Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий

Кафедра математики и цифровых технологий

**ОТЧЕТ**

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ»

**Проектирование и описание архитектуры приложения**

ОГУ 38.04.08. 4015 О

Руководитель

Старший преподаватель

\_\_\_\_\_\_\_\_\_Минина И. В.

«\_\_» \_\_\_\_\_\_\_\_\_\_\_2025г.

Студенты группы 23ПМИ(б)ППКС

\_\_\_\_\_\_\_\_\_\_Никитин Илья

\_\_\_\_\_\_\_\_\_\_Керенцев Михаил

\_\_\_\_\_\_\_\_\_\_Носов Владислав

\_\_\_\_\_\_\_\_\_\_Козаченко Олег

«\_\_» \_\_\_\_\_\_\_\_\_\_\_2025г.

Оренбург 2025

**Содержание**

[1 Тип приложения 3](#_Toc9991)

[2 Архитектурный стиль 3](#_Toc6899)

[3 Прототип архитектуры 3](#_Toc21143)

[4 Выбранные технологии 4](#_Toc2467)

[5 Параметры качества 4](#_Toc16904)

[6 Сквозная функциональность 4](#_Toc23018)

**Проектирование и описание архитектуры приложения**

# **1 Тип приложения**

Разрабатываемый проект представляет собой десктопное приложение для игры в шахматы в 3D формате с улучшенной визуализацией и новыми возможностями взаимодействия.

Ключевые характеристики приложения:

1. Платформа: Windows.
2. 3D-графика – объемная доска, фигуры, анимации.
3. Локальная игра – два игрока на одном ПК.
4. Категория: развлекательное приложение.
5. Целевая аудитория: студенты, преподаватели, киберспортсмены, шахматисты-любители, новички в шахматах.

# **2 Архитектурный стиль**

Архитектура приложения будет монолитной – весь функционал шахмат (интерфейс, логика) будет реализован в рамках единого приложения без разделения на микросервисы или внешние сервисы.

Особенности монолитного подхода в проекте:

1. Единая кодовая база – все компоненты (логика передвижения фигур, интерфейс поля, правила шахмат) собраны в одном приложении.
2. Локальная работа – приложение не требует подключения к серверам или облачным сервисам (все операции выполняются на клиенте).
3. Простота развертывания - один исполняемый файл(\*.exe для Windows).

# **3 Прототип архитектуры**

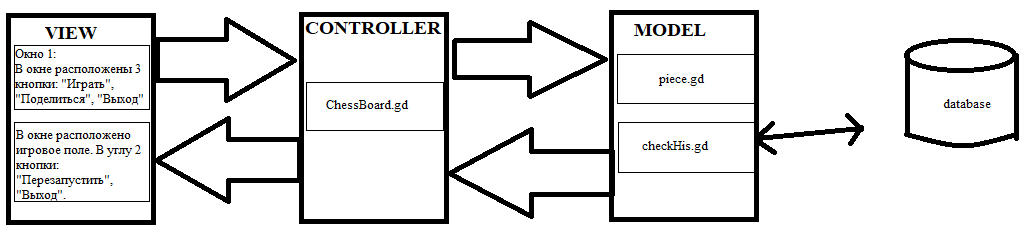


Рисунок 1 – Схема архитектуры приложения в стиле MVC

# **4 Выбранные технологии**

Написание программного кода происходит на встроенном высокоуровневом языке программирования GDScript с использованием мощного и современного оптимизированного движка Godot.

1. GDScript — высокоуровневый, [объектно-ориентированный](https://en.wikipedia.org/wiki/Object-oriented_programming), [императивный](https://en.wikipedia.org/wiki/Imperative_programming) язык программирования с [последовательной типизацией](https://en.wikipedia.org/wiki/Gradual_typing), созданный для Godot. Его цель — оптимизировать и тесно интегрировать с Godot Engine, обеспечивая большую гибкость при создании контента.
2. Godot — это универсальный 2D и 3D игровой движок, спроектированный для поддержки всех видов проектов. Вы можете использовать его для создания игр или приложений, которые сможете затем выпускать на настольных или мобильных платформах, а также web.
3. Blender — это профессиональный инструмент для создания и скульптинга 2D и 3D моделей с возможностью экспорта с различными расширениями. Это позволяет интегрировать созданные модели в свои проекты.

# **5 Параметры качества**

1. Производительность.

Приложение должно обеспечивать пользователю плавную и бесперебойную работу на оборудовании, соответствующем минимальным заявленным требованиям. Осуществлять все заявленные в техническом задании, функции.

2. Надежность.

Минимизация критических ошибок, таких как попадание фигуры на уже занятую ячейку поля, выход фигуры за пределы доски. Вывод об ошибке, в случае некорректного хода пользователя.

3. Удобство использования.

Интерфейс должен быть интуитивно понятным, чтобы пользователи с разными уровнями подготовки могли быстро освоить основные функции.

# **6 Сквозная функциональность**

**Обработка ошибок.**

Проверка корректности ходов(например, нельзя поставить короля под шах)

Защита от «падения» приложения (try-catch для критических операций).

**Валидация данных.**

Проверка корректности: валидация правил шахмат(например, проверка координат хода) с предоставлением понятных сообщений об ошибках.

**Кэширование.**

Хранение результатов провёденной партии (сохранение истории ходов пользователей).