

Национальный исследовательский университет
«Высшая школа экономики»

Лицей

Индивидуальная выпускная работа
Приложение “PlainStereo”

Выполнил Медведев Андрей Олегович

Москва 2026

Введение

Проект PlainStereo представляет собой десктопное приложение, разработанное с целью решения актуальной проблемы в образовательной сфере. Приложение предназначено для визуализации стереометрических фигур, построения сечений и анализа взаимосвязей между геометрическими объектами. Идея создания этого продукта возникла из наблюдения разрыва между потребностями учащихся в инструменте для визуализации сложных геометрических конструкций и ограниченностью существующих решений. Существующие приложения либо чрезмерно усложнены в использовании, либо ограничены в функциональности, что приводит к значительным временными потерям при изучении стереометрии. PlainStereo был задуман как инструмент, который объединил бы простоту интерфейса с более широким функционалом, позволяя пользователям быстро и эффективно решать задачи, связанные со стереометрией.

Проблемное поле

В образовательной и профессиональной сферах, связанных с математикой, инженерией и дизайном, стереометрия играет ключевую роль для понимания пространственных отношений и расчетов. Учащиеся и преподаватели регулярно сталкиваются с необходимостью визуализировать сечения фигур, определять углы, расстояния и объемы в 3D-пространстве. При решении этих задач возникают проблемы, не встречающиеся в алгебре или планиметрии. Например, даже у людей с большим опытом редко получается построить «удобное» изображение для стереометрической задачи с первого раза. При подготовке к экзаменам по математике многие бы были не против использовать инструмент, который позволил бы быстрее найти ошибки в их решениях или в начале изучения стереометрии визуализировать задачи, чтобы понять их лучше. Однако отсутствует удобный инструмент, который позволял бы интуитивно визуализировать задачу без больших потерь времени,

существующие решения либо слишком сложны в использовании, либо ограничены базовым функционалом, что приводит к большим временным затратам на понимание стереометрии, ошибкам в понимании материала и снижению эффективности подготовки. Недостаток простоты интерфейса и отсутствие некоторых функций приводят к тому, что использовать существующие инструменты невыгодно из-за потерь времени. Данный программный продукт позволит решить эти проблемы и обеспечит пользователям широкий функционал и простой интерфейс.

Для обоснования актуальности проблемного поля и валидации целевой аудитории был проведён опрос среди 23 респондентов из целевой аудитории (учащихся старших классов). Опрос включал 5 закрытых и 1 открытый вопросы, направленных на выявление потребностей и оценку актуальности разработки инструмента. Результаты показали высокую заинтересованность целевой аудитории в инструменте, который был бы одновременно простым в использовании и функционально полноценным. Большинство респондентов указали, что текущие инструменты либо сложны в освоении, либо требуют излишних действий для простых операций. Анализ ответов выявил основные проблемы текущих решений: сложность ввода данных, нет возможности ввода точных данных, сложность интерфейса, а также небольшой функционал. Эти выводы послужили основанием для формирования требований к функциональности PlainStereo.

Целевая аудитория

Продукт PlainStereo ориентирован на учащихся старших классов школ в возрасте от 15 до 18 лет, изучающих геометрию в рамках школьной программы. Целевая аудитория обладает минимальным уровнем навыков работы с персональным компьютером, достаточным для пользования стандартным программным обеспечением, однако не имеет специального опыта программирования или математического моделирования. Интернет-соединение

требуется только для функций аутентификации и синхронизации данных, что позволяет пользователям работать в автономном режиме при необходимости.

Функциональные требования

Разработанное приложение предоставляет широкий спектр функциональных возможностей, специально подобранных для решения выявленных потребностей целевой аудитории.

Функция аутентификации позволяет выбирать между входом с синхронизацией объектов на сервер или работой в локальном режиме без синхронизации, что позволяет использовать программу без подключения к интернету. Для авторизованных пользователей доступно создание файлов, сохраняемых на сервере проекта, что позволяет сохранять и восстанавливать работу на разных устройствах. При восстановлении или регистрации аккаунта на email отправляется код, что позволяет удостовериться в принадлежности пользователю данного email или убедиться в возможности восстановить аккаунт в дальнейшем благодаря доступу к почте.

Одним из функциональных требований является выбор и визуализация стереометрических фигур из предустановленного списка, включающего куб, призму, пирамиду, конус, цилиндр, шар и другие фигуры, с поддержкой синтаксиса КaTeX для наименования вершин, что позволяет адаптировать обозначения для конкретной задачи.

Самое важное функциональное требование, интуитивный ввод данных, реализовано через текстовое поле с поддержкой специального синтаксиса для точного ввода координат и объектов. Приложение предоставляет возможность построения плоскостей множеством методов с помощью того, что называется в программе инструкциями. С помощью них программа понимает запрос пользователя и вычисляет углы и расстояния между различными объектами, позволяет описывать более сложные условия. В этих и других инструкциях легко разобраться благодаря наличию в программе окна с помощью по инструкциям. Сечения могут быть построены как через текстовое поле, так и

через визуальный ввод через клики на трёхмерной модели для быстрого построения сечения.

Используемый стек технологий

Для разработки PlainStereo был выбран современный стек технологий, обеспечивающий кроссплатформенность, производительность и удобство разработки. Three.js выступает в качестве основной библиотеки для трёхмерной визуализации, обеспечивая высокопроизводительный рендеринг трёхмерных объектов и интерактивное взаимодействие с ними. HTML, CSS и JavaScript использовались для разработки интерфейса приложения, обеспечивая его отзывчивость и удобство использования. Tauri выбран в качестве оболочки приложения, позволяющей упаковать веб-приложение в самостоятельное приложение с минимальными накладными расходами и доступом к системным ресурсам. На данный момент доступен только вариант приложения для Windows, однако благодаря Tauri есть возможность сделать приложение доступным на большинстве операционных систем при исправлении всех недочетов, которые могут возникнуть для каждой операционной системы. Серверная часть приложения разработана на PHP, что обеспечивает простоту в установке и расширения серверного функционала. MySQL использована в качестве основной базы данных для хранения данных пользователей, синхронизированных объектов и истории работы. GitHub использован для контроля версий кода, обеспечивая возможность совместной разработки и отслеживания изменений на протяжении всего проекта.

Этапы работы над проектом

Январь 2024 года - начало разработки и работы над проектом

Я начал работу над проектом с анализа существующих решений. На этом этапе я изучил все доступные платформы, которые предлагали свое решение визуализации стереометрических задач, однако я понял, что они либо не имеют

достаточного функционала, либо являются слишком сложными и требуют много времени на работу с ними.

Затем я приступил к реализации проекта. К середине января я написал функционал: выбор стереометрической фигуры из списка (куб, пирамида, конус, цилиндр, шар и другие) и их визуализация с возможностью масштабирования и просмотра с разных углов

Июнь 2025 года — работа над проектом и разработка некоторых функциональных требований

Далее я определил для себя функциональные требования для программы и сделал первые шаги к реализации ввода и обработки инструкций: я написал парсер некоторых инструкций под их синтаксис и обработку некоторых из них. Реализовал полностью инструкции запроса расстояния между точками, длины отрезка и некоторые другие

Август 2025 года — разработка ввода и обработки параметров фигуры

Я реализовал ввод и обработку параметров фигуры, включая возможность выбора и изменения параметров неправильных фигур

Октябрь 2025 года — разработка проектной заявки

Была полностью готова и отправлена проектная заявка, которую я в дальнейшем отредактировал в финальной части этапа работы

Ноябрь 2025 года — завершение реализации ввода и обработки инструкций

Реализовал ввод и обработку всех существующих инструкций

Декабрь 2025 года — реализация аутентификации

Изначально планировал писать серверную часть на Node.js, однако решил, что в моей ситуации более правильным решением будет использовать PHP, так как это был язык, в котором уже имелся опыт реализации аутентификации

Реализовал регистрацию, авторизацию, восстановление аккаунта, включая отправку кодов по email и frontend

Январь 2026 года - итоговая версия приложения

Реализовал работу с файлами: их выбор, удаление, изменение названия, экспорт, импорт

Исправил в аутентификации пользователя проблему: если пользователь не смог завершить регистрацию на этапе ввода кода из письма email, то у него оставался неподтвержденный email и он не смог бы пользоваться сервисом далее с этим email. Решением было использовать восстановление аккаунта для подтверждения email, о чем выводиться пользователю при попытке войти в аккаунт с неподтвержденным email

Исправил некоторые недочеты в продукте, немного изменил проектную заявку под текущую реализацию проекта, написал отчет и другую документацию для моего IT проекта

Краткий анализ аналогичных продуктов

Анализ 3 программных продуктов, которые максимально приближены к заданному функционалу, показал, что:

- <https://www.geogebra.org/3d>: интерфейс неинтуитивен — ввод данных через текстовое поле имеет более сложную систему, требующую явно указывать вид объектов (точки, отрезки и т.д.), что усложняет быстрый ввод; построение сечений сильно усложнено: для этого необходимо использовать команду вида $p = \text{ПересечьКонтуры}(a, b)$, есть упрощённый аналог этого действия: построить плоскость по трём точкам, однако таким способом нет возможности построить само сечение (нет возможности выделения его контура для улучшения визуального качества и т.д.); нет возможности быстро и интуитивно точно выбирать/изменять размеры для многогранников: для этого необходимо изменять координаты точек, по которым был построен многогранник; используются менее привычные для целевой аудитории моего проекта и более редкие для задач наименования для вершин в алфавитном порядке
- <https://repetitorka.ru/math>: Позволяет только выбрать фигуру и три точки для сечения с четким отображением контура, то есть функционал ограничен
- <https://www.desmos.com/3d?lang=ru>: Фокусируется на графиках, но для стереометрических фигур функционал минимален — отсутствует построение сечений, нет анализа взаимного расположения (углы, расстояния)

Рефлексия

В процессе работы над проектом мне пришлось встретиться с несколькими трудностями. Я вначале использовал диаграмму Ганта, однако потом сбился с неё из-за того, что реализация некоторых частей заняла сильно больше времени, чем я ожидал. Я считаю, что можно было бы сделать проект нацеленным на планиметрию, так как это бы сильно упростило мои задачи и позволило бы завершить разработку проекта раньше. Помимо этого, я сначала не продумал, что желательно использовать CSS фреймворк, так как в таком случае не надо выходить из HTML файла во время части frontend разработки, что делает разработку проекта проще, кроме случаев, когда нужно отредактировать JavaScript файл. Поэтому до определенного времени я использовал CSS без фреймворка, однако позже решил потратить несколько дней на то, чтобы перейти с чистого CSS на такой CSS фреймворк, как Tailwind. Решением такой проблемы будет то, что в будущих проектах я буду заранее продумывать то, что может сделать написание кода проще.

Пока я работал над проектом я приобрел важный навык - использование функций для повторяющегося кода и понятные названия для большинства объектов. Нередко я использовал переводчик на английский в тех случаях, когда не знал перевода с русского для того, чтобы название было понятно другим. Я также старался придерживаться одного стиля написания кода и некоторых стандартов при наименовании переменных в JavaScript и PHP.

Если бы я начинал проект заново, я бы уделил больше внимания выбору стека. Например, Three.js на первый взгляд хорошо подходил для моего проекта, однако координаты хранятся исключительно как вещественные числа, поэтому я не могу выдавать пользователем моей программы точные значения величин (в формате с корнями и/или дробями). Возможно, что это не было бы проблемой, если бы у меня были бы идеи и знания, как это исправить. Однако программа всё еще хорошо подходит для изучения стереометрии, особенно на ранних

этапах. Я бы также уделил больше внимания тому, чтобы выпустить это приложение в формате веб-сайта, а не только Desktop-приложения, так как для многих это было бы удобнее и пользоваться можно было бы со всех устройств. Tauri позволяет выпускать приложения для большинства известных операционных систем, однако есть свои особенности. Например, на Android возможно использование только Android System WebView, работа которого сильно отличается от WebView2 (Windows), из-за чего могут происходить некоторые баги, которых не было на Windows. Таким образом, я бы возможно выбрал другой стек для приложения.

Планирую развивать этот проект далее. На данный момент, вводить параметры фигуры необходимо через форму под текстовым полем для выражений. Однако я бы хотел реализовать ввод только через текстовое поле, так как я считаю, что было бы удобнее пользоваться только одним элементом для ввода данных, чем сразу несколькими. Это бы подвело меня к реализации еще одного значительного обновления в программе: добавление инструкций, задающих длину отрезка или величину угла. Это позволило бы охватить гораздо больший диапазон задач, которые можно было бы рассмотреть с помощью программы очень быстро по сравнению с аналогичными продуктами. Однако реализовать это весьма сложно с моими текущими знаниями и потребуется значительное время в будущем.