**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

**«Tough growth»**

по курсу

**«Конструирование программного обеспечения»**

Аналитик:

студент группы 9413

Соколова Ю. А.

Разработчик:

студент группы 9413

Соколова Ю. А.

Проверил:

доцент кафедры ВПМ

Столчнев В. К.

Рязань, 2022

Оглавление

[1 Версии документа и изменения 3](#_Toc272312955)

[2 Термины и определения 3](#_Toc272312956)

[3 Общее описание задачи 3](#_Toc272312957)

[3.1 Описание требуемой функциональности 3](#_Toc272312958)

[3.1.1 Архитектура приложения 3](#_Toc272312959)

[3.1.2 Разработка интерфейса приложения 3](#_Toc272312960)

[3.1.3 Алгоритмы и логика работы приложения 3](#_Toc272312961)

[Приложения 3](#_Toc272312962)

# Версии документа и изменения

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Описание изменений | Автор |
| 1.0 | 10.10.2022 | Создан | Соколова Ю. А. |
| 1.1 | 10.10.2022 | Добавлены основные термины и определения, общее описание задачи и интерфейс программы. | Соколова Ю. А. |
| 1.2 | 23.10.2022 | Добавление описания требуемой функциональности. | Соколова Ю. А. |
| 1.3 | 24.10.2022 | Обновление описания требуемой функциональности. | Соколова Ю. А. |
| 1.4 | 25.20.2022 | Добавление описания архитектуры, алгоритмов и логики работы приложения. | Соколова Ю. А. |
| 1.5 | 26.10.2022 | Добавление приложения. | Соколова Ю. А. |

# Термины и определения

[сюда можно добавлять различные непонятные термины и определения, чтобы было понятно, о чём идёт речь при прочтении документа]

Tough Growth – игра, где все игровые объекты являются простыми геометрическими фигурами.

Термины, использующиеся для описания процесса игры:

* игровой квадрат - квадрат, которым управляет пользователь;
* постоянный съедобный квадрат - квадрат, который всегда доступен пользователю для «съедения»;
* шестиугольник, круг, треугольник, прямоугольник – игровые объекты, которые предоставляют препятствия игровому квадрату до момента их «съедения»;
* стрелка, выстрел – препятствия, генерируемые игровыми объектами (круг выпускает стрелку, шестиугольник – короткий выстрел, треугольник – длинный выстрел), которые приводят к смерти игрового квадрата;
* размер – под этим термином подразумевается площадь игровых объектов;
* короткий выстрел – это короткая линия, быстро проходящая через игровое поле;
* длинный выстрел – это длинная линия, медленно проходящая через игровое поле;
* съесть - пересечение игрового квадрата с игровым объектом меньшего размера;
* смерть игрового квадрата – начало уровня заново (если уровень больше 8, то начало предыдущего уровня);

# Общее описание задачи

[В данном разделе необходимо дать общее описание документа]

В данном документе описываются основные алгоритмы и принципы работы игры Tough Growth для разработки игровой программы.

Описание игры.

В игре есть 10 уровней. Задача пользователя – пройти все уровни игры и набрать наименьшее количество «смертей» при условии, что на каждом следующем уровне находится всё большее количество препятствий. «Смерть» игрового квадрата не завершает игру, а заставляет пользователя начать неудачный уровень заново до 8 уровня. Как только пользователь достигнет 8 уровень, при каждой «смерти» его будет отбрасывать на 1 уровень назад.

Чтобы перейти на следующий уровень игровому квадрату необходимо «съесть» все игровые объекты, находящиеся на игровом поле. При этом игровой квадрат не сможет «съесть» другой игровой объект, если его размеры меньше игрового объекта. Для набора размеров пользователю всегда доступен «постоянный съедобный квадрат».

Все игровые объекты, кроме квадратов, создают пользователю различные препятствия:

* шестиугольник – раз в 2 секунды делает короткий выстрел в ту точку, в которой находился игровой квадрат в момент начала выстрела;
* круг – выпускает стрелку, которая преследует игровой квадрат с задержкой в 1 секунду. Если игровой квадрат будет стоять на одном месте, стрелка догонит и разобьёт его (наступит «смерть» игрового квадрата). Стрелка исчезнет только тогда, когда круг, выпустивший её, будет съеден;
* треугольник – каждую секунду делает длинный выстрел в ту точку, в которой находился игровой квадрат в момент начала выстрела;
* прямоугольник – перемещается вдоль оси, на которой он расположен. Является препятствием до тех пор, пока его размер больше размера игрового квадрата.

Смерть игрового квадрата наступает в следующих случаях:

* если игровой квадрат столкнулся с игровым объектом больших размеров (в этом случае можно считать игровой объект препятствием);
* если игровой квадрат столкнулся с препятствием (преследующая стрелка круга, выстрел шестиугольника или выстрел треугольника).

## Описание требуемой функциональности

[В данном разделе и его подразделах необходимо описать требуемую функциональность]

1. Функциональность, связанная с управлением игрового приложения.

Управление игровым приложением.

Управление меню игрового приложения осуществляется с помощью стрелок клавиатуры «Вверх», «Вниз». С помощью этих клавиш происходит переключение фокуса между пунктами меню.

Подтверждение действий, таких как, переход в раздел выбранного пункта меню или сохранение рекорда, в приложении осуществляется нажатием клавиши «Enter».

Управление игровым квадратом.

Движение квадрата осуществляется при нажатии и удержании стрелок клавиатуры «Вверх», «Вниз», «Вправо», «Влево» в соответствующие стороны. Другими словами, игровой квадрат будет двигаться до тех пор, пока нажата клавиша, после её отпускания, квадрат остановится. Скорость квадрата во время движения остается постоянной.

1. Общая функциональность.

Возможность игроком ввода своего никнейма для сохранения рекорда после завершения игры.

Сохранение рекордов пользователей в файл.

Отображение рекордов десяти лучших пользователей в порядке возрастания (чем меньше смертей понадобилось пользователю для прохождения всех уровней, тем выше его место в таблице рекордов) на странице рекордов. Загрузка данных для таблицы рекордов графического и консольного вариантов приложения происходит из файла.

Текст справки для графического и консольного вариантов также загружается из файлов.

1. Функциональность, связанная с игровым процессом.

В самом начале любого уровня недоступны все игровые объекты, кроме игрового квадрата и постоянного съедобного квадрата. Игровые объекты становятся доступными и начинают выдавать препятствия после первого съедения постоянного съедобного квадрата. Пока игровые объекты неактивны они не являются препятствиями и сами не выдают препятствия. Когда игровые объекты активны они могут быть доступными для съедения или недоступными (зависит от текущих размеров игрового квадрата). Требуется реализовать возможность изменения состояния игровых объектов, находящихся на поле, по событиям.

Переход на новый уровень возникает после съедения постоянного съедобного квадрата в том случае, если на поле отсутствуют другие игровые объекты.

При столкновении игрового квадрата с препятствием должен увеличиваться счетчик смертей игрока и уровень, на котором произошло столкновение начинается сначала. В случае, если уровень игрока, на котором произошло столкновение равен 8 или выше, то при столкновении его отбрасывает на один уровень назад.

1. Функциональность, связанная с игровыми объектами.

Каждый игровой объект (кроме квадратов) в игре выдает какие-либо препятствия, либо же сам объект выполняет какие-либо функции.

Постоянный съедобный квадрат. Требуется реализовать появление квадрата в новом месте игрового поля каждый раз после его съедения. При появлении он не должен пересекаться с другими объектами, находящимися на игровом поле. Появление должно быть реализовано от начала уровня до его окончания на всех уровнях.

Шестиугольник. Требуется реализовать функцию по генерации короткого выстрела раз в 2 секунды. Начальными выстрела является центр шестиугольника. Конечными координатами (координатами цели) выстрела являются координаты игрового квадрата на поле в момент начала выстрела. Скорость выстрела равна скорости игрового квадрата при движении. В представлении выстрел является линией. Длина линии на графическом отображении равна половине стороны постоянного съедобного квадрата. По аналогии, в консольном варианте линия выстрела должна быть короткой. Генерация выстрелов производится с момента активации шестиугольника до момента его съедения.

Круг. Требуется реализовать функцию по генерации стрелки. Стрелка выпускается один раз спустя 1 секунду после активации игрового объекта «Круг» и исчезает после съедения круга, выпустевшего её. Стрелка движется по тем же координатам, что и игровой квадрат. Скорость движения стрелки равна скорости игрового квадрата при движении.

Треугольник. Требуется реализовать функцию по генерации длинных выстрелов раз в секунду. Начальными выстрела является центр треугольника. Конечными координатами (координатами цели) выстрела являются координаты игрового квадрата на поле в момент начала выстрела. Скорость выстрела равна ¼ скорости игрового квадрата при движении. В представлении выстрел является линией. Длина выстрела на графическом отображении равна стороне треугольника. По аналогии, в консольном варианте линия выстрела должна быть длинной. Генерация выстрелов производится с момента активации треугольника до момента его съедения.

Прямоугольник. Прямоугольник является препятствием только пока его размер меньше размера игрового квадрата, в противном случае он будет доступен для съедения и перестанет быть препятствием. У прямоугольника есть линейная траектория (невидимая линия, вдоль которой прямоугольник движется от начала до конца и обратно). Требуется реализовать движение прямоугольника по линейной траектории.

### Архитектура приложения

[Здесь необходимо разработать общую архитектуру приложения (обязательно приложить графический вариант). Также необходимо описать взаимосвязь и что должны делать различные части приложения. Не нужно описывать буквально, с точностью до имён классов и методов. Это описательная часть, но от неё зависит насколько правильно будет разработано приложение.]

Архитектура приложения строится на шаблоне MVC (Модель – Представление – Контроллер).

Модель отвечает за логику работы приложения.

Контроллер осуществляет взаимодействие с пользователем, передает команды модели, создает представления, а также осуществляет взаимодействие с другими контроллерами.

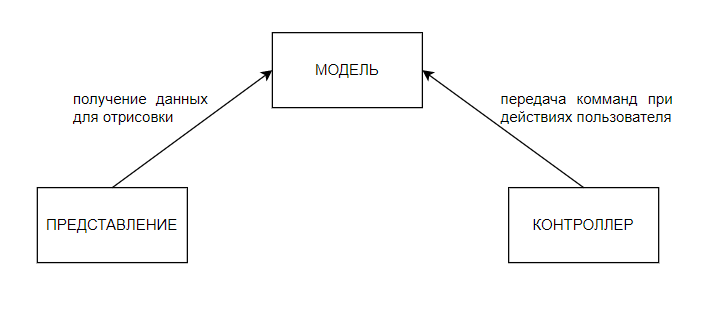


Рисунок 1 - Общий вид взаимодействия модели, представления и контроллера в приложении

Под представлением, моделью и контроллером подразумевается не один класс, а группа классов, относящихся к представлению, модели и контроллеру.

В игровом приложении ToughGrowth каждая часть приложения организована с помощью MVC.

Приложение можно разделить на следующие **базовые** части:

* главное меню;
* рекорды;
* справка;
* игра:
  + уровень 1;

. . .

* + уровень 10.

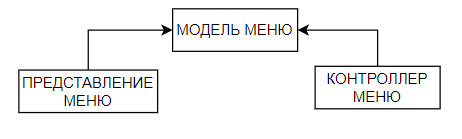


Рисунок 2 – Базовая архитектура MVC меню

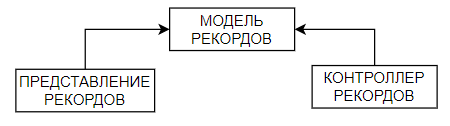


Рисунок 3 – Базовая архитектура MVC рекордов

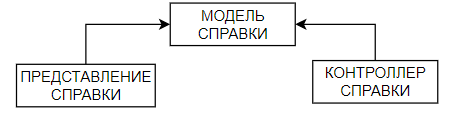


Рисунок 4 – Базовая архитектура MVC справки

Логика игры для каждого уровня остаётся постоянной: на игровом поле есть группа игровых объектов, которая генерирует препятствия, игровой квадрат и постоянных съедобный квадрат, поэтому модель для всех уровней игры одна, так же, как и контроллер, так как управление игрой на всех уровнях одинаковое.

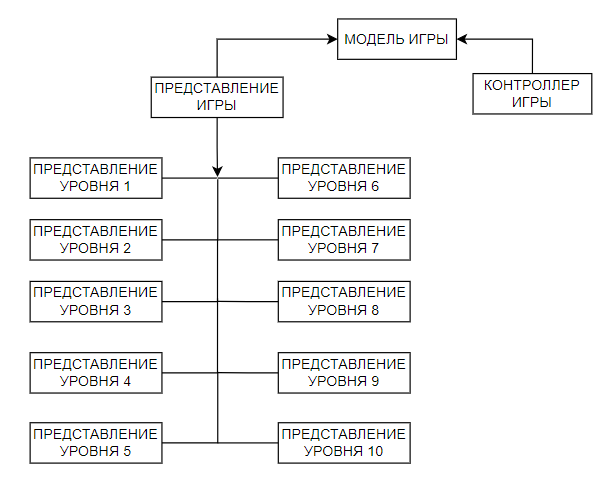


Рисунок 5 – Базовая архитектура MVC для игры

Приложение должно быть в двух вариантах: графический и консольный. К функциям контроллера относится создание представлений. Таким образом, представления и контроллеры будут различаться для консольного графического вариантов, но при этом вся логика работы останется в одном представлении.

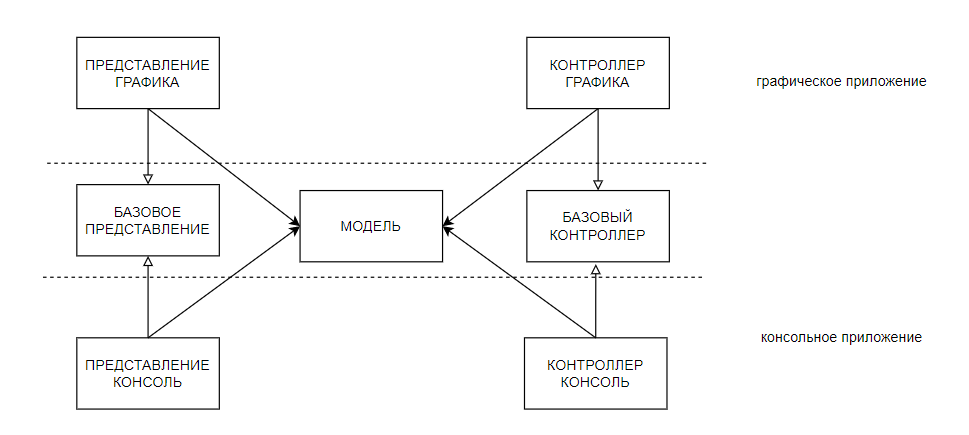


Рисунок 6 - Общее представление MVC для двух вариантов приложения

Каждая часть приложения графического или консольного варианта наследуется от своих базовых классов. Это относится только к представлениям и контроллерам, так как модели остаются постоянными, ведь логика работы приложения не зависит от варианта его представления.

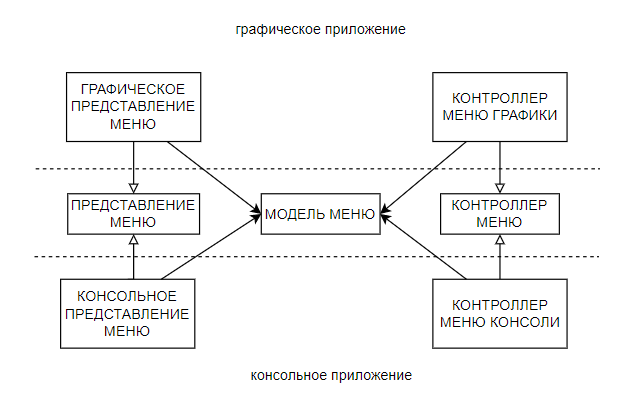


Рисунок 7 - Архитектура MVC меню для двух вариантов приложения

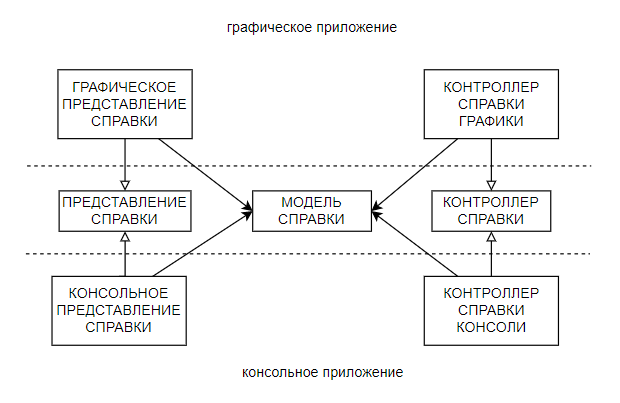


Рисунок 8 - Архитектура MVC справки для двух вариантов приложения

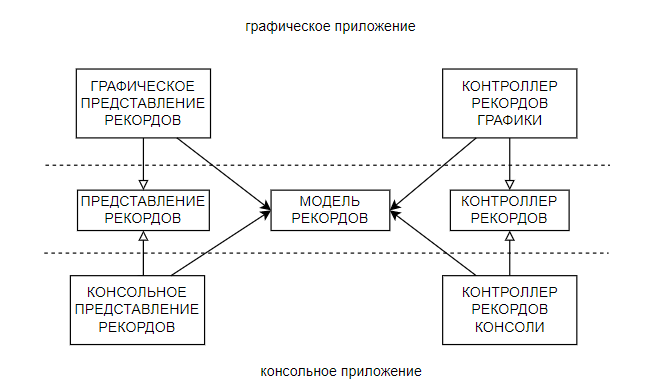


Рисунок 9 - Архитектура MVC рекордов для двух вариантов приложения

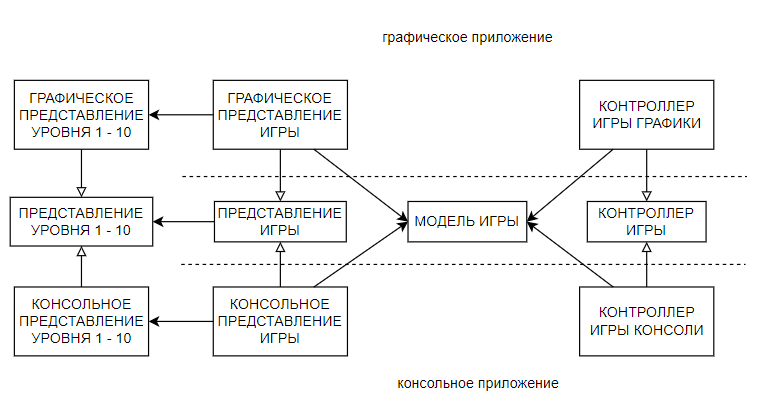


Рисунок 10 - Архитектура MVC игры для двух вариантов приложения

Организация взаимодействия контроллеров между собой организована с помощью менеджера контроллеров, который по событиям будет делегировать работу нужным контроллерам. Например, при нажатии на пункт меню «Рекорды» генерируется событие перехода в раздел и управление из контроллера меню передается контроллеру рекордов.

У каждого варианта приложения есть свой набор контроллеров, поэтому у каждого варианта должен быть реализован свой менеджер для корректного управления контроллерами.

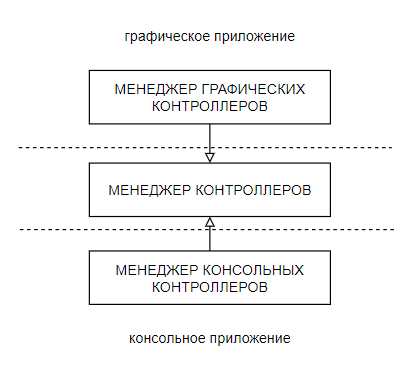


Рисунок 11 - Менеджер контроллеров для разных вариантов приложений

### Разработка интерфейса приложения

Интерфейс игрового приложения будет представлен следующим образом.

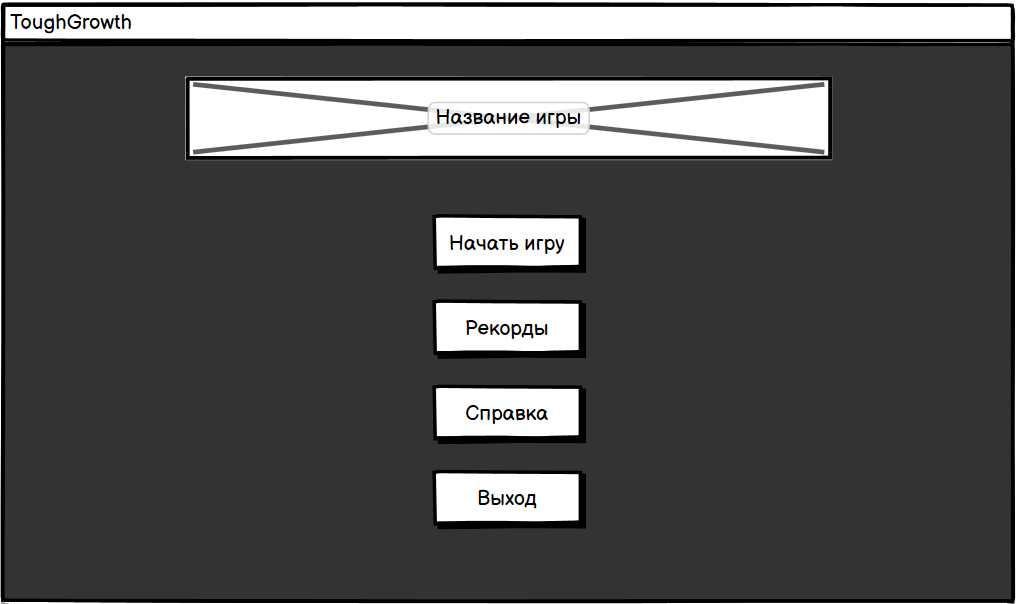


Рисунок 12 – Интерфейс главного меню игрового приложения

Главное меню (рисунок 12) содержит кнопки:

* «Начать игру»;
* «Рекорды»;
* «Справка»;
* «Выход».

Каждая кнопка осуществляет переход к основным разделам программы. Кнопка «Рекорды» позволяет пользователю перейти в раздел, где указаны 10 игроков, набравших наилучшее (наименьшее) количество баллов. Кнопка «Рекорды» направляет пользователя в соответствующий раздел, где указаны правила игры и особенности управления. Нажатие на кнопку «Выход» завершает программу, а нажатие на кнопку «Начать игру» начинает игровой процесс.

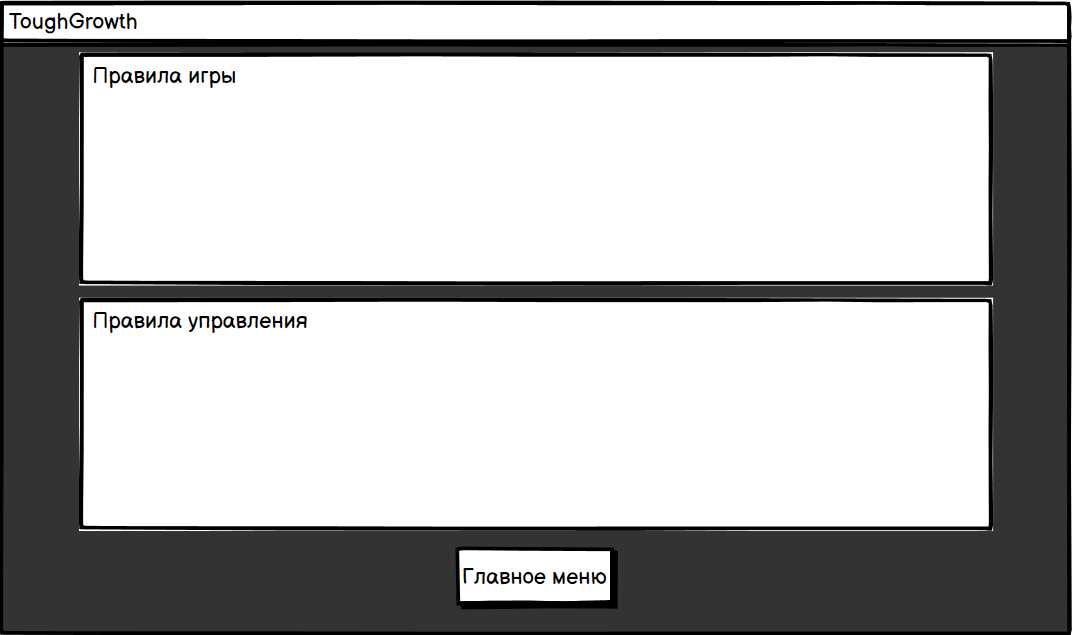


Рисунок 13 - Раздел "Справка"

При переходе в раздел «Справка» (рисунок 13) пользователь получает информацию о правилах игры и об особенностях управления в игре. Из этого раздела пользователь может также выйти в раздел главного меню нажатием на кнопку «Главное меню».

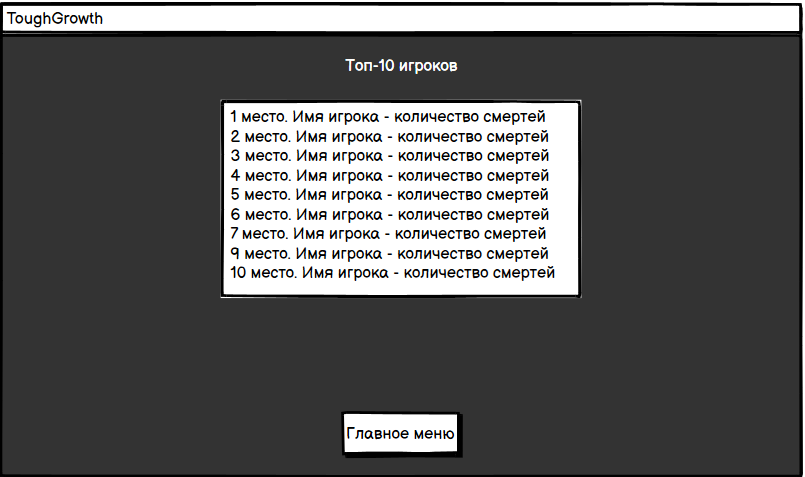


Рисунок 14 - Раздел "Рекорды"

Каждому пользователю доступен раздел рекордов (рисунок 14), где отображаются 10 игроков, набравшие наилучшее количество баллов. В данном случае считается, что лучшим будет пользователь, набравший наименьшее количество смертей в процессе игры. Из этого раздела можно также попасть в главное меню, нажав на кнопку «Главное меню».

В игре есть 10 уровней. На каждом уровне находятся игровые объекты в различных комбинациях. Распределение игровых объектов и их расположение на каждом уровне представлено на рисунках 15-24.

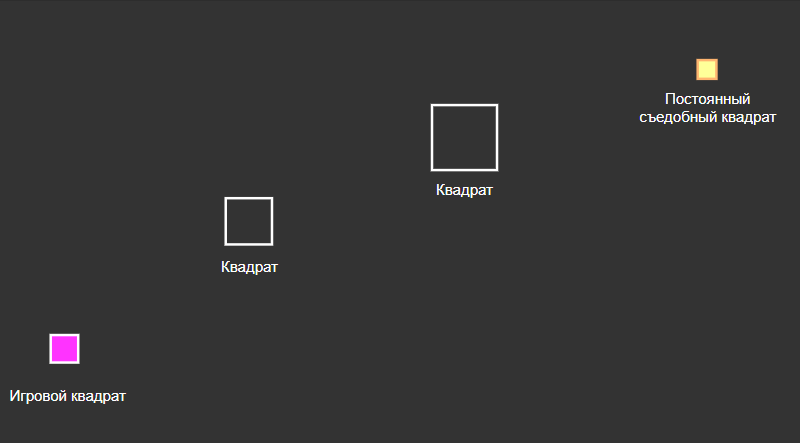


Рисунок 15 - Интерфейс уровня 1

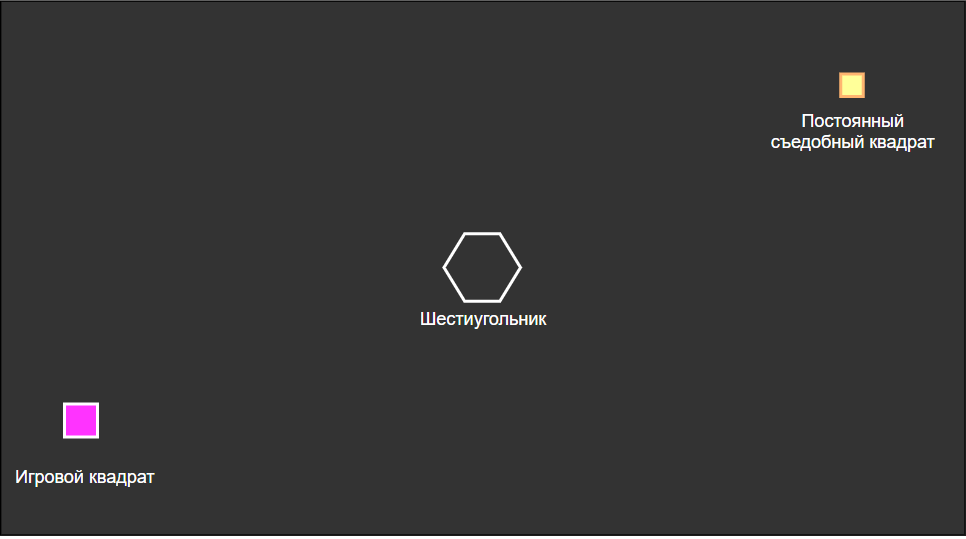


Рисунок 16 - Интерфейс уровня 2

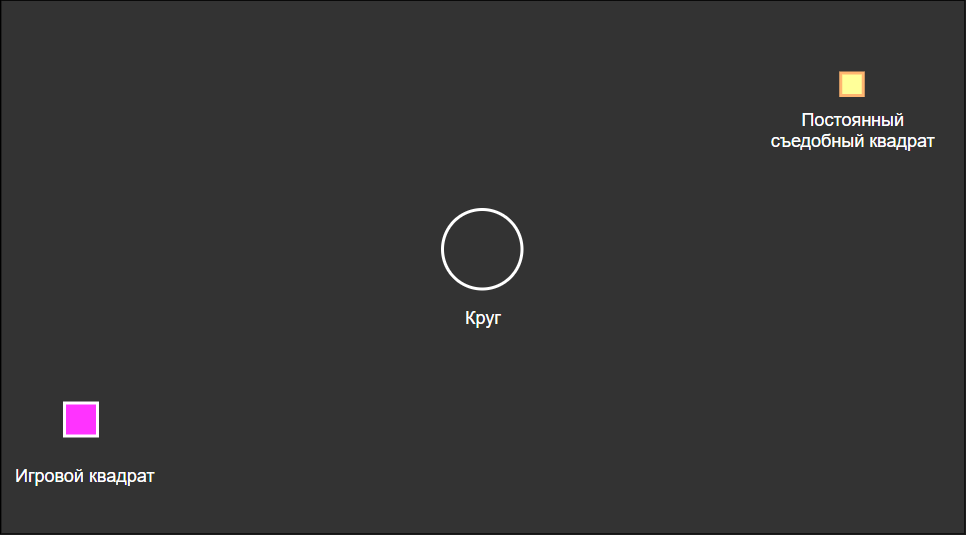


Рисунок 17 - Интерфейс уровня 3

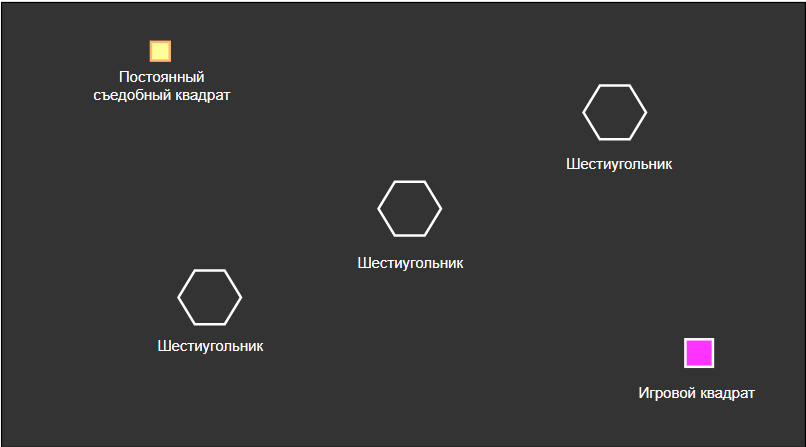


Рисунок 18 - Интерфейс уровня 4

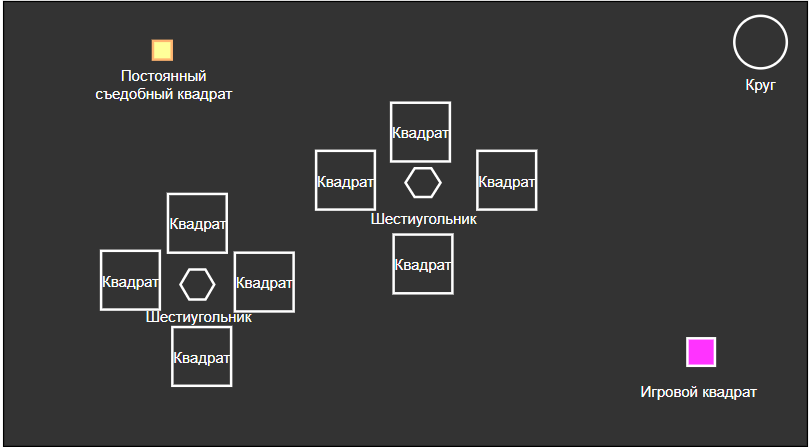


Рисунок 19 - Интерфейс уровня 5

Каждый игровой объект выпускает своё препятствие. Как видно из рисунка 9, если на игровом поле находятся 4 круга, то каждый из них выпустит свою стрелку.

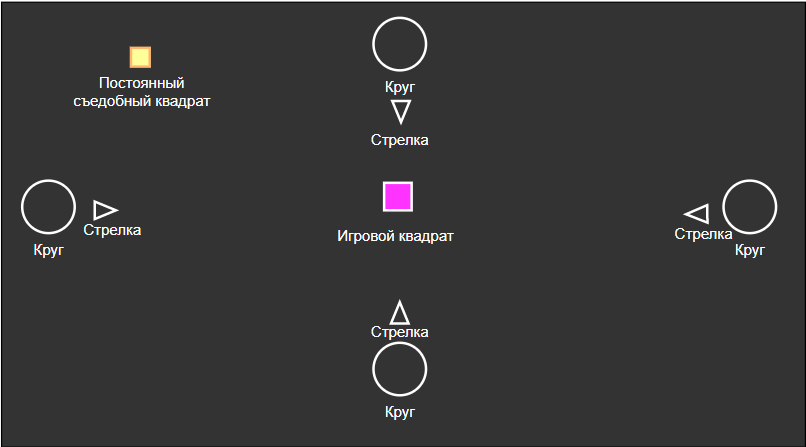


Рисунок 20 - Интерфейс уровня 6

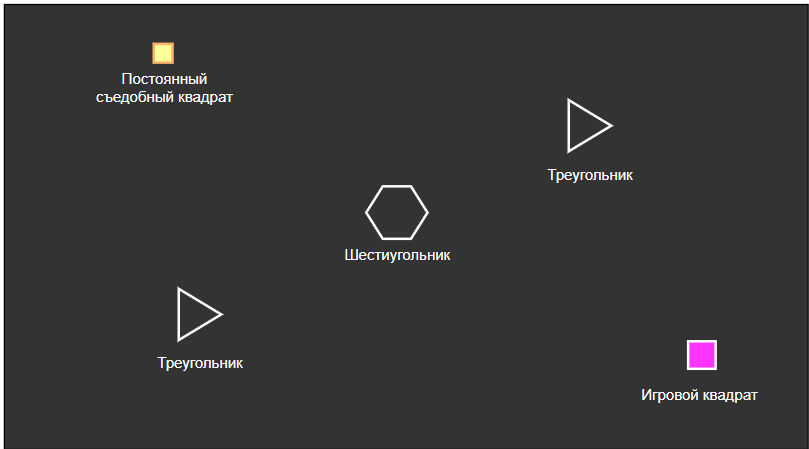


Рисунок 21 - Интерфейс уровня 7

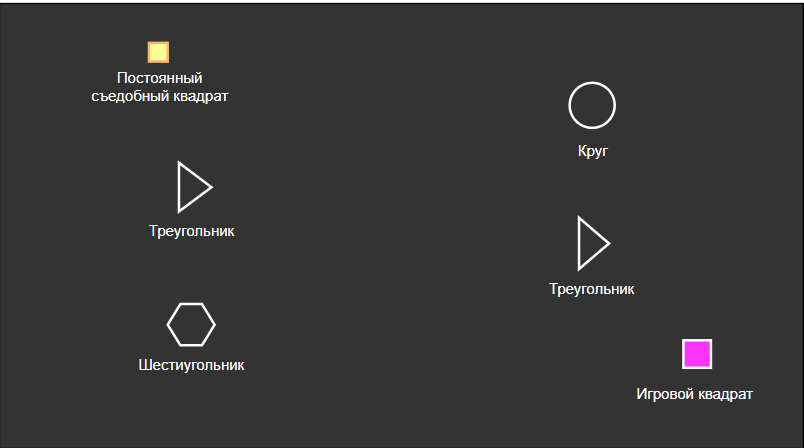


Рисунок 22 - Интерфейс уровня 8

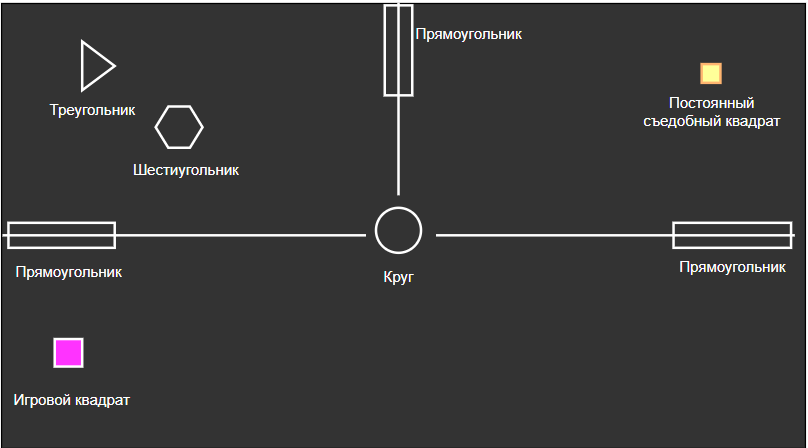


Рисунок 23 – Интерфейс уровня 9

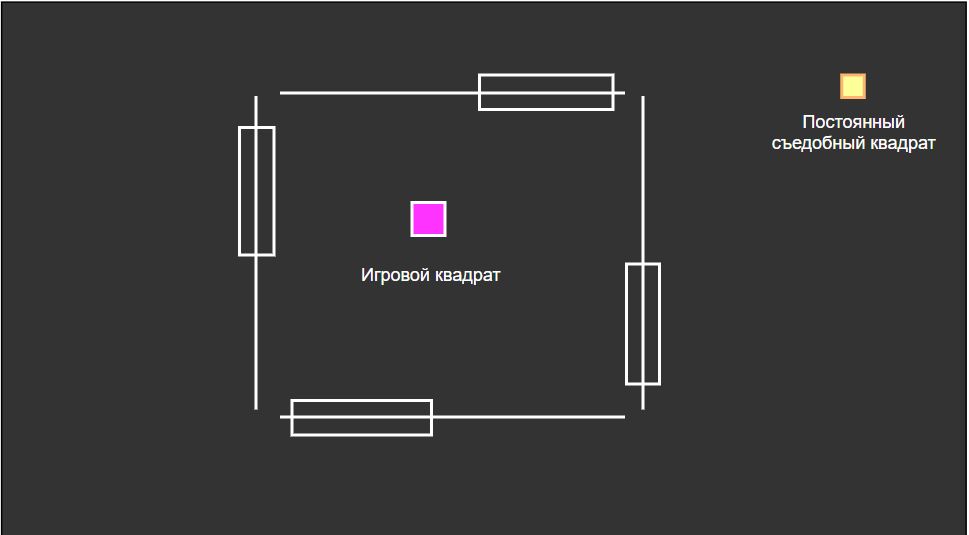


Рисунок 24 - Интерфейс уровня 10

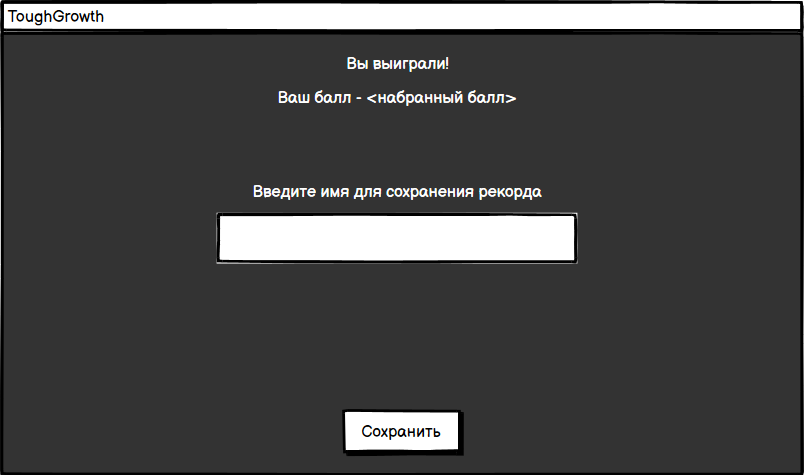


Рисунок 25 - Интерфейс окончания игры

После успешного прохождения 10 уровней игра завершается и пользователю отображается окно о завершении игры (рисунок 25), в котором отображается количество смертей, которые он набрал. Также пользователю предлагается ввести своё имя, для сохранения его рекорда. Нажатие на кнопку «Сохранить» переводит пользователя в раздел «Рекорды».

### Алгоритмы и логика работы приложения

[В данном разделе необходимо описать основные алгоритмы работы приложения, а также описать логику работы отдельных частей приложения в виде диаграмм последовательности. Вплоть до логики работы игрового процесса].

Во время работы программы пользователь может взаимодействовать с различными частями приложения.

При запуске приложения пользователю доступно меню, из которого он может переходить в различные разделы приложения (рисунок 26).

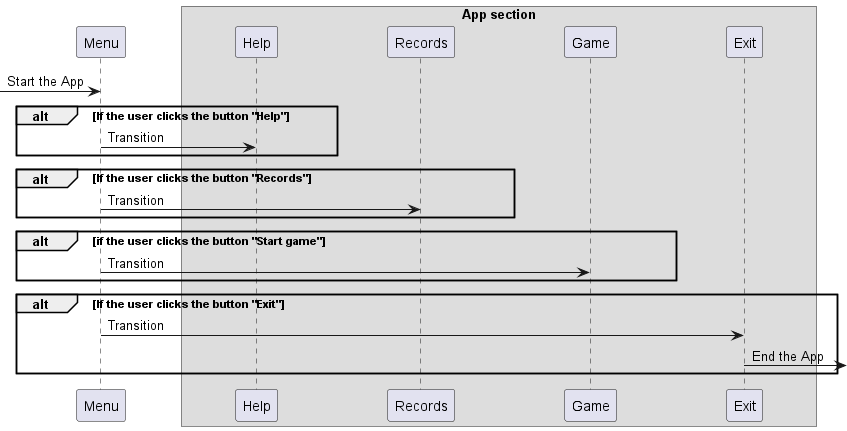


Рисунок 26 - Диаграмма последовательности перехода пользователя из главного меню в выбранный раздел

Более подробное описание действий, происходящих при переходе в каждый раздел.

Переход в раздел «Рекорды».

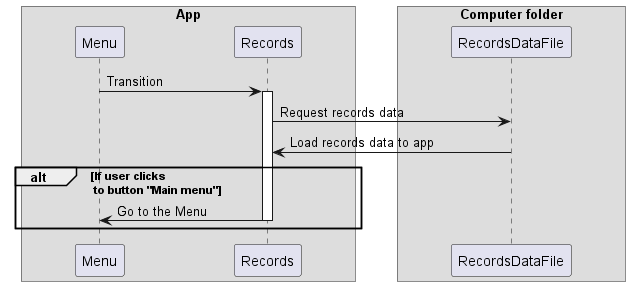


Рисунок 27 - Диаграмма последовательности взаимодействия пользователя с разделом "Рекорды"

При нажатии кнопки Enter, когда в фокусе находится пункт меню «Рекорды» пользователь переходит в соответствующий раздел. При этом все данные рекордов загружаются из файла. Пользователю отображаются первые 10 рекордов в порядке убывания. Выйти из раздела «Рекорды» обратно в меню пользователь может с помощью кнопки «Главное меню». Аналогичные действия происходят, если пользователь переходит в раздел «Рекорды» после завершения игры.

В модели рекордов полученные из файла данные сортируются в порядке убывания и первые 10 записей передаются представлению для отрисовки раздела рекордов.

Данные рекордов в файле хранятся в следующем виде:

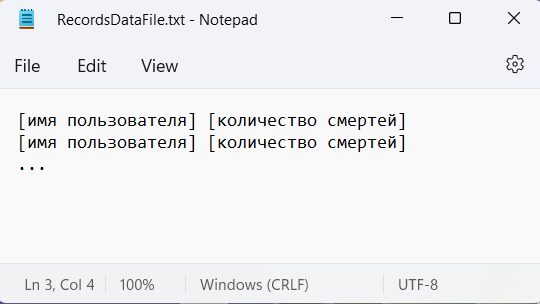


Рисунок 28 - Формат хранения рекордов в файле

Переход в раздел «Справка».

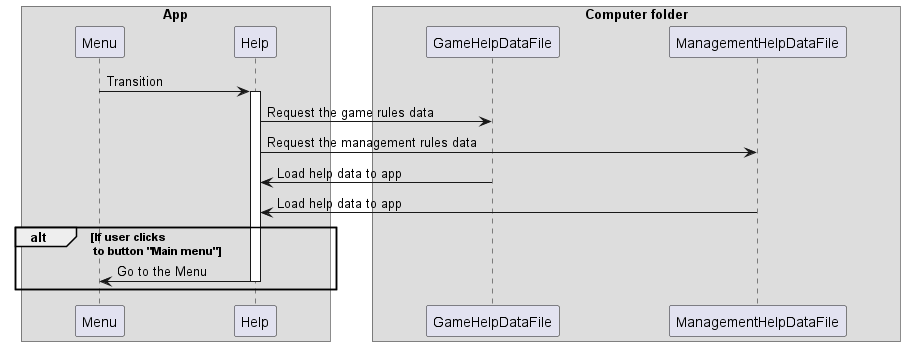


Рисунок 29 - Диаграмма последовательности взаимодействия пользователя с разделом "Справка"

При нажатии кнопки Enter, когда в фокусе находится пункт меню «Справка» пользователь переходит в соответствующий раздел. При этом все справочные данные (правила игры и правила управления) загружаются из файлов. Выйти из раздела «Справка» обратно в меню пользователь может с помощью кнопки «Главное меню».

Переход в раздел «Выход».

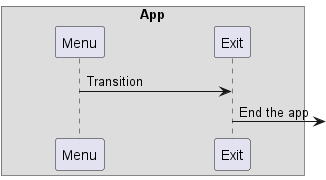


Рисунок 30 - Диаграмма последовательности взаимодействия пользователя с разделом "Выход"

При нажатии кнопки Enter, когда в фокусе находится пункт меню «Выход» пользователь завершает работу приложения.

Переход в раздел «Начать игру».

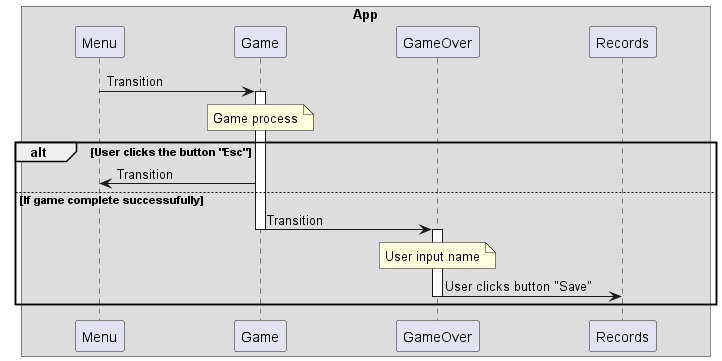


Рисунок 31 - Общая диаграмма последовательности взаимодействия пользователя с разделом "Начать игру"

Подробное рассмотрение игры.

Игра содержит в себе 10 уровней. При успешном прохождении уровня пользователь переходит на следующий уровень. В противно случае есть два варианта развития событий:

1. Если текущий уровень игрока меньше 8: уровень начинается заново.

2. Если текущий уровень игрока больше или равен 8: пользователя отбрасывает на один уровень назад.

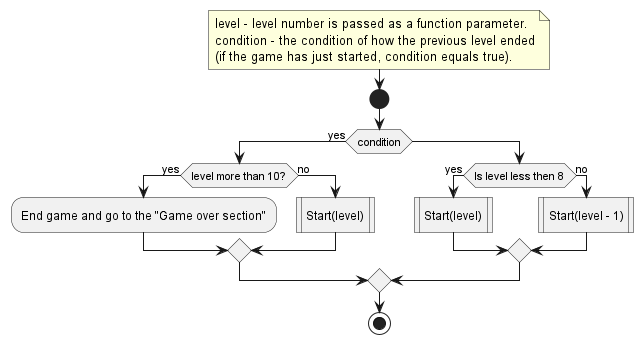


Рисунок 32 - Диаграмма активности перехода пользователя по уровням

Каждый игровой объект в течение уровня может находиться в 4 состояниях (рисунок 33). Каждый игровой объект проходит все эти стадии на каждом уровне. Переход по стадиям осуществляется при наступлении соответствующих событий.

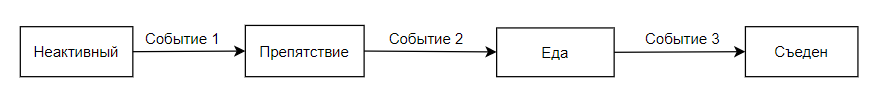


Рисунок 33 - Состояния игрового объекта

**Неактивный**. В этом состоянии игровой объект находится в самом начале уровня. Он не является препятствием и не выдает препятствия в этом состоянии. Игровой квадрат может проходить сквозь него.

**Событие 1**: первое столкновение (съедение) игрового квадрата. (Переход в состояние «Препятствие»).

**Препятствие**. В этом состоянии игровой объект сам является препятствием и начинает выдавать другие препятствия, если у него есть такая функция. При столкновении игрового квадрата с игровым объектом в этом состоянии приведёт к смерти игрового квадрата.

**Событие 2**: размер игрового квадрата стал равен (или больше) размера игрового объекта. (Переход в состояние «Еда»)

**Еда**. В этом состоянии объект уже не является препятствием, но всё также продолжает выдавать препятствие, если он имеет такую функцию.

**Событие 3**: игровой объект столкнулся (пересекся) с игровым квадратом. (Переход в состояние «Съеден»)

**Съеден**. В этом состоянии игровой объект исчезает с игрового поля, он перестает выдавать препятствия, все выданные им ранее препятствия также исчезают с игрового поля.

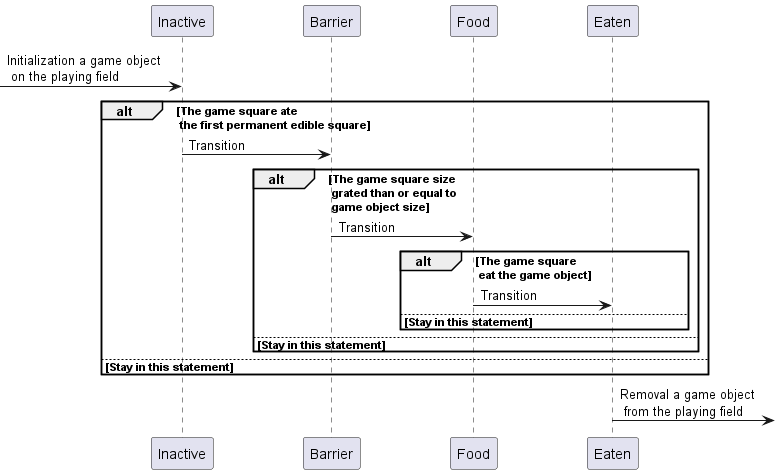


Рисунок 34 - Диаграмма последовательности изменения состояний игрового объекта

В процессе игры игровые объекты (шестиугольник, треугольник, круг, прямоугольник, квадрат) и постоянный съедобный квадрат никак не взаимодействуют между собой. Только игровой квадрат может взаимодействовать с другими игровыми объектами.

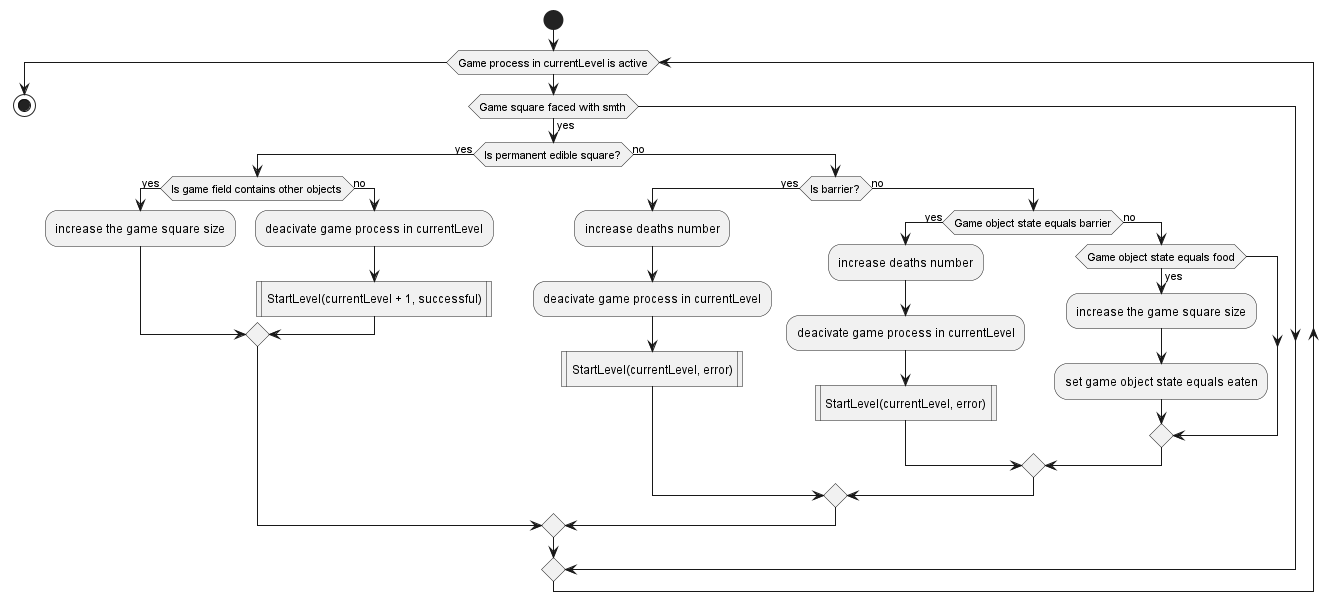


Рисунок 35 - Диаграмма активности, описывающая общее взаимодействие игрового квадрата с другими объектами при столкновении на игровом поле

Приложения

[Сюда можно добавлять различные списки, либо описания, которые не нужны в основной части, но помогут разобраться с общими принципами].

При съедении игрового объекта размеры игрового квадрата увеличиваются. Каждый игровой объект имеет свою площадь, поэтому в качестве коэффициента для увеличения игрового квадрата можно взять площадь съеденного игрового объекта. После съедения к площади игрового квадрата прибавляется площадь съеденного объекта. Исходя из новой площади рассчитывается новая длина стороны игрового квадрата и затем перерисовывается представлением.



Рисунок 36 - Пример увеличения размера игрового квадрата