**ВВЕДЕНИЕ**

На данный момент игр на развитие памяти достаточно много и все они пользуются популярностью в той или иной мере. Игра, в которой нужно запоминать карточки и выбирать одинаковые одна из самых популярных. Чем моя игра отличается от тысячи таких же? По смыслу нечем, но в ней есть свои «изюминки».

Во-первых, в ней есть регистрация и при успешной регистрации будет высвечиваться ник играющего, а также сохраняться статистика.

Во-вторых, в ней есть выбор, а тоесть можно выбрать цвета или картинки (животные, фрукты и овощи).

В-третьих, можно выбрать размер самого поля.

Можно перечислять многое, но в целом игра достаточно интересная, развивающая с разнообразными картинками.

На мое мнение, игры не должны быть только развлекающими, они должны нас чему-то учить.

Данная игра рассчитана не только для детей, но и для взрослых. Все мы вправе выбирать ту или иную развлекательную программу, но мой проект – это то, что может заинтересовать многих.

Данная игра была написана с использованием Angular, Node js и MongoDb.

**1 ПОСТАНОВКА ЗАДАЧИ**

Цель данной работы – создание развлекающей и в тоже время поучительной игры, с помощью которой можно развивать память, а также просто и комфортно проводить время.

Перед разработкой данного приложения были поставлены следующие задачи:

■ Выбор размера поля.

■ Начальная расстановка карт выполняется компьютером.

■ У каждого пользователя игры есть свой логин и пароль для входа.

■ Статистика:

• количество переворотов карт.

**2 ТРЕБОВАНИЯ К ПРОГРАМНОМУ ОБЕСПЕЧЕНИЮ**

**2.1 Требование к функциональным характеристикам системы**

Программа должна обеспечить возможность выполнения следующих функций:

- получения доступа к MongoDb для создания базы данных на компътере (заносится статистика по каждому пользователю) и сохранение данных в БД;

- правильная настройка подключения MongoDb на компъютере

**2.2 Требования к составу и параметров технических средств**

Программа должна работать на на базе операционной системы Windows.

Минимальная конфигурация:

Тип процессора: IntelCore i3 и выше.

Объем оперативной памяти: 8Гб и више.

**2.3 Требование к програмным характеристикам**

Для корректной работы программы, машины, на которой будет происходить ее запуск должны отвечать следующим программным требованиям:

* операционная система Windows 10;
* платформа – 64х.
* .Net версия 4.7.2

**3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ**

**3.1 Проектирование интерфейса**

При запуске приложения открывается сразу главное окно (компонент loginForm), в котором находятся такие кнопки, как:

- Без регистрации – тоесть пользователь заходит в игру без регистрации и все его результаты будут сохраняться под ником «Неизвестный»

- Зарегестрироваться – пользователь имеет право зарегестрироваться и все его результаты сохраняются под его ником.

Далее, не зависимо от выбора появляется другое окно (компонент mainForm), в котором и находится непосредственно сама игра. На данном компоненте находятся:

- RadioButton – которые позволяют задать размер самого поля

- Select – которые позволяют выбрать тип отображаемых картинок (цвета или же картинки)

- Lable – в котором оторбражается логин самого пользователя (если пользователь не зарегестрировался, то логин будет «Неизвестный»)

- Table – само игровое поле

Изначально игровое поле размером 6X6 и заполнено картинками – цветами. Игрок сразу же уже может начать играть. Под полем также находится еще один lable, который хранит статичтику, а тоесть количество сделанных ходов. После того, как игрок прошел данную игру сохраняется статистика в таблицу с количеством ходов и логином данного игрока.

**3.1.1 Интерфейс формы loginForm**

На данной форме находятся такие кнопки, как:

- Без регистрации – тоесть пользователь заходит в игру без регистрации и все его результаты будут сохраняться под ником «Неизвестный»

- Зарегестрироваться – пользователь имеет право зарегестрироваться и все его результаты сохраняются под его ником.

**3.1.2 Интерфейс формы mainForm**

На данном компоненте находятся:

- RadioButton – которые позволяют задать размер самого поля

- Select – которые позволяют выбрать тип отображаемых картинок (цвета или же картинки)

- Lable – в котором оторбражается логин самого пользователя (если пользователь не зарегестрировался, то логин будет «Неизвестный»)

- Table – само игровое поле

**3.2 Проектирование диаграммы классов**

Диаграмма классов ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Static Structure diagram) — структурная [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)) языка моделирования [UML](https://ru.wikipedia.org/wiki/UML), демонстрирующая общую структуру иерархии [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) системы, их коопераций, [атрибутов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0) (полей), [методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

В данном проекте всего два класса, это

- Picture – который имеет такие поля, как:

* mainPicture – основная обложка,
* backPicture – задняя обложка
* middleState – та обложка, которая будет показана в зависимости от того, было ли нажато на картинку или нет (основная или задняя обложка)
* isOpacity – отвечает за прозрачность, когда угаданы две картинки, они становяться невидимыми.

- User – который хранит в себе три поля (login, password, email). Нужен для записи данных в бд.

Диаграмму классов данного приложения выглядит следующим образом (рисунок 3.2.1).

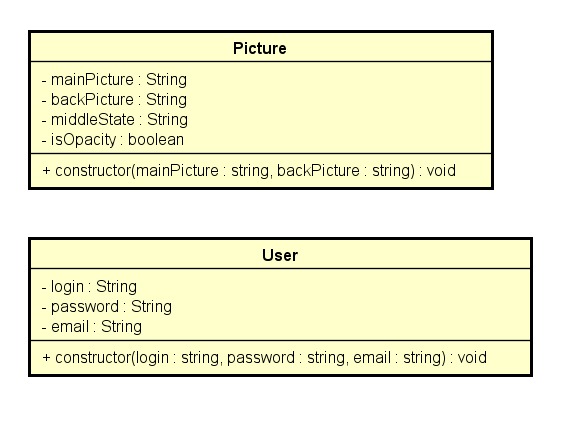


Рисунок 3.2.1 Диаграмма классов

**3.3 Проектирование базы данных**

База данных (БД) - это совокупность данных, связанных между собой определенную тематику. Эти данные хранятся на машинных носителях в упорядоченном виде. Наряду с понятием баз данных существует понятие системы управления базами данных (СУБД) - это компьютерная программа, которая позволяет управлять базой данных, то есть создавать и работать с ней.

В этом приложении работа ведется с MongoDB. Для удобного управление базами данных используется Node js.

Node.js - программная платформа, основанная на движке [V8](https://ru.wikipedia.org/wiki/V8_(%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA_JavaScript)) (транслирующем [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript) в [машинный код](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4)), превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

База данных создается в том случае, если она еще не создана. В ней храниться две таблицы:

Users – которая хранит в себе такие поля, как id, login, password

Statistics – которая хранит в себе стратистику по каждому пользователю, и тоесть login и numberOfMoves (количество ходов). Создание, а также запись в бд можно увидеть в файле app.js.

Для начала настраиваем прослушку нашего порта на Node js, а также создаем клиент MongoDb (рисунок 3.3.1).

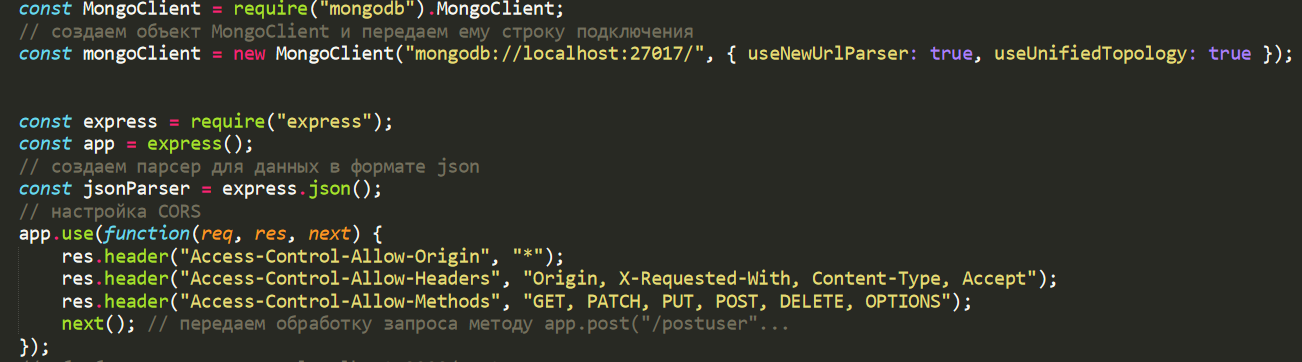


Рисунок 3.3.1 необходимые настройки для получение данных через POST метод, а также для подключения к бд.

Если рассматривать регистрацию пользователей, то последовательность действий такая:

1. С angular мы передаем через POST метод необходимые данные на Node js (рисунок 3.3.2), а также получаем ответ и выводим соответствующую информацию (занят логин, регистрация прошла успешно или же необработанная ошибка).

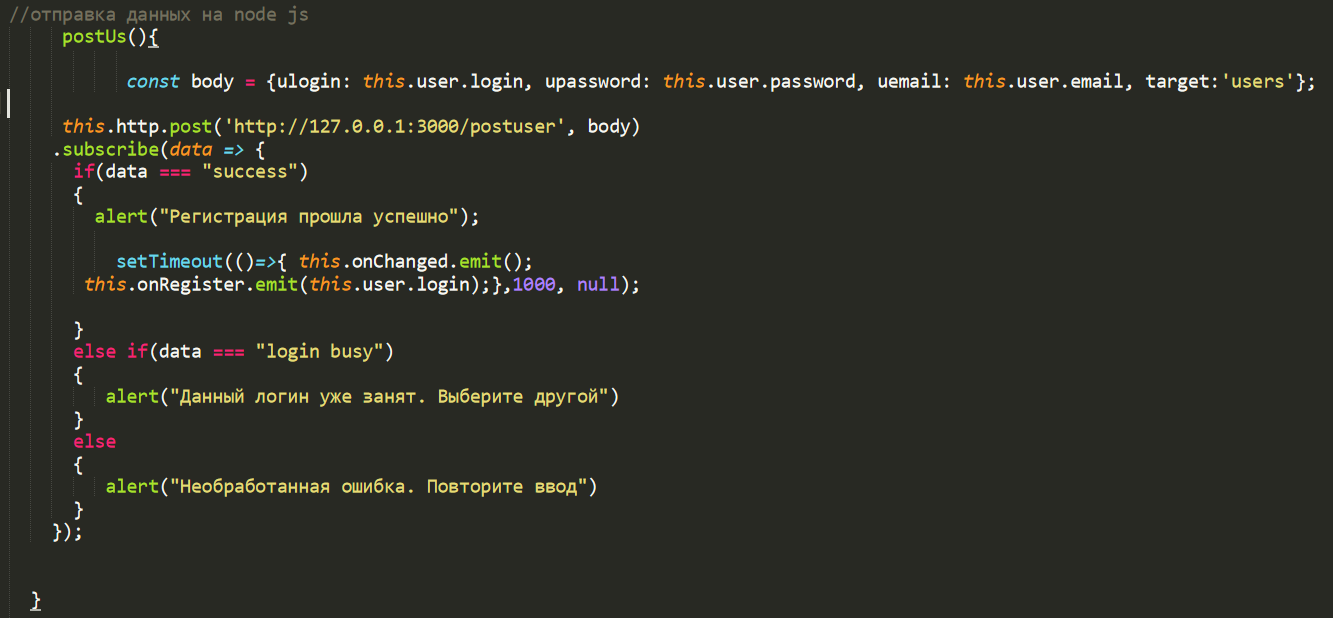


Рисунок 3.3.2 отправка данных на Node js с mainForm и обработка после ответа

Если рассматривать запись в статистику, также передаем необходимые данные POST методом (рисунок 3.3.3)

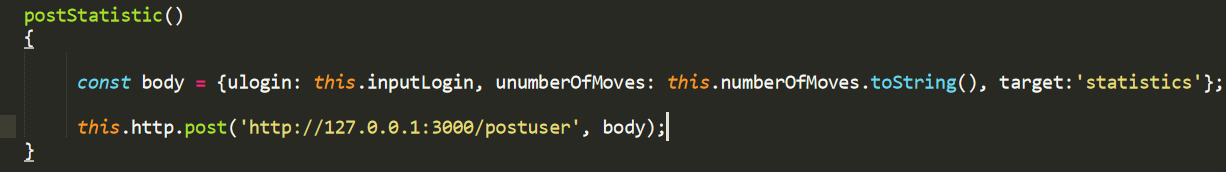


Рисунок 3.3.3 отправка данных на Node js с loginForm

На самом Node js мы обрабатываем данные (так как прослушка порта работает постоянно, код прописан таким способом, чтобы можно было записывать данные как в таблицу Users, так и в Statistic, для этого создана переменная isUsers) рисунок 3.3.4, 3.3.5 и 3.3.6.



Рисунок 3.3.4 Получение данных



Рисунок 3.3.5 обработка данных для таблицы Users

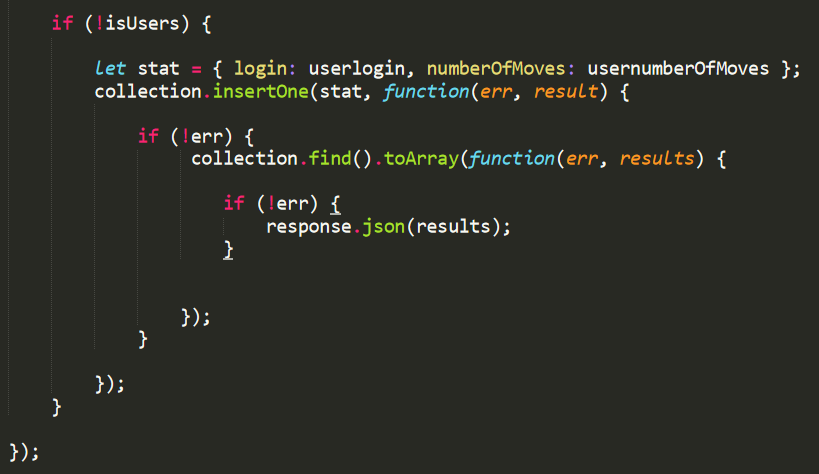


Рисунок 3.3.6 обработка данных для таблицы Statistics

**3.4 Проектирование алгоритмов**

В данной игре самым важным алгоритмом является обработка нажатия на картинку, а также сравнение картинок между собой. Для этого я использую класс Picture (сам класс был описан выше).

Рассмотрим диаграмму деятельности (рисунок 3.4.1). Приложение начинается с входа (регистрация или без регистрации), настроек самой игры и окончания игры. Если пользователь захочет еще раз сыграть он может продолжить.

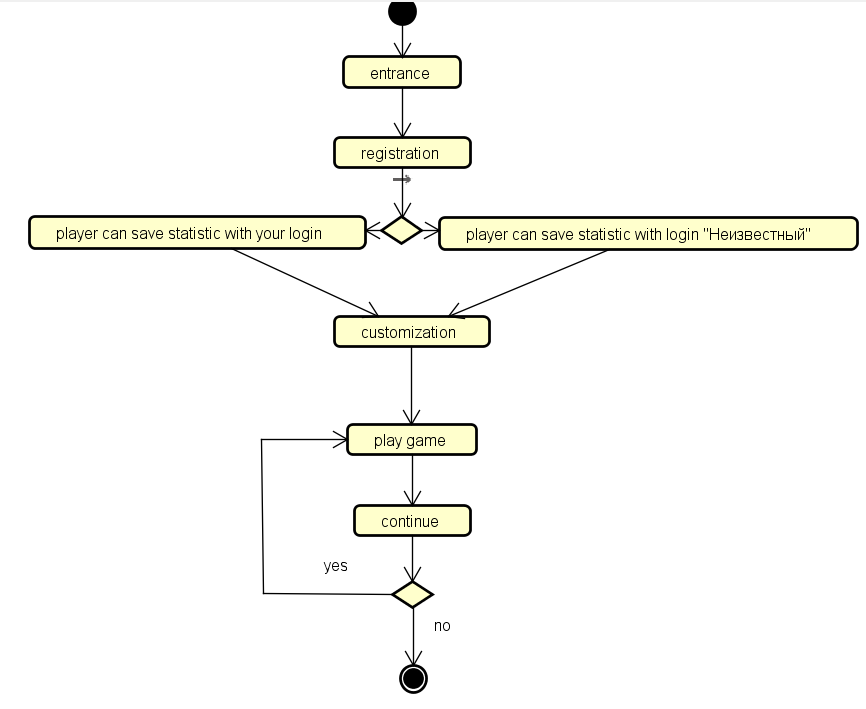


Рисунок 3.4.1 Диаграмма деятельности

Рассмотрим также диаграмму Вариантов использования (рисунок 3.4.2).

На диаграмме видно, что существует две сущности (Игрок и игра) и между собой они взаимодействуют, а именно:

- Игрок может войти в игру (с регистрацией и без нее)

- Может настроить игру (выбрать размер поля и отображаемые картинки)

- и естветственно может в нее сыграть

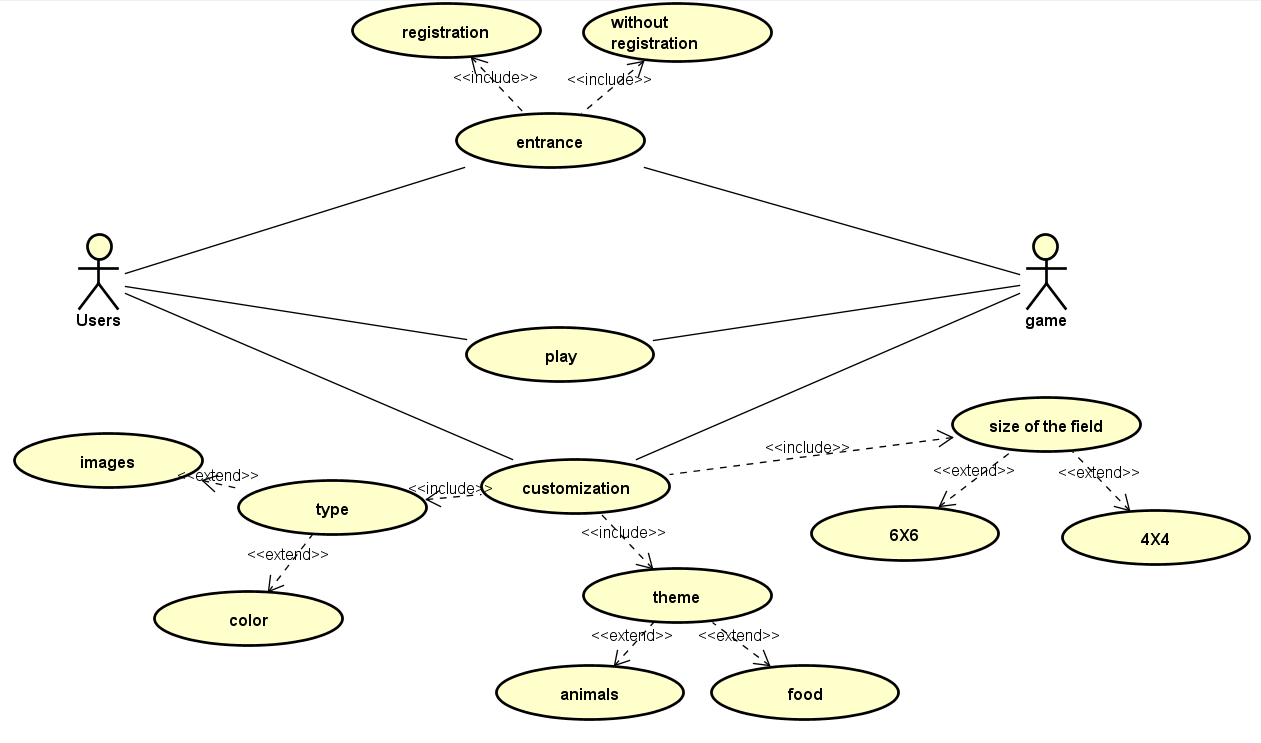


Рисунок 3.4.2 Диаграмма вариантов использования

Таким образом, мы рассмотрели что из себя представляет данная игра.

**4 ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММЫ**

Отладка программы - этап в разработке программы, который состоит из выявления и устранения программных ошибок, факт наличия которых все установлен. Протестируем наше приложение. При входе в игру мы увидим такое окно (рисунок 4.1)

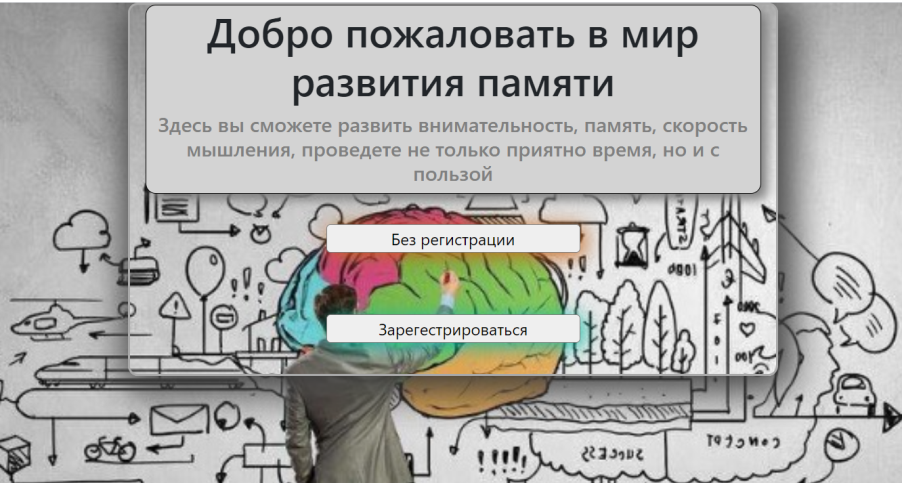


Рисунок 4.1 Форма при входе в игру

Для начала попробуем зайти без регистрации. После нажатия на кнопку «Без регистрации» появляется такое диалоговое окно (рисунок 4.2)

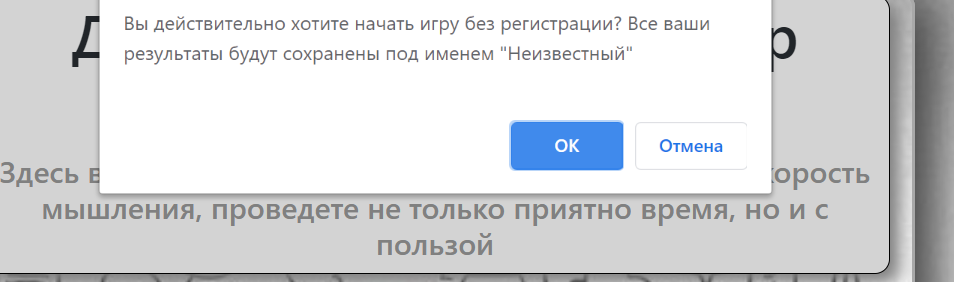


Рисунок 4.2 Вход без регистрации

Тоесть, все отлично отрабатывает.

Далее рассмотрим вход с регистрацией (также присутствует авторизация) рисунок 4.3

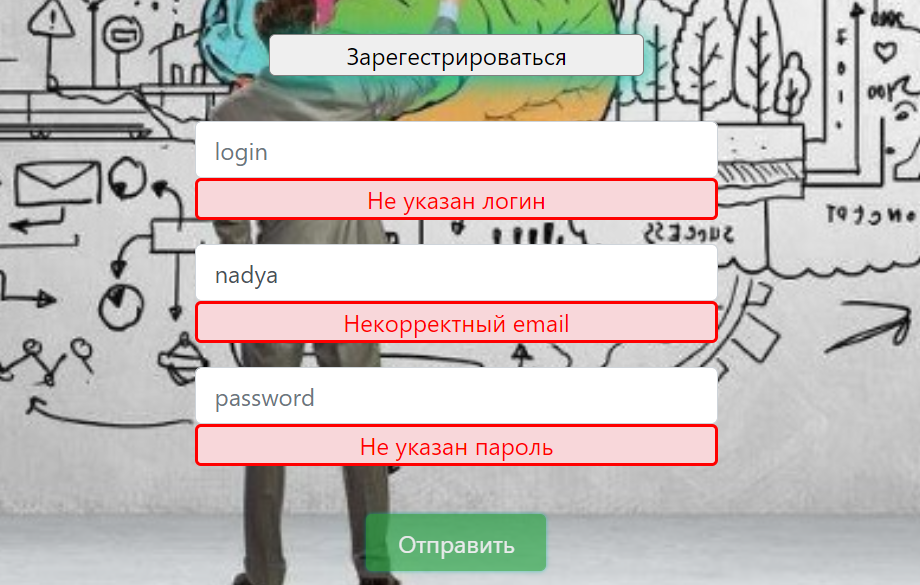


Рисунок 4.3 Вход с регистрацией и сама регистрация

И тут все отлично работает, ошибок не выявлено.

Если все прошло успешно тогда увидим такое диалоговое окно (рисунок 4.4)

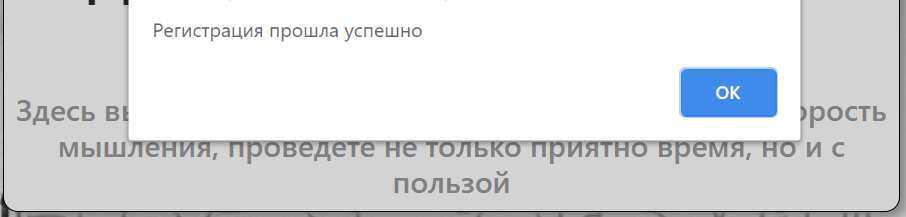


Рисунок 4.4 Регистрация прошла успешно

Попробуем настроить и сыграть в игру (рисунок 4.5)

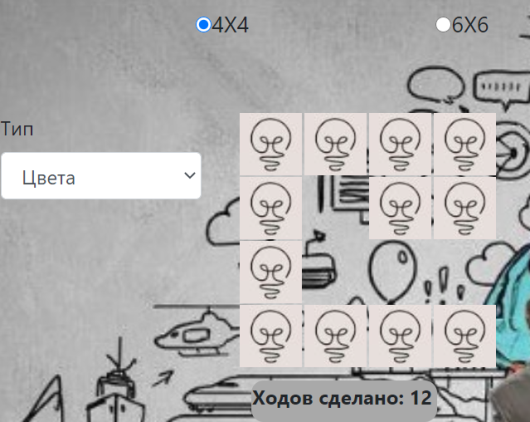


Рисунок 4.5 Настройка игры, а также ее прохождение.

Если игра закончена, будет высвечено такое диалоговое окно (рисунок 4.6)

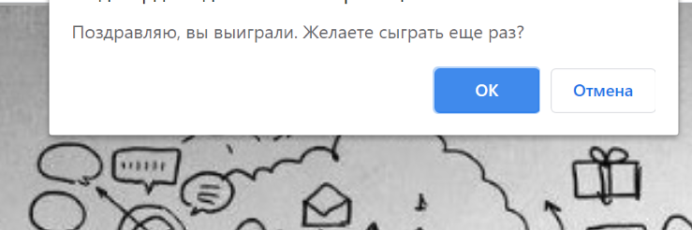


Рисунок 4.6 Окончание игры

Тоесть, можно сделать вывод, что все отлично работает без ошибок

**5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ И ПРИЛОЖЕНИЕ**

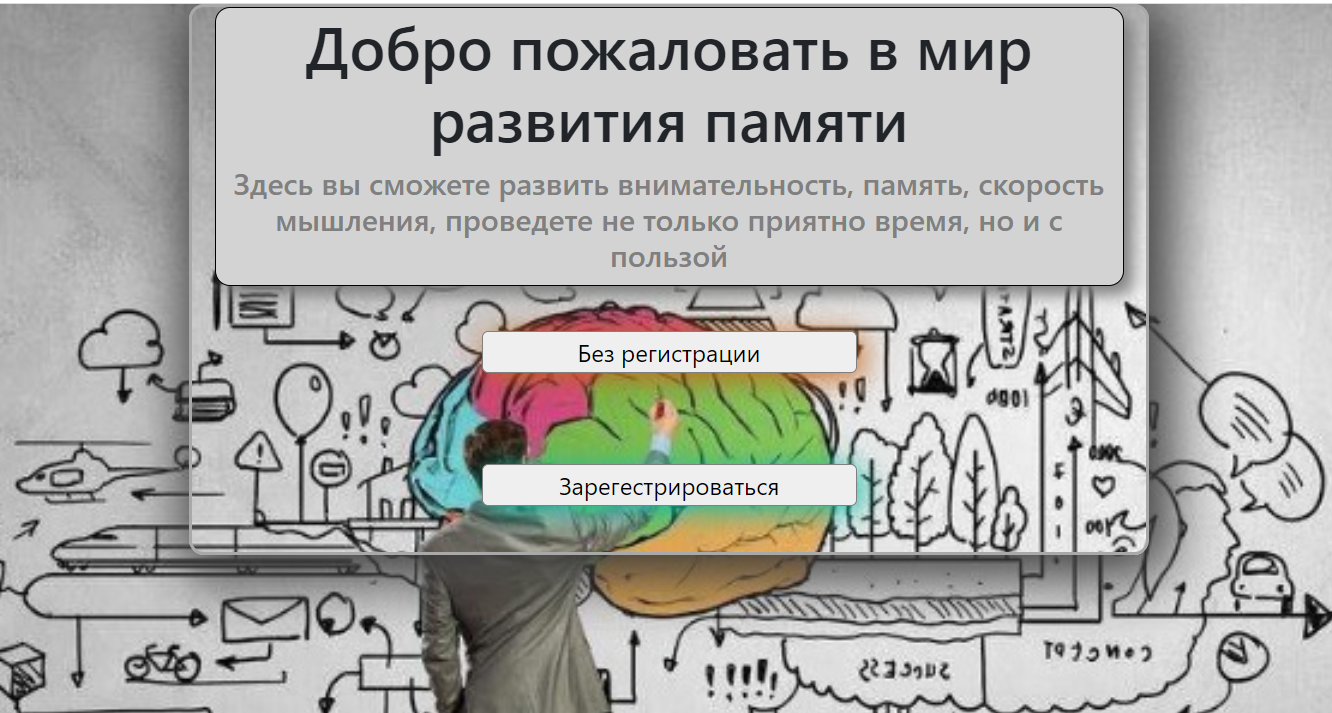
При запуске приложения пользователь сразу увидит вот такое окно (рисунок 5.1)

Рисунок 5.1 Форма при входе в игру

Где все достаточно понятно. У пользователя есть два варианта

- либо зайти в игру без регистрации

- либо зарегестрироваться

Если же пользователь выберет «Без регистрации», он увидит такое диалоговое окно (рисунок 5.2 )

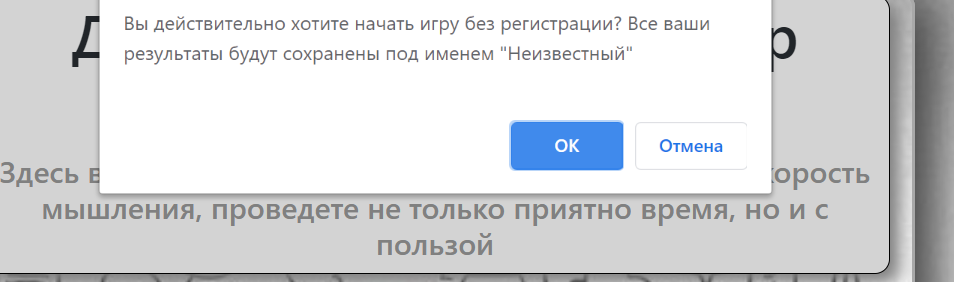


Рисунок 5.2 Вход в игру без регистрации

Данный рисунок говорит о том, что вся статистика будет сохраняться под именем «неизвестный», что не даст увидеть и узнать именно результат пользователя.

Но есть другой вариант, вход с регистрацией, который позволяет зарегистрироваться и запомнить ник пользователя. И в статистику будет уже внесен результат с логином пользователя. Также присутствует проверка на ввод корректности данных (рисунок 5.3)

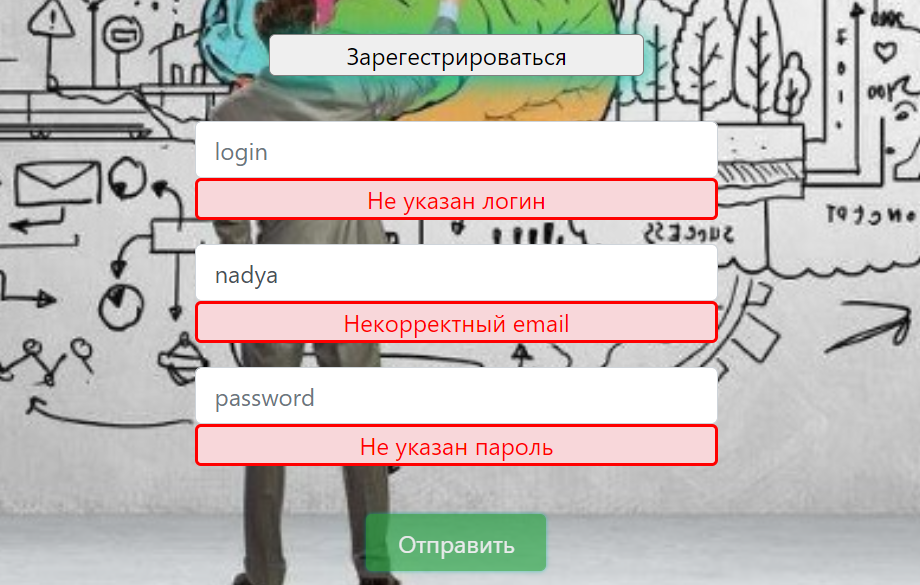


Рисунок 5.3 Проверка на ввод корректности данных

И если регистрация прошла успешно, тогда пользователь увидит такое диалоговое окно (рисунок 5.4)

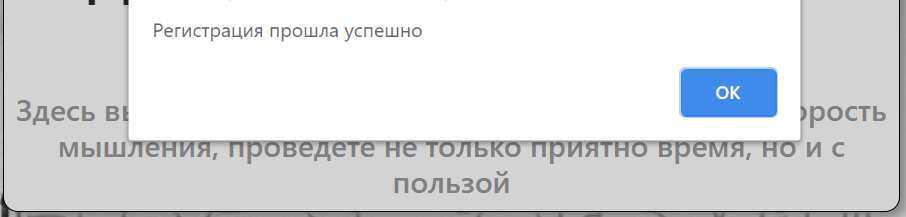


Рисунок 5.4 Регистрация прошла успешно.

Есть также и другие варианты, такие как «Данный логин уже занят. Выберите другой» и «Необработанная ошибка. Повторите ввод». Вся информация будет выведена также в диалоговом окне при тех или иных случаях.

Далее, пользователь видит саму игру, где может сам ее настроить и следовательно сыграть. Настроить очень просто.

Во-первых, пользователь может выбрать размер игрового поля (доступно 4Х4 и 6Х6).

Во-вторых, он сам может выбрать тип картинок (а тоесть цвета или картинки) и если пользователь выбирает картинки у него появляется возможность выбрать тип (фрукты и овощи или же животные). Все картинки достаточно забавны и понравятся не только детям, но и взрослым.

Все настройки, а также картинки изображены на рисунках 5.5, 5.6, 5.7, 5.8.

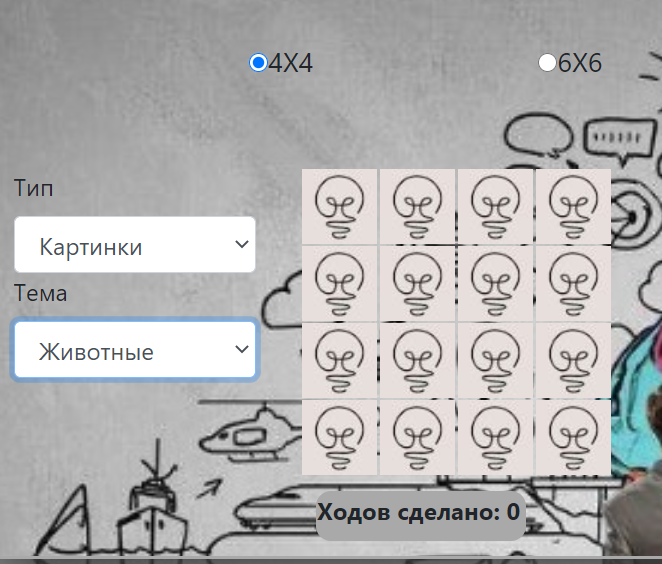


Рисунок 5.5 Настройки размера поля и выбор картинок

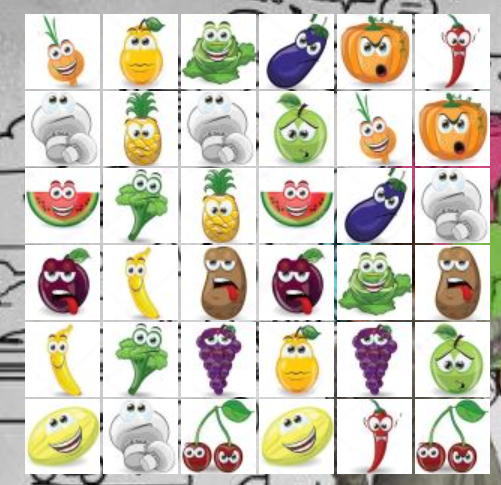


Рисунок 5.6 Картинки фрукты и овощи



Рисунок 5.7 картинки животные



Рисунок 5.8 картинки цвета

Далее пользователь может начать игру. Если пользователь выбирает две одинаковые картинки, они становятся прозрачными , а также сразу считается количество сделанных ходов (ниже игрового поля) (рисунок 5.9)



Рисунок 5.9 Прохождение игры

Обратите внимание, что в верхнем правом углу (при условии, что пользователь зарегистрировался) виден логин пользователя. (рисунок 5.10)

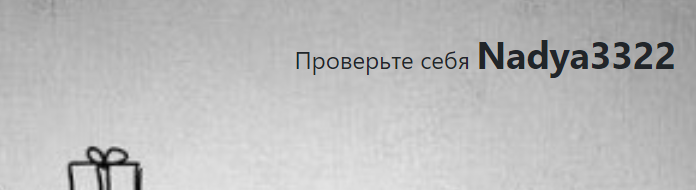


Рисунок 5.10 Отображение логина пользователя

После того, как пользователь пройдет игру высветиться такое диалоговое окно (рисунок 5.11)

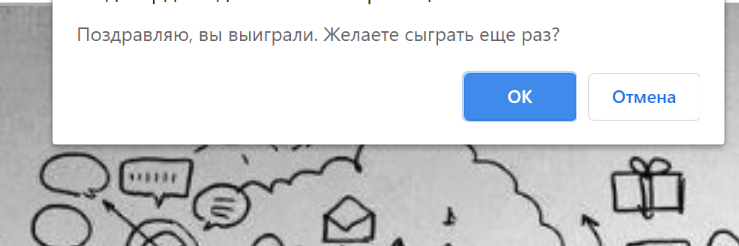


Рисунок 5.11 Диалоговое окно при завершении игры

Если пользователь выбирает «Ок», тогда поле снова заполняется и пользователь может снова сыграть. Иначе поле остается пустым.

Посмотреть всю статистику можно нажав на кнопку «Посмотреть статистику» (рисунок 5.12).

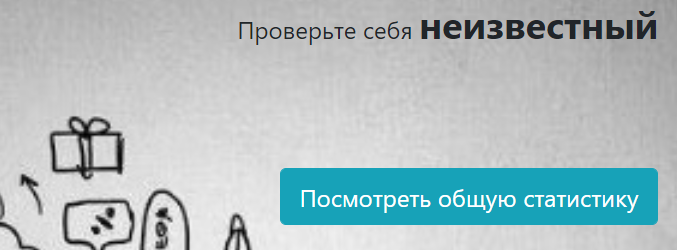


Рисунок 5.12 Посмотреть статистику

В целом игра довольно проста, очень интересна, развивающая память, внимание и концентрацию. Подходит не только детям, но и взрослым.

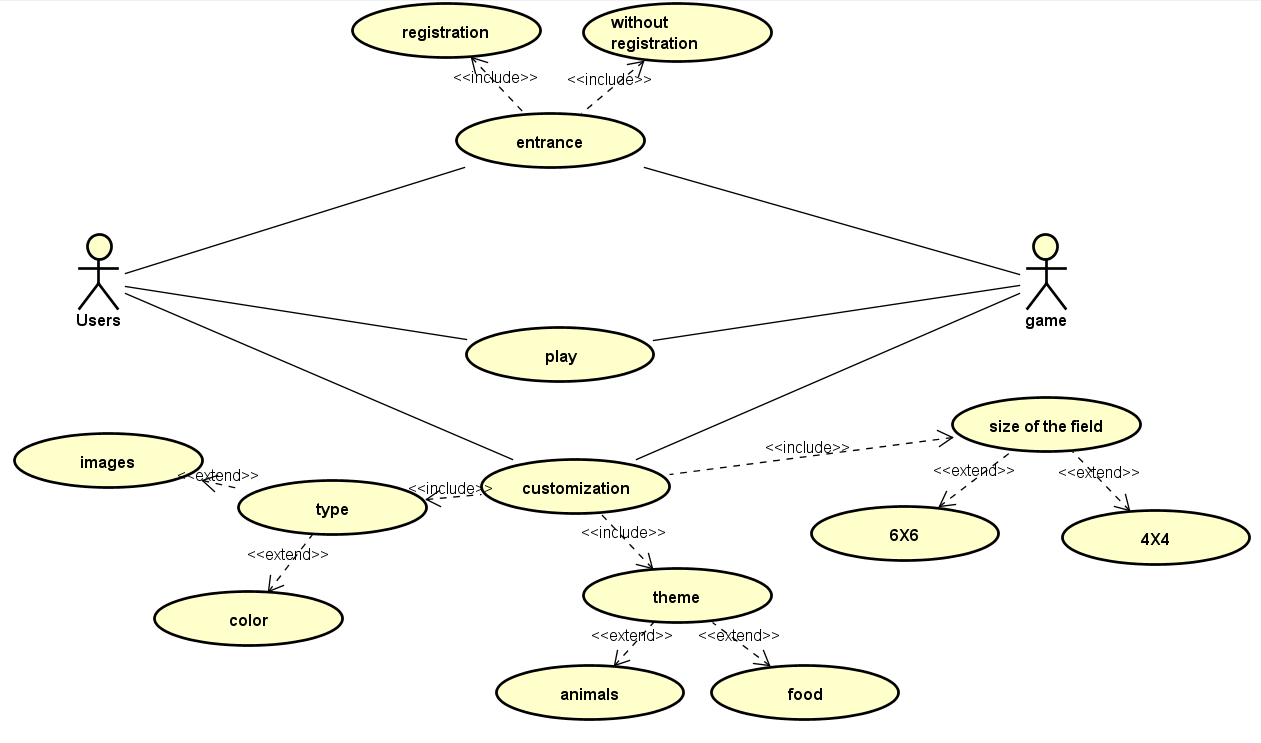
**6 ВЫВОД**

Таким образом, можно сделать вывод, что

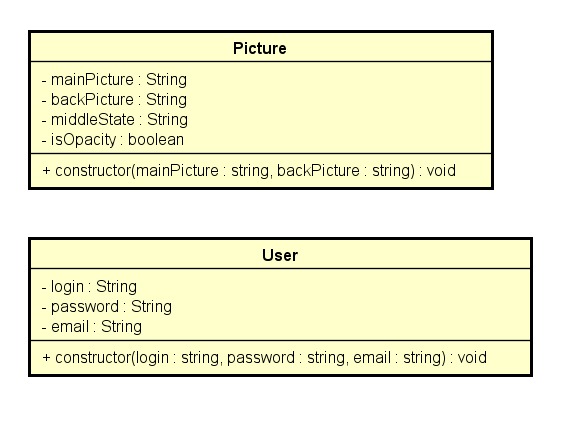
* игра довольно интересна, проста, развивает память, внимание и концентрацию
* игра подходит не только детям, но и взрослым
* можно провести время не только приятно, но и с пользой

**7 ПРИЛОЖЕНИЯ**

Приложение 1. Диаграмма вариантов использования



Приложение 2. Диаграмма классов



Приложение 3. Диаграмма деятельности

