Оглавление

[Клиент-серверное приложение (примитивный чат) 2](#_Toc184854233)

[Learning Calculations 3](#_Toc184854234)

[PainTint 4](#_Toc184854235)

[Colors Game 6](#_Toc184854236)

[Переводной Дурак 8](#_Toc184854237)

[Simple 3D Viewer 9](#_Toc184854238)

[Database App 10](#_Toc184854239)

[CRUD веб-приложение 11](#_Toc184854240)

[Dino Game 15](#_Toc184854241)

[Веб-приложение Unity 17](#_Toc184854242)

[Физический движок 2D 18](#_Toc184854243)

[Системный мониторинг с Telegram API 19](#_Toc184854244)

[Калькулятор «Calcul» 20](#_Toc184854245)

# Клиент-серверное приложение (примитивный чат)

**GitHub:** [github.com/r0masaN/Client-Server-Test-App](https://github.com/r0masaN/Client-Server-Test-App)

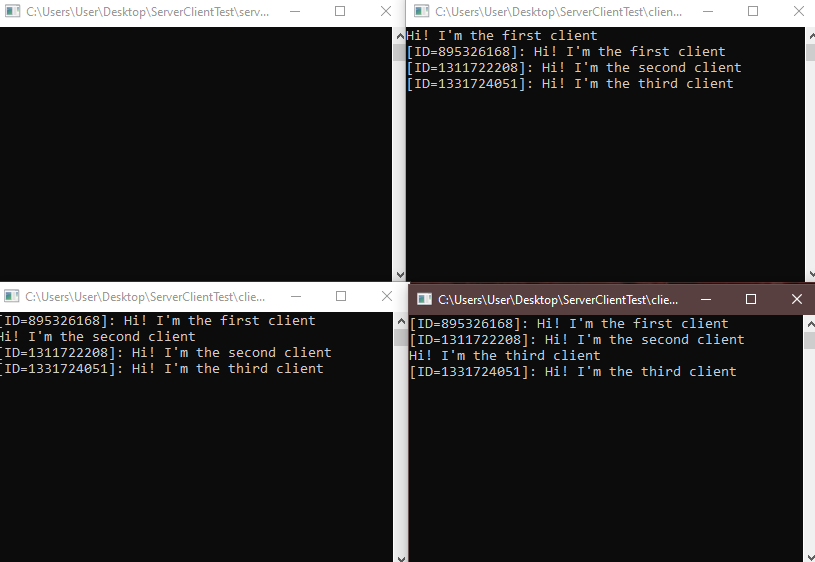
**Описание:**Тестовое приложение, демонстрирующее основы клиент-серверной архитектуры. Состоит из двух программ: серверного и клиентского портативных приложений. Сервер принимает подключение нескольких клиентов. Все сообщения, отправленные одним клиентом, ретранслируются всем остальным в режиме реального времени через сервер.

**Технологии:** Java, Sockets API.

**Функционал:**

* Подключение нескольких клиентов к одному серверу.
* Передача сообщений между клиентами через сервер.
* Реализация на основе потоков для обработки подключений.

**Роль:**  
Full Stack. Разработка серверной и клиентской частей, реализация потоков для одновременной обработки нескольких соединений.



# Learning Calculations

**GitHub:** [github.com/r0masaN/Learning-Calculations-App](https://github.com/r0masaN/Learning-Calculations-App)

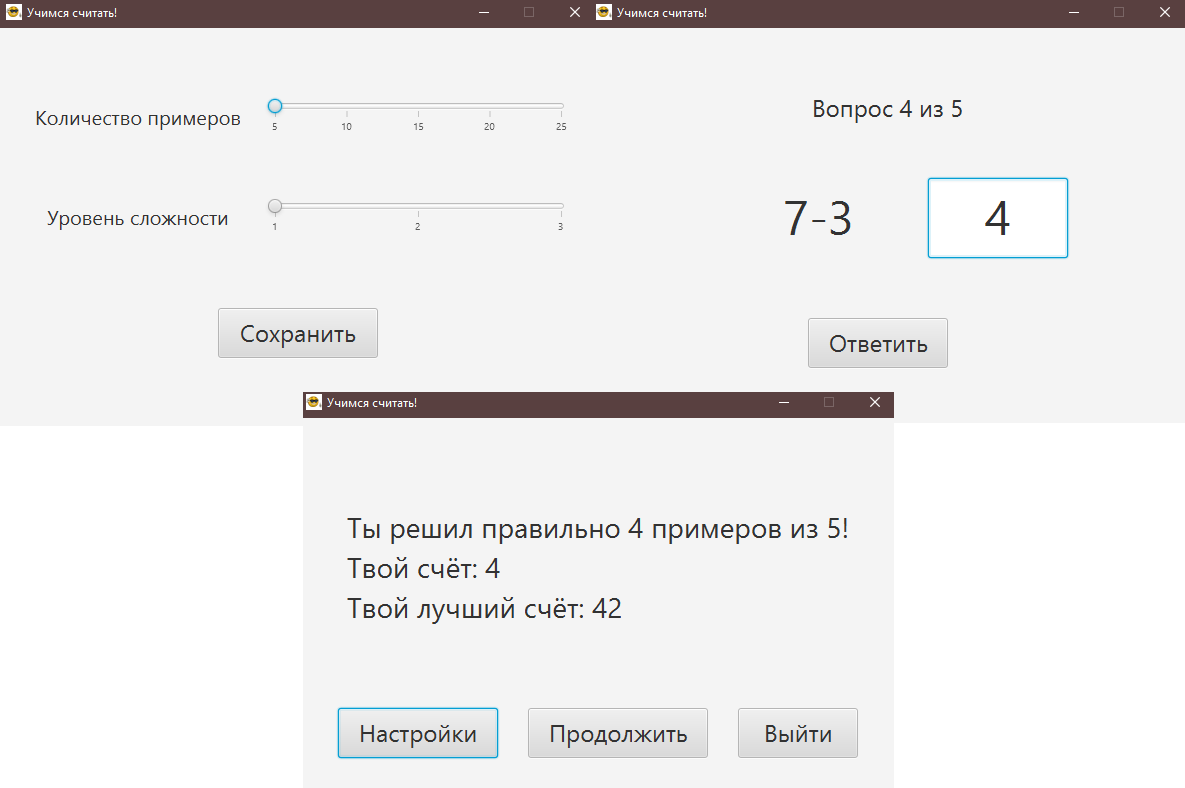
**Описание:**  
Игра разработана для помощи ученикам начальных классов в тренировке математических навыков. Пользователь может выбрать сложность и количество примеров, решать их со знаками «+» и «-». Итоговый результат оценивается программой, учитывая сложность примеров и количество правильно решённых, а лучший счёт сохраняется в текстовый файл.

**Технологии:** Java, JavaFX.

**Функционал:**

* Генерация математических примеров (сложение и вычитание).
* Выбор уровня сложности (от этого зависит диапазон чисел в примерах).
* Подсчёт правильных ответов и очков с учётом сложности.
* Сохранение лучшего результата в .txt файл.

**Роль:**  
Full Stack. Разработка интерфейса, алгоритма генерации примеров и логики подсчёта результатов.



# PainTint

**GitHub:** [github.com/r0masaN/PainTint-App](https://github.com/r0masaN/PainTint-App)

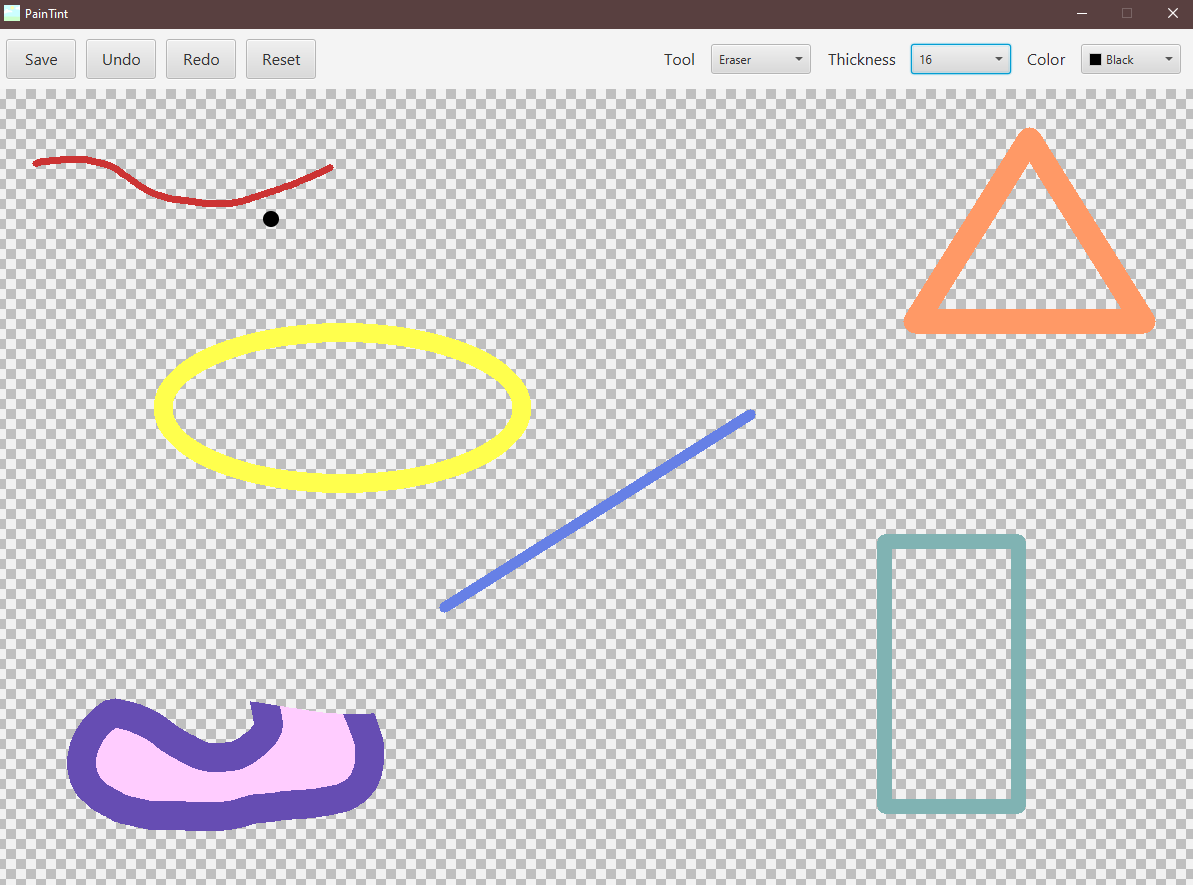
**Описание:**  
Приложение для рисования, созданное как упрощённый аналог Paint. Оно предоставляет базовые инструменты для редактирования изображений, включая различные инструменты для рисования, действия Undo и Redo, настройку цвета и толщины линий, а также сохранение результата в .png файл.

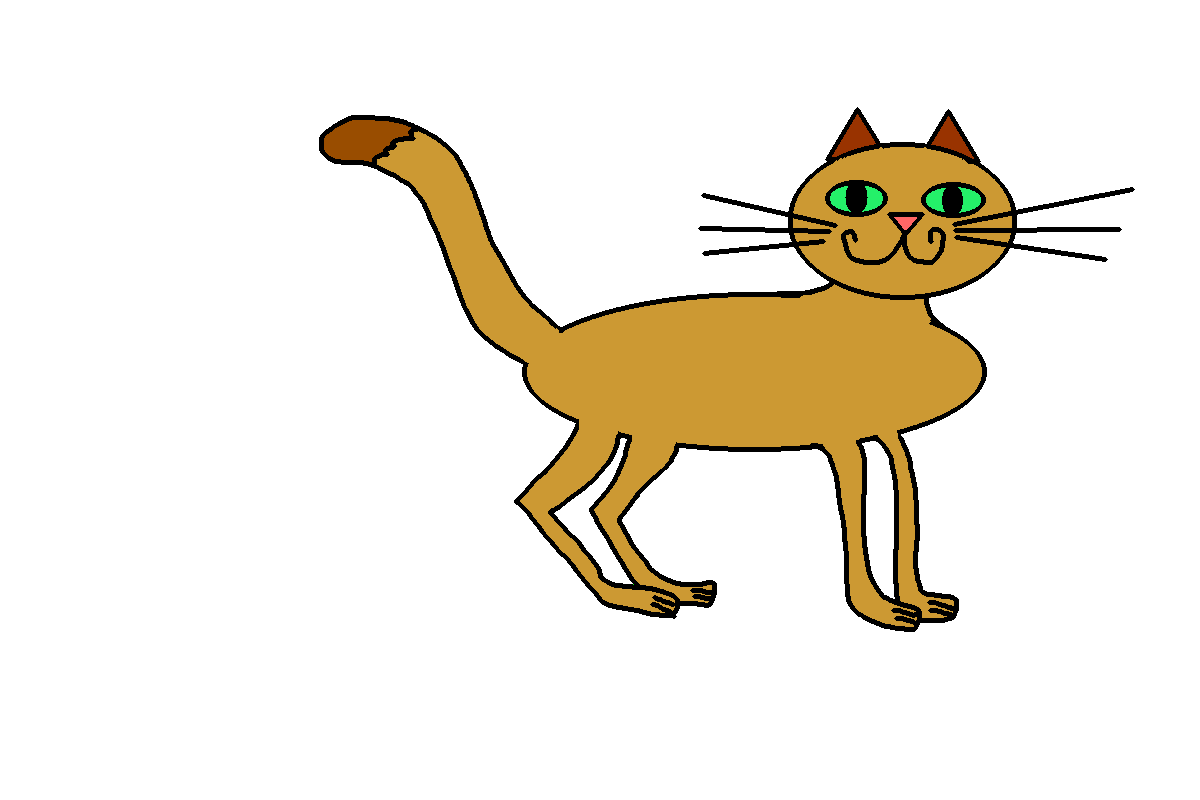
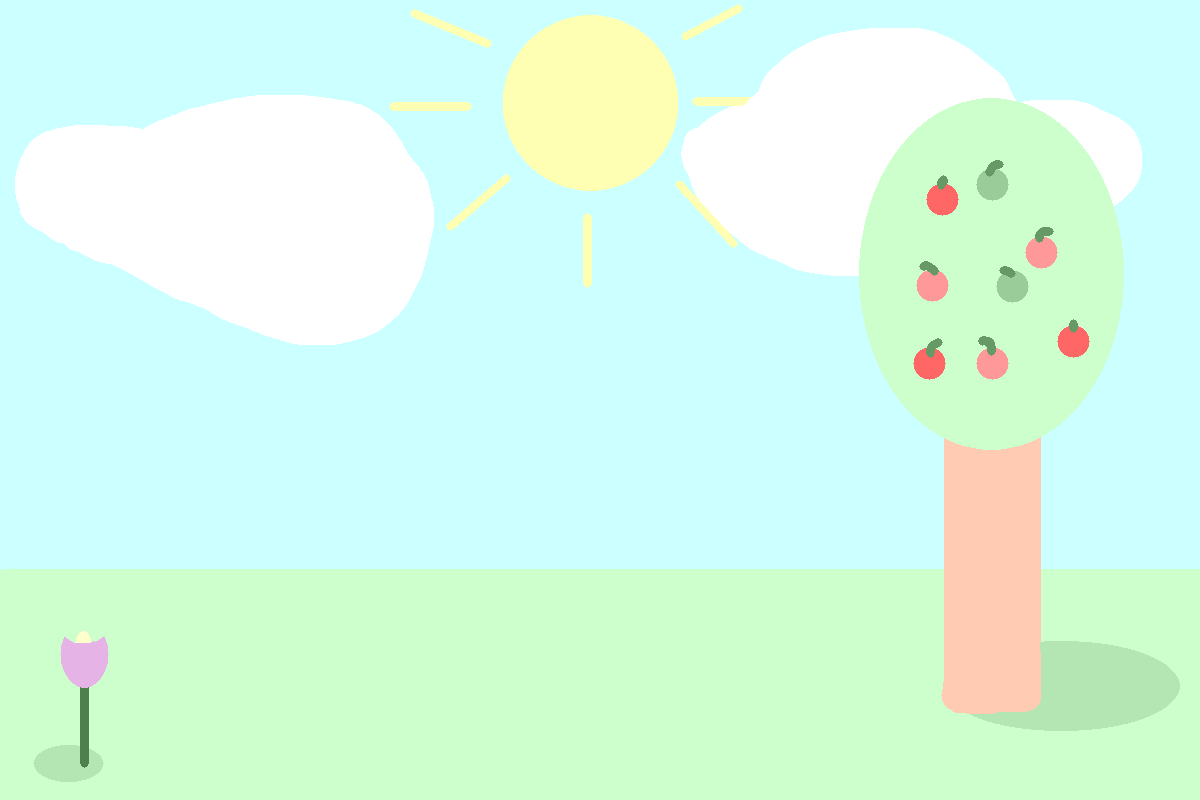
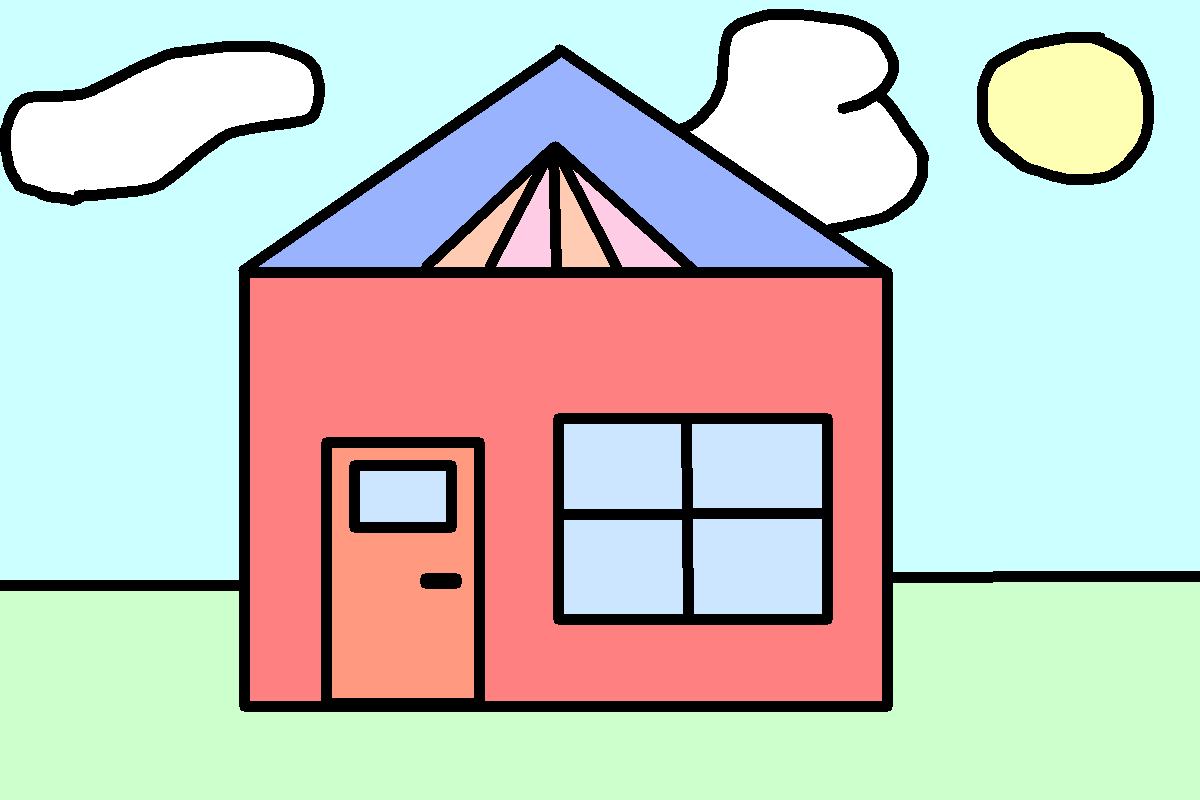
**Технологии:** Java, JavaFX.

**Функционал:**

* Различные инструменты рисования: ластик, кисть, фигуры (прямоугольник, окружность, треугольник), прямая линия, заливка цветом.
* Поддержка отмены и повтора действий.
* Полная очистка холста.
* Настройка толщины линии и её цвета (RGB).
* Сохранение рисунка в .png файл.

**Роль:**  
Full Stack. Разработка интерфейса, логики холста и инструментов рисования, системы сохранения изображений.





# Colors Game

**GitHub:** [github.com/r0masaN/Colors-Game](https://github.com/r0masaN/Colors-Game)

**Описание:**  
Игра-головоломка, в которой игроку нужно закрасить всё поле одним цветом, выбирая цвета из доступных. Поле представляет собой сетку квадратов разного цвета. Изменение цвета начинается с верхнего левого квадрата и затрагивает соседние (по горизонтали и вертикали) квадраты такого же цвета. Игрок побеждает, если сможет закрасить поле одним цветом за указанное количество ходов.

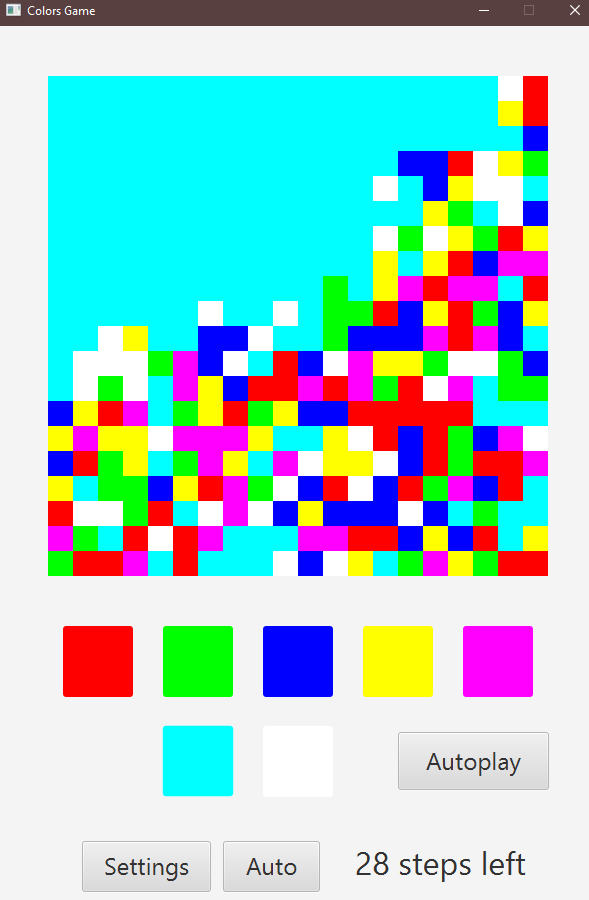
**Технологии:** Java, JavaFX.

**Функционал:**

* Настройка игры: количество цветов (от 3 до 7) и размеры поля (от 5x5 до 20x20).
* Кнопки для выбора цветов, окрашенные в соответствующие цвета.
* Автоматический подсчёт допустимого количества ходов в зависимости от размера поля и количества цветов.

**Роль:**  
Full Stack. Разработка интерфейса, логики игры и механики изменения цветов.





# Переводной Дурак

**GitHub:** [github.com/r0masaN/Transferable-Fool-Game](https://github.com/r0masaN/Transferable-Fool-Game)

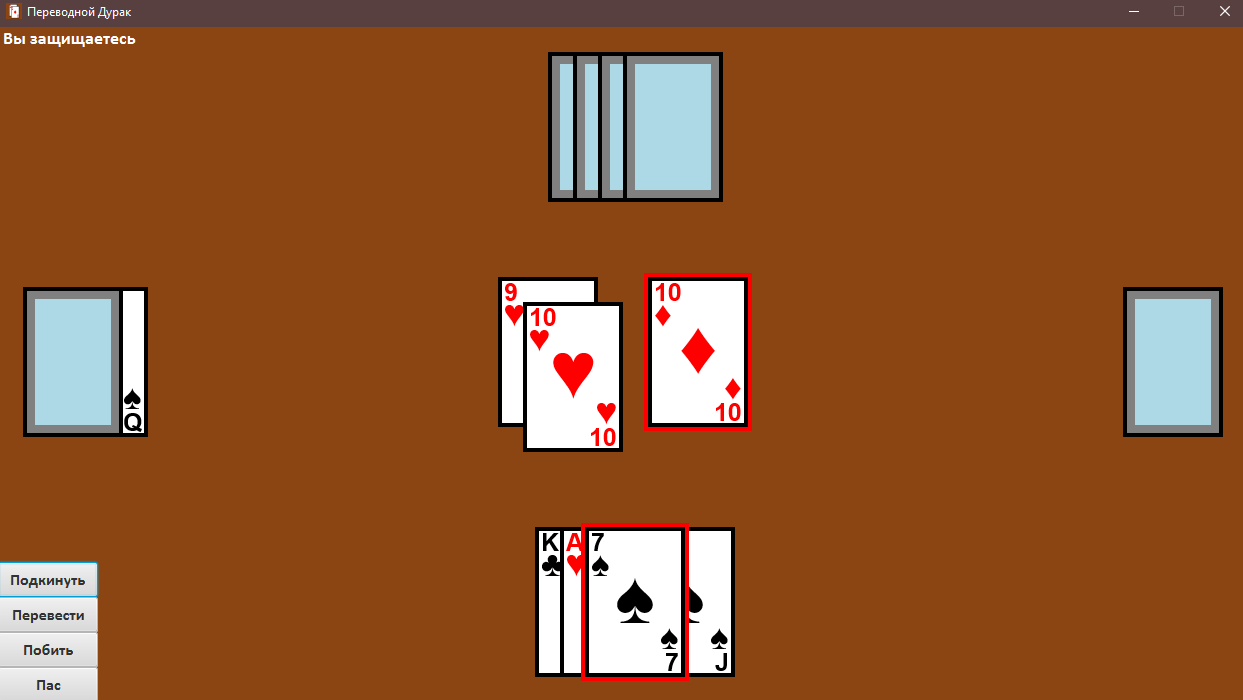
**Описание:**  
Классическая карточная игра с возможностью игры против бота. Реализованы все базовые правила переводного дурака, включая подкидывание, перевод и отбивание карт. Бот имеет алгоритмы, определяющие приоритет действий и выбор карт, делая его более интересным соперником и создавая реалистичный игровой процесс.

**Технологии:** Java, JavaFX.

**Функционал:**

* Управление картами с помощью мыши (клик для выделения/снятия выделения).
* Игровое поле включает колоду, козырь, карты на столе (кон), карты игрока, карты бота (рубашкой вверх) и биту.
* Кнопки действий: "Подкинуть", "Перевести", "Побить", "Пас".
* Бот с базовым ИИ, принимающим решения на основе текущей игровой ситуации.

**Роль:**  
Full Stack. Разработка логики игры, алгоритмов для бота и пользовательского интерфейса.



# Simple 3D Viewer

**GitHub:** [github.com/r0masaN/Simple-3D-Viewer](https://github.com/r0masaN/Simple-3D-Viewer)

**Описание:**  
Учебное портативное приложение, демонстрирующее основы работы с 3D-графикой. Программа позволяет загружать и отображать модели в формате .obj и их текстуры в формате .jpg, а также взаимодействовать с ними через простое управление. Проект реализован с целью изучения работы с 3D графикой, матрицами переходов и Java.

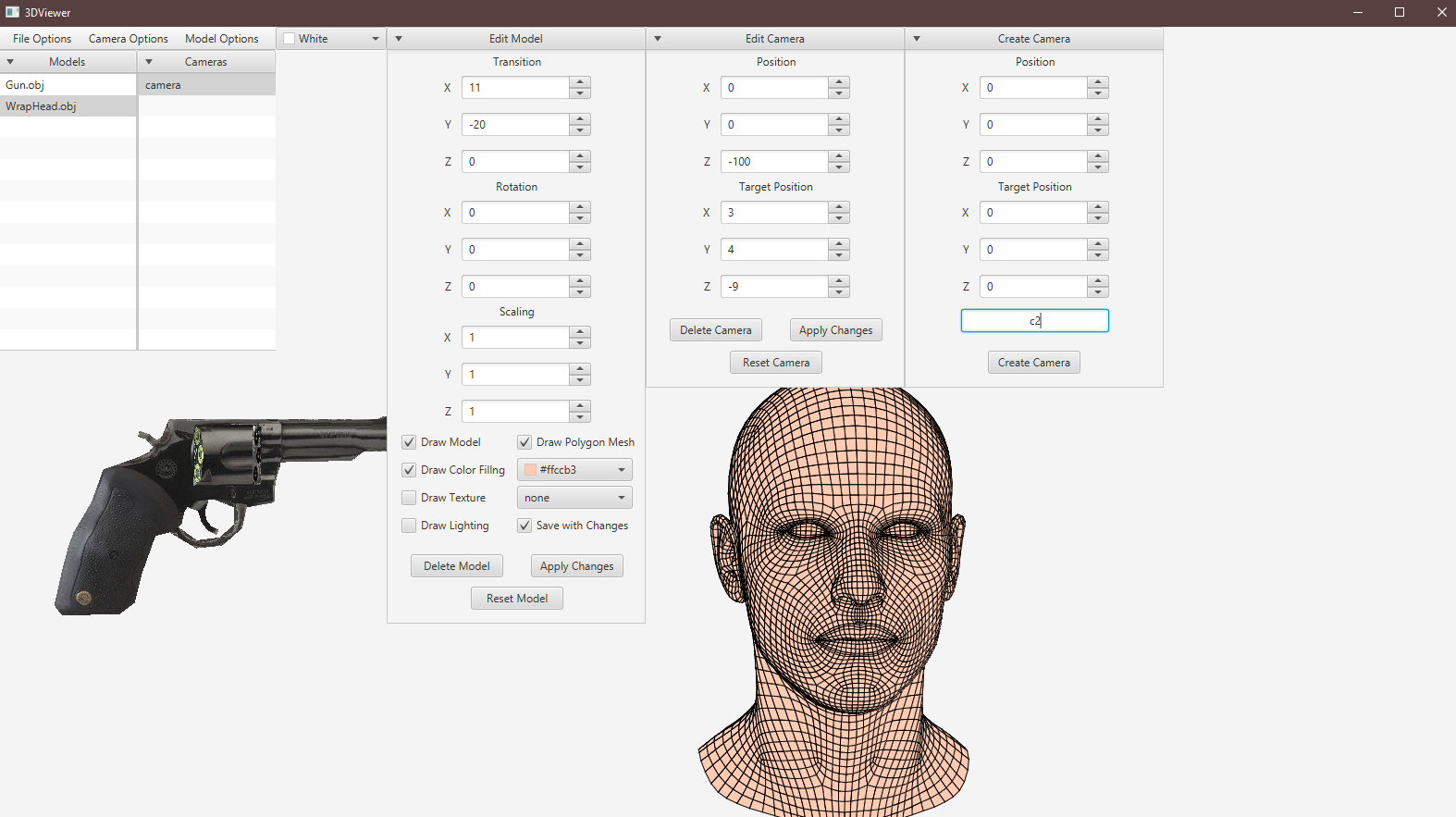
**Технологии:** Java, JavaFX.

**Функционал:**

* Загрузка моделей в формате .obj.
* Визуализация моделей с использованием Canvas (JavaFX).
* Управление камерой и объектами сцены: вращение, масштабирование и перемещение.
* Обработка пользовательского ввода через мышь и клавиатуру.

**Роль в проекте:**

* Реализация загрузчика файлов .obj.
* Написание логики управления объектами и камерой.
* Моделирование и реализация пользовательского интерфейса.
* Программирование контроллеров элементов интерфейса, связывание элементов интерфейса и соответствующих методов в коде.



# Database App

**GitHub:** [github.com/r0masaN/Database-App](https://github.com/r0masaN/Database-App)

**Описание:**  
Портативное приложение для управления локальной базой данных, состоящей из таблиц с взаимосвязанными данными. Позволяет выполнять основные операции CRUD (создание, чтение, обновление, удаление) с записями. Таблицы базы данных реализованы в PostgreSQL с помощью pgAdmin, связь и обработка данных выполняются через JDBC с использованием Java.

**Технологии:** Java, JavaFX, PostgreSQL.

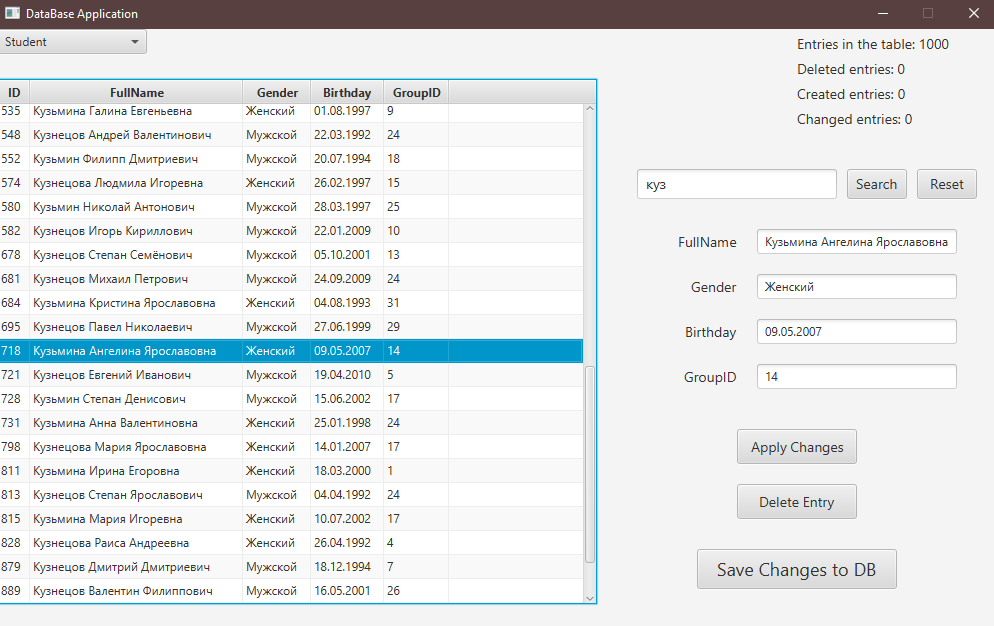
**Функционал:**

* Просмотр записей таблицы через интерактивный TableView (JavaFX).
* Добавление, редактирование и удаление записей.
* Возможность выбора любой таблицы из БД.
* Удобный графический интерфейс для работы с данными.
* Возможность поиска по строке и сортировки по полям записей.

**Роль в проекте:**

Full Stack.

* Создание структуры базы данных и написание SQL-запросов.
* Реализация пользовательского интерфейса с использованием JavaFX.
* Настройка связи приложения с базой данных через JDBC.



# CRUD веб-приложение

**GitHub:** [github.com/r0masaN/PurchaseTask](https://github.com/r0masaN/PurchaseTask)

**Описание:**  
Веб-приложение для работы с базой данных, предоставляющее полный набор операций CRUD (создание, чтение, обновление, удаление). Функционал включает поиск, сортировку записей и удобный интерфейс для управления данными.

**Технологии:**

* **Backend:** Java, Spring Boot.
* **Frontend:** TypeScript, Vue.js, Consta.
* **База данных:** PostgreSQL, DBeaver

**Функционал:**

* Полный набор операций CRUD для работы с данными.
* Возможность поиска записей по ключевым словам.
* Сортировка записей по любым полям.
* REST API для взаимодействия между фронтендом и бэкендом.
* Использование компонентов из библиотеки Consta, обеспечивающих удобный и эстетичный интерфейс.

**Роль:**  
Full Stack.

* Реализация REST API с использованием Spring Boot.
* Интеграция фронтенда и бэкенда.
* Разработка интерфейса на базе Vue.js и библиотеки Consta.



**Case Simulator**

**GitHub:** [github.com/r0masaN/Case-Simulator-Game](https://github.com/r0masaN/Case-Simulator-Game)

**Описание:**  
Портативная игра. Симулятор кейсов из CS:GO с реалистичными шансами выпадения предметов. Поддерживает полный функционал для управления скинами: от открытия кейсов до работы с инвентарём и торговой площадкой.

**Основные особенности:**

**Кейсы:**

* Реальные шансы выпадения дропа.
* Полный набор кейсов из CS:GO с точным содержимым.
* Поиск кейсов по названию и типу (обычный, сувенирный).

**Инвентарь:**

* Возможность сортировать скины по флоту, цене и редкости.
* Фильтрация по параметрам: износ, оружие, редкость, статусы (StatTrak/сувенир).
* Возможность осмотреть скин в увеличенном виде.
* Продажа ненужных скинов.

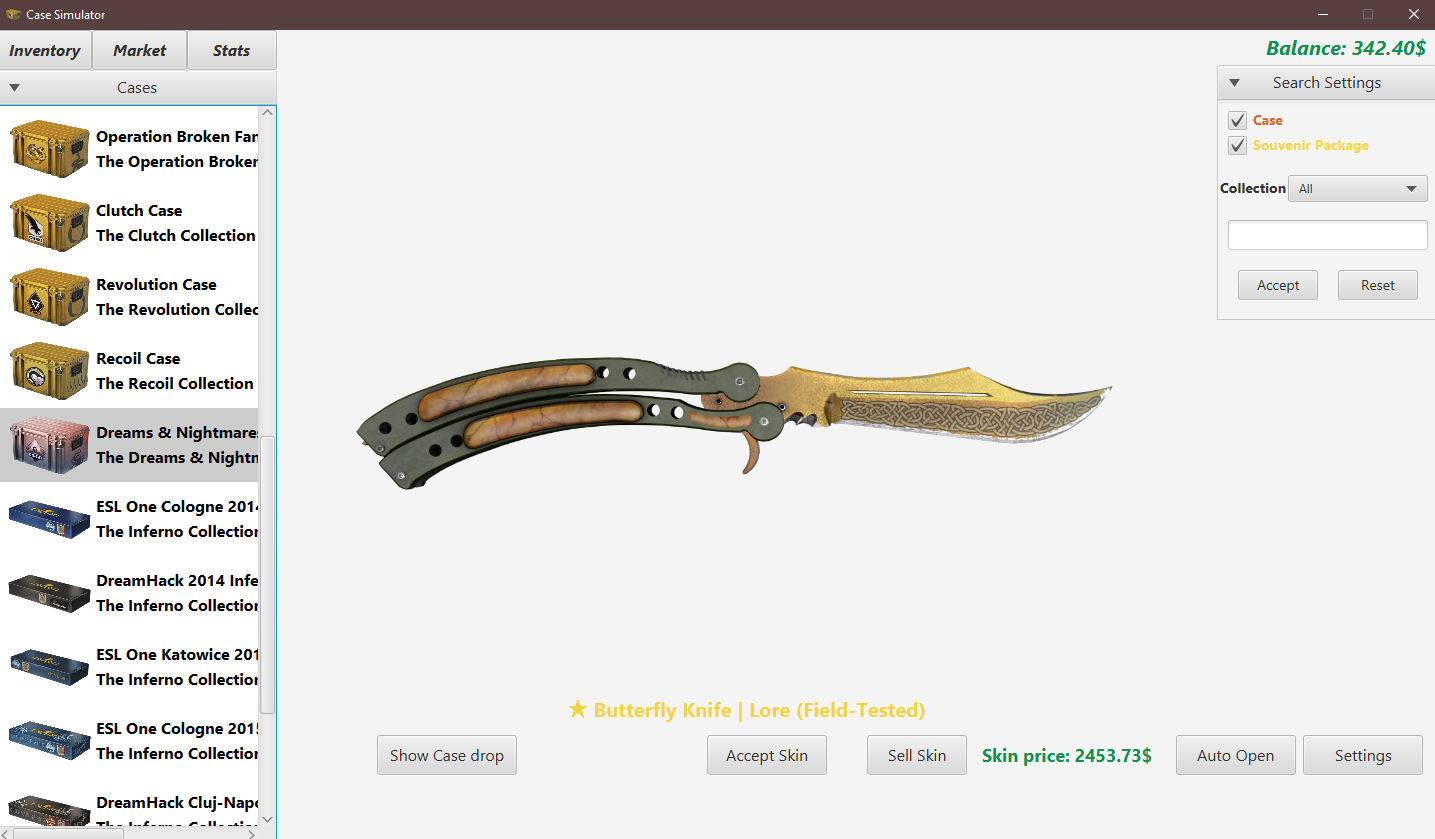
**Торговая площадка:**

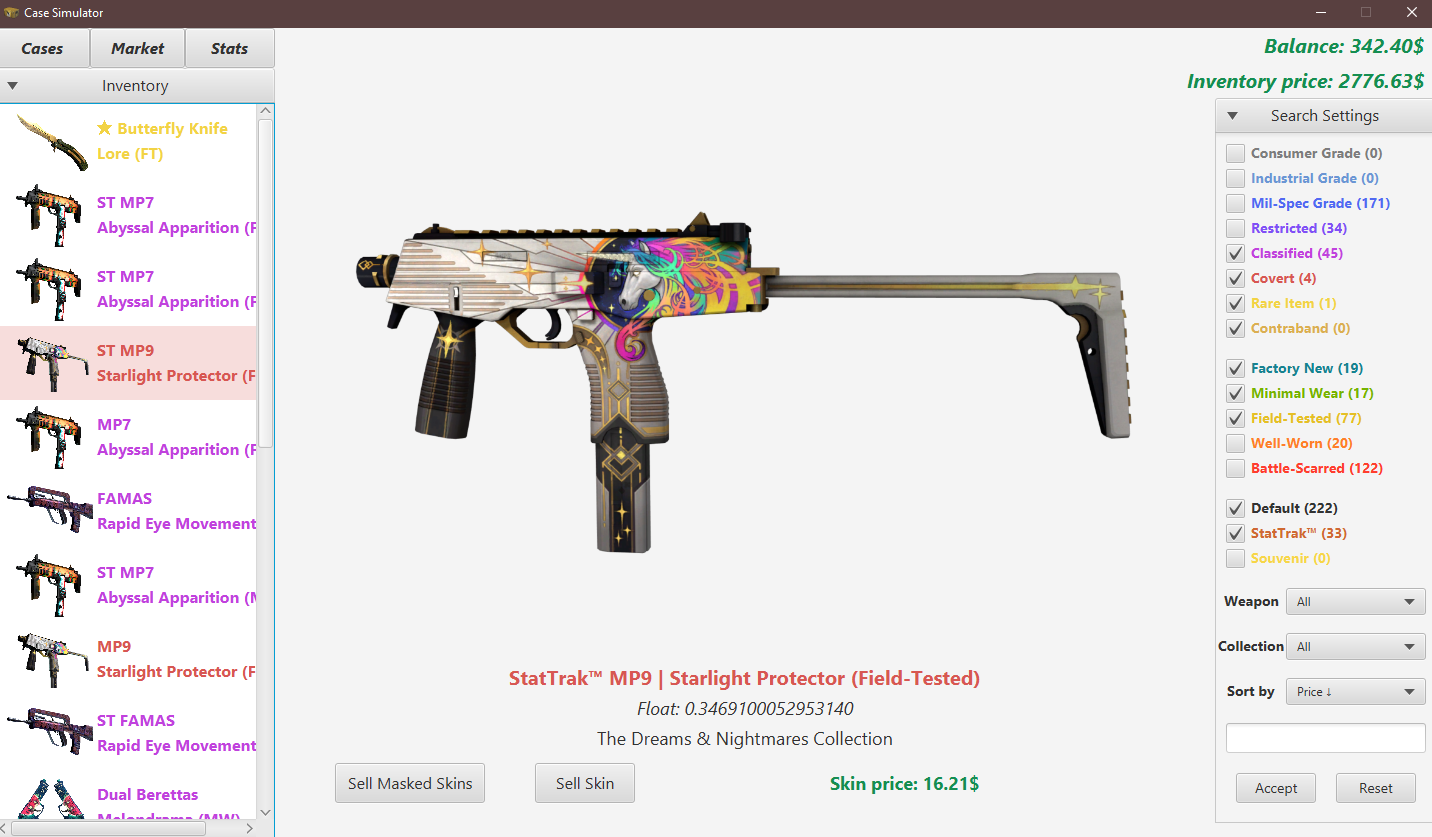
* Покупка скинов.
* Сортировка и фильтрация, аналогичная инвентарю.

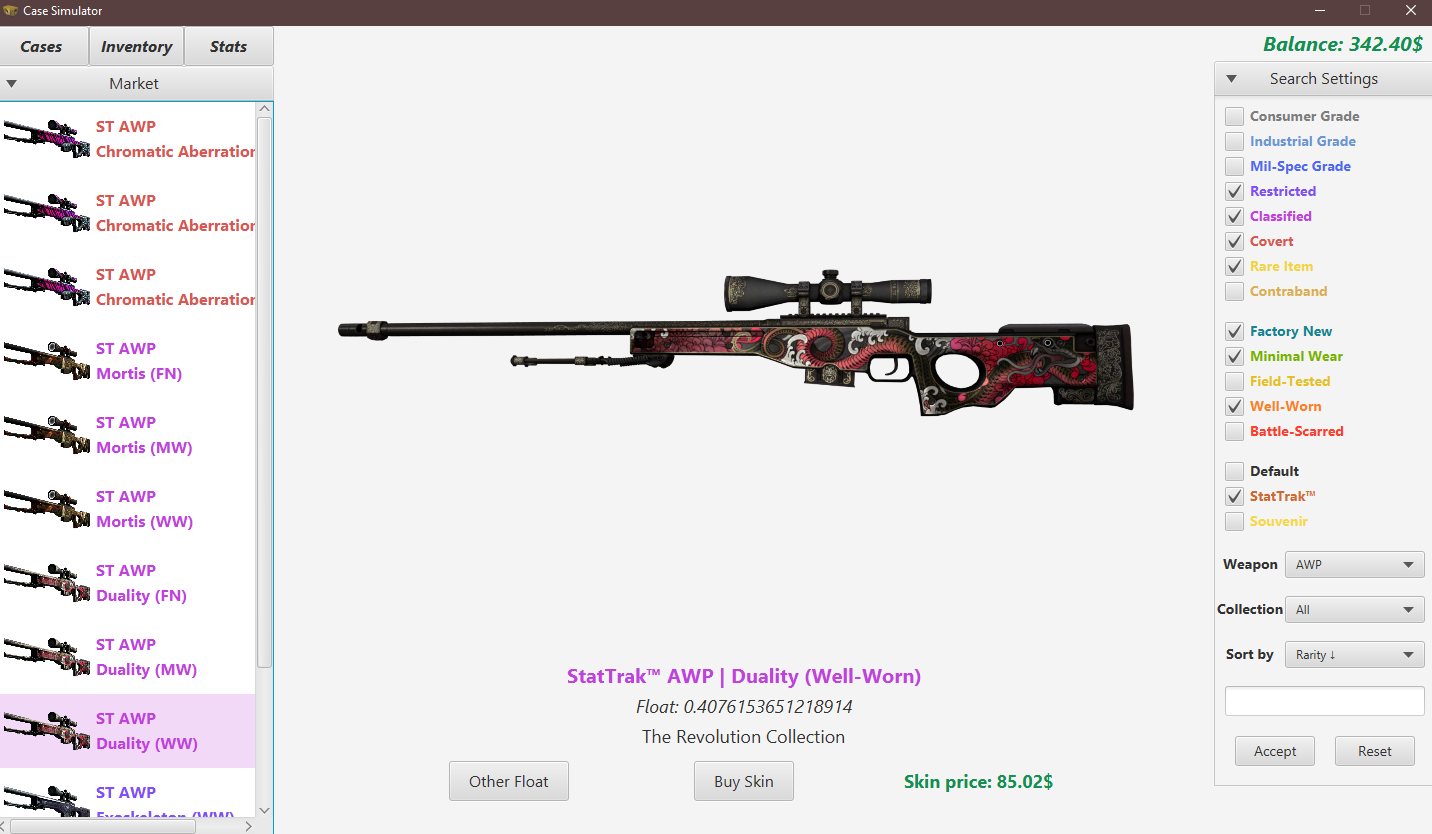
**Графика:**  
Для каждого скина реализованы превью и изображения степеней износа.

**Технологии:** Java, JavaFX.

**Роль:**  
Full Stack. Реализовал логику выпадения скинов, сортировки, фильтрации, а также пользовательский интерфейс.







# Dino Game

**GitHub:** [github.com/r0masaN/Dino-Game](https://github.com/r0masaN/Dino-Game)

**Описание:**  
**Динамическая** аркада в духе классического «гугловского» динозаврика, но с массой улучшений и новых возможностей. Игра погружает игрока в пустынный мир, полный препятствий, бонусов и атмосферных деталей. Главная цель – выжить, преодолеть как можно больше препятствий и набрать максимальное количество очков.

**Технологии:** Java, JavaFX.

**Функционал:**

**Основной геймплей:**

* Дино перемещается по пустыне, перепрыгивая кактусы и собирая бонусы.
* Прогрессирующая сложность: скорость Дино увеличивается с течением времени.
* Система очков: начисление очков за собранные монеты и пройденное расстояние.

**Бонусы и антибонусы:**

* **Монетки:** +100 очков за каждую.
* **Аптечки:** Восстанавливают 1 жизнь (максимум – 3).
* **Щит:** Даёт временную защиту от препятствий, «ломается» при столкновении.
* **Магнит:** Увеличивает радиус сбора монет.
* **Слепота:** Временно снижает видимость.

**Косметические элементы:**

* Пальмы, пирамиды, облака, цветы, змейки, птеродактили.
* Лужи с реалистичными отражениями.
* Цветы на кактусах.

**Управление:**

* Прыгнуть: W, Пробел, Стрелка вверх.
* Упасть: S, Shift, Стрелка вниз.
* Увеличенный прыжок: быстрое чередование «прыжок» – «падение» – «прыжок».
* Пауза: P, Рестарт: R, Отключение HUD: H.

**HUD:**

* Отображение жизней, текущих очков, собранных монет и активных бонусов.
* Плавное уменьшение индикаторов времени действия бонусов.

**Роль:**

**Full Stack.**

* Разработка механики игры, включая логику генерации объектов, бонусов и антибонусов.
* Реализация системы прогрессии сложности.
* Отрисовка визуальных элементов, включая косметические детали и анимации.
* Создание удобного и простого для пользователя интерфейса.
* Деплой финальной версии игры в формате .exe.





# Веб-приложение Unity

**Описание:**  
Веб-приложение для создания и просмотра интерактивных обучающих курсов. Unity объединяет семьи, предоставляя платформу для совместного изучения материалов и взаимодействия с сообществом.

**Технологии:**

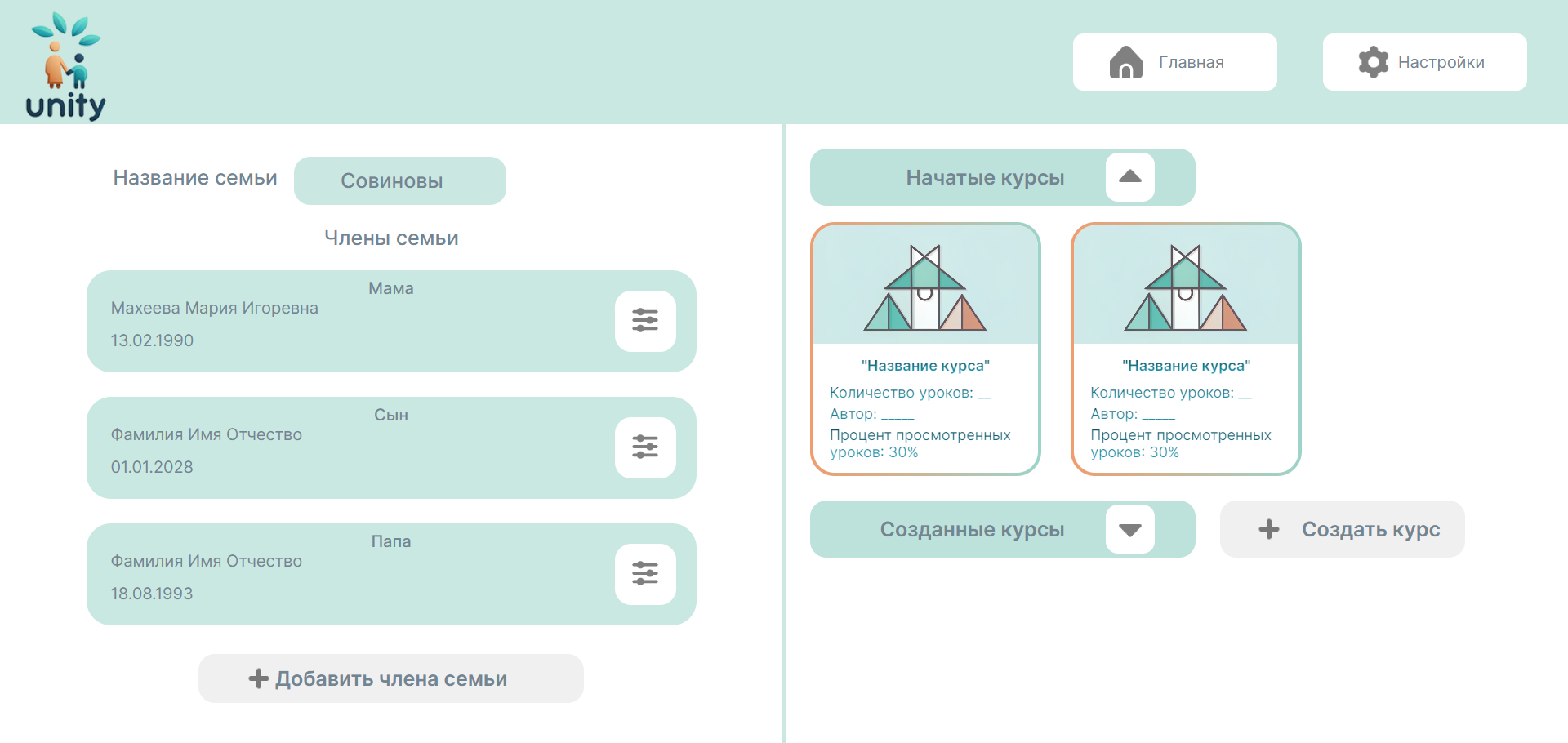
* Сервер: Python, Django, PostgreSQL.
* Клиент: Vue.js, HTML, CSS.

**Функционал:**

* Роли пользователей: гости, авторизованные пользователи, модераторы.
* Поиск курсов по ключевым словам и категориям, сортировка по популярности.
* Управление семейными аккаунтами.
* Создание и редактирование курсов и уроков.
* Интерактивные элементы: комментарии, оценки, модерация контента.

**Роль:**

* Разработка серверной части (Python, Django).
* Реализация REST API для взаимодействия клиентской и серверной частей.
* Обработка запросов для операций создания, чтения, обновления и удаления данных (CRUD).
* Архитектурное проектирование базы данных (PostgreSQL).
* Руководство проектом в роли Teamlead: постановка задач, контроль выполнения, координация работы команды.
* Организация и проведение тестирования функционала.



# Физический движок 2D

**Описание:**  
Простой двумерный физический движок для обработки базовых физических взаимодействий. Проект разработан для использования в личных проектах, таких как Dino Game. Движок поддерживает базовые физические законы, взаимодействия и параметры для прямоугольных объектов.

**Технологии:** Java.

**Функционал:**

**Физические силы:**

* Сила тяжести
* Сила реакции опоры
* Сила трения

**Физические взаимодействия:**

* Упругие и неупругие столкновения
* Закон сохранения импульса

**Объекты и параметры:**

* Прямоугольные хитбоксы
* Параметры: масса, скорость, силы, импульс
* Система флагов: неподвижность, подверженность гравитации, участие в расчёте коллизий

**Расчёты:**

* Изменение скорости и координат
* Обработка столкновений: расчёт векторов изменения скорости
* Учёт соседей для расчёта сил взаимодействия

**Ограничения:**

* Только 2D пространство
* Нет вращений объектов
* Поддержка только прямоугольных хитбоксов.

**Роль:**

Full Stack.

* Проектирование и реализация структуры движка.
* Разработка алгоритмов для расчётов физических взаимодействий и обновления состояний объектов

# Системный мониторинг с Telegram API

**Описание:**  
Портативное приложение для автономного мониторинга системы с автоматической отправкой отчётов в Telegram. Программа работает в фоне, собирая информацию о состоянии системы и отправляя её через API Telegram-бота. Этот проект был разработан исключительно в исследовательских целях и использовался только с разрешения владельцев компьютеров.

**Технологии:** Java, Telegram API.

**Функционал:**

* Автономный запуск: Автоматически запускается при включении компьютера.
* Сбор данных: Использование командной строки для сбора информации о состоянии системы (процессор, оперативная память, запущенные процессы).
* Формирование отчёта: Генерация текстового отчёта с информацией о системе.
* Передача данных: Отправка отчёта через Telegram API напрямую в личные сообщения через бота.
* Периодичность: Формирование и отправка отчёта выполняются каждые 10 минут.

**Роль:**

Full Stack.

* Разработка архитектуры и всей функциональности приложения.
* Реализация взаимодействия с Telegram API.
* Написание команд для сбора системной информации через cmd.
* Создание логики для автоматического запуска приложения и его работы в фоновом режиме.

# Калькулятор «Calcul»

**GitHub:** [github.com/r0masaN/Calcul](https://github.com/r0masaN/Calcul)

**Описание:**  
Приложение для выполнения арифметических операций с поддержкой длинных выражений и скобок. Работает на основе оригинального алгоритма рекурсивного разбиения выражений, разработанного самостоятельно.

**Технологии:** Java, JavaFX, CSS.

**Функционал:**

* Поддержка сложения, вычитания, умножения, деления, возведения в степень и группировки выражений в скобки.
* Обработка многоуровневых выражений.
* Встроенный алгоритм для приоритетного выполнения операций.
* Удобный минималистичный пользовательский интерфейс.

**Роль:**

Full Stack.

* Разработка и оптимизация рекурсивного алгоритма для вычислений.
* Реализация пользовательского интерфейса на JavaFX.

