

**Министерство науки и высшего образования Российской Федерации**

**Рязанский государственный радиотехнический университет  
им. В.Ф. Уткина**

**Кафедра «Вычислительная и прикладная математика» (ВПМ)**

**Техническое задание к курсовому проекту**

на тему

**«Игровая программа: Battle City»**

по курсу

**«Конструирование программного обеспечения»**

Аналитик:

студент группы 143

Соколов Е.А.

Разработчик:

студент группы 143

Соколов Е.А.

Проверил:

ст. преподаватель кафедры ВПМ

Васин А.О.

Рязань, 2024

## Оглавление

1	Версии документа и изменения .....	3
2	Термины и определения .....	3
3	Общее описание задачи .....	3
3.1	Описание требуемой функциональности .....	4
3.1.1	Общее описание функциональности .....	4
3.1.2	Архитектура приложения .....	6
3.1.3	Разработка интерфейса приложения .....	7
3.1.4	Алгоритмы и логика работы приложения .....	12
Приложения .....		15

## 1 Версии документа и изменения

Версия	Дата	Описание изменений	Автор
1.0	23.10.2024	Создан	Соколов Е.А.
1.1	23.10.2024	Добавлены термины и определения, а также заполнен раздел с описанием требуемой функциональности	Соколов Е.А.
1.2	23.10.2024	Добавлена первоначальная версия графического интерфейса игрового поля	Соколов Е.А.
1.2.1	24.10.2024	Добавлены начальные эскизы интерфейса программы	Соколов Е.А.
1.3	04.11.2024	Эскизы интерфейса готовы, добавлена полная совокупная диаграмма взаимодействия	Соколов Е.А.
1.4	05.11.2024	Добавлено высокоуровневое описание архитектуры приложения	Соколов Е.А.
1.5	05.11.2024	Добавлены алгоритмы и приложения, исправления опечаток в тексте документа	Соколов Е.А.
2.0	08.01.2025	Исправлены описания функциональности и архитектуры приложения, несколько изменены эскизы меню игры и совокупная диаграмма взаимодействия, исправлены алгоритмы. Добавлен раздел с описанием снарядов.	Соколов Е.А.
2.1	09.01.2025	Исправлены опечатки в тексте и схемах	Соколов Е.А.

## 2 Термины и определения

**Игра** – программное обеспечение для игры «Battle City», разрабатываемое в рамках данной курсовой работы;  
**Игровое поле** – пространство, где происходит действие игры. Представлено клетками в виде прямоугольной сетки;

**Карта** – конкретная уникальная конфигурация игрового поля;

**Пиксель** – мера расстояния в игре;

**Блок, клетка** – минимальная квадратная часть игрового поля, в которую может поместиться препятствие. Размер клетки 16 на 16 пикселей;

**Тик** – минимальный отрезок времени в игре;

**Север, юг, запад, восток** – соответственно верхняя, нижняя, левая и правая части игрового поля;

**Танк** – управляемый игроком (собственный) или компьютером (вражеский) объект, который перемещается по игровому полю и способен выпускать снаряды;

**Снаряды (пули)** – объекты, выпускаемые танками во время стрельбы. Могут поражать другие танки или препятствия;

**Препятствие** – объект на поле, который может блокировать движение танков и/или их выстрелы;

**База** – объект, который нужно защищать танку игрока;

**Уровень** – логически завершенная часть игры, представляющее собой карту с уникальными относительно других уровней расположением, количеством и характеристиками объектов игры;

**Здоровье** – характеристика танка, отражающая степень его устойчивости к повреждениям, прежде чем танк будет уничтожен;

**Количество жизней** – характеристика танка Игрока, уменьшаемая при уничтожении танка. При равенстве ее нулю Игрок теряет возможность продолжать Игру, иначе танк восстанавливается;

**Спаун (спавн)** – точка на карте, в которой может появиться танк игрока после того, как он был уничтожен.

## 3 Общее описание задачи

В данном разделе рассматриваются вопросы функциональности, архитектуры, внешнего вида и основных алгоритмов работы разрабатываемой Игры.

## **3.1 Описание требуемой функциональности**

### **3.1.1 Общее описание функциональности**

Пользователь Игры «Battle City» управляет танком с возможностью перемещаться по игровому полю, которое состоит из клеток. Танки могут двигаться в четырех направлениях: вверх, вниз, влево и вправо, при этом поворот танка всегда происходит на угол, кратный 90 градусам. Основной задачей Игрока является защита своей базы, обозначенной на карте, от вражеских танков, которые также перемещаются по полю и могут атаковать. Игрок должен уничтожать вражеские танки снарядами, при этом стараться избежать их атак, чтобы не потерять свои жизни. При уничтожении всех вражеских танков Пользователь попадает на новый уровень Игры. Игра завершается либо победой Игрока, если все вражеские танки на последнем уровне уничтожены, либо его поражением, если на одном из уровней база Игрока была разрушена или он потерял все свои жизни.

#### **3.1.1.1 Описание объектов игры**

##### **3.1.1.1.1 Игровое поле**

Игровое поле состоит из 50 блоков по горизонтали и 40 по вертикали. Блок может быть либо пустым, либо содержать один из блоков (препятствие или базу).

##### **3.1.1.1.2 Препятствия**

Могут мешать стрельбе либо перемещению танков. Препятствия бывают разрушаемыми и не разрушаемыми. Первые могут быть частично или полностью разрушены снарядами, вторые – нет. До уничтожения разрушаемых препятствий их блоки ловят снаряды и не пропускают их дальше. Также препятствия бывают проходимыми и непроходимыми для танка.

Виды препятствий в Игре:

- кирпичная стена – разрушаемый объект, до уничтожения непроходим для танка,
- неразрушимая стена – неуничтожимый непроходимый объект,
- кусты – неуничтожимый проходимый блок. Снаряды пропускает свободно,
- вода – неуничтожимый и непроходимый блок. Снаряды пропускает свободно.

##### **3.1.1.1.3 База**

База – частный случай препятствия, является непроходимым уничтожимым блоком, при разрушении всех его экземпляров на игровом поле Пользователь терпит поражение в Игре. База может быть уничтожена одним снарядом, в том числе и «дружественным огнем» самого пользователя.

##### **3.1.1.1.4 Снаряд (пуля)**

Игровой объект, выпускаемый танком. Имеет заданную скорость и наносимый урон. Двигается по прямой линии в выбранном направлении. Снаряды могут попадать как в другие танки, так и в различные препятствия на карте. В случае попадания в другой танк (коллизии с родительским танком не проверяются) у последнего отнимается здоровье на величину урона, а снаряд уничтожается. В случае попадания в разрушимый блок он и снаряд уничтожаются. Снаряд может пролетать через блоки кустов и воды без их повреждения. Коллизии снарядов друг с другом не проверяются: пули могут одновременно пролетать через одинаковые координаты.

##### **3.1.1.1.5 Танк**

Основная боевая единица, может управляться пользователем либо компьютером. Танк может двигаться по игровому полю с определенной скоростью, а также поворачиваться и стрелять с частотой, не превышающей максимальную допустимую для данного вида танка. За границу игрового поля танк заехать не может.

Каждый выстрел блокирует возможность нового выстрела на некоторое время (в том числе существует ограничение на максимальное количество еще летящих снарядов, выпущенных данным танком).

В Игре присутствует 4 вида танков:

- **обычный** – стандартная скорость, 100 единиц здоровья, может выпускать не более 1 снаряда до его попадания, стандартный урон,
- **скоростной** – скорость выше обычного на 25%, в остальном ведет себя как обычный танк,
- **скорострельный** – может выпускать дополнительный снаряд к первому до попадания первого боеприпаса, максимальная частота стрельбы увеличена до 10 раз в секунду, в остальном ведет себя как обычный танк,
- **бронированный** – скорость 75% от стандартной, 200 единиц здоровья, урон на 50% выше стандартного, в остальном ведет себя как обычный танк.

Стандартная скорость равна 40 пикселям в секунду, урон равен 10 единицам. Запасы снарядов у танков не ограничены. Танки могут поворачиваться и стрелять с частотой не выше, чем 2 раза в секунду (у танка Игрока нет ограничений на частоту поворота).

### 3.1.1.1.6 Вспомогательные элементы

К вспомогательным элементам игрового процесса относят счетчик очков пользователя, количества его жизней, здоровья, и текущего уровня.

### 3.1.1.2 Описание игрового процесса

Пользователь управляет танком с помощью клавиш «стрелки» и *w a s d* для движения соответственно вверх, вниз, влево и вправо по игровому полю, а также пробела для стрельбы. При отпускании клавиш перемещения танка он останавливается. Танк может стрелять снарядами, которые движутся по прямой линии в выбранном направлении. Снаряды могут попадать как во вражеские танки, так и в различные препятствия на карте. Скорострельность танка ограничена, как и его скорость. Тип танка игрока – обычный. Танки противника управляются компьютером. В случае попадания в танк его здоровье уменьшается; если оно опустилось до 0, танк уничтожается. Если уничтожается танк игрока, то при наличии хотя бы одной жизни танк восстанавливается на точке спауна с характеристиками по умолчанию, а его количество жизней уменьшается на один. За каждый уничтоженный танк противника пользователь получает определенное количество очков (в зависимости от вида уничтоженного танка, 100 для обычного, 125 скоростного или скорострельного, 150 для бронированного). При уничтожении всех танков на уровне Игрок переходит на следующий уровень.

Игра завершается в одном из следующих случаев:

**Победа игрока:** на последнем уровне все вражеские танки уничтожены, причем база Игрока осталась целой.

**Поражение игрока:** на одном из уровней либо база Игрока оказалась разрушена, либо Игрок потерял все жизни.

В случае завершения Игры выводится информационное сообщение с информацией о победе или поражении Игрока. В случае победы пользователя система предлагает ему зарегистрировать свое имя в списках рекордов, куда попадает информация и о количестве набранных им очков.

### 3.1.1.3 Режимы работы Игры

Игроку доступен запуск Игры в нескольких режимах сложности:

- **легкий** – у танка пользователя 3 жизни, 200 единиц здоровья, скорость увеличена на 50%, все танки противника – обычные,
- **обычный** – у танка пользователя 2 жизни, 100 единиц здоровья, скорость увеличена на 25%, танки противников могут быть любого типа и на уровне их больше,
- **сложный** – у танка пользователя 1 жизнь, 100 единиц здоровья, танки противников могут быть любого типа и на уровне их еще больше.

### 3.1.1.4 Дополнительные возможности Игры

Игра должна предоставлять пользователю возможность:

- просматривать справку о правилах Игры и управления танком,
- просматривать таблицу рекордов Игры (топ 10 лучших результатов),
- досрочно выходить из Игры в главное меню при помощи нажатия клавиши ESC,
- при досрочном выходе из Игры ее состояние не должно утрачиваться и при попытке запуска нового игрового процесса Пользователь должен иметь возможность не только начать новую Игру, но продолжить прежнюю.

### 3.1.2 Архитектура приложения

В основе архитектуры разрабатываемого приложения лежит шаблон проектирования MVC, состоящий из:

- model (модель) – класс, ответственный за логику работы Игры, существует в единственном экземпляре. Модель включает в себя как движок Игры, так и вспомогательные утилиты (классы для работы с рекордами, загрузки уровней), модели различных для меню, ресурсы уровней,
- view (вид/представление) – класс, отображающий пользователю текущее состояние модели. У каждого меню существует два вида: консольный и графический (используется технология WPF),
- controller (контроллер) – класс, позволяющий пользователю изменять состояние модели.

Дополнительно стоит отметить, что:

- старт программы начинается с запуска контроллера, который создаёт свое представление, которое уже и создает окно, показывает его пользователю.
- вид знает о существовании модели, а модель не является зависимой от других классов,
- модель создает события, на которые подписывается вид,
- для каждого окна приложения реализован своя собственная реализация шаблона модель-представление-контроллер. При переходе с одного экрана Игры на другой контроллер первого экрана передает управление контроллеру второго.

Высокоуровневая схема архитектуры приложения представлена на рисунке 1.

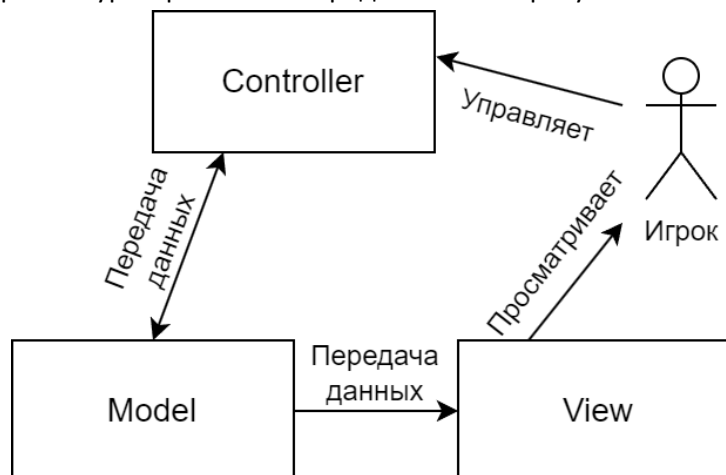


Рисунок 1 – Высокоуровневая схема архитектуры приложения

Диаграмма последовательности, отражающая данный подход на примере главного меню, представлена на рисунке 2.

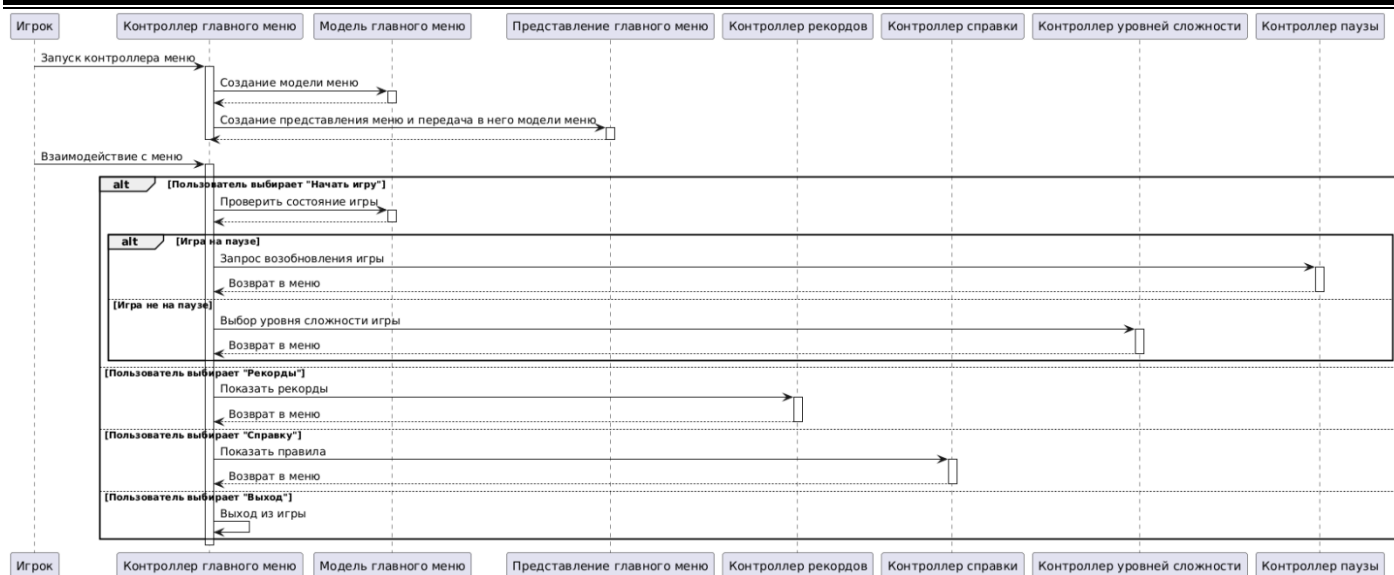


Рисунок 2 – Алгоритм работы главного меню приложения

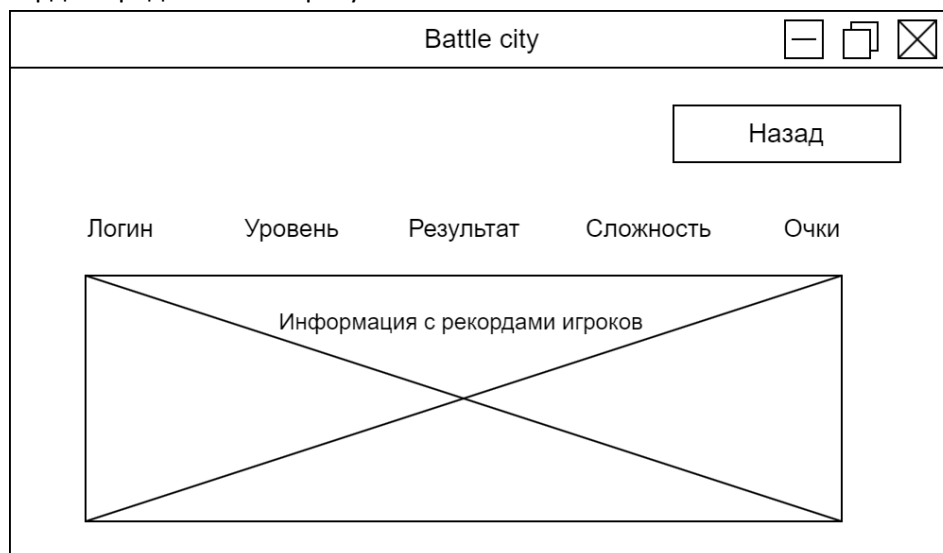
### 3.1.3 Разработка интерфейса приложения

Эскиз главного меню приложения представлен на рисунке 3.



Рисунок 3 – Главное меню Игры

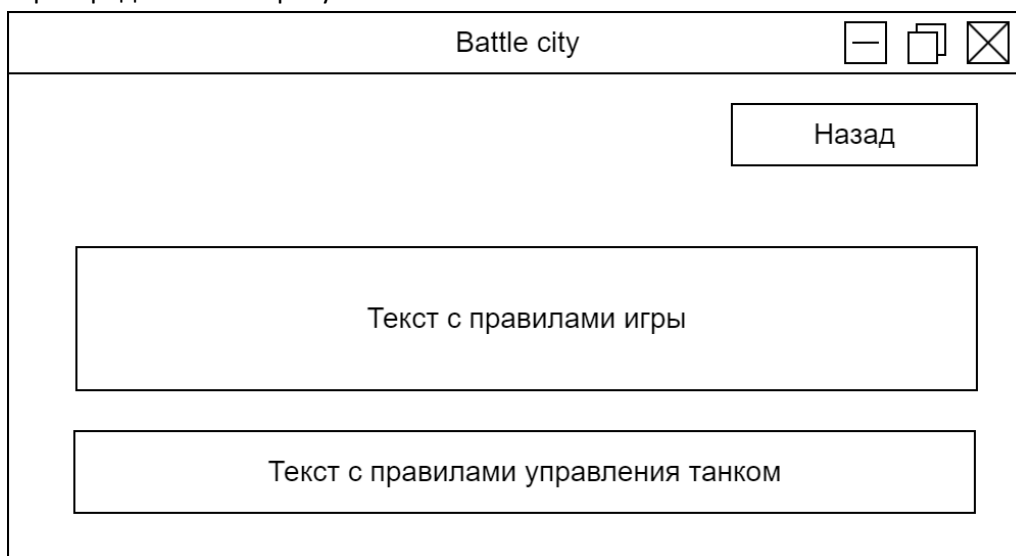
Эскиз таблицы рекордов представлен на рисунке 4.



The sketch shows a window titled "Battle city" with standard window controls (minimize, maximize, close). Inside the window, there is a "Назад" (Back) button in the top right corner. Below the button, there are five column headers: "Логин" (Login), "Уровень" (Level), "Результат" (Result), "Сложность" (Difficulty), and "Очки" (Points). In the center of the window, there is a large rectangular area with a diagonal cross, labeled "Информация с рекордами игроков" (Information with player records).

Рисунок 4 – Таблица рекордов

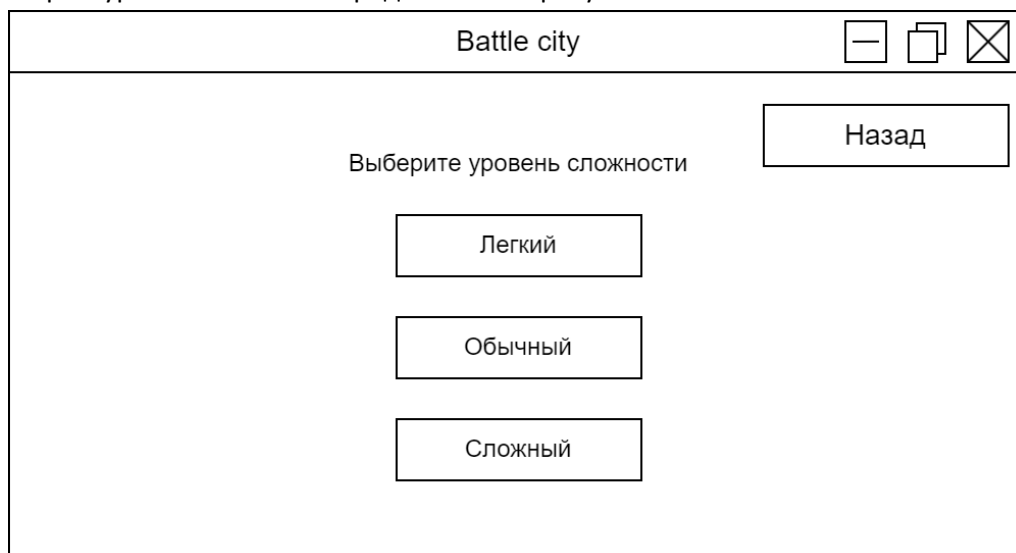
Эскиз справки Игры представлен на рисунке 5.



The sketch shows a window titled "Battle city" with standard window controls (minimize, maximize, close). Inside the window, there is a "Назад" (Back) button in the top right corner. Below the button, there are two large rectangular text areas. The first area is labeled "Текст с правилами игры" (Text with game rules) and the second area is labeled "Текст с правилами управления танком" (Text with tank control rules).

Рисунок 5 – Справка Игры

Эскиз окна с выбором уровня сложности представлен на рисунке 6.



The sketch shows a window titled "Battle city" with standard window controls (minimize, maximize, close). Inside the window, there is a "Назад" (Back) button in the top right corner. Below the button, there is a label "Выберите уровень сложности" (Select difficulty level). Underneath the label, there are three buttons stacked vertically, labeled "Легкий" (Easy), "Обычный" (Normal), and "Сложный" (Hard).

Рисунок 6 – Окно с выбором уровня сложности



Эскиз окна с игровым полем представлен на рисунке 7.

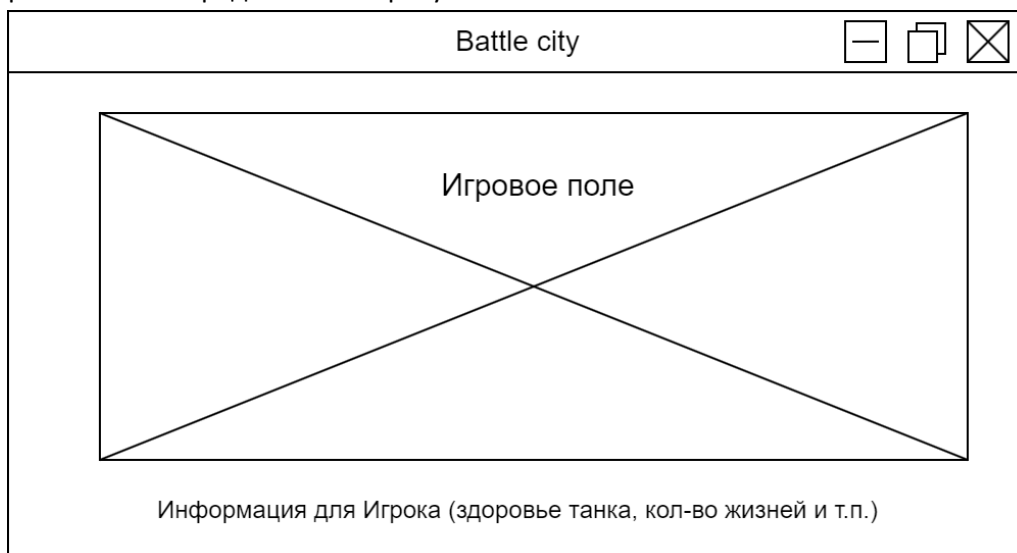


Рисунок 7 – Эскиз окна с игровым полем

Эскиз окна с паузой представлен на рисунке 8.

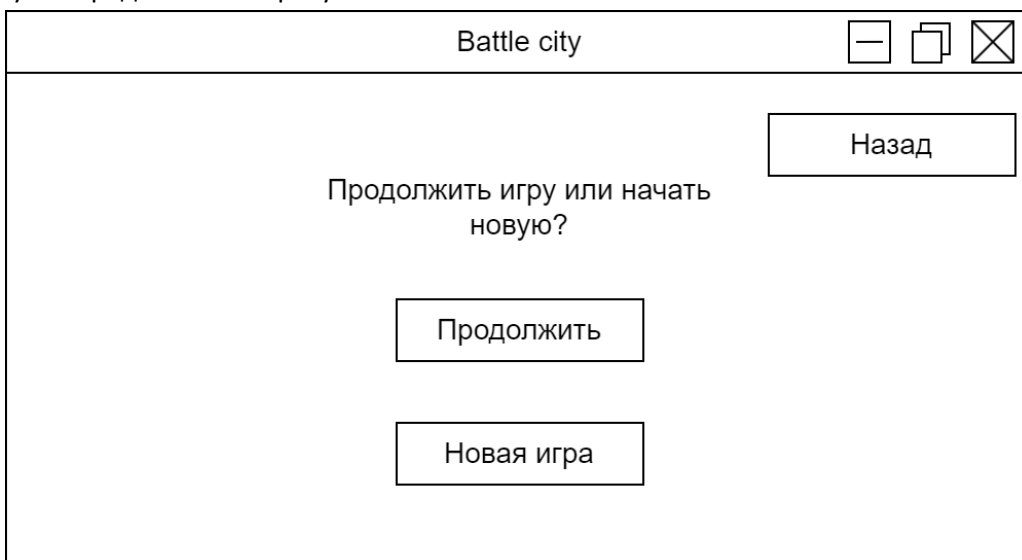
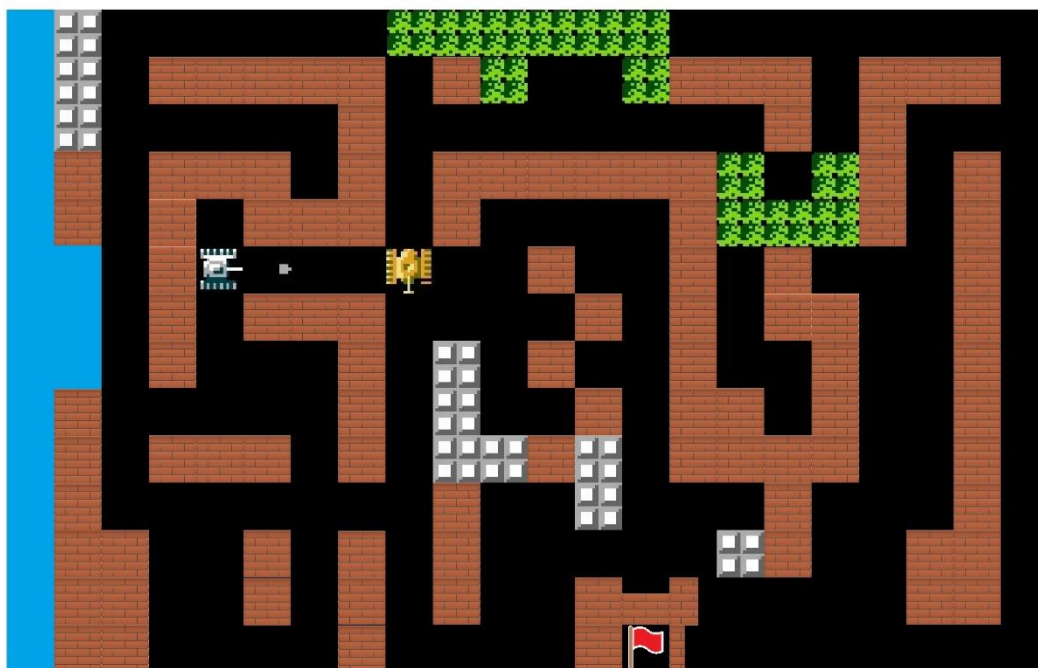


Рисунок 8 – Эскиз окна с паузой

Эскиз самого игрового поля представлен на рисунке 9.



Здоровье = 80, Жизней = 1, Уровень = First, Счет = 200

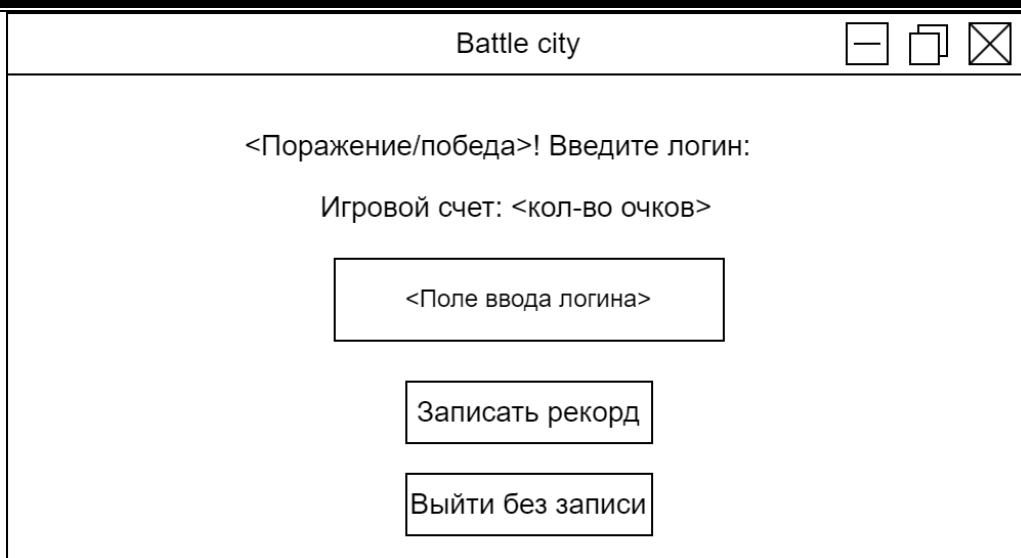
Рисунок 9 – Эскиз игрового поля

Эскизы элементов игрового поля представлены на рисунке 10.



Рисунок 10 – Эскизы элементов игрового поля

Эскиз окна с информацией о результате завершения Игры и предложением сохранить результаты представлен на рисунке 11.



The screenshot shows a window titled "Battle city" with standard Windows window controls (minimize, maximize, close). The main content area displays the following text and controls:

<Поражение/победа>! Введите логин:

Игровой счет: <кол-во очков>

<Поле ввода логина>

Записать рекорд

Выйти без записи

Рисунок 11 – Окно с информацией о результате завершения Игры

Полная совокупная диаграмма взаимодействия представлена на рисунке 12.

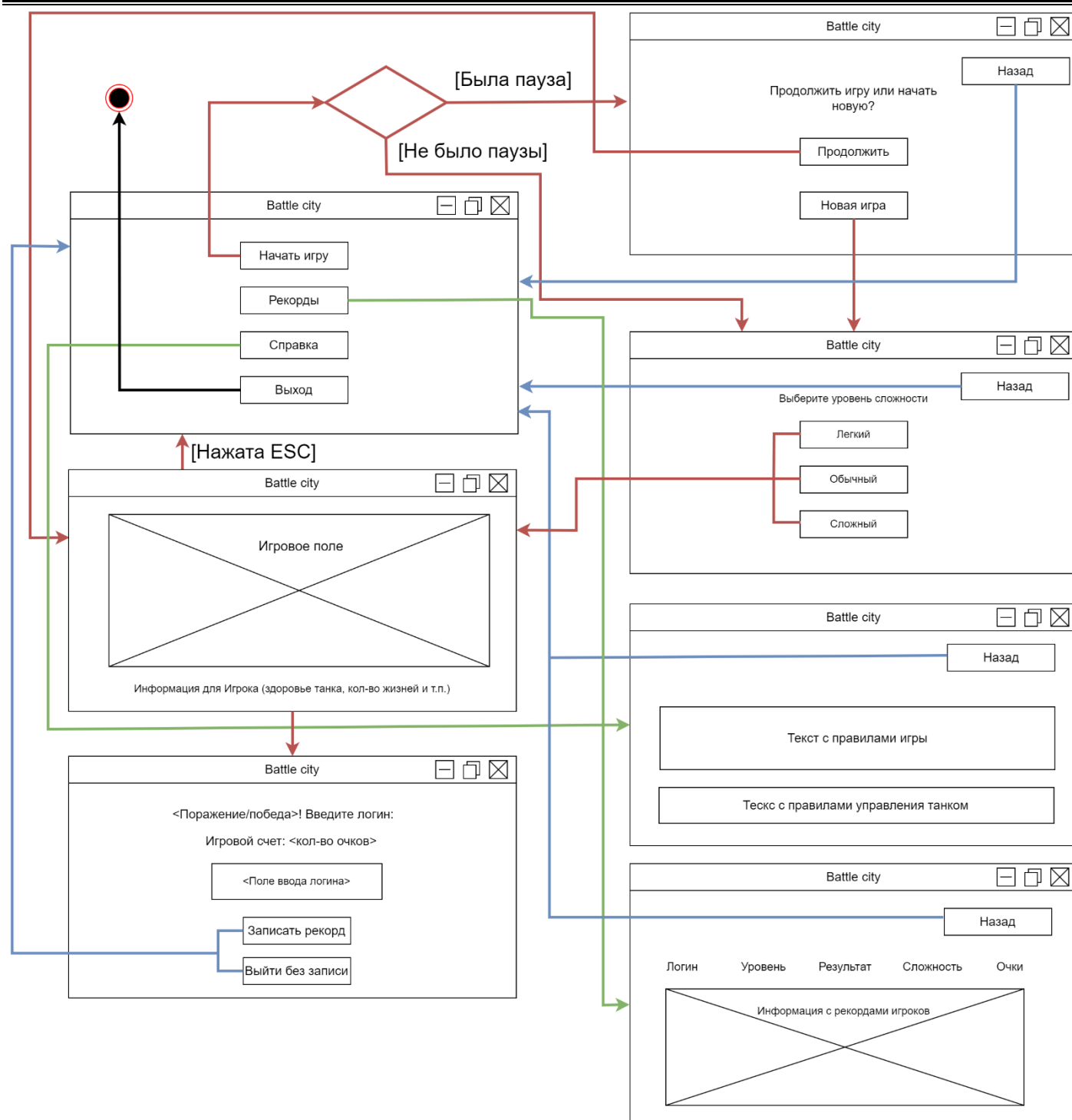


Рисунок 12 – Полная совокупная диаграмма взаимодействия для Игры

### 3.1.4 Алгоритмы и логика работы приложения

#### 3.1.4.1 Алгоритм интеллекта вражеских танков

Вражеский танк стремится уничтожить игрока или базу. Он вычисляет кратчайший путь до танка Игрока и начинает двигаться к нему, периодически проверяя досягаемость цели (танка Игрока или блока базы) для снарядов. Если цель может быть поражена, танк открывает огонь. Если цель не находится в зоне досягаемости, то танк с вероятностью 25% на данном шаге может переключиться на новую цель (приоритетно на танк Игрока). Данный алгоритм в виде диаграммы последовательности представлен на рисунке 13.

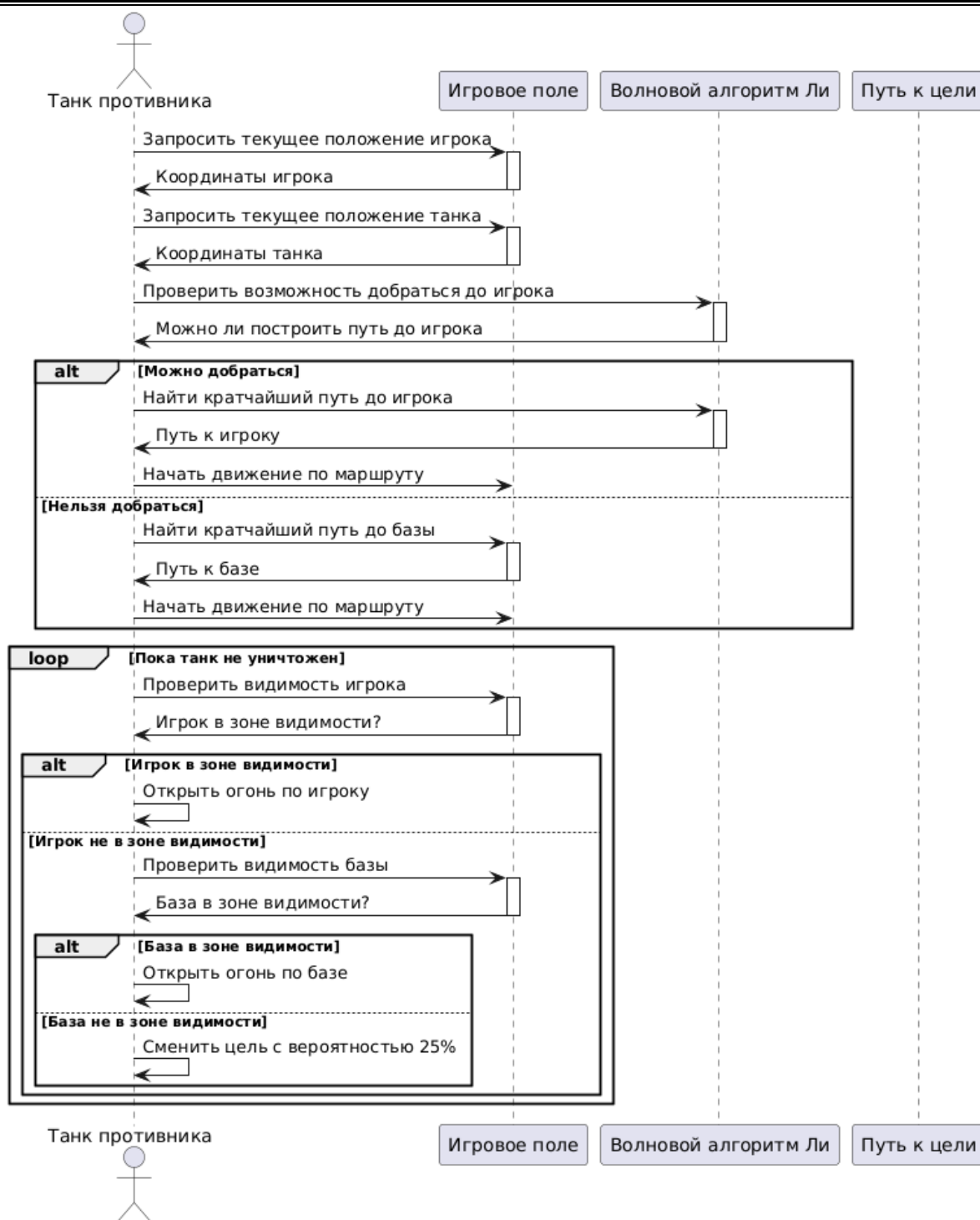


Рисунок 13 – Алгоритм интеллекта вражеских танков

### 3.1.4.2 Поиск кратчайшего пути до базы

Для поиска танками противника кратчайшего пути до базы используется волновой алгоритм Ли, работающий по принципу постепенного «распространения волны» от стартовой точки (где располагается танк) к целевой (базе), заполняя доступные ячейки шаг за шагом, пока не достигнет цели или не заполнит все возможные пути. Сам алгоритм можно кратко описать следующим образом:

- 1) инициализация: всем ячейкам игрового поля соответствует значение -1, также задается начальная ячейка, ей присваивается значение 0,
- 2) распространение волны: на каждом шаге алгоритм помечает соседние ячейки, которые не были еще посещены (т.е. им соответствует -1) значением, увеличенным на 1 по сравнению с текущей,

- 3) проверка завершения и восстановление пути: если числовая «волна» достигла целевой точки, то путь существует; восстановить его кратчайшую версию можно, двигаясь в обратном направлении по ячейкам с убывающими значениями.

Блок-схема алгоритма поиска кратчайшего пути танками противника представлена на рисунке 14.

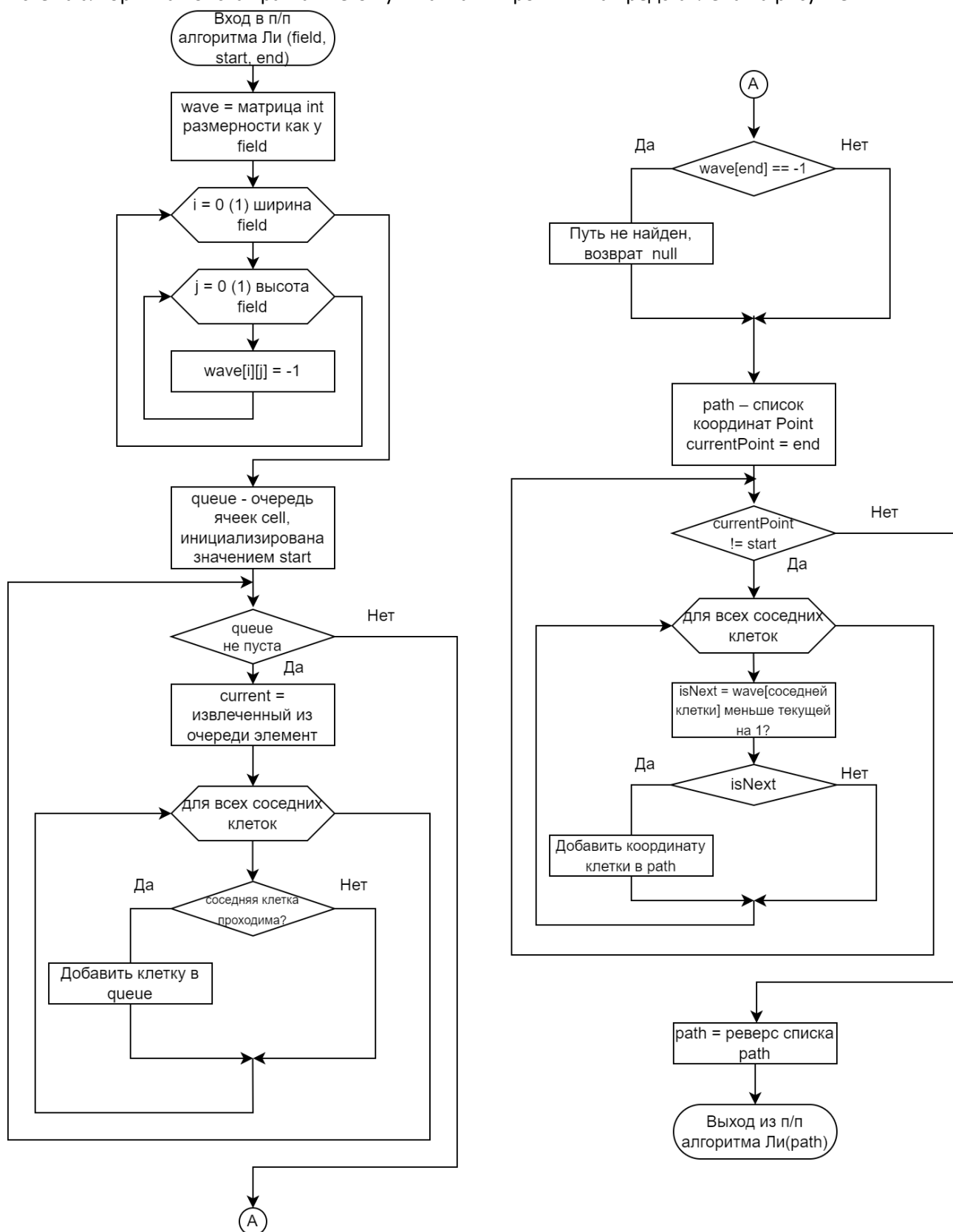


Рисунок 14 – Волновой алгоритм Ли

### 3.1.4.3 Алгоритм взаимодействия снаряда с объектами на игровой карте

Снаряды, выпущенные танком, могут поражать другие танки, базу и разрушимые препятствия. Некоторые неразрушимые препятствия снаряд может преодолеть, а некоторые – нет. При встрече с любым непреодолимым препятствием, а также танком или базой снаряд уничтожается, нанося урон (если снаряд попал в блок и этот блок разрушимый, то он уничтожается от первого попадания пули, а если попал в танк, то у него отнимается здоровье на величину урона, хранимого в объекте пули). Снаряды никак не взаимодействуют с друг с другом.

Алгоритм взаимодействия снаряда с объектами на игровой карте в виде диаграммы последовательности представлен на рисунке 15.

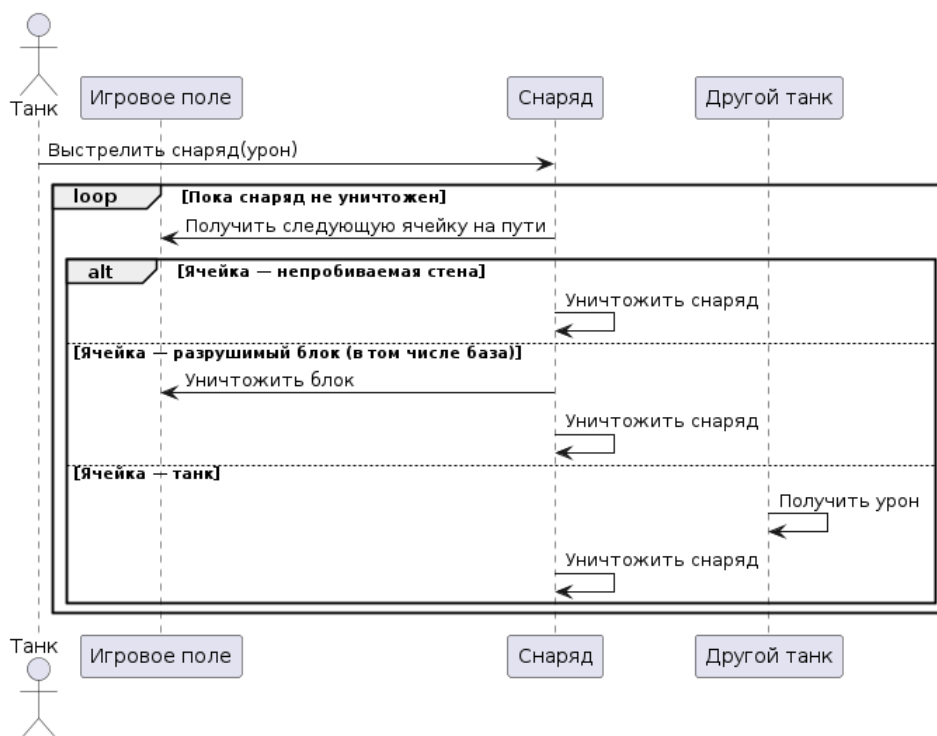


Рисунок 15 – Алгоритм взаимодействия снаряда с объектами на игровой карте

## Приложения

### Исходный код диаграммы последовательности для главного меню приложения

```

@startuml
participant "Игрок" as User
participant "Контроллер главного меню" as MenuController
participant "Модель главного меню" as Menu
participant "Представление главного меню" as MenuView
participant "Контроллер рекордов" as AchievementController
participant "Контроллер справки" as HelpController
participant "Контроллер уровней сложности" as GameController
participant "Контроллер паузы" as PauseController
    
```

```

User -> MenuController: Запуск контроллера меню
activate MenuController
MenuController -> Menu: Создание модели меню
activate Menu
return
    
```

MenuController -> MenuView: Создание представления меню и передача в него модели меню

```
activate MenuView
return
```

deactivate MenuController

User -> MenuController: Взаимодействие с меню

activate MenuController

alt Пользователь выбирает "Начать игру"

MenuController -> Menu: Проверить состояние игры

activate Menu

return

alt Игра на паузе

MenuController -> PauseController: Запрос возобновления игры

activate PauseController

PauseController --> MenuController: Возврат в меню

deactivate PauseController

else Игра не на паузе

MenuController -> GameController: Выбор уровня сложности игры

activate GameController

GameController --> MenuController: Возврат в меню

deactivate GameController

end

else Пользователь выбирает "Рекорды"

MenuController -> AchievementController: Показать рекорды

activate AchievementController

AchievementController --> MenuController: Возврат в меню

deactivate AchievementController

else Пользователь выбирает "Справку"

MenuController -> HelpController: Показать правила

activate HelpController

HelpController --> MenuController: Возврат в меню

deactivate HelpController

else Пользователь выбирает "Выход"

MenuController -> MenuController: Выход из игры

end

deactivate MenuController

@enduml

### Исходный код диаграммы ИИ танка

@startuml

actor "Танк противника" as EnemyTank

participant "Игровое поле" as Field

participant "Волновой алгоритм Ли" as Algorithm

participant "Путь к цели" as Path

EnemyTank -> Field: Запросить текущее положение игрока

activate Field

Field -> EnemyTank: Координаты игрока

deactivate Field

EnemyTank -> Field: Запросить текущее положение танка



```
activate Field
Field -> EnemyTank: Координаты танка
deactivate Field

EnemyTank -> Algorithm: Проверить возможность добраться до игрока
activate Algorithm
Algorithm -> EnemyTank: Можно ли построить путь до игрока
deactivate Algorithm

alt Можно добраться
    EnemyTank -> Algorithm: Найти кратчайший путь до игрока
    activate Algorithm
    Algorithm -> EnemyTank: Путь к игроку
    deactivate Algorithm
    EnemyTank -> Field: Начать движение по маршруту
else Нельзя добраться
    EnemyTank -> Field: Найти кратчайший путь до базы
    activate Field
    Field -> EnemyTank: Путь к базе
    deactivate Field
    EnemyTank -> Field: Начать движение по маршруту
end

loop Пока танк не уничтожен
    EnemyTank -> Field: Проверить видимость игрока
    activate Field
    Field -> EnemyTank: Игрок в зоне видимости?
    deactivate Field
    alt Игрок в зоне видимости
        EnemyTank -> EnemyTank: Открыть огонь по игроку
    else Игрок не в зоне видимости
        EnemyTank -> Field: Проверить видимость базы
        activate Field
        Field -> EnemyTank: База в зоне видимости?
        deactivate Field
        alt База в зоне видимости
            EnemyTank -> EnemyTank: Открыть огонь по базе
        else База не в зоне видимости
            EnemyTank -> EnemyTank: Сменить цель с вероятностью 25%
        end
    end
end
end
@enduml
```

**Исходный код диаграммы поиска кратчайшего пути до базы:**

```
@startuml
actor "Танк противника"
participant "Игровое поле" as Field
participant "Волновой алгоритм Ли" as Algorithm
participant "Путь к базе" as Path

"Танк противника" -> Field: Запросить текущее положение базы
```

```
activate Field
Field -> "Танк противника": Координаты базы
deactivate Field

"Танк противника" -> Field: Запросить текущее положение танка
activate Field
Field -> "Танк противника": Координаты танка
deactivate Field

"Танк противника" -> Algorithm: Запуск волнового алгоритма\n(начало с текущих
координат)
activate Algorithm

loop Пока путь не найден
    Algorithm -> Field: Получить соседние ячейки
    activate Field
    Field -> Algorithm: Соседние ячейки
    deactivate Field

    alt Если соседняя ячейка достижима
        Algorithm -> Field: Отметить ячейку с новым шагом волны
        Algorithm -> Algorithm: Добавить ячейку в очередь на обработку
    else Если ячейка непроходима (стена, препятствие)
        Algorithm -> Algorithm: Игнорировать ячейку
    end
end

Algorithm -> "Path": Вернуть кратчайший путь к базе
deactivate Algorithm
"Танк противника" -> "Path": Получить путь к базе
activate Path
"Path" -> "Танк противника": Список координат для перемещения
deactivate Path
"Танк противника" -> Field: Начать движение по маршруту
@enduml
```

**Исходный код диаграммы взаимодействия снаряда с объектами игрового поля:**

```
@startuml
actor "Танк" as Player
participant "Игровое поле" as Field
participant "Снаряд" as Projectile
participant "Другой танк" as Enemy

Player -> Projectile: Выстрелить снаряд(урон)

loop Пока снаряд не уничтожен
    Projectile -> Field: Получить следующую ячейку на пути

    alt Ячейка – непробиваемая стена
        Projectile -> Projectile: Уничтожить снаряд
    else Ячейка – разрушимый блок (в том числе база)
        Projectile -> Field: Уничтожить блок
    end
end
```

---

```
    Projectile -> Projectile: Уничтожить снаряд
else Ячейка — танк
    Enemy -> Enemy: Получить урон
    Projectile -> Projectile: Уничтожить снаряд
end
end
@enduml
```