

Программная реализация (на языке JavaScript) алгоритмов генерации ФОС ЕГЭ по геометрии в 2024 году

Суматохина Александра 4 курс Кафедра Теории функции и геометрии
Научный руководитель: Авдеев Н.Н.

23 апреля 2024

Здравствуйте, меня зовут Суматохина Александра, я обучаюсь на 4 курсе, мой научный руководитель Семёнов Евгений Михайлович, научный консультант Авдеев Николай Николаевич. Тема моей курсовой работы Программная реализация (на языке JavaScript) алгоритмов генерации ФОС (Фонды оценочных средств) ЕГЭ по геометрии в 2024 году.

Существующие проблемы

Но за время обучения в 10 и 11 классе при подготовке к ЕГЭ школьники сталкиваются с дефицитом заданий по определённым категориям. Так в конце 2021 года в список заданий ЕГЭ были добавлены новые задания под номером 11 по теме «Графики функций», а в конце 2023 — задание №2 по теме «Вектора», количество которых для прорешивания было очень мало. А по теме «Производная и первообразная» банк заданий расходуется при подготовке с невероятной скоростью: так как это преимущественно графические задания, решение их занимает менее минуты, а их составление вручную занимает несоразмерно много времени.

Так как это преимущественно графические задания, решение их занимает менее минуты, а их составление вручную занимает несоразмерно много времени.

ЕГЭ является относительно неизменяемым экзаменом, поэтому все материалы, которые уже были выложены в открытый доступ имеют полные решения, что приводит к списыванию учениками.

При этом существуют задания с вспомогательным чертежом. Чаще всего для целого ряда заданий используется одна и та же иллюстрация, которая не всегда соответствует условиям задачи, а иногда отвлекают от решения. Проект «Час ЕГЭ» позволяет решить все эти проблемы.

Первая глава этой работы посвящена обзору вспомогательных функций, которые ускоряют написание шаблонов по теме «Планиметрия». Также приведён алгоритм написания шаблона с чертежом.

Вторая глава представляет решение проблемы отрисовки фигур в трёхмерном пространстве на языке программирования JavaScript; рассказывает о применении объектно-ориентированного программирования для упрощения написания шаблонов с чертежом; затрагивает вопрос об написании программного кода при помощи нейросетей; приводит обзор вспомогательных функций и алгоритм написания шаблона по теме «Стереометрия».

Проект «Час ЕГЭ»

«Час ЕГЭ» — компьютерный образовательный проект, разрабатываемый при математическом факультете ВГУ в рамках «OpenSource кластера» и предназначенный для помощи уча-

щимся старших классов, учителями и репетиторам при подготовке к тестовой части единого государственного экзамена.

Задания в «Час ЕГЭ» генерируются случайным образом по специализированным алгоритмам, называемых шаблонами, каждый из которых охватывает множество вариантов соответствующей ему задачи.

Достижения

- Полностью покрыт банк заданий ФИПИ по теме «Планиметрия»

В ядро добавлены функции для отрисовки:

- Условных обозначений на чертежах, такие как штрих-метка
- Углов, в отдельности прямых
- Обозначений для равных углов
- Отрезков заданной длины под некоторым углом
- Строк на середине отрезка

0.0.1. Элементы декларативного программирования

Определение. Декларативное программирование — парадигма программирования, в которой задается спецификация решения задачи, то есть описывается конечный результат, а не способ его достижения. [?]

Во время разработки шаблонов по теме «Графики функции» требовалось много раз переопределять коэффициенты функций через циклы `while` или `do...while`, пока они не начнут соответствовать заданным условиям (видимость графика, сливание его с осями, видимость целых точек). Это часто приводило к бесконечной работе шаблона, при этом сложно было определить, какое условие не выполняется.

Для этого было разработано окружение `retryWhileUndefined` для шаблонов, которое бы перезапускало их не более чем некоторое количество раз, если одно из условий не удовлетворено.

Но всё равно было тяжело определить, почему шаблон перезапускается. Для этого было разработано более совершенное окружение `retryWhileError`, которое не только могло бы ограничивать количество перезапусков, но и фиксировать, какие проверки не были пройдены и выводить их на экран (ошибки видны только для разработчика при отладке).

Для окружения были написаны функции-утверждения, которые имеют структуру : условие не выполнено - записать ошибку - перезапустить шаблон. Если максимальное количество повторений достигнуто, то вывести накопившиеся ошибки и количество их появлений.

Проблема отрисовки многогранников в JavaScript

- Отсутствуют встроенные средства для изображения трёхмерных фигур
- На данный момент существует только одна подходящая библиотека `Three.js`, которая могла бы выполнить проецирование координат фигуры на плоскость с учётом положения наблюдателя.
- Но при этом для создания любого объекта необходима не только камера, но и сцена с рендерингом, что значительно замедляет работу проекта. Поэтому эта библиотека нам не подходит.

- Подобные ей проводят проецирование на плоскость с поворотом только вокруг осей OX и OZ
- Ранне уже попытки написания функции проекций фигур. Но чертежи не соответствовали условиям задач.

С учётом прогресса современных технологий, встал закономерный вопрос, сможет ли нейросеть ChatGPT 3.5 сгенерировать код проектора. За несколько шагов удалось получить корректный, оптимизированный код.

Сокращение кода и введение канонических координат

Стоит отметить, что задач по теме "Стереометрия" огромное множество. Поэтому одной из первостепенных задач было сократить код шаблонов и исключить вычислительные ошибки. Для этого были разработаны классы многогранников, которые содержат в себе длины рёбер, объем, площади оснований, а так же тернарную матрицу связности и канонические координаты вершин.

Матрица может содержать значения: 1, 0, либо специальное значение, указывающий на отображении ребра пунктиром.

Каноническим положением будем называть такое расположение многогранника, когда его высота, проходящая через центр масс его основания, совпадает с осью аппликата и начало координат делится пополам.

При таком расположении, начало координат можно расположить в центре иллюстрация. Тогда чертёж не будет смещён ни в одну из сторон.

Так написание шаблона было сведено к 5 пунктам.

Этапы генерации

- Создание объекта нужного класса (фигуры)
- Преобразование канонических координат на двумерную плоскость при помощи функции `project3DTo2D`
- Масштабирование координат функцией `autoScale`
- Корректирование матрицы связей (добавление диагоналей или сечений)
- Отрисовка фигуры `drawFigure`

Достижения

- Полностью покрыт открытый банк заданий ФИПИ по темам: параллелепипеды, призмы, кубы.
- Разработано 35 шаблонов
- Проведён эксперимент по написанию проектора из $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ с помощью ChatGPT 3.5 на языке программирования JavaScript.
- Написана функция отрисовки фигуры на основе её координат и матрицы связности вершин
- Написана функция автомасштабирования фигуры.

Итоги

За этот год был полностью покрыт открытый банк заданий ФИПИ по темам:

- Планиметрия — 26 шаблонов принято.
- Вектора — 18 шаблонов (10 принято, 8 на внутреннем рецензировании).
- Стереометрия — 56 шаблонов (7 принято, 49 на внутреннем рецензировании).
- Теория вероятности — 10 шаблонов на внутреннем рецензировании.
- Теория вероятности (повышенной сложности) — 11 шаблонов (1 принят 10 на внутреннем рецензировании).

В ядро проекта добавлены:

- Функции, упрощающие написание шаблонов по темам «Планиметрия» и «Стереометрия».
- Класс многогранников.
- Линейный проектор из $\mathbb{R}^3 \rightarrow \mathbb{R}^2$.
- Сокращён технический долг проекта
- Поставлена цели в будущем добавить в проект класс плоских геометрических фигур и использовать в заданиях по теме «Планиметрия» динамические изображения