**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 «Electric Cage»»**

по курсу

**«Конструирование программного обеспечения»**

Аналитик:

студент группы № 943

Бобылева Е. В.

Разработчик:

студент группы № 943

Бобылева Е. В.

Проверил:

доцент кафедры ВПМ

Столчнев В. К.

Рязань, 2022

Содержание

[1 Версии документа и изменения 3](#_Toc118239342)

[2 Термины и определения 3](#_Toc118239343)

[3 Общее описание задачи 3](#_Toc118239344)

[3.1 Описание требуемой функциональности 3](#_Toc118239345)

[3.1.1 Архитектура приложения 4](#_Toc118239346)

[3.1.2 Разработка интерфейса приложения 5](#_Toc118239347)

[3.1.2.1 Интерфейс главного меню 5](#_Toc118239348)

[3.1.2.2 Интерфейс игры 6](#_Toc118239349)

[3.1.2.3 Интерфейс игры после поражения 7](#_Toc118239350)

[3.1.2.4 Интерфейс «Правил игры» 9](#_Toc118239351)

[3.1.2.5 Интерфейс «Рекордов» 10](#_Toc118239352)

[3.1.3 Алгоритмы и логика работы приложения 11](#_Toc118239353)

[3.1.3.1 Общая логика взаимодействия пользователя с приложением 11](#_Toc118239354)

[3.1.3.2 Алгоритм при переходе в «Правила игры» 12](#_Toc118239355)

[3.1.3.3 Алгоритм при переходе в «Рекорды» 12](#_Toc118239356)

[3.1.3.4 Алгоритм для определения столкновение космического корабля с электрическим шаром 13](#_Toc118239357)

[3.1.3.5 Алгоритм работы приложения после поражения 13](#_Toc118239358)

[Приложения 14](#_Toc118239359)

# Версии документа и изменения

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Описание изменений | Автор |
| 1.0 |  | Создан | Бобылева Е.В. |
| 1.1 |  | Добавление общего описания задачи | Бобылева Е.В. |
| 1.2 |  | Добавление описания требуемой функциональности | Бобылева Е.В. |
| 1.3 |  | Добавление терминов | Бобылева Е.В. |
| 1.4 |  | Добавление описания архитектуры приложения в описание требуемой функциональности | Бобылева Е.В. |
| 1.5 |  | Добавление разработки интерфейса приложения | Бобылева Е.В. |
| 1.6 |  | Добавление алгоритмов и логики работы приложения | Бобылева Е.В. |

# Термины и определения

* **Электрическая клетка** – прямоугольная область внутри окна, зависит от размеров окна.
* **Электрический шар** – круг внутри электрической клетки, движется сверху экрана вниз.
* **Космический корабль** – сущность управляемая пользователем. Движется под углом 45 градусов (по диагонали) и может поворачивать только влево. При столкновении космического корабля со стенкой электрической клетки или с электрическим шаром, игра оканчивается.

# Общее описание задачи

Данный документ предназначен для разработки игрового приложения «Electric Cage».

«Electric Cage» относится к категории игр-аркад. В данном игровом приложении пользователю, который управляет космическим кораблем, необходимо избегать столкновения со стенками электрической клетки и электрическими шарами.

Однако космический корабль поворачивает только в левую сторону.

Очки начисляются пользователю в зависимости от того, столкновение со сколькими электрическими шарами он сумел избежать.

Игра заканчивается в 2 случаях:

* космический корабль столкнулся с электрическим шаром;
* космический корабль столкнулся со стеной электрической клетки.

## Описание требуемой функциональности

**Сущности:**

* Электрическая клетка:
* размер (статический, зависит от размера окна, не изменяется во время игры)
* Электрический шар:
* номер
* координаты центра (по х и у)
* размер
* скорость движения (шаг, на сколько перемещается)
* Космический корабль:
* координаты (по х и у)
* скорость движения (шаг, на сколько перемещается)
* направление движения
* Рекорды:
* имя пользователя
* очки пользователя

Электрическая клетка представляет собой пространство, в которой может перемещаться космический корабль. Если космический корабль столкнется с любой из стен электрической клетки, то игра будет прекращена.

Космический корабль – это объект, которым управляет пользователь. Управление осуществляется посредством нажатия на клавишу «Пробел» на клавиатуре. После нажатия на пробел космический корабль должен изменить направление своего движения. Корабль может двигаться только по диагонали и поворачивать только в одном направлении (влево). Т.е. например если космический корабль движется вверх влево, то после нажатия на «Пробел», он должен изменить траекторию своего движения и двигаться вниз влево. При повторном нажатии на «Пробел» - вниз вправо, после вверх вправо. Таким образом, мы можем выделить 4 возможных траектории движения для космического корабля:

* вверх влево;
* вниз влево;
* вниз вправо;
* вверх вправо.

Электрический шар – объект, который движется от верхней границы электрической клетки к ее нижней границе по прямой. Если космический корабль столкнется с электрическим шаром, то игра также будет прекращена. Электрический шар имеет свой номер, в зависимости от которого будут рассчитываться очки пользователя.

### Архитектура приложения

Архитектура приложения основывается на шаблоне проектирования MVC (Модель-Вид-Контроллер). Схема шаблона проектирования MVC представлена на рисунке 1.

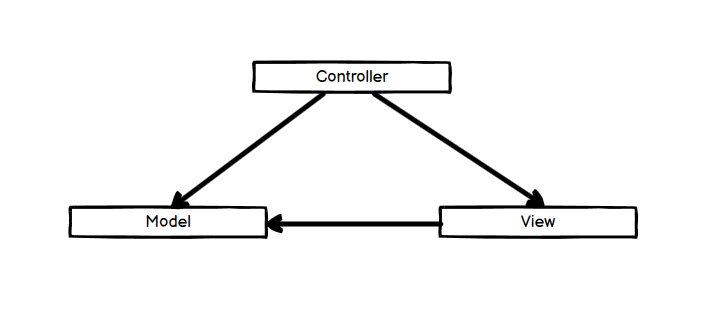


Рисунок – Шаблон проектирования MVC

Model (модель) содержит всю логику приложения. Модель не должна ссылаться на Controller(контроллер) и View (вид).

View (вид) отвечает за отображение данных пользователю. View (вид) реагирует на изменения, произошедшие в модели, и изменяет данные, отображаемые пользователю.

Controller(контроллер) отвечает за обработку действий пользователя и оповещает модель о необходимых изменениях.

Модель, вид и контроллер реализуются группой классов относящихся к модели, виду и контроллеру.

Так как реализация игрового приложения должна осуществляться и в WPF, и в консоли, то для каждого из интерфейсов должны быть реализованы свои View и Controller. Model не изменяется поскольку логика приложения не зависит от выбранного интерфейса. View и Controller должны наследоваться от базовых View и Controller соответственно.

Меню, игра, рекорды и правила должны иметь свои Model, View и Controller.

### Разработка интерфейса приложения

Интерфейс должен быть реализован в двух реализациях интерфейса: WPF – полностью графический интерфейс и консольное приложение.

**3.1.2.1 Интерфейс главного меню**

Интерфейс главного меню представлен на рисунке 2.

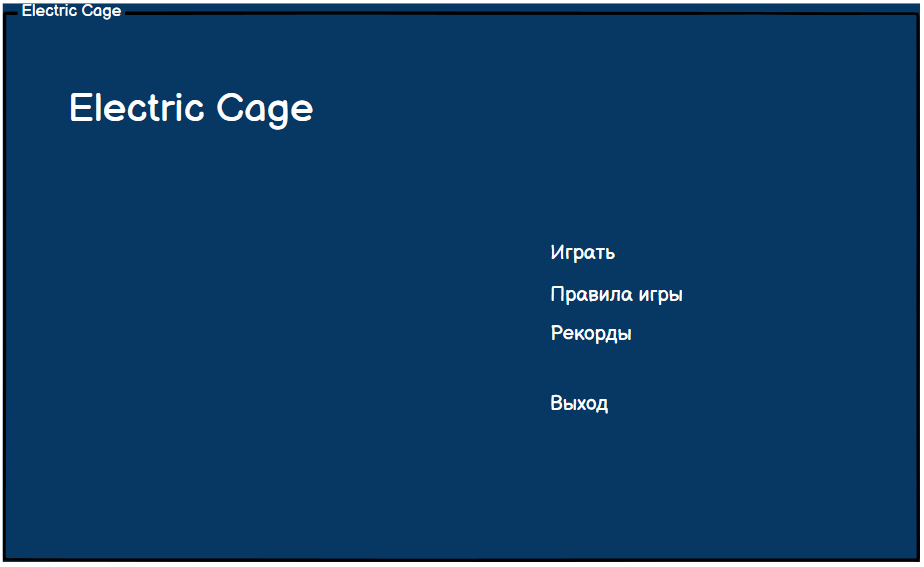


Рисунок 2 – Главное окно игры

На данном окне располагается название игры «Electric Cage» в верхнем левом углу. В правом нижнем углу расположены 4 кнопки – «Играть», «Правила игры», «Рекорды» и «Выход». Перемещение между пунктами меню осуществляется посредством нажатия на клавиши стрелок «↑» и «↓» на клавиатуре. Перемещение зациклено, т.е. при нажатии стрелки «↓», если пользователь находится на пункте «Выход», он перемещается на пункт «Играть», а при нажатии стрелки «↑», если пользователь находится на пункте «Играть», он перемещается на пункт «Выход».

При выборе пункта меню «Играть» пользователь переходит в интерфейс игры. При выборе пункта меню «Правила игры» пользователь переходит в интерфейс правил игры. При выборе пункта меню «Рекорды» пользователь перемещается в интерфейс рекордов. При выборе пункта меню «Выход» пользователь закрывает игровое приложение.

**3.1.2.2 Интерфейс игры**

Интерфейс игры показан на рисунке 3.

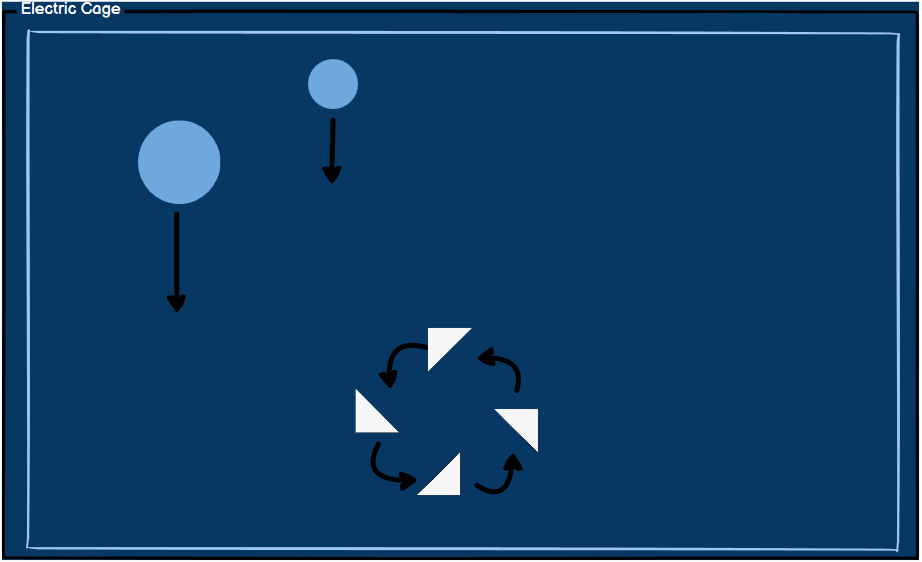


Рисунок 3 – Окно игры

После нажатия на кнопку «Играть» на главном меню, игрок переходит на игровое поле. Игрок управляет космическим кораблем, который движется в пределах игрового поля (ограничено экраном) – электрической клетки. Особенностью движения космического корабля является то, что он движется под углом 45 градусов и может поворачивать только в левую сторону. Игрок должен уворачиваться от электрических шаров, которые движутся сверху вниз. Очки рассчитываются исходя из того от какого количества электрических шаров сможет увернуться космический корабль.

**3.1.2.3 Интерфейс игры после поражения**

Если космический корабль сталкивается с электрическим шаром или врезается в электрическую клетку, то игра считается оконченной.

В случае если количество очков, набранных пользователем, больше чем у игроков в таблице рекордов, то прежде чем вывести результат, у пользователя запрашиваются имя, которое будет отображаться в таблице с рекордами. Интерфейс данного окна показан на рисунке 4. Если же количества очков недостаточно для записи результата в таблицу рекордов, то пользователь переходит в интерфейс, показанный на рисунке 5.



Рисунок 4 – Игровое поле игрока при входе в игру (идентификация пользователя)

Игрок (пользователь) может ввести любое имя длиной не более 20 символов. Если пользователь сразу нажимает на кнопку «Записать» или вводит пробелы вместо символов, то в таблицу рекордов будет записан результат с именем «Неизвестный». После нажатия на кнопку «Записать» пользователю показывают окно приложения, представленное на рисунке 5.

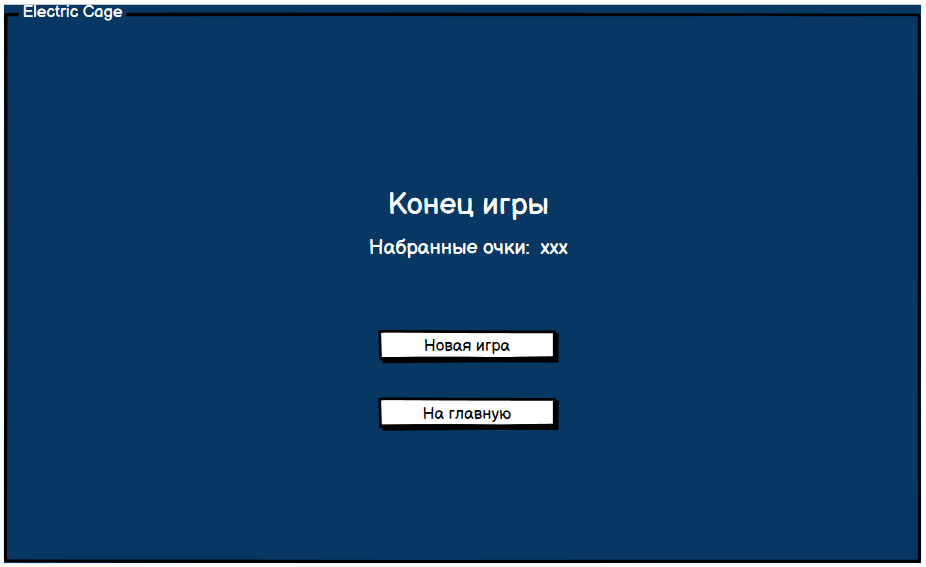


Рисунок 5 – Окно игры после завершения игры (поражения)

Игрок видит надпись «Конец игры» и количество набранных очков. Также в данном окне расположено меню, состоящее из 2 пунктов: «Новая игра» и «На главную». При нажатии на пункт меню «Новая игра» пользователь снова переходит в окно игры. Если пользователь выбирает пункт меню «На главную», то он переходит в окно с главным меню приложения.

**3.1.2.4 Интерфейс «Правил игры»**

Интерфейс приложения, после выбора пункта меню «Правила игры», показан на рисунке 6.

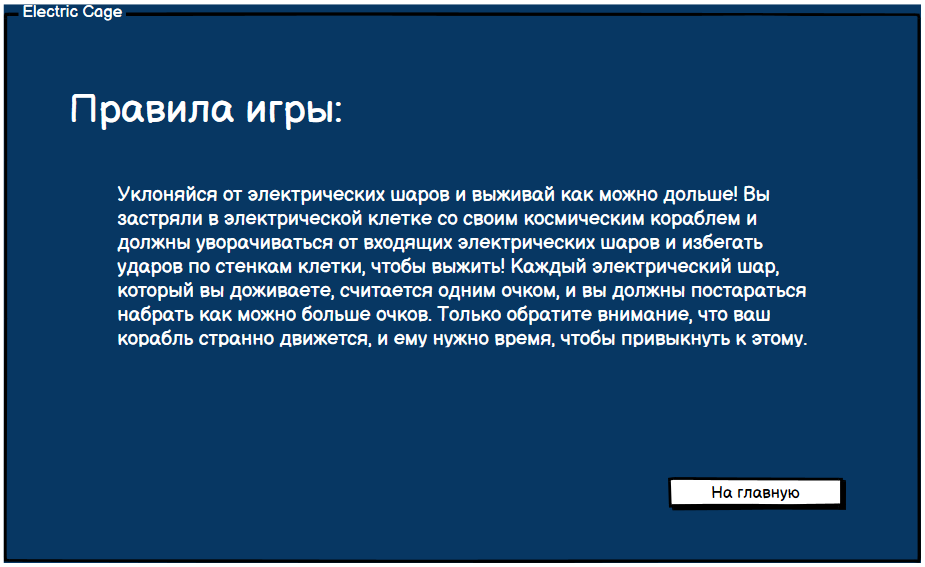


Рисунок 6 – Правила игры

При нажатии на кнопку «Правила игры», расположенную на главном окне, пользователь переходит в окно с правилами игры. В данном окне описаны правила игры, а также в нижнем правом углу окна расположена кнопка «На главную», после нажатия на которую пользователь возвращается в главное окно приложения.

**3.1.2.5 Интерфейс «Рекордов»**

Интерфейс приложения, после выбора пункта меню «Рекорды», показан на рисунке 7.

После нажатия на кнопку «Рекорды», расположенную в главном окне приложения, пользователь попадает в окно с рекордами. В центре данного окна расположена таблица с именами игроков и их очками. В таблицу записываются только три лучших результата. В случае если рекорды пока не записаны будет отображена пустая таблица. Если текущий игрок набрал большее количество очков, чем последний игрок, то результат текущего игрока заносится в таблицу. В нижнем правом углу окна расположена кнопка «На главную», после нажатия на которую пользователь возвращается в главное окно приложения.

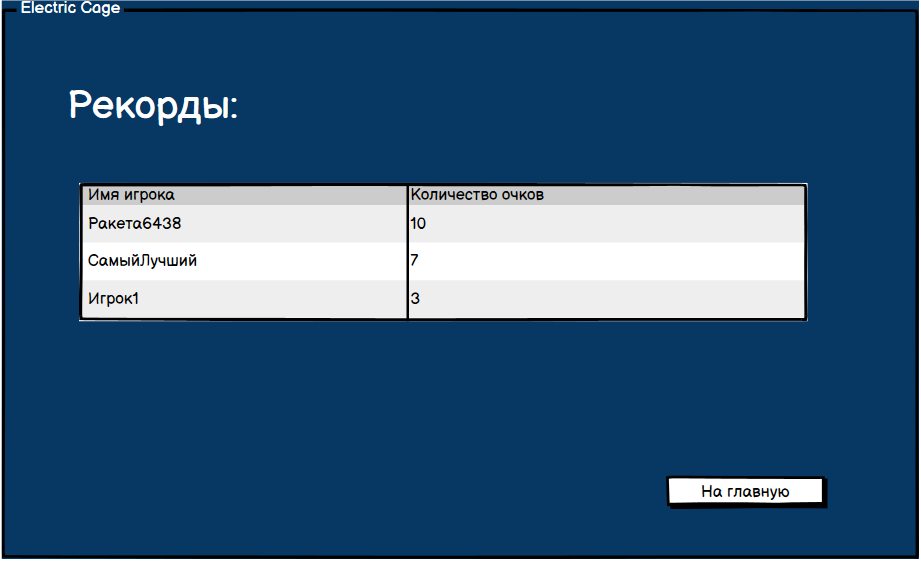


Рисунок 7 – Окно игры с рекордами

### Алгоритмы и логика работы приложения

**3.1.3.1 Алгоритм при переходе в «Играть»**

Диаграмма последовательности, представленная на рисунке 8, показывает взаимодействие пользователя с игровым приложением.

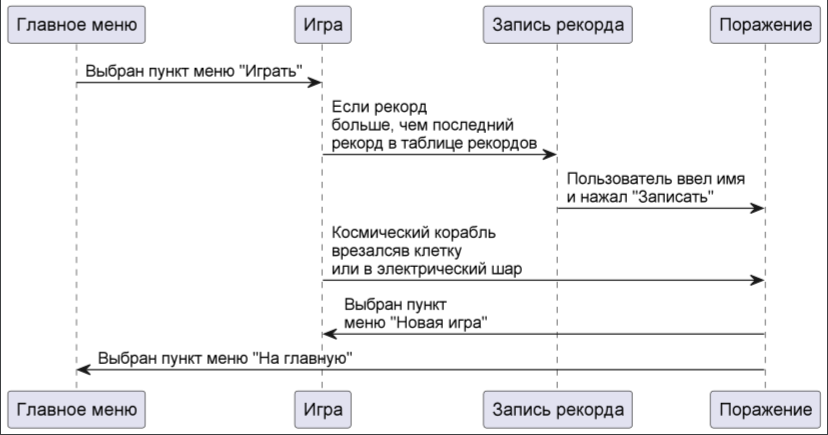


Рисунок 8 – Диаграмма последовательности при взаимодействии с игрой

При выборе пункта меню «Играть» на главном меню пользователь переходит в раздел с игрой. После поражения пользователь, если его текущие очки больше, чем очки в последней записи в таблице рекордов, переходит в раздел с записью рекордов. В этом разделе пользователю необходимо ввести имя и нажать «Записать». После этого он переходит в окно с поражением. В случае если его текущие набранные очки меньше, то пользователь сразу переходит в окно с поражением.

**3.1.3.2 Алгоритм при переходе в «Правила игры»**

При выборе пункта меню «Правила игры» пользователь переходит в раздел с правилами игры. Текст правил записан в файле. При переходе в раздел данные загружаются из этого файла. На рисунке 9 показана диаграмма последовательности при взаимодействии с разделом «Правила игры».

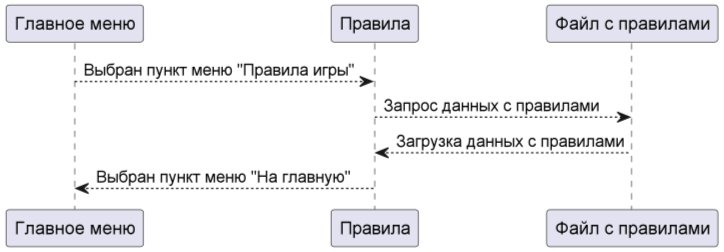
****

Рисунок 9 – Диаграмма последовательности при взаимодействии с разделом «Правила»

**3.1.3.3 Алгоритм при переходе в «Рекорды»**

При выборе пункта меню «Рекорды» пользователь переходит в раздел с рекордами. Рекорды пользователей записаны в файл. При переходе в раздел с рекордами данные загружаются из этого файла в таблицу.

Рекорды записаны в файл в виде (имя пользователя, тире и очки):

[имя пользователя] [-] [очки]

На рисунке 10 показана диаграмма последовательности при взаимодействии с разделом «Рекорды».

****

Рисунок 10 – Диаграмма последовательности при взаимодействии с разделом «Рекорды»

**3.1.3.4 Алгоритм для определения столкновение космического корабля с электрическим шаром**

Возможные случаи столкновение космического корабля с электрическим шаром показаны на рисунке 11.

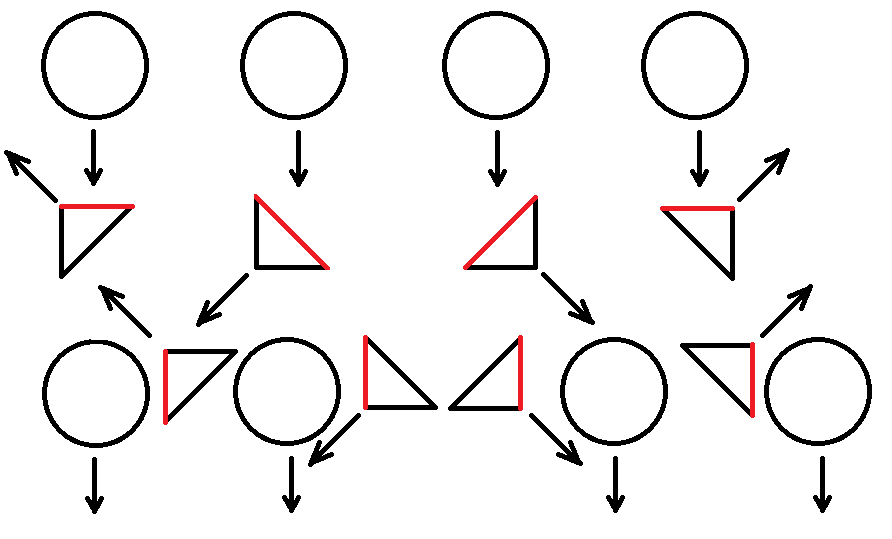


Рисунок 11 – Возможные столкновения электрического шара и космического корабля

Исходя из данного рисунка, необходимо определить случаи коллизии для окружности и отрезка.

Сначала определим, есть ли точки пересечения у окружности и линии. После чего проверим, принадлежит ли точка пересечения отрезку.

Пусть точки A(x1;y1) и B(x2;y2) – координаты начала и конца отрезка, точка O(x0;y0) – координаты центра окружности с радиусом R. Тогда точку пересечения можно вычислить, используя формулы:

Если lec > R, то точек нет.

Если lec = R, то возможна 1 точка пересечения E с координатами (ex;еу). Если ex>=x1 и ex<=x2 и ey>=y1 и ey<=y2, то точка пересечения есть, то есть столкновение электрического шара и космического корабля

Если lec < R, то возможны 2 точка пересечения F и G. Для вычисления координат точек воспользуемся формулами:

По аналогии с точкой Е определяем принадлежат ли точки нашему отрезку. Если хотя бы 1 точка принадлежит, то считаем что электрический шар столкнулся с космическим кораблем.

**3.1.3.5 Алгоритм работы приложения после поражения**

На диаграмме деятельности, показанной на рисунке 8, представлен алгоритм, выполняемый приложением после поражения пользователя. Если текущий результат пользователя больше последнего рекорда записанного в таблицу рекордов, то текущий результат пользователя будет записан. Чтобы записать рекорд необходимо ввести имя пользователя. В случае если пользователь ничего не ввел или же ввел пробелы, то результат будет записан под именем «Неизвестный». Диаграмма деятельности, отражающая алгоритм работы приложения после поражения, показана на рисунке 11.

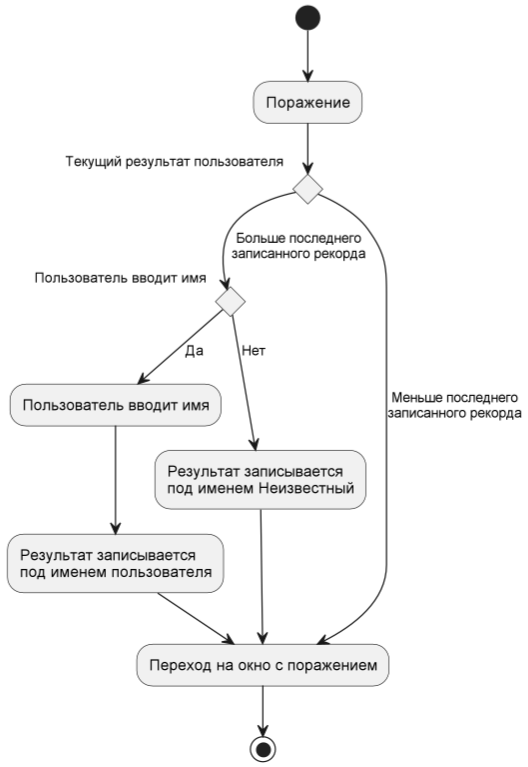


Рисунок 12 – Диаграмма деятельности

Приложения