

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»  
Факультет информационных технологий  
Кафедра программной инженерии

Утверждаю  
Заведующий кафедрой ПИ  
\_\_\_\_\_  
подпись                      инициалы и фамилия  
“    ” \_\_\_\_\_ 2023г.

**ЗАДАНИЕ**  
**к курсовому проектированию**  
**по дисциплине "Объектно-ориентированное**  
**программирование"**

Специальность: 1-40 01 01 Программное обеспечение  
информационных технологий

Группа: 4

Студент: Точило О.В.

**Тема: Симулятор сражений Второй Мировой войны для компьютерной игры "День победы"**

**1. Срок сдачи студентом законченной работы: "20 мая 2023 г."**

**2. Исходные данные к проекту:**

**2.1. Функционально ПП поддерживает:**

- Функции администратора:
  - Изменение характеристик рот и батальонов
- Функции пользователя:
  - Создание собственных шаблонов дивизий из батальонов и рот поддержки
  - Сохранение шаблонов в базе данных
  - Создание набора дивизий из готовых или собственных шаблонов для обеих сторон
  - Добавление, удаление и изменение модификаторов и генералов, влияющих на эффективность дивизий
  - Изменение типа местности, на котором будет проходить сражение

**2.2.** При выполнении курсового проекта необходимо использовать принципы проектирования ООП. Приложение разрабатывается под ОС Windows и представляет собой настольное приложение. Отображение, бизнес логика должны быть максимально независимы друг от друга для возможности расширения. Диаграммы вариантов использования, классов реализации задачи, взаимодействия разработать на основе UML. Язык разработки проекта – С#. Управление программой должно быть интуитивно понятным и удобным. При разработке использовать несколько наиболее подходящих шаблонов проектирования ПО.

**3. Содержание расчетно-пояснительной записки**

(перечень вопросов подлежащих разработке)

- Введение
- Постановка задачи и обзор литературы (алгоритмы решения, обзор прототипов, актуальность задачи)

- Проектирование архитектуры проекта (структура модулей, классов).
- Разработка функциональной модели и модели данных программного средства (выполняемые функции)
- Тестирование
- Заключение
- Список используемых источников
- Приложения

#### 4. Форма представления выполненной курсовой работы:

- Теоретическая часть курсового проекта должны быть представлены в формате docx. Оформление записки должно быть согласно выданным правилам.
- Листинги программы представляются частично в приложении.
- Пояснительную записку, листинги, проект (инсталляцию проекта) необходимо загрузить на диск, указанный преподавателем.

#### **Календарный план**

№ п/п	Наименование этапов курсового проекта	Срок выполнения этапов проекта	Примечание
1	Введение	19.02.2023	
2	Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования	12.03.2023	
3	Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных)	26.03.2023	
4	Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса	2.04.2023	
5	Кодирование программного средства	23.04.2023	
6	Тестирования и отладка программного средства	30.04.2023	
7	Оформление пояснительной записки	7.05.2023	
9	Сдача проекта	20.05.2023	

5. Дата выдачи задания 14.02.2023

Руководитель \_\_\_\_\_  
(подпись)

*Ю.А. Чистякова*

Задание принял к исполнению \_\_\_\_\_  
(дата и подпись студента)

*О.В. Точило*

## Содержание

Введение .....	3
1 Аналитический обзор известных решений .....	4
2 Анализ требований к программному средству и разработка функциональных требований .....	8
2.1 Описание функциональности программного средства.....	8
2.2 Спецификация функциональных требований.....	9
3 Проектирование программного средства .....	10
3.1 Архитектура программного средства .....	10
3.2 Проектирование базы данных .....	10
3.3 Взаимодействие программного средства с базой данных.....	13
4 Реализация программного средства.....	14
4.1 Основные классы программного средства.....	14
4.2 Реализация паттерна MVVM.....	14
4.3 Реализация сражения.....	17
5 Тестирование .....	19
5.1 Тестирование изменения характеристик рот и батальонов.....	19
5.2 Тестирование создания пользовательских дивизий и сохранения их в базе данных.....	20
5.3 Тестирование прикрепления и открепления командующих .....	22
5.4 Тестирование добавления и удаления модификаторов .....	24
5.5 Тестирование изменения типа местности .....	25
6 Руководство пользователя .....	27
Заключение .....	32
Список использованных источников.....	33
Приложение А .....	34
Приложение Б.....	35
Приложение В .....	47

## Введение

Видеоигра «День Победы» – это стратегическая игра, предоставляющая чрезвычайно сильную вариативность игрового процесса. Эффективность игрока зависит от множества параметров, что затрудняет планирование своих действий. С другой стороны, игра также предоставляет возможность играть против других людей, и высокая степень соревновательности таких матчей требует наличия заранее заготовленной стратегии и шаблонов дивизий, наиболее эффективных в конкретных условиях. Сама игра не предоставляет возможности удобно испытывать различные конфигурации дивизий: для этого необходимо либо тратить очень много времени, готовя эти дивизии и создавая благоприятные условия для испытаний, либо пользоваться консольными командами, что также не вполне удобно. В таких условиях существование инструмента, позволяющего настраивать количественный и качественный состав противников за несколько нажатий левой кнопкой мыши, существенно сократило бы временные затраты на испытание своих конфигураций дивизий.

Целью данного курсового проекта является разработка приложения, позволяющего удобно создавать и испытывать наборы дивизий для комфортной игры в «День Победы».

Данная пояснительная записка будет включать в себя в том числе следующие разделы:

- Аналитический обзор литературы
- Анализ требований к программному продукту
- Проектирование программного средства
- Реализация программного средства
- Тестирование программного продукта и анализ полученных результатов
- Методика использования программного средства

## 1 Аналитический обзор известных решений

Основой данного курсового проекта является видеоигра «День Победы», система сухопутных сражений обладает следующими достоинствами и недостатками:

- гибкость: игра позволяет игрокам настраивать состав своей армии в соответствии с конкретными потребностями. Они могут создавать специализированные подразделения, такие как танки, пехота или артиллерия, и распределять их по разным фронтам в зависимости от ситуации;
- стратегическая глубина: игра предлагает широкий спектр тактик и стратегий, которые игроки могут использовать, чтобы получить преимущество в бою. Например, игроки могут использовать окружение, чтобы отрезать вражеские поставки и ослабить свои позиции, или использовать комбинацию воздушных и наземных атак, чтобы прорвать вражескую оборону, заманивать вражеские войска в неудобные провинции и отрезать пути снабжения;
- управление снабжением: система снабжения является важнейшим аспектом сухопутных боев. Игроки должны управлять своими линиями снабжения, чтобы гарантировать, что у их войск будет достаточно ресурсов для эффективного противостояния противнику. Они могут распределять ресурсы по разным направлениям в зависимости от важности каждой цели;
- эффекты местности: игра учитывает тип местности при просчёте сухопутных боев. Игроки должны учитывать местность, на которой они сражаются, и соответствующим образом корректировать свою тактику. Например, гористая местность дает преимущества в обороне, в то время как равнинная местность лучше подходит для наступательных операций;
- ограниченная видимость: туман войны может ограничить способность игрока принимать обоснованные решения, например, скрывая количество вражеских дивизий и их состав. Игроки без созданного и улучшенного разведывательного агентства практически никогда не имеют полного представления о поле боя, и в результате они могут совершать ошибки, которые будут стоить им победы;
- ограничения искусственного интеллекта: искусственный интеллект игры иногда может быть предсказуемым, из-за чего тактика сражений может сводиться к использованию одних и тех же шаблонов и манёвров. Кроме того, искусственный интеллект не всегда может принимать оптимальные решения, что может снизить сложность игры.

Окно сухопутного сражения представлено на рисунке 1.1.



Рисунок 1.1 – окно сухопутного сражения

Пример типа местности представлен на рисунке 1.2.

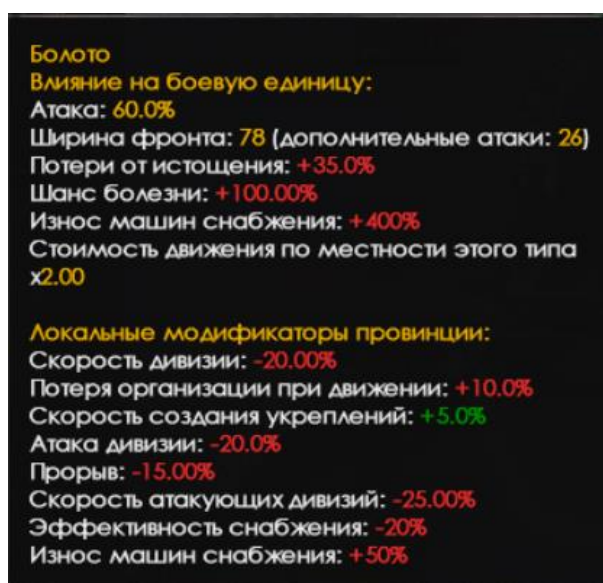


Рисунок 1.2 – пример типа местности

Было принято решение не реализовывать механики снабжения и разведки из-за сложности реализации и малого воздействия на боевой процесс. При этом оставлена гибкость в создании боевых единиц и механика местности.

Ещё одним недостатком «Дня Победы» является следующая особенность конструктора дивизий: при желании добавить несколько одинаковых батальонов игрок вынужден добавлять их по одному, каждый раз проходя одни и те же пункты меню добавления батальона.

Другим аналогичным решением может служить «Stellaris»: похожая на «День Победы» стратегическая игра, действия которой разворачиваются в космосе. Её боевая система обладает схожими достоинствами и недостатками:

- гибкость: игроки могут создавать различные конфигурации кораблей и флотов, назначать и откреплять адмиралов;
- стратегическая глубина: игроки могут совершать различные манёвры, чтобы запутать противника и связать его боем или, наоборот, избежать его;
- модернизация технологий: игра позволяет игрокам исследовать новые технологии, которые улучшают мощь их флота. Они могут разблокировать новое вооружение, двигатели, щиты, системы наведения, бортовые компьютеры и модификаторы, которые дают их кораблям преимущество в бою;
- ограничение видимости;
- ограничения искусственного интеллекта.

Окно сражения представлено на рисунке 1.3.



Рисунок 1.3 – окно сражения

Преимущество интерфейса в «Stellaris» заключается в том, что однотипные боевые единицы можно добавлять одним щелчком мыши. Недостатками же являются разделение боевых единиц (кораблей) по категориям, как в «Дне Победы», что может привести к затруднению поиска нужного корабля среди десятков других кораблей в одном списке, однако обычно проекты кораблей унифицированы и данная проблема не наблюдается, и отсутствие сводных характеристик флота (скорость, атака больших, средних и малых орудий, дальность стрельбы и так далее), что затрудняет создание флота и заставляет полагаться на рассчитываемую игрой характеристику «мощь флота», которая не отражает действительность.

Сравнение окон создания боевых единиц представлено на рисунке 1.4.



Рисунок 1.4 – сравнение окон создания боевых единиц в «День Победы» (сверху) и «Stellaris» (снизу)

Итак, на основе рассмотренных существующих решений было принято решение внести следующие изменения в UI/UX и функционал приложения:

- заменить систему полков с вынужденным выбором одинаковых батальонов по одному на умный счётчик с инкрементацией и декрементацией;
- упразднить системы разведки, снабжения и снаряжения;
- упразднить систему навыков генералов.



## 2 Анализ требований к программному средству и разработка функциональных требований

### 2.1 Описание функциональности программного средства

Функционал программного средства можно разделить на доступный всем пользователям и доступный лишь администраторам. К первой категории относятся:

- Создание дивизий из батальонов
- Создание армий из дивизий
- Создание групп армий из армий
- Изменение названия и иконки дивизий, армий и групп армий
- Назначение и открепление генералов от армий и фельдмаршалов от групп армий
- Сохранение, изменение и удаление дивизий, армий и групп армий в базе данных
- Создание, удаление и изменение боевых модификаторов
- Сражение между wybranнми войсками

Ко второй категории относится изменение характеристик батальонов и рот в базе данных.

Диаграмма вариантов использования представлена на рисунке 2.1.

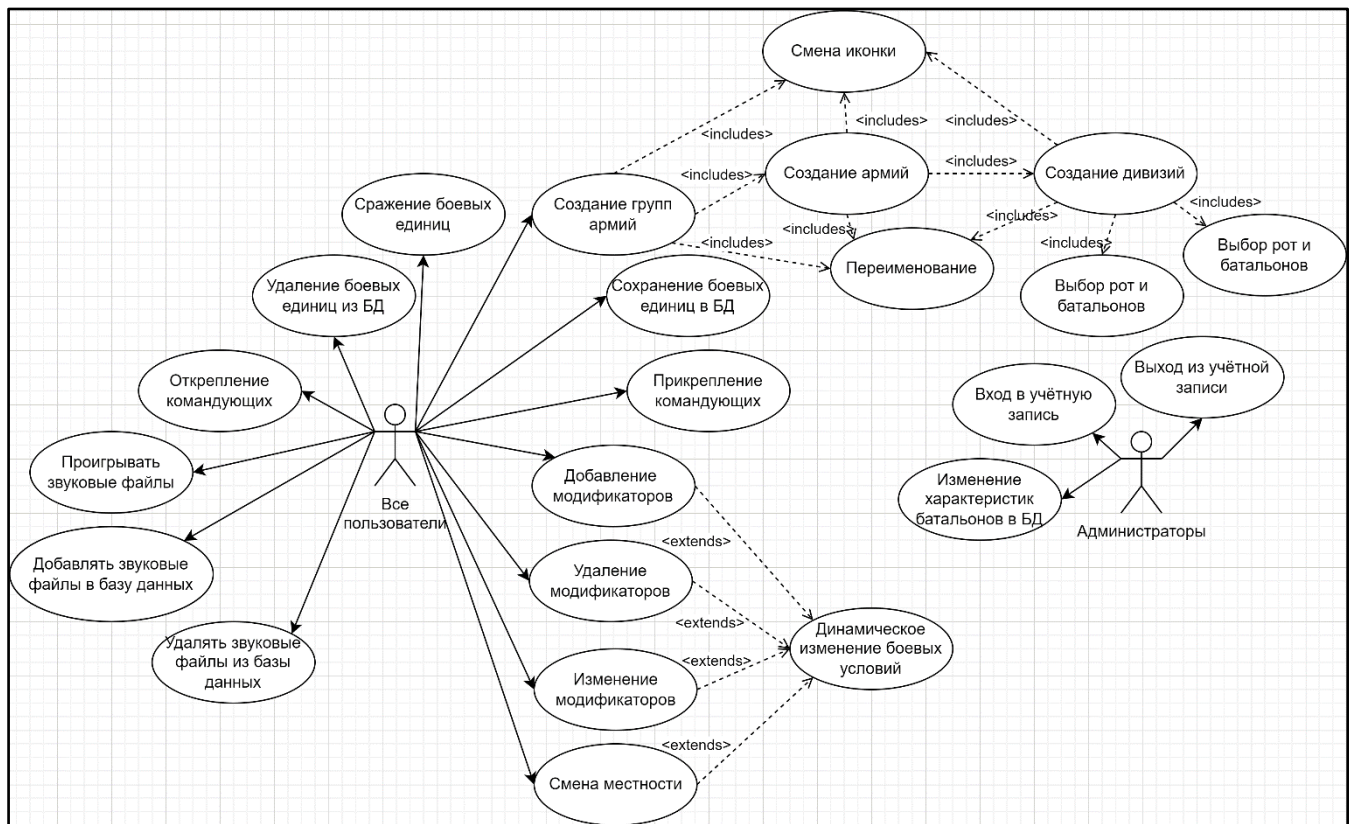


Рисунок 2.1 – Диаграмма вариантов использования

Дополнительно было добавлено требование наличия музыкального проигрывателя с функциями перемешивания и повтора, чтобы пользователь мог разнообразить использование программного средства.

## **2.2 Спецификация функциональных требований**

Для детализации функций проектируемого программного средства необходимо разбить функционал на блоки, каждый из которых будет выполнять свои функции:

- Блок авторизации и регистрации. Данный блок обеспечивает идентификацию пользователей и администраторов системы, а также их регистрацию. После успешной аутентификации пользователь получает доступ к соответствующим функциям программного средства. Администратор может зарегистрироваться, нажав на специальный чекбокс и введя кодовую фразу в появившееся текстовое поле для ввода.

- Блок управления базой данных. Данный блок обеспечивает хранение и управление данными, необходимыми для функционирования программного средства. В базе данных хранятся сведения о батальонах, ротах, модификаторах, генералах, местности и других объектах, используемых в системе. Данный блок содержит функции для получения и изменения данных в базы данных.

- Блок управления армиями. Данный блок позволяет пользователям создавать собственные группы армий из армий, армии из дивизий и шаблоны дивизий из батальонов и рот поддержки, а также сохранять и удалить их из базы данных. Также блок позволяет редактировать названия и иконки боевых единиц и прикреплять и откреплять генералов и фельдмаршалов от боевых единиц.

- Блок управления модификаторами. Данный блок позволяет пользователям создавать, удалять, изменять и применять различные модификаторы, влияющие на боевую эффективность дивизий.

- Блок сражения. Данный блок позволяет начать, остановить и перезапустить сражение между двумя сторонами, просмотреть лог сражения, изменить тип местности.

## 3 Проектирование программного средства

### 3.1 Архитектура программного средства

При проектировании программного средства первоочерёдной задачей является выбор архитектуры приложения, от которой будет зависеть весь процесс конструирования приложения. С учётом особенностей выбранной технологии WPF (таких, как возможность привязки данных к контексту данных) была выбрана архитектура MVVM (Model-View-ViewModel).

Данная архитектура предполагает разделение каждого блока на три слоя: слой представления, отвечающий за визуальное представление интерфейса, слой модели представления, отвечающий за обработку событий, получение и обновление данных в модели и предоставление свойств для привязки к ним элементов управления в представлении, и слой модели, хранящий данные блока. Структура шаблона MVVM представлена на рисунке 3.1.

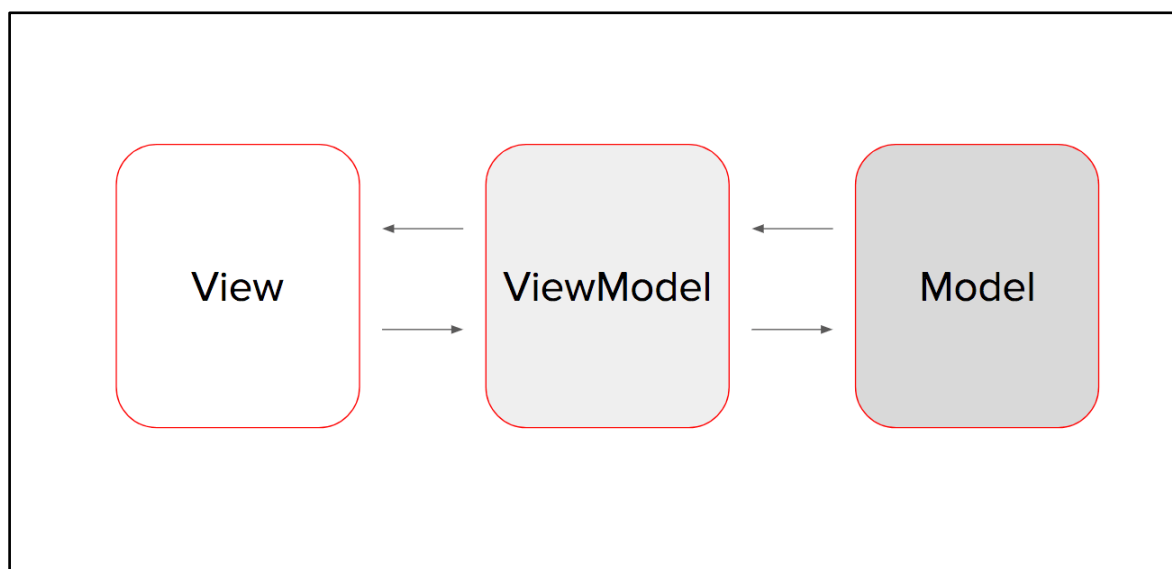


Рисунок 3.1 – Структура шаблона MVVM

Основным компонентом программного средства является главное окно, с которым взаимодействует пользователь. Оно содержит остальные страницы (представления), такие как главное меню, меню выбора стран, меню создания армий, меню выбора модификаторов и меню сражения. В свою очередь эти страницы могут содержать дочерние окна (например, для выбора новой иконки и выбора или создания новой дивизии). У каждой страницы есть своя модель представления, которая может, в зависимости от возможностей, предоставляемых страницей пользователю, иметь или не иметь модель.

### 3.2 Проектирование базы данных

Приложение использует базу данных в Microsoft SQL Server для хранения данных о боевых единицах, их характеристиках, генералах и модификаторах. Для

хранения необходимых данных были определены следующие сущности: группа армий, армия, дивизия, батальон и рота поддержки (хранятся в одной таблице из-за того, что имеют одинаковый набор параметров), генерал (генералы и фельдмаршалы хранятся в одной таблице из-за того, что отличаются только тем, к каким сущностям их можно прикреплять), тип местности. Также были добавлены таблицы для хранения содержимого и путей к иконкам групп армий, армий и дивизий и модификаторов атаки и защиты для различных типов местности каждого батальона. Логическая диаграмма базы данных представлена на рисунке 3.3. SQL-сценарий создания базы данных представлен в Приложении Б.

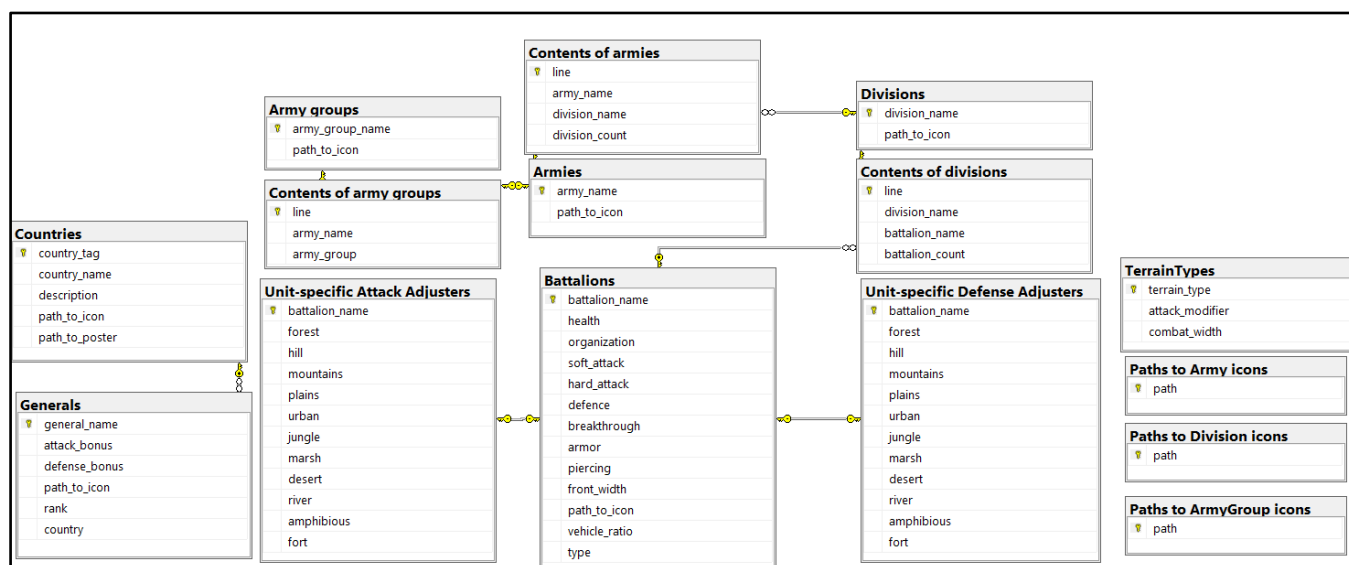


Рисунок 3.3 – Логическая диаграмма базы данных

Таблица Divisions содержит следующие столбцы:

- division\_name название дивизии, первичный ключ;
- path\_to\_icon путь к иконке дивизии.

Таблица Armies содержит следующие столбцы:

- army\_name название армии, первичный ключ;
- path\_to\_icon путь к иконке.

Таблица Army groups содержит следующие столбцы:

- army\_group\_name название группы армий, первичный ключ;
- path\_to\_icon путь к иконке.

Таблицы Divisions, Armies и Army groups содержат списки дивизий, армий и групп армий и путей к их иконкам.

Таблица Contents of divisions содержит следующие столбцы:

- line первичный ключ;
- division\_name название дивизии, вторичный ключ;
- battalion\_name название батальона, вторичный ключ;
- battalion\_count количество данных батальонов в данной дивизии.

Таблица Contents of armies содержит следующие столбцы:

- line первичный ключ;

- army\_name название армии, вторичный ключ;
- division\_name название дивизии, вторичный ключ;
- division\_count количество данных дивизий в данной армии.

Таблица Contents of army groups содержит следующие столбцы:

- line первичный ключ;
- army\_group\_name название группы армий, вторичный ключ;
- army\_name название армии, вторичный ключ.

В таблице Contents of divisions хранятся названия дивизий и названия и количество данных батальонов в данных дивизиях. В таблице Contents of armies хранятся названия армий и названия и количество данных дивизий в данных армиях. В таблице Contents of army groups хранятся названия групп армий и названия армий, входящих в данные группы армий. Данные таблицы связаны внешними ключами с таблицами Divisions и Battalions, Armies и Divisions, Army groups и Armies соответственно.

Таблица Battalions содержит следующие столбцы:

- battalion\_name название батальона или роты, первичный ключ;
- health прочность (показывает, сколько урона может принять боевая единица до того, как будет уничтожена);
- organization организация (показывает боевую готовность и организованность боевой единицы. Боевая единица без организации не может эффективно воевать и перемещаться);
- soft\_attack противопехотная атака (показывает способность боевой единицы атаковать цели без брони или с очень тонкой бронёй, например пехоту);
- hard\_attack противотанковая атака (показывает способность боевой единицы атаковать бронированные цели);
- defense защита (показывает, сколько вражеских атак попытается избежать обороняющаяся боевая единица, что позволит ей дольше держать оборону. Когда боевая единица больше не может избегать атак, она начинает получать больше урона и не может держать оборону);
- breakthrough прорыв (показывает, сколько вражеских атак попытается избежать наступающая боевая единица, что позволит ей дольше атаковать. Когда боевая единица больше не может избегать атак, она начинает получать больше урона и не может атаковать);
- armor броня (показывает броню боевой единицы. Если значение вашей брони выше, чем вражеская бронейность, вы получите меньше урона и сможете чаще атаковать в бою, поскольку боевая единица обладает более высокой свободой передвижения);
- piercing бронейность (показывает способность боевой единицы пробить броню. Если ваша бронейность равна или выше значения вражеской брони, вы сможете нанести цели больше урона и эффективнее нейтрализовать бронетехнику);
- front\_width ширина фронта (каждая дивизия занимает определённую часть фронта, на основании этой информации можно понять сколько дивизий одновременно могут вести активные боевые действия);

- path\_to\_icon путь к иконке;
- vehicle\_ratio доля пехоты (показывает, какую часть дивизии составляет бронированная или хотя бы защищённая техника);
- type тип боевой единицы (батальон или рота поддержки).

Таблица Countries содержит следующие столбцы:

- country\_tag сокращённое название страны, первичный ключ;
- country\_name полное название страны;
- description описание страны;
- path\_to\_icon путь к флагу страны;
- path\_to\_poster путь к плакату страны.

Таблица Generals содержит следующие столбцы:

- general\_name имя генерала, первичный ключ;
- attack\_bonus модификатор атаки;
- defense\_bonus модификатор обороны;
- path\_to\_icon путь к портрету;
- rank звание (генерал или фельдмаршал);
- country страна принадлежности, вторичный ключ.

В таблицах Paths to Division icons, Paths to Army icons и Paths to ArmyGroup icons хранятся пути к иконкам дивизий, армий и групп армий соответственно.

Таблица TerrainTypes содержит следующие столбцы:

- terrain\_type тип местности, первичный ключ;
- attack\_modifier модификатор атаки в данной местности;
- combat\_width ширина фронта в данной местности.

### 3.3 Взаимодействие программного средства с базой данных

Для организации работы с базой данных используется Entity Framework, позволяющий взаимодействовать с сущностями базы данных как с классами C#, с полями таблиц – как со свойствами классов, а синтаксис SQL-запросов заменен на LINQ. Entity Framework берет на себя обязанности по преобразованию кода C# в код SQL. В качестве посредника между Entity Framework и моделями представлений был разработан класс DataBaseInteraction, инкапсулирующий формирование LINQ-запросов и реализующий паттерны «Репозиторий» и «Единица работы». Для разработки программного средства удобно использовать диаграммы последовательности, показывающие, как пользователь, программное средство и база данных должны взаимодействовать. Пример диаграммы последовательности входа пользователя в качестве администратора представлен в Приложении А.

## 4 Реализация программного средства

### 4.1 Основные классы программного средства

Для реализации программного средства необходимо конструирование вспомогательных классов для реализации паттерна MVVM, сущностей базы данных и взаимодействия с базой данных и конвертеров. Так, для представления боевых единиц были разработаны классы Unit, Battalion, SupportCompany и Division. Класс Unit является абстрактным, от него остальные классы наследуют характеристики. Класс Division содержит списки батальонов и рот поддержки, поля, хранящие значения максимальной прочности и максимальной организации и метод вычисления суммарных характеристик дивизии на основе входящих в её состав батальонов и рот поддержки. Диаграмма классов боевых единиц представлена на рисунке 4.1.

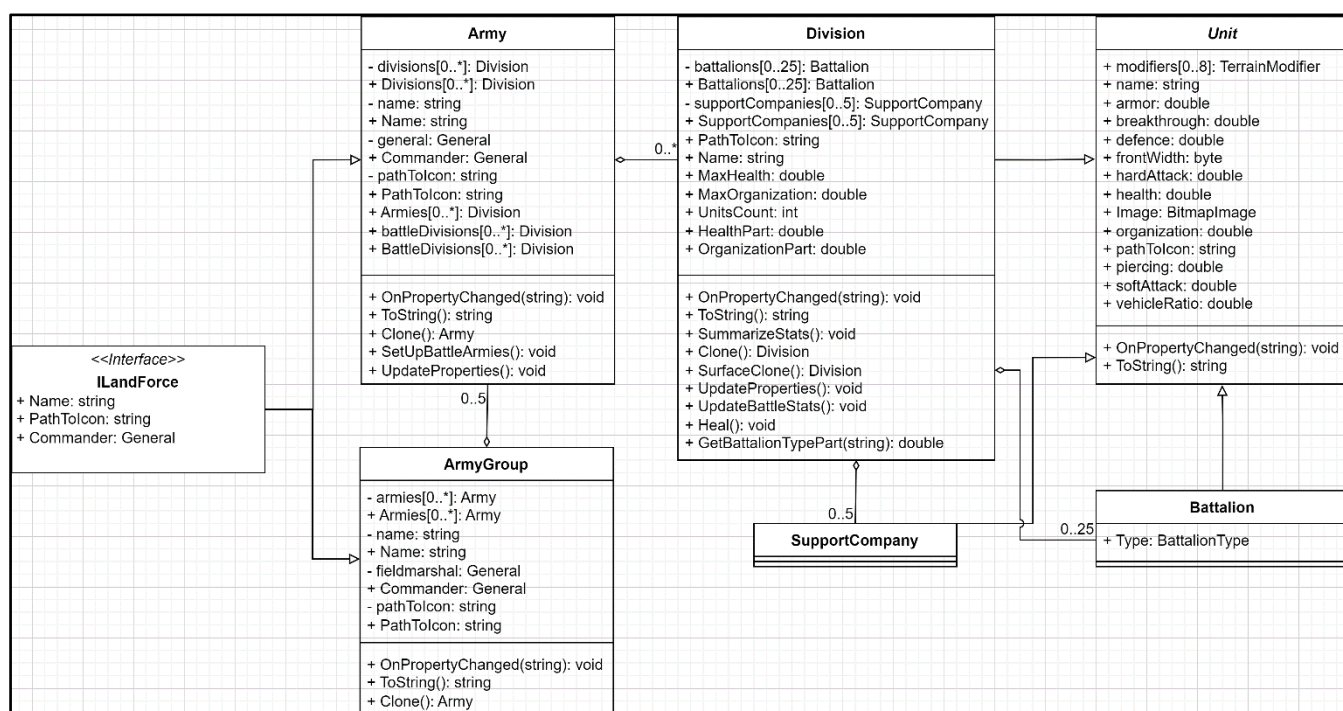


Рисунок 4.1 – Диаграмма классов боевых единиц

### 4.2 Реализация паттерна MVVM

Для реализации паттерна MVVM классы приложения были разделены на три группы по выполняемой роли: представления, модели представлений, модели. Структура директорий данных классов представлена на рисунке 4.2.



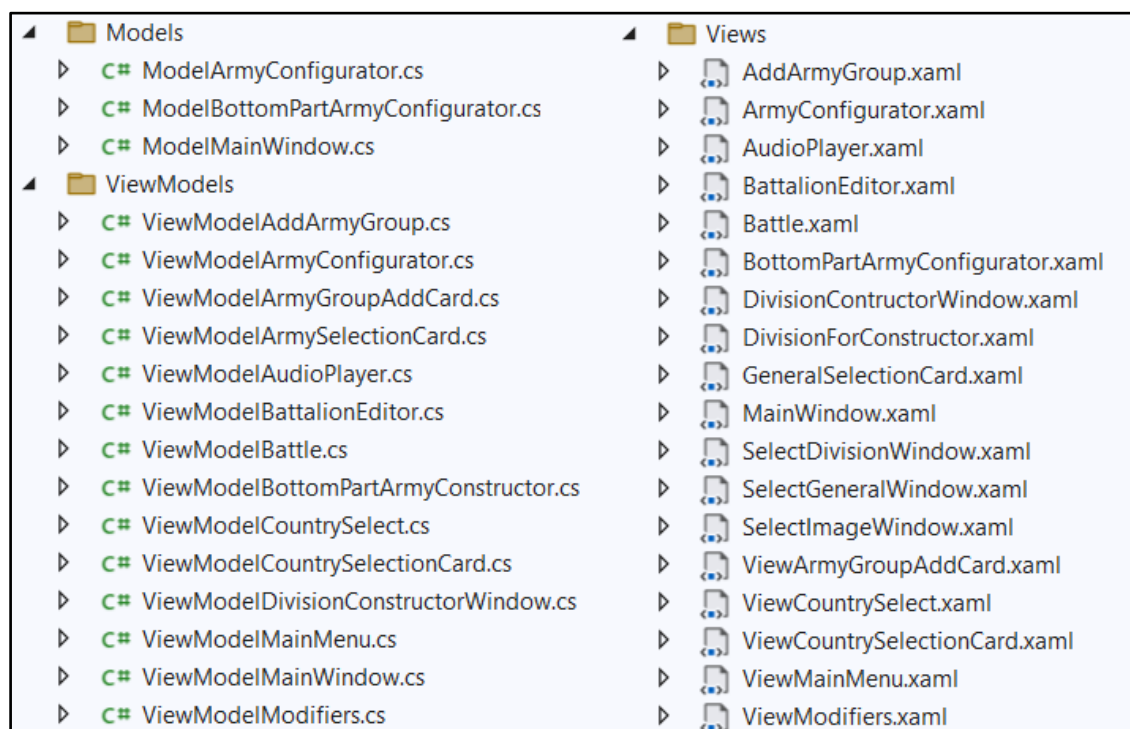


Рисунок 4.2 – Структура директорий классов паттерна MVVM

Представления представляют собой интерфейс приложения для взаимодействия с пользователем, они содержат код разметки и обработчики событий, которые не могут быть привязаны к командам в модели представления и которые перенаправляются в модель представления. Пример шаблона элемента управления с привязками значений элементов управления к свойствам модели представления представлен на рисунке 4.3.

```
<ControlTemplate x:Key="DivisionControlTemplate" TargetType="Button">
    <Grid Height="60"
        x:Name="Header"
        Background="{StaticResource LightGreen}"
        Width="{Binding ActualWidth, RelativeSource={RelativeSource AncestorType={x:Type ListBox}},
            Converter={StaticResource SummatorConverter}, ConverterParameter=60}">
        <Grid.ColumnDefinitions...>
        <Image Source="{Binding PathToIcon, Converter={StaticResource StringToImageConverter}}"
            Grid.Column="0"/>
        <TextBlock Text="{Binding Name}"
            Grid.Column="1"
            Style="{StaticResource Title}"/>
        <Grid Grid.Column="2"...>
        <Grid Grid.Column="3"...>
    </Grid>
</ControlTemplate>
```

Рисунок 4.3 – Шаблон элемента управления, демонстрирующий использование привязок значений к свойствам модели представления

Модели представлений содержат методы бизнес-логики, обработчики событий и свойства, к которым привязываются значения элементов управления в



представлении. Модели представлений получают сообщения от представлений, обрабатывают их, при необходимости получают и изменяют данные в модели и обновляют свойства, к которым совершается привязка. Пример кода модели представления с командами и свойствами, к которым производится привязка значений в представлении, представлен на рисунке 4.4.

```
private DispatcherTimer _timer = new DispatcherTimer();

Ссылка: 9
public ObservableCollection<Tracks> Tracks{...}
Ссылка: 21
public ObservableCollection<Tracks> SortedTracks { get; set; }

private MediaPlayer mediaPlayer = new();

Ссылка: 1
public ICommand RevealCommand { get; set; }
Ссылка: 1
public ICommand PauseCommand { get; set; }
Ссылка: 1
public ICommand NextTrackCommand { get; set; }
Ссылка: 1
public ICommand PreviousTrackCommand { get; set; }
Ссылка: 1
public ICommand ShuffleCommand { get; set; }
Ссылка: 1
public ICommand RepeatCommand { get; set; }
Ссылка: 1
public ICommand AddCommand { get; set; }
Ссылка: 1
public ICommand ConfirmAddCommand { get; set; }
Ссылка: 1
public ICommand CancelAddCommand { get; set; }
```

Рисунок 4.4 – Фрагмент кода модели представления, демонстрирующий команды и свойства, к которым привязываются значения в представлении

Модели хранят в себе данные приложения и предоставляют моделям представлений доступ к этим данным. В данном программном средстве модели хранят в себе списки армий обеих сторон и объекты окон, вызываемых в моделях представлений. Пример модели представлен на рисунке 4.5.

```
Ссылка: 3
public class ModelMainWindow
{
    public List<int> selectedCountries = new() { 0, 1 };

    public LandForces Attackers = new();
    public LandForces Defenders = new();

    Ссылка: 1
    public ModelMainWindow(){}
}
```

Рисунок 4.5 – Пример модели

Такое использование моделей позволяет хранить данные, общие для всего программного средства в одном месте, и не выделять дополнительную память.

### 4.3 Реализация сражения

Сражение состоит из последовательных атак и контратак. Атаковать первыми могут не все дивизии атакующей стороны, а лишь помещающиеся в ширину фронта, определяемую типом местности, на которой происходит сражение. Это значит, что атакуют только те дивизии, которые находятся в начале списка атакующих дивизий, с суммарной занимаемой шириной фронта меньшей либо равной максимальной ширине фронта для данной местности. Ширина фронта дивизии вычисляется как сумма данных характеристик входящих в дивизию батальонов.

Атакующие дивизии по очереди выбирают случайным образом свою цель из всех защищающихся дивизий и проводят одну атаку. При этом сравнивается показатель защиты защищающейся дивизии и количество атак атакующей дивизии. Количество атак вычисляется как произведение модификатора атаки выбранной местности, модификатора атаки дивизии на данной местности и суммы произведения противопехотной атаки и разности единицы и доли техники защищающейся дивизии и произведения противотанковой атаки и доли техники защищающейся дивизии.

Затем рассчитывается количество атак, поразивших цель. Это значение вычисляется как сумма произведения базового шанса уклонения при наличии защиты и количества атак, если защита больше количества атак, и суммы произведения защиты защищающейся дивизии и разности единицы и базового шанса уклонения при наличии защиты и произведения разности единицы и шанса уклонения без защиты и разности количества атак и защиты защищающейся дивизии.

Затем производится расчёт урона, наносимого прочности и организации защищающейся дивизии. Для расчёта урона прочности вычисляется модификатор урона от текущей прочности атакующей дивизии. Он равен частному произведения частного текущей прочности дивизии и её максимальной прочности и десяти и десяти. Для получения урона прочности полученный модификатор умножается на количество попаданий, базовый модификатор урона прочности и частное математического ожидания значения брошенной двугранной игральной кости и двух. Если броня защищающейся дивизии превосходит бронепробитие атакующей дивизии, урон делится на два.

Урон организации вычисляется сходным образом, однако в формуле отсутствует модификатор урона от прочности и количество граней игральной кости равно четырём в случае, если броня атакующей дивизии не превосходит бронепробитие защищающейся дивизии, и равно шести в противном случае, и вместо базового модификатора урона прочности используется базовый модификатор урона организации. Полученные значения отнимаются от текущих значений прочности и организации защищающейся дивизии, и в случае падения любого из этих показателей до нуля дивизия выходит из боя. В противном случае

защищающаяся дивизия проводит контратаку, точно так же, как атаковали её, однако вместо защиты бывшая атакующая дивизия использует показатель прорыва.

Листинг кода, отвечающего за сражение, представлен в Приложении В.

## 5 Тестирование

### 5.1 Тестирование изменения характеристик рот и батальонов

Для изменения характеристик рот и батальонов пользователь должен войти в учётную запись администратора. Для этого он должен нажать на кнопку входа в качестве администратора и ввести логин и пароль. При вводе неправильных данных должно выводиться сообщение об ошибке. На рисунке 5.1 представлено главное окно приложения после нажатия на кнопку входа при вводе правильных данных (сверху) и неправильных данных (снизу).

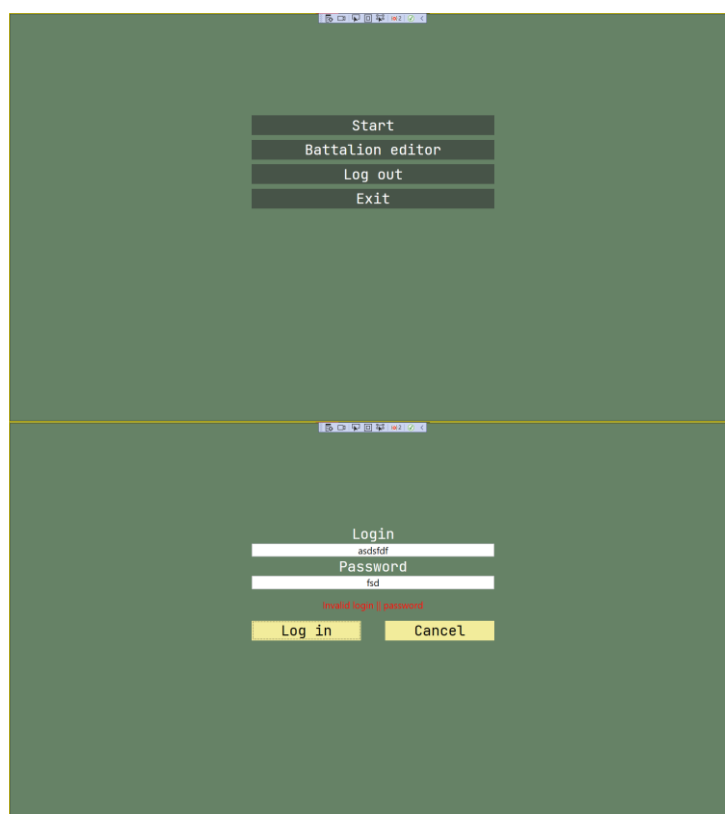


Рисунок 5.1 – Вид главного окна при различных введённых данных входа в учётную запись администратора

После входа в учётную запись администратора в меню изменения характеристик батальонов нужно выбрать нужную ячейку и ввести новое значение. При потере фокуса ячейки будет вызван метод модели представления, выполняющий валидацию нового значения и в случае его правильности вызывающий метод класса DataBaseInteraction, сохраняющий новое значение в базу данных. На рисунке 5.2 представлено состояние базы данных до изменения значения организации батальона Anti-Air (сверху) и после (снизу). В случае неправильности нового значения изменение отменяется и в ячейку вносится старое значение.

`select * from Battalions where battalion_name = 'Anti-Air'`

результаты Сообщения

battalion_name	health	organization	soft_attack	hard_attack	defence	breakthrough	armor	piercing	front_width	path_to_icon	vehicle_ratio	type
Anti-Air	0,6	0	3	7	4	1	0	60	1	./././Images/Battalion Icons/AA.png	0	inf

Name	Health	Organization	SoftAttack	HardAttack	Defense	Breakthrough
Anti-Air	0.6000000238418579	0	3	7	4	1

Name	Health	Organization	SoftAttack	HardAttack	Defense	Breakthrough
Anti-Air	0.6000000238418579	5	3	7	4	1

`select * from Battalions where battalion_name = 'Anti-Air'`

результаты Сообщения

battalion_name	health	organization	soft_attack	hard_attack	defence	breakthrough	armor	piercing	front_width	path_to_icon	vehicle_ratio	type
Anti-Air	0,6	5	3	7	4	1	0	60	1	./././Images/Battalion Icons/AA.png	0	inf

Рисунок 5.2 – Состояния базы данных до и после изменения значения организации

## 5.2 Тестирование создания пользовательских дивизий и сохранения их в базе данных

Для создания дивизии пользователь должен выбрать необходимые батальоны и роты поддержки. При этом допустимое количество рот поддержки – от нуля до пяти, а батальонов – от одного до двадцати пяти. Если не выбран ни один батальон, сохранение не произойдёт. При нажатии на кнопку сохранения шаблон сразу сохраняется в базе данных. Тест-кейс добавления новой дивизии в выбранную армию представлен в таблице 5.1.

Таблица 5.1 – Тест-кейс добавления новой дивизии в выбранную армию

Номер	1
Заголовок	Добавление новой дивизии
Предусловие	Открыто меню конфигуратора армий, выбрана группа армий и армия
Шаг	Ожидаемый результат
Нажать на кнопку добавления дивизии	Появилось окно выбора дивизии
Нажать на кнопку добавления новой дивизии	Появилось окно конструктора дивизий с пустой дивизией
Нажать на кнопку изменения названия	Кнопка изменения изменила задний фон
Ввести новое название дивизии	В поле названия дивизии отображается название дивизии
Нажать на кнопку изменения дивизии	Кнопка изменения изменила задний фон
Нажать на кнопку изменения иконки	Появилось окно выбора иконки
Нажать на иконку	Окно выбора иконки закрылось, иконка изменилась

Продолжение таблицы 5.1

Нажать на кнопку добавления батальона	Область с ротами и батальонами исчезла, появилось меню выбора батальонов
Нажать на батальон	Батальон исчез из списка доступных, характеристики дивизии изменились
Нажать на кнопку «Назад»	Меню выбора батальонов исчезло, появилась область с ротами и батальонами
Нажать на кнопку увеличения количества батальона	Количество батальонов данного типа увеличилось на 1
Нажать на кнопку «Сохранить»	Название дивизии изменилось на название по умолчанию, список батальонов очистился, характеристики обнулились, в окне выбора дивизий появилась новая дивизия
Нажать на кнопку «Закрыть»	Окно редактора дивизий закрылось

Содержимое таблицы Contents of Divisions до (сверху) и после (снизу) создания новой дивизии представлено на рисунке 5.3.

select \* from [Contents of divisions]

108 %

Результаты Сообщения

	line	division_name	battalion_name	battalion_count
1	103	div 2	Camelry	3
2	104	div 2	Artillery	1
3	129	div 1	Infantry	9
4	130	div 1	Engineer Company	1
5	131	div 1	Support Artillery	1
6	132	Armored Division	Mechanized Infantry	12
7	133	Armored Division	Motorized Artillery	6
8	134	Armored Division	Support Artillery	1
9	135	Armored Division	Support Rocket Artillery	1
10	136	Armored Division	Engineer Company	1

select \* from [Contents of divisions]

108 %

Результаты Сообщения

	line	division_name	battalion_name	battalion_count
1	103	div 2	Camelry	3
2	104	div 2	Artillery	1
3	129	div 1	Infantry	9
4	130	div 1	Engineer Company	1
5	131	div 1	Support Artillery	1
6	132	Armored Division	Mechanized Infantry	12
7	133	Armored Division	Motorized Artillery	6
8	134	Armored Division	Support Artillery	1
9	135	Armored Division	Support Rocket Artillery	1
10	136	Armored Division	Engineer Company	1
11	137	Demonstration	Modern SP Artillery	3
12	138	Demonstration	Light Flame Tank Company	1

Рисунок 5.3 – Демонстрация изменений в базе данных при сохранении новой дивизии.

Таким образом, создание и сохранение шаблонов работает корректно.

### 5.3 Тестирование прикрепления и открепления командующих

Без прикрепленного командующего на месте портрета должно отображаться соответствующее изображение. Представление карточки группы армий без прикрепленного фельдмаршала представлено на рисунке 5.4.



Рисунок 5.4 – Внешний вид группы армий без прикрепленного фельдмаршала

При нажатии на портрет командующего должно появляться окно выбора командующего, содержащее всех доступных командующих соответствующего ранга (генералы для армий, фельдмаршалы для групп армий) и соответствующей страны. Внешний вид данного окна для французских фельдмаршалов при условии отсутствия уже прикрепленных к какой-либо группе армий командующих представлен на рисунке 5.5.

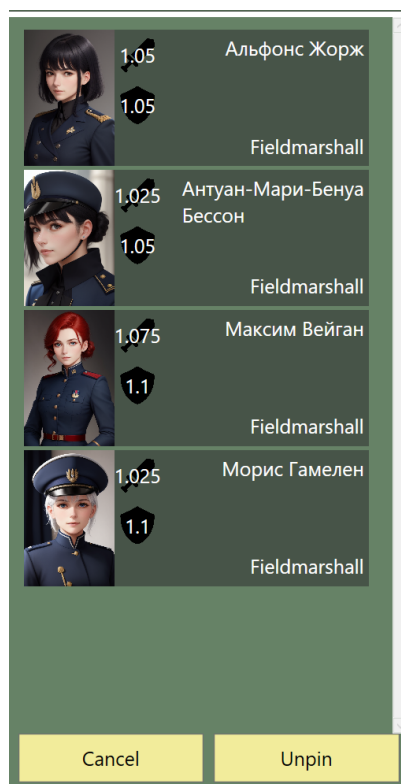


Рисунок 5.5 – Внешний вид окна выбора командующего до прикрепления

При нажатии на командующего он прикрепляется к той армии или группе армий, через которую было вызвано окно выбора. Внешний вид карточки группы армий после прикрепления фельдмаршала представлен на рисунке 5.6.

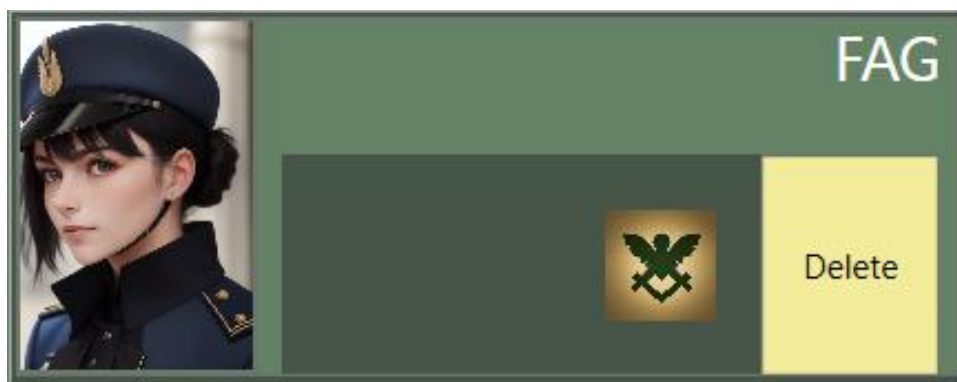


Рисунок 5.6 – Внешний вид группы армий с прикреплённым фельдмаршалом

После прикрепления командующего к боевой единице он пропадает из списка доступных для выбора в окне добавления командующего. Внешний вид данного окна после прикрепления фельдмаршала представлен на рисунке 5.7.



Рисунок 5.7 – Внешний вид окна выбора командующего после прикрепления















Таким образом, прикрепление и открепление командующих работает корректно

#### 5.4 Тестирование добавления и удаления модификаторов

При добавлении и удалении модификатора он сразу сохраняется или удаляется из базы данных. Строки таблицы Modifiers и содержимое списка модификаторов в приложении до добавления нового модификатора (сверху) и после (снизу) представлены на рисунке 5.8.

select * from Modifiers					
% <					
Результаты Сообщения					
id	name	battalion_type	property	modifier	path_to_icon
1	Infantry Attack for Soviets	inf	Soft attack	1,25	../Images/Modifiers/15.png
2	Tank Defense for Germany	tnk	Defense	1,3	../Images/Modifiers/1.png

	Infantry Attack for Soviets		1.25				
	Tank Defense for Germany		1.3				

select * from Modifiers					
, <					
Результаты Сообщения					
id	name	battalion_type	property	modifier	path_to_icon
1	Infantry Attack for Soviets	inf	Soft attack	1,25	../Images/Modifiers/15.png
2	Tank Defense for Germany	tnk	Defense	1,3	../Images/Modifiers/1.png
9	New modifier	mob	Soft attack	1,05	../Images/Modifiers/19.png



















	Infantry Attack for Soviets		1.25				
	Tank Defense for Germany		1.3				
	New modifier		1.05				

Рисунок 5.8 – Демонстрация изменений в базе данных и приложении при добавлении нового модификатора

Внешний вид списка модификаторов в приложении и строк таблицы Modifiers в базе данных до (сверху) и после (снизу) нажатия на кнопку удаления модификатора New modifier представлен на рисунке 5.9.

The image shows two screenshots of a database interface and an application UI. The top screenshot shows the state before deletion, and the bottom screenshot shows the state after deletion.

**Top Screenshot (Before Deletion):**

SQL Query: `select * from Modifiers`

id	name	battalion_type	property	modifier	path_to_icon
1	Infantry Attack for Soviets	inf	Soft attack	1,25	../Images/Modifiers/15.png
2	Tank Defense for Germany	tnk	Defense	1,3	../Images/Modifiers/1.png
9	New modifier	mob	Soft attack	1,05	../Images/Modifiers/19.png

**Bottom Screenshot (After Deletion):**

SQL Query: `select * from Modifiers`

id	name	battalion_type	property	modifier	path_to_icon
1	Infantry Attack for Soviets	inf	Soft attack	1,25	../Images/Modifiers/15.png
2	Tank Defense for Germany	tnk	Defense	1,3	../Images/Modifiers/1.png

The application UI below each table shows three rows of modifier data. Each row includes an icon, the name, a property icon, a numerical value, and two action buttons (edit and delete). In the top UI, the third row corresponds to the 'New modifier' entry. In the bottom UI, this row has been removed.

Рисунок 5.9 – Демонстрация изменений в базе данных и приложении при удалении модификатора

### 5.5 Тестирование изменения типа местности

Перед боем пользователь может изменить тип местности, выбрав его из выпадающего меню. После начала сражения изменение типа местности автоматически отменяется до окончания боя. Внешний вид выпадающего меню (сверху) и фрагмент исходного кода, рассчитывающий значение атаки дивизии с учётом текущей местности, (снизу) представлены на рисунке 5.10.

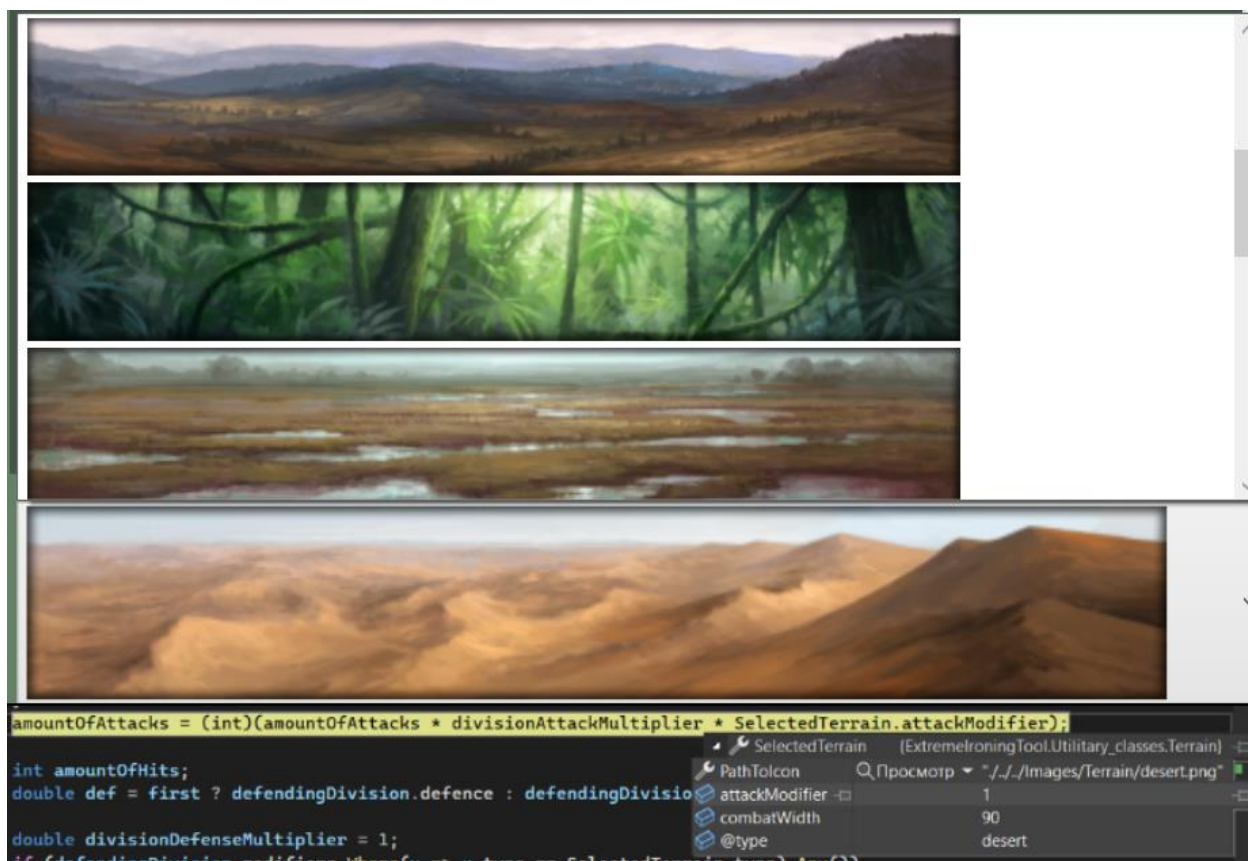


Рисунок 5.10 – Демонстрация влияния типа местности на показатели дивизии в бою

Таким образом, было доказано выполнение всех заявленных требований к программному средству.

## 6 Руководство пользователя

При запуске программного средства отображается главное меню, из которого пользователь может выйти из приложения (кнопка «Exit»), войти в качестве администратора (кнопка «Log in as admin»), попасть в меню изменения характеристик рот и батальонов (кнопка «Battalion editor») и затем выйти из учётной записи (кнопка «Log out») и перейти в меню выбора стран (кнопка «Start»).

Для получения прав администратора пользователь должен нажать на кнопку «Log in as admin». Появится меню ввода логина и пароля. Из этого меню пользователь может выйти в главное меню, нажав на кнопку «Cancel», или ввести данные для входа (например, Admin1 и 1234) и, нажав на кнопку «Log in», вернуться в главное меню с правами администратора. При вводе данных, не совпадающих с имеющимися в базе данных, выводится сообщение об ошибке.

После получения пользователем прав администратора в главном меню появляется кнопка, нажатие на которую производит переход в меню изменения характеристик рот и батальонов. Оно представляет собой таблицу, содержащую характеристики всех рот и батальонов в базе данных. Пользователь может вводить любые значения в любые ячейки, кроме ячеек, содержащих названия рот и батальонов. При потере ячейкой фокуса значение либо сохранится в базе данных, если оно валидно, либо изменение отменится. Для возврата в главное меню пользователь должен нажать на кнопку «Back».

В меню выбора стран отображаются четыре кнопки с названиями и изображениями стран, название, описание и плакат последней выбранной страны и значки, отображающие текущий выбор стран. Значок с цифрой «1» означает, что выбранная страна является атакующей, а значок с цифрой «2» – защищающейся. При нажатии на страну вне зависимости от выбранных стран название, описание и плакат изменяется на название, описание и плакат соответствующей страны. Если при нажатии на страну она была выбрана, выбор снимается. Если она не была выбрана, и атакующая или защищающаяся страна не выбрана, она становится выбранной. Для возврата в главное меню пользователь должен нажать на кнопку «Back», для перехода к конфигуратору армий – «Forward».

В меню конфигулятора армий отображаются наборы групп армий, армий и дивизий для обеих стран. Пользователь может переключаться между представлениями разных стран нажатием на кнопку с флагом нужной страны в верхней части окна. Кнопка выбранной страны увеличивается в размерах, а кнопка не выбранной – уменьшается. Каждая группа армий и армия представляет собой карточку с названием, портретом командующего, списком входящих в данную группу армий или армию армий или дивизий и кнопку удаления из текущего набора войск. Эта кнопка не удаляет группу армий или армию из базы данных. Для выбора командующего пользователь должен нажать на портрет командующего. Появится окно выбора командующего, содержащее список командующий выбранной страны и ранга, зависящего от того, выбрана ли какая-нибудь группа армий (если группа армий выбрана, список содержит генералов, иначе – фельдмаршалов). При нажатии на командующего он прикрепляется к боевой единице, на портрет командующего которой нажимал пользователь. Если данная боевая единица уже имеет

прикреплённого командующего, он заменяется новым командующим, а старый командующий снова становится доступным для выбора. Новый командующий пропадает из списка доступных для выбора для других боевых единиц. Если пользователь хочет открепить генерала от его боевой единицы, он должен нажать на портрет нужного командующего и в окне выбора командующего нажать на кнопку «Unpin». Командующий будет откреплён от боевой единицы и будет снова доступен для прикрепления. Также командующий открепляется от удаляемой боевой единицы (и при удалении из текущего набора боевых единиц, и при удалении из базы данных).

Для добавления новой группы армий пользователь должен нажать на кнопку добавления группы армий, представленную на рисунке 6.1. Появится окно добавления группы армий, в котором пользователь может добавить пустую группу армий, нажав на кнопку «Add empty army group», закрыть окно, нажав на кнопку «Cancel» или добавить группу армий, загруженную из базы данных, нажав на её карточку. Карточка группы армий содержит её название, иконку и список иконок входящих в неё армий.



Рисунок 6.1 – Кнопка добавления группы армий

Для перехода к списку армий нужной группы армий пользователь должен нажать на любую область внутри карточки группы армий, кроме портрета командующего и кнопки удаления. Пропадет список групп армий и появится список армий выбранной группы армий. Кнопка добавления группы армий сменится на кнопку добавления пустой армии, представленную на рисунке 6.2. В центральной части экрана появится иконка и название выбранной группы армий и кнопка изменения названия.



Рисунок 6.2 – Кнопка добавления пустой армии

Карточки армий ничем не отличаются от карточек групп армий. Для возврата к списку групп армий пользователь должен нажать на кнопку, представленную на рисунке 6.3.



Рисунок 6.3 – Кнопка возврата к списку групп армий

Для добавления новой армии к нужной группе армий пользователь должен выбрать нужную группу армий и нажать на кнопку добавления армии. К выбранной группе армий добавится пустая армия с названием по умолчанию.

Для сохранения или удаления нужной боевой единицы пользователь должен выбрать её и нажать на кнопку сохранения или удаления из базы данных. Кнопки сохранения и удаления из базы данных представлены на рисунке 6.4. После нажатия на обе данные кнопки произойдёт возврат к списку групп армий. Меню конфигуратора армий также содержит строку состояния, в которую выводятся временные сообщения о результатах сохранения и удаления дивизий, армий и групп армий.



Рисунок 6.4 – Кнопки сохранения (слева) и удаления (справа) из базы данных

Для перехода к списку дивизий выбранной армии пользователь должен выбрать армию так же, как группу армий. В центральной части экрана справа от иконки и названия выбранной группы армий появится иконка и название армии, кнопка изменения названия армии и кнопка добавления дивизий в выбранную армию. Под иконками и названиями группы армий и армии появится список дивизий. Карточка дивизии состоит из иконки и названия дивизии, количества дивизий данного шаблона в выбранной армии и кнопок увеличения и уменьшения количества дивизий данного шаблона в выбранной армии на единицу и удаления всех дивизий данного шаблона из выбранной армии.

Пользователь может изменять названия и иконки любых групп армий и армий. Для изменения названия он должен нажать на кнопку, представленную на рисунке 6.5. Кнопка изменит своё изображение на галочку, и текстовое поле с названием армии или группы армий станет доступно для изменения. После изменения названия пользователь должен нажать на кнопку с галочкой, чтобы сохранить изменения. В случае существования боевой единицы с введённым названием введённое название заменяется на название по умолчанию. Для изменения иконки пользователь должен нажать на неё и в появившемся окне, содержащем список иконок для данного типа боевых единиц нажать на нужную иконку. Для отмены изменения иконки пользователь должен нажать на кнопку «Cancel».



Рисунок 6.5 – Кнопка изменения названия до нажатия (слева) и после нажатия (справа)



Для добавления новой дивизии к текущей армии пользователь должен нажать на кнопку добавления дивизии, представленную на рисунке 6.6.



Рисунок 6.6 – Кнопка добавления новой дивизии

Появится окно выбора дивизий, в котором пользователь может открыть окно конструктора дивизий, нажав на кнопку «Create new division», добавить существующую дивизию, нажав на неё в списке дивизий, удалить дивизию из базы данных, нажав на кнопку с корзиной в карточке нужной дивизии, открыть окно конструктора для изменения дивизии, нажав на кнопку изменения в карточке нужной дивизии, и закрыть окно добавления дивизии, нажав на кнопку «Cancel».

Окно конструктора содержит кнопку закрытия в правом верхнем углу окна, кнопку изменения иконки в левом верхнем углу окна, название дивизии, кнопку изменения названия дивизии, область с параметрами и модификаторами атаки и защиты для каждого типа местности, а также области рот и батальонов. Если количество рот меньше пяти, а количество батальонов меньше двадцати пяти, то области рот и батальонов содержат кнопки добавления новой роты или батальона. При превышении количеством рот и батальонов данных значений кнопки пропадают. При нажатии на кнопку добавления роты или батальона области рот и батальонов пропадают, и появляется область добавления роты или батальона. Для добавления пользователь должен нажать на элемент списка с нужным значком. Рота или батальон будет добавлен в состав дивизии и пропадёт из списка доступных для выбора в области добавления. Если пользователь добавляет пятую роту или двадцать пятый батальон, область добавления пропадает, и появляются области рот и батальонов. Пользователь может применять фильтр типа при добавлении батальонов, нажав на нужную радиокнопку в верхней части области добавления. При изменении количественного и качественного состава дивизии обновляются параметры и модификаторы типов местности. Переименование и смена иконки дивизии работает аналогично группе армий и армии. Для сохранения дивизии пользователь должен нажать на кнопку «Save». Пользователь не может сохранить дивизию, не имеющую ни одного батальона. При успешном сохранении дивизии она добавится в список в окне добавления дивизий.

Для перехода в меню модификаторов пользователь должен нажать на кнопку «Forward». В меню выбора модификаторов находится список модификаторов, существующих в базе данных и кнопки возврата в меню конфигурирования армий, перехода в меню сражения и добавления нового модификатора. Карточка модификатора содержит его иконку, название, иконку характеристики, на которую влияет данный модификатор, значение, используемое в вычислениях, кнопки изменения и удаления модификатора и два чекбокса, отвечающих за страны, к которым будет применён данный модификатор. Для добавления модификатора пользователь должен нажать на кнопку добавления, представленную на рисунке 6.7,

в верхней части окна. Список модификаторов исчезнет, и появится область ввода параметров модификатора. Пользователь должен ввести название модификатора, выбрать характеристику из выпадающего списка, ввести числовое значение, выбрать тип батальонов, к которым будет применяться модификатор. Если название модификатора будет занято, оно будет заменено на название по умолчанию. При ошибке в введенных данных будет выведено сообщение об ошибке. Процесс изменения иконки модификатора аналогичен изменению иконки групп армий, армий и дивизий. Для сохранения модификатора в базе данных пользователь должен нажать на кнопку «Save», для отмены создания – на «Cancel». Для изменения модификатора пользователь должен нажать на кнопку изменения, аналогичную представленной на рисунке 6.3 (слева). Появится область добавления нового модификатора, но значения параметров модификатора будут заполнены. Если пользователь изменит название модификатора и нажмёт на кнопку сохранения, в базе данных будет сохранён новый модификатор, если же название не будет изменено, то будет обновлён существующий модификатор. При нажатии на кнопку удаления, аналогичную представленной на рисунке 6.4 (справа) модификатор будет удалён из базы данных и из списка в меню модификаторов.

Для перехода в меню сражения пользователь должен нажать на кнопку «Forward». Меню сражения содержит кнопки перехода в меню модификаторов, начала (видна только вне сражения) и остановки сражения (видна только во время сражения), списки боевых единиц атакующей и защищающейся страны, текстовое поле, в которое записывается каждое действие дивизий и выпадающий список, содержащий типы местности, на которых будет происходить сражение. Карточки боевых единиц содержат иконку и название. Карточки дивизий также содержат два индикатора: прочности (цвет #FFA556, слева) и организации (цвет #00FF50, справа). Когда у дивизии заканчивается прочность или организация, она выбывает из сражения и больше не вступает в бой. Пользователь может выбрать тип местности, и начать сражение, нажав на кнопку «Battle». Дивизии начнут атаковать друг друга до момента, когда у одной из сторон не закончатся боеспособные дивизии. В лог записывается победитель сражения. Пользователь не может изменять тип местности во время сражения: изменение отменится.



## Заключение

В результате выполнения курсового проекта было разработано программное средство «ExtremeIroningTool» на языке C# с использованием технологий Entity Framework и Windows Presentation Foundation. Был реализован весь заявленный функционал программного средства, а именно:

- Изменение характеристик рот и батальонов администратором
- Создание собственных шаблонов дивизий из батальонов и рот поддержки
- Сохранение шаблонов в базе данных
- Создание набора дивизий из готовых или собственных шаблонов для обеих сторон
- Добавление, удаление и изменение модификаторов и генералов, влияющих на эффективность дивизий
- Изменение типа местности, на котором будет проходить сражение

Дополнительно был реализован MP-3 проигрыватель с функциями добавления и удаления треков, перемешивания и выбора различных способов повтора.

При проектировании использовался паттерн MVVM. Для облегчения взаимодействия с базой данных использовались паттерны «Репозиторий» и «Единица работы».

По результатам тестирования программного средства был сделан вывод о его работоспособности и пригодности к использованию пользователями.

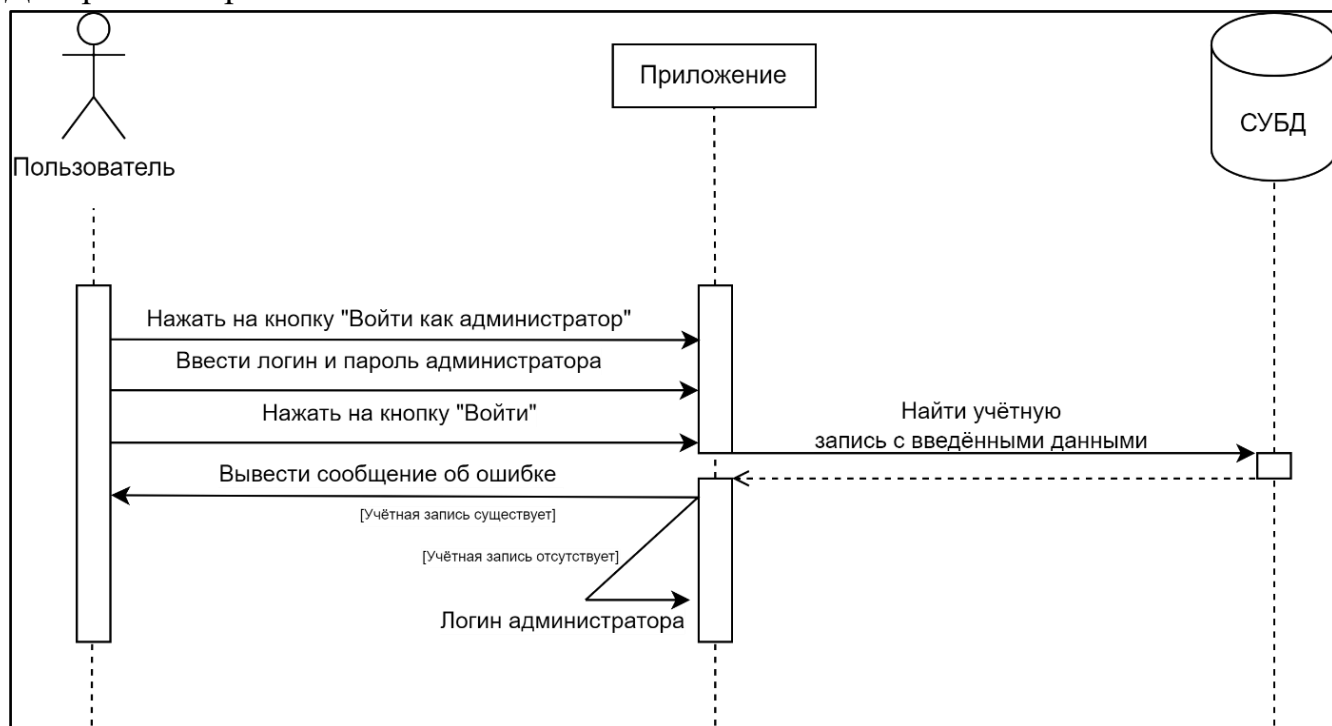
В соответствии с полученным результатом работы программы можно сделать вывод, что разработанное программное средство работает верно, а требования технического задания выполнены в полном объеме.

### Список использованных источников

1. Полное руководство по языку программирования C# 11 и платформе .NET 7 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://metanit.com/sharp/tutorial/>. Дата доступа: 18.05.2023
2. Руководство по WPF [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://metanit.com/sharp/wpf/>. Дата доступа: 18.05.2023
3. Изучение C# WPF на практике [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://www.youtube.com/playlist?list=PL0lO\\_mIqDDFVI0xwaYbm7h9ewYu5hftfA](https://www.youtube.com/playlist?list=PL0lO_mIqDDFVI0xwaYbm7h9ewYu5hftfA). Дата доступа: 18.05.2023
4. Руководство по MS SQL Server 2022 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://metanit.com/sql/sqlserver>. Дата доступа: 18.05.2023
5. Land battle [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://hoi4.paradoxwikis.com/Land\\_battle](https://hoi4.paradoxwikis.com/Land_battle). Дата доступа: 19.05.2023

## Приложение А

### Диаграмма вариантов использования



## Приложение Б

### SQL-сценарий создания базы данных программного средства

```

USE [master]
GO
/***** Object:  Database [ExtremeIroningDatabase]    Script Date:
18.05.2023 19:01:39 *****/
CREATE DATABASE [ExtremeIroningDatabase]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'ExtremeIroningDatabase', FILENAME =
N'D:\Programs\SQL\MSSQL16.MSQL_SERVER_V1\MSSQL\DATA\ExtremeIroningDa
tabase.mdf' , SIZE = 73728KB , MAXSIZE = UNLIMITED, FILEGROWTH =
65536KB )
    LOG ON
    ( NAME = N'ExtremeIroningDatabase_log', FILENAME =
N'D:\Programs\SQL\MSSQL16.MSQL_SERVER_V1\MSSQL\DATA\ExtremeIroningDa
tabase_log.ldf' , SIZE = 73728KB , MAXSIZE = 2048GB , FILEGROWTH =
65536KB )
    WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET COMPATIBILITY_LEVEL =
160
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [ExtremeIroningDatabase].[dbo].[sp_fulltext_database] @action =
'enable'
end
GO
ALTER DATABASE [ExtremeIroningDatabase] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET ANSI_NULLS OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET ANSI_PADDING OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET ARITHABORT OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET AUTO_UPDATE_STATISTICS
ON
GO

```

```
ALTER DATABASE [ExtremeIroningDatabase] SET CURSOR_CLOSE_ON_COMMIT
OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [ExtremeIroningDatabase] SET CONCAT_NULL_YIELDS_NULL
OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET DISABLE_BROKER
GO
ALTER DATABASE [ExtremeIroningDatabase] SET
AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET
DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET ALLOW_SNAPSHOT_ISOLATION
OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [ExtremeIroningDatabase] SET READ_COMMITTED_SNAPSHOT
OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET HONOR_BROKER_PRIORITY
OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET RECOVERY FULL
GO
ALTER DATABASE [ExtremeIroningDatabase] SET MULTI_USER
GO
ALTER DATABASE [ExtremeIroningDatabase] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [ExtremeIroningDatabase] SET DB_CHAINING OFF
GO
ALTER DATABASE [ExtremeIroningDatabase] SET FILESTREAM(
NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [ExtremeIroningDatabase] SET TARGET_RECOVERY_TIME =
60 SECONDS
```

```

GO
ALTER DATABASE [ExtremeIroningDatabase] SET DELAYED_DURABILITY =
DISABLED
GO
ALTER DATABASE [ExtremeIroningDatabase] SET
ACCELERATED_DATABASE_RECOVERY = OFF
GO
EXEC sys.sp_db_vardecimal_storage_format N'ExtremeIroningDatabase',
N'ON'
GO
ALTER DATABASE [ExtremeIroningDatabase] SET QUERY_STORE = ON
GO
ALTER DATABASE [ExtremeIroningDatabase] SET QUERY_STORE
(OPERATION_MODE = READ_WRITE, CLEANUP_POLICY =
(STALE_QUERY_THRESHOLD_DAYS = 30), DATA_FLUSH_INTERVAL_SECONDS =
900, INTERVAL_LENGTH_MINUTES = 60, MAX_STORAGE_SIZE_MB = 1000,
QUERY_CAPTURE_MODE = AUTO, SIZE_BASED_CLEANUP_MODE = AUTO,
MAX_PLANS_PER_QUERY = 200, WAIT_STATS_CAPTURE_MODE = ON)
GO
USE [ExtremeIroningDatabase]
GO
/***** Object: Table [dbo].[Admins]      Script Date: 18.05.2023
19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Admins](
    [UserName] [nvarchar](50) NOT NULL,
    [Password] [nvarchar](100) NULL,
    CONSTRAINT [PK_Admins] PRIMARY KEY CLUSTERED
(
    [UserName] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Armies]      Script Date: 18.05.2023
19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Armies](
    [army_name] [nvarchar](50) NOT NULL,
    [path_to_icon] [nvarchar](200) NOT NULL,
    CONSTRAINT [PK_Armies] PRIMARY KEY CLUSTERED

```

```

(
    [army_name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Army groups]      Script Date:
18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Army groups](
    [army_group_name] [nvarchar](50) NOT NULL,
    [path_to_icon] [nvarchar](200) NOT NULL,
    CONSTRAINT [PK_Army groups] PRIMARY KEY CLUSTERED
(
    [army_group_name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Battalions]      Script Date: 18.05.2023
19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Battalions](
    [battalion_name] [nvarchar](50) NOT NULL,
    [health] [real] NOT NULL,
    [organization] [real] NOT NULL,
    [soft_attack] [real] NOT NULL,
    [hard_attack] [real] NOT NULL,
    [defence] [real] NOT NULL,
    [breakthrough] [real] NOT NULL,
    [armor] [real] NOT NULL,
    [piercing] [real] NOT NULL,
    [front_width] [tinyint] NOT NULL,
    [path_to_icon] [nvarchar](200) NOT NULL,
    [vehicle_ratio] [real] NOT NULL,
    [type] [nchar](3) NULL,
    CONSTRAINT [PK_Batallions] PRIMARY KEY CLUSTERED
(
    [battalion_name] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Contents of armies]      Script Date:
18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Contents of armies](
    [line] [int] IDENTITY(1,1) NOT NULL,
    [army_name] [nvarchar](50) NOT NULL,
    [division_name] [nvarchar](50) NOT NULL,
    [division_count] [int] NOT NULL,
    CONSTRAINT [PK_Contents of armies] PRIMARY KEY CLUSTERED
(
    [line] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Contents of army groups]  Script
Date: 18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Contents of army groups](
    [line] [int] IDENTITY(1,1) NOT NULL,
    [army_name] [nvarchar](50) NOT NULL,
    [army_group] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Contents of army groups] PRIMARY KEY CLUSTERED
(
    [line] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
    CONSTRAINT [unique_army_names] UNIQUE NONCLUSTERED
(
    [army_name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

```



```

/***** Object: Table [dbo].[Contents of divisions]    Script Date:
18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Contents of divisions](
    [line] [int] IDENTITY(1,1) NOT NULL,
    [division_name] [nvarchar](50) NOT NULL,
    [battalion_name] [nvarchar](50) NOT NULL,
    [battalion_count] [tinyint] NOT NULL,
    CONSTRAINT [PK_Contents of divisions] PRIMARY KEY CLUSTERED
(
    [line] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Countries]    Script Date: 18.05.2023
19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Countries](
    [country_tag] [nchar](3) NOT NULL,
    [country_name] [nvarchar](50) NOT NULL,
    [description] [nvarchar](1000) NOT NULL,
    [path_to_icon] [nvarchar](100) NOT NULL,
    [path_to_poster] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Countries] PRIMARY KEY CLUSTERED
(
    [country_tag] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Divisions]    Script Date: 18.05.2023
19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Divisions](
    [division_name] [nvarchar](50) NOT NULL,
    [path_to_icon] [nvarchar](200) NULL,

```

```

CONSTRAINT [PK_Divisions] PRIMARY KEY CLUSTERED
(
    [division_name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Generals]    Script Date: 18.05.2023
19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Generals](
    [general_name] [nvarchar](50) NOT NULL,
    [attack_bonus] [real] NOT NULL,
    [defense_bonus] [real] NOT NULL,
    [path_to_icon] [nvarchar](200) NOT NULL,
    [rank] [nvarchar](20) NOT NULL,
    [country] [nchar](3) NOT NULL,
    CONSTRAINT [PK_Generals] PRIMARY KEY CLUSTERED
(
    [general_name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Modifiers]    Script Date: 18.05.2023
19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Modifiers](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [name] [nvarchar](50) NULL,
    [battalion_type] [char](3) NULL,
    [property] [nvarchar](50) NULL,
    [modifier] [real] NULL,
    [path_to_icon] [nvarchar](200) NULL,
    CONSTRAINT [PK_Modifiers] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

```

```

) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Paths to Army icons]    Script Date:
18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Paths to Army icons](
    [path] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Path to Army icons] PRIMARY KEY CLUSTERED
(
    [path] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Paths to ArmyGroup icons]    Script
Date: 18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Paths to ArmyGroup icons](
    [path] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Paths to ArmyGroup icons] PRIMARY KEY CLUSTERED
(
    [path] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Paths to Division icons]    Script
Date: 18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Paths to Division icons](
    [path] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Paths to Division icons] PRIMARY KEY CLUSTERED
(
    [path] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

```

```

) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Paths to Modifiers icons]      Script
Date: 18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Paths to Modifiers icons](
    [path] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[TerrainTypes]      Script Date:
18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TerrainTypes](
    [terrain_type] [nvarchar](10) NOT NULL,
    [attack_modifier] [real] NULL,
    [combat_width] [tinyint] NULL,
    CONSTRAINT [PK_TerrainTypes] PRIMARY KEY CLUSTERED
(
    [terrain_type] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Tracks]      Script Date: 18.05.2023
19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tracks](
    [TrackName] [nvarchar](100) NOT NULL,
    [TrackPath] [nvarchar](200) NULL,
    CONSTRAINT [PK_Tracks] PRIMARY KEY CLUSTERED
(
    [TrackName] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

/***** Object: Table [dbo].[Unit-specific Attack Adjusters]
Script Date: 18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Unit-specific Attack Adjusters](
    [battalion_name] [nvarchar](50) NOT NULL,
    [forest] [real] NULL,
    [hill] [real] NULL,
    [mountains] [real] NULL,
    [plains] [real] NULL,
    [urban] [real] NULL,
    [jungle] [real] NULL,
    [marsh] [real] NULL,
    [desert] [real] NULL,
    [river] [real] NULL,
    [amphibious] [real] NULL,
    [fort] [real] NULL,
    CONSTRAINT [PK_Unit-specific Attack Adjusters] PRIMARY KEY
    CLUSTERED
    (
        [battalion_name] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
    = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
    OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Unit-specific Defense Adjusters]
Script Date: 18.05.2023 19:01:39 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Unit-specific Defense Adjusters](
    [battalion_name] [nvarchar](50) NOT NULL,
    [forest] [real] NULL,
    [hill] [real] NULL,
    [mountains] [real] NULL,
    [plains] [real] NULL,
    [urban] [real] NULL,
    [jungle] [real] NULL,
    [marsh] [real] NULL,
    [desert] [real] NULL,
    [river] [real] NULL,
    [amphibious] [real] NULL,
    [fort] [real] NULL,

```

```

CONSTRAINT [PK_Unit-specific Defense Adjusters] PRIMARY KEY
CLUSTERED
(
    [battalion_name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Contents of armies] WITH CHECK ADD CONSTRAINT
[FK_Contents of armies_Armies] FOREIGN KEY([army_name])
REFERENCES [dbo].[Armies] ([army_name])
GO
ALTER TABLE [dbo].[Contents of armies] CHECK CONSTRAINT [FK_Contents
of armies_Armies]
GO
ALTER TABLE [dbo].[Contents of armies] WITH CHECK ADD CONSTRAINT
[FK_Contents of armies_Divisions] FOREIGN KEY([division_name])
REFERENCES [dbo].[Divisions] ([division_name])
GO
ALTER TABLE [dbo].[Contents of armies] CHECK CONSTRAINT [FK_Contents
of armies_Divisions]
GO
ALTER TABLE [dbo].[Contents of army groups] WITH CHECK ADD
CONSTRAINT [FK_Contents of army groups_Armies] FOREIGN
KEY([army_name])
REFERENCES [dbo].[Armies] ([army_name])
GO
ALTER TABLE [dbo].[Contents of army groups] CHECK CONSTRAINT
[FK_Contents of army groups_Armies]
GO
ALTER TABLE [dbo].[Contents of army groups] WITH CHECK ADD
CONSTRAINT [FK_Contents of army groups_Army groups] FOREIGN
KEY([army_group])
REFERENCES [dbo].[Army groups] ([army_group_name])
GO
ALTER TABLE [dbo].[Contents of army groups] CHECK CONSTRAINT
[FK_Contents of army groups_Army groups]
GO
ALTER TABLE [dbo].[Contents of divisions] WITH CHECK ADD
CONSTRAINT [FK_Contents of divisions_Batallions] FOREIGN
KEY([battalion_name])
REFERENCES [dbo].[Battalions] ([battalion_name])
GO
ALTER TABLE [dbo].[Contents of divisions] CHECK CONSTRAINT
[FK_Contents of divisions_Batallions]
GO

```

```

ALTER TABLE [dbo].[Contents of divisions] WITH CHECK ADD
CONSTRAINT [FK_Contents of divisions_Divisions] FOREIGN
KEY([division_name])
REFERENCES [dbo].[Divisions] ([division_name])
GO
ALTER TABLE [dbo].[Contents of divisions] CHECK CONSTRAINT
[FK_Contents of divisions_Divisions]
GO
ALTER TABLE [dbo].[Generals] WITH CHECK ADD CONSTRAINT
[FK_Generals_Countries] FOREIGN KEY([country])
REFERENCES [dbo].[Countries] ([country_tag])
GO
ALTER TABLE [dbo].[Generals] CHECK CONSTRAINT
[FK_Generals_Countries]
GO
ALTER TABLE [dbo].[Unit-specific Attack Adjusters] WITH CHECK ADD
CONSTRAINT [[FK_Unit-specific Attack Batallions] FOREIGN
KEY([battalion_name])
REFERENCES [dbo].[Battalions] ([battalion_name])
GO
ALTER TABLE [dbo].[Unit-specific Attack Adjusters] CHECK CONSTRAINT
[[FK_Unit-specific Attack Batallions]
GO
ALTER TABLE [dbo].[Unit-specific Defense Adjusters] WITH CHECK ADD
CONSTRAINT [FK_Unit-specific Defense Adjusters_Batallions] FOREIGN
KEY([battalion_name])
REFERENCES [dbo].[Battalions] ([battalion_name])
GO
ALTER TABLE [dbo].[Unit-specific Defense Adjusters] CHECK CONSTRAINT
[FK_Unit-specific Defense Adjusters_Batallions]
GO
USE [master]
GO
ALTER DATABASE [ExtremeIroningDatabase] SET READ_WRITE

```

GO

## Приложение В

Листинг кода, отвечающего за сражение

```
private bool Attack(Division division)//if attacking division
was destroyed, return true

    {

        void OneAttack(Division attackingDivision, Division
defendingDivision, bool first)

            {

                int amountOfSoftAttacks =
(int)(attackingDivision.softAttack * (1 -
defendingDivision.vehicleRatio));

                int amountOfHardAttacks =
(int)(attackingDivision.hardAttack *
defendingDivision.vehicleRatio);

                int amountOfAttacks = amountOfHardAttacks +
amountOfSoftAttacks;

                amountOfAttacks = (int)(amountOfAttacks *
SelectedTerrain.attackModifier);

                double divisionAttackMultiplier = 1;

                if (attackingDivision.modifiers.Where(u =>
u.type == SelectedTerrain.type).Any())

                    {

                        divisionAttackMultiplier =
attackingDivision.modifiers.Where(u => u.type ==
SelectedTerrain.type).First().attackModifier;

                    }

                amountOfAttacks = (int)(amountOfAttacks *
divisionAttackMultiplier * SelectedTerrain.attackModifier);
```



```

        int amountOfHits;

        double def = first ? defendingDivision.defence
: defendingDivision.breakthrough;

        double divisionDefenseMultiplier = 1;

        if (defendingDivision.modifiers.Where(u =>
u.type == SelectedTerrain.type).Any())
        {

            divisionDefenseMultiplier =
defendingDivision.modifiers.Where(u => u.type ==
SelectedTerrain.type).First().attackModifier;

        }

        def = (int)(amountOfAttacks *
divisionDefenseMultiplier);

        if (amountOfAttacks > def)
        {

            amountOfHits = (int)((1 -
BASE_CHANCE_TO_AVOID_HIT) * def +

                (1 - CHANCE_TO_AVOID_HIT_AT_NO_DEF) *
(amountOfAttacks - def));

        }

        else
        {

            amountOfHits = (int)((1 -
BASE_CHANCE_TO_AVOID_HIT) * amountOfAttacks);

        }

```

```

        int healthDieSize = 2;

        int orgDieSize = attackingDivision.armor >
defendingDivision.piercing ? 6 : 4;

        int healthParts =
(int)(attackingDivision.health / attackingDivision.MaxHealth *
10);

        double attackHealthFactor = (double)healthParts
/ 10;

        double armorDamageFactor =
defendingDivision.armor > attackingDivision.piercing ?
ARMOR_DEFLECTION_FACTOR : 1;

        double mathExpectHealthDamage = (1 +
healthDieSize) / 2 * amountOfHits * attackHealthFactor *

        HEALTH_DAMAGE_MODIFIER * armorDamageFactor;

        double mathExpectOrgDamage = (1 + orgDieSize) /
2 * amountOfHits * ORG_DAMAGE_MODIFIER * armorDamageFactor;

        defendingDivision.health -=
mathExpectHealthDamage;

        defendingDivision.organization -=
mathExpectOrgDamage;

```

```

        WriteToLog($"{attackingDivision.Name}
attacks{(!first ? " back" : "")} {defendingDivision.Name},
{Math.Round(mathExpectHealthDamage, 2)} DMG to Health,
{Math.Round(mathExpectOrgDamage, 2)} DMG to Organization,
terrain: {SelectedTerrain.type}");

    }

    //determine defending division

    Division defendingDivision =
Defenders.AllDivisions[random.Next(0,
Defenders.AllDivisions.Count)];

    //divisions attack each other

    OneAttack(division, defendingDivision, true);

    OneAttack(defendingDivision, division, false);

    division.UpdateBattleStats();

    defendingDivision.UpdateBattleStats();

    //remove destroyed divisions

    if (defendingDivision.health < 0 ||
defendingDivision.organization < 0)

Defenders.AllDivisions.Remove(defendingDivision);

    if (division.health < 0 || division.organization <
0)

        return true;

```

```
        else return false;

    }

    public void StartBattle()

    {

        ContinueBattle = true;

        LogText = string.Empty;

        OnPropertyChanged(nameof(LogText));

OnPropertyChanged(nameof(StartBattleButtonVisibility));

OnPropertyChanged(nameof(ResetBattleButtonVisibility));

        Battle();

    }

    const string logFilePath = "./logFile";

    private readonly FileStream logFileStream =
File.Create(logFilePath);

    public async void Battle()
```

```

    {

        logFileStream.Close();

        LogText = string.Empty;

        while (Attackers.AllDivisions.Count > 0 &&
Defenders.AllDivisions.Count > 0 && ContinueBattle)

            {

                for (int i = 0, frontCapacity = 0; i <
Attackers.AllDivisions.Count; i++)

                    {

                        if (frontCapacity <
SelectedTerrain.combatWidth)

                            {

                                if (Defenders.AllDivisions.Count < 1)
break;

                                if (Attack(Attackers.AllDivisions[i]))
Attackers.AllDivisions.RemoveAt(i--); //if division was
destroyed,

//remove it

                                else frontCapacity +=
Attackers.AllDivisions[i].frontWidth;

                                    }

                                else

                                    {

```

```
                break;

            }

        }

        await Task.Delay(Speed);

    }

    LogText = File.ReadAllText(logFilePath);

    if (ContinueBattle) LogText +=
Attackers.AllDivisions.Count > 0 ? "Attackers won\n" :
"Defenders won\n";

    ResetBattleCommand.Execute(null);

}
```