёл измерения потребляемой памяти инструментами разработчика в браузере Firefox и теория подтвердилась. Мною была написана функция, которая складывает ссылки на все тяжёлые ресурсы в отдельный буфер и вызывает очистку уже ненужных объектов из памяти. Пример работы функции - *Рисунок 4* и сама функция - *Таблица 2*.

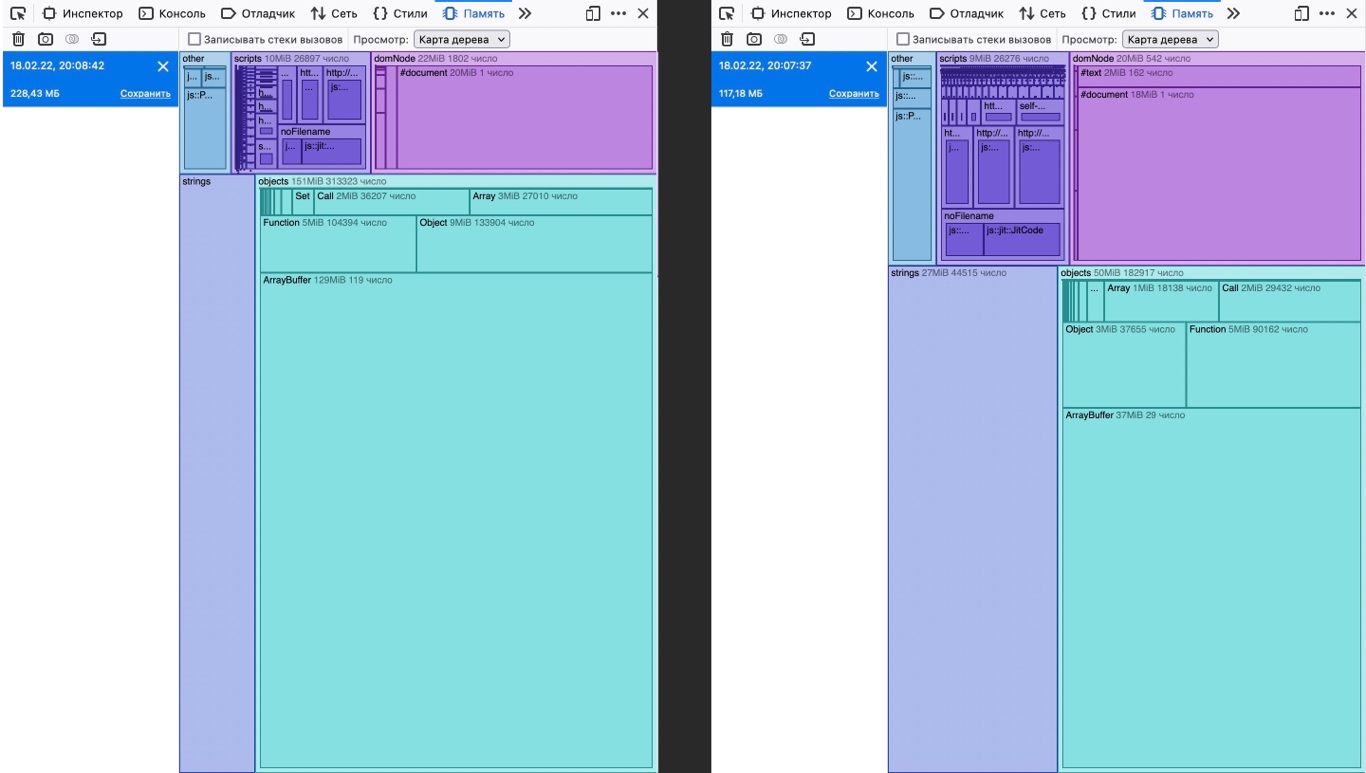


Рисунок 4 - Дамп памяти до и после

Таблица 2 - Функция очистки буфера

|  |
| --- |
| /\*\*  \* Clear assets { Object3D, Texture, Geometry } from memory and from scene.  \*/ clearAssetsFromMemory() {   /\*\*  \* Full clearing of the memory occupied by the object. If the geometry or material is being used by another object,  \* it will be destroyed.  \*/  Object3D.prototype.fullDispose = function() {  this.traverse( child => {  if (child.isMesh) {  child.geometry.dispose()  child.material.dispose()  }  })  }    const { scene, texturesBuffer, model, renderer } = this  model?.fullDispose()  texturesBuffer.forEach( it => it.dispose() )  renderer?.dispose()  scene?.clear() } |

Также меня смутило «размытие» полученного рендера. Такое бывает, когда разрешение изображения меньше разрешения экрана. Проведя исследование, я выяснил, что вызов функции, которая устанавливает разрешение рендера происходит неправильно – не учитывается DPR браузера (URL: <https://developer.mozilla.org/ru/docs/Web/API/Window/devicePixelRatio>). Я исправил ошибку, пример - *Рисунок 5.*

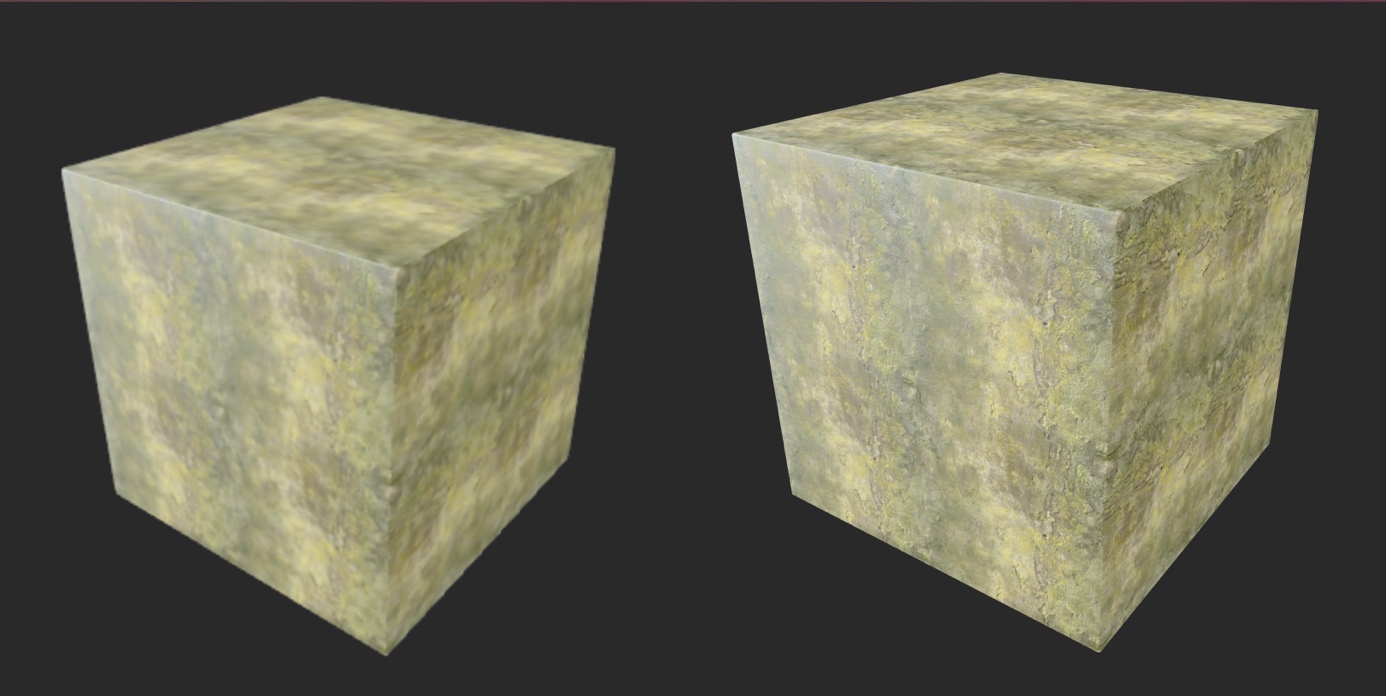


Рисунок 5 - Разрешение рендера до и после

Последним этапом оптимизации стал процесс уменьшения статических ресурсов – файлов текстур и моделей. Изначально их качество было избыточно для целей рендера в браузере. Я использовал фоторедактор Pixelmator Pro для сжатия изображений с потерями и Blender для усечения полигонов в 3D-моделях. Итоговый размер всех ресурсов уменьшился более чем в три раза, при этом разницу заметить трудно.

Я оформил pull-request и отправил изменения на согласование руководителю. Проблем изменения не вызвали, и они были приняты. В рамках этого этапа индивидуального задания я понял кодовую база рабочего проекта и научился работать с ним.

## 2.4 Создание нового UI среды просмотра

Вторую рабочую задачу – создание нового пользовательского интерфейса - я начал с изучения существующих css-правил и компонентов сайта. После изучения мною был разработан первый прототип нового интерфейса, согласно главному требованию задачи – сделать его более наглядным: меньше текста (старый интерфейс - *Рисунок 6*), больше визуальной информации. Мною был подготовлен первый прототип () и представлен команде 3D-artist.

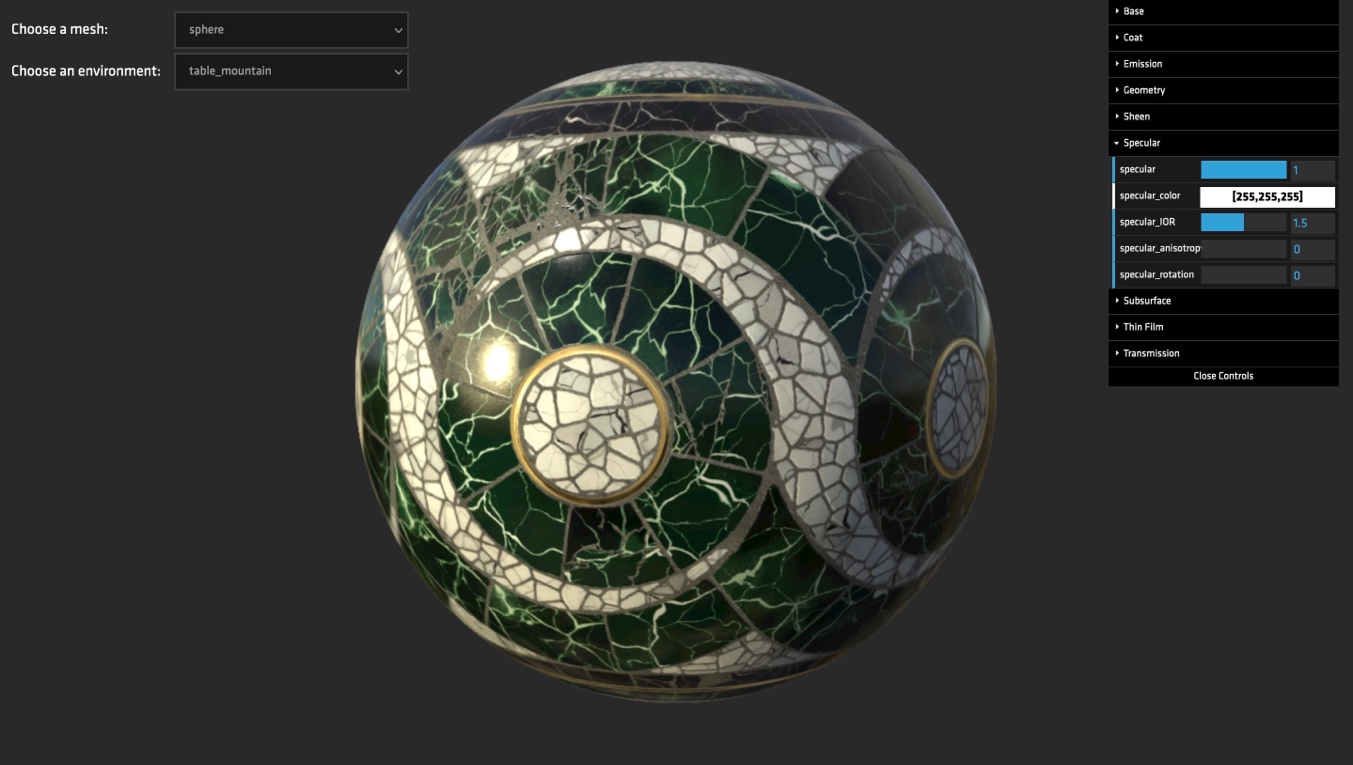


Рисунок 6 - Старый интерфейс

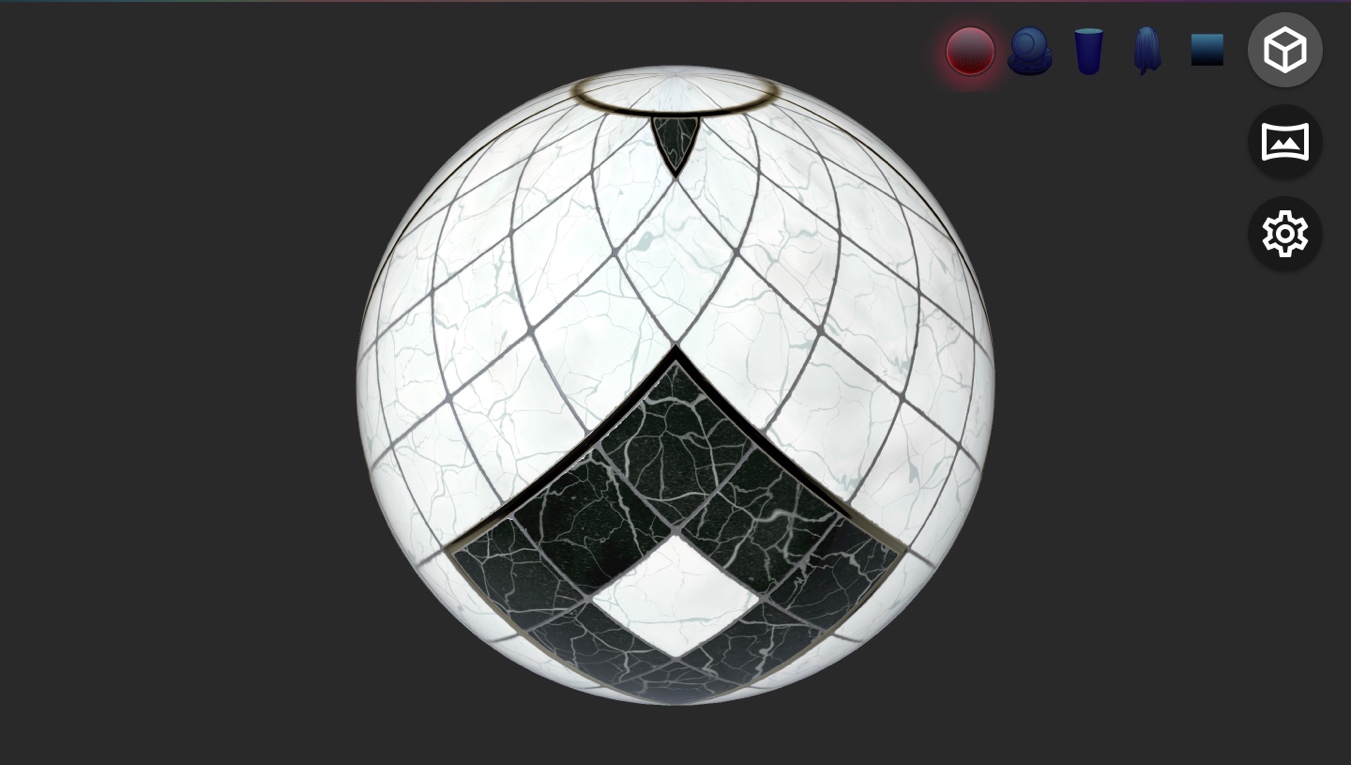


Рисунок 7 - Прототип интерфейса

Получив обратную связь, о необходимости сделать интерфейс менее ярким и ещё больше соотнести его со стилем основного сайта, после обсуждений возможных вариантов, я внёс необходимые изменения и показал новый прототип художникам. Он и стал финальным (*Рисунок 8*).



Рисунок 8 - Финальный интерфейс

В результате работы над этим этапом практики я получил опыт работы в команде, мы вместе обсуждали прототипы дизайна интерфейсов и принимали решение об изменениях.

## 2.5 Получение обратной связи по проделанной работе

После выполнения заданий, руководитель практики организовал звонок, где дал обратную связь по выполненной работе: работа выполнена в срок и качественно. Процедура код-ревью не понадобилась, так как я следовал принятым внутри команды стандартам оформления кода на JavaScript, а команда QA не обнаружила критических багов, только недочёты, которые мне надо будет поправить во время технологической практики.

## 2.6 Оформлением отчётности

Последний день практики целиком ушёл на написание отчёта и формирования задания на технологическую практику – обсуждения с руководителем плана развития проекта и моих задач в нём. Этот этап научил меня эмпирически определять сроки выполнения рабочих задач в рамках методологии Agile Sprint (URL: <https://www.atlassian.com/ru/agile/scrum/sprints>).

# 3. ЗАКЛЮЧЕНИЕ

Цель практики – ознакомить меня с рабочими процессами и интегрировать в них – была выполнена и даже перевыполнена: я поработал над настоящим аутсорс проектом для AMD, а сделанные мною изменения отправятся в релиз. К тому же, я наконец смог и поработать с трёхмерной графикой. Это оказалось не так сложно, как я себе представлял раньше и даже возможно на понятном мне технологическом стеке, что поспособствовало моему профессиональному развитию и мотивировало и дальше изучать эту предметную область.