Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

КУРСОВОЙ ПРОЕКТ

Разработка игры "Memory"

по дисциплине «Технологии программирования»

Выполнил студент гр. 3530901/10006 _____Лазарьков Д.С.
(подпись)

Преподаватель _____Степанов Д.С.
(подпись)

“___”_____ 2023 г.

Санкт-Петербург
2023

ЗАДАНИЕ

НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА

студенту группы 3530901/10006

1. Тема проекта: создание игры Memory с графическим интерфейсом

2. Срок сдачи законченного проекта: 31 мая

3. Исходные данные к проекту: требования к реализовываемому проекту

4. Содержание пояснительной записки: введение с описанием правил игры, основная часть (технологии JavaFX, MVC и их применение в приложении), заключение, список используемых источников.

Дата получения задания: «17» апреля 2023 г

Руководитель Степанов Д.С.

Задание принял к исполнению Лазарьков Д.С.

17 апреля 2023

СОДЕРЖАНИЕ

ВВЕДИНЕ

Цель работы: создать и протестировать игру Memory с графическим интерфейсом.

Правила игры:

Мемори или Найди пару – настольная игра, оригинально играющаяся, со специальными или простыми игральными картами. На доске расположено чётное число карт, рубашкой вверх. У каждой карты есть пара – ровно такая же карта. За ход игроку разрешается перевернуть 2 карты, если оказавшиеся карты оказались одинаковы, он забирает их себе, иначе кладёт обратно рубашкой вверх.

В моем варианте игра будет реализована для одного человека, суть игры будет в наборе очков, очки будут зависеть от числа ходов, которые игрок потратит на полную очистку игрового поля.

Игра будет иметь 2 режима игры:

- со звуками (одинаковые карточки при переворачивании издают одинаковый звук)
- с цветами (одинаковые карточки при перевороте имеют одинаковый цвет).

ОПИСАНИЕ РЕШЕНИЯ

Для создания графического пользовательского интерфейса (GUI) использовалась библиотека JavaFX, которая обладает большим числом заготовок элементов интерфейса. Взаимодействие пользователя с графическим элементом описывается в событиях для этого элемента.

Программа была написана с использованием паттерна MVC (model-view-controller) для отделения бизнес-логики от визуализации, поэтому весь код разбит на три пакета: view, mode, controller. Кроме того, так как игра и система подсчёта очков отделены друг от друга каждая из них реализуется своей моделью, а значит и своим контроллером, и отображением.

Работа программы начинается в классе MainApp, задающем параметры окна и создающим View, последующие операции выполняются вне MainApp.

В пакете View описан графический интерфейс программы, включающий в себя основной класс View, а также 2 пакета с классами отображения игрового поля и таблицы игроков. Отображение игры является слушателем модели игры, а отображение подсчёта очков является слушателем модели подсчёта очков.

В пакете Controller также находятся 2 пакета с контроллерами, отвечающими за события внутри игрового поля и события касающиеся таблицы игроков и набора очков соответственно.

В пакете Model содержатся также 2 пакета с бизнес-логикой для самой игры и для действий связанных с подсчётом очков. Модель подсчёта очков, является слушателем модели игры, для возможности подсчитывать очки и изменять данные в базе данных.

В соответствии с выбранным шаблоном разработки, пользователь взаимодействует с отображением, все команды от пользователя обрабатывает контроллер, который в свою очередь обращается к модели, которая подаёт данные об изменениях своим слушателям.

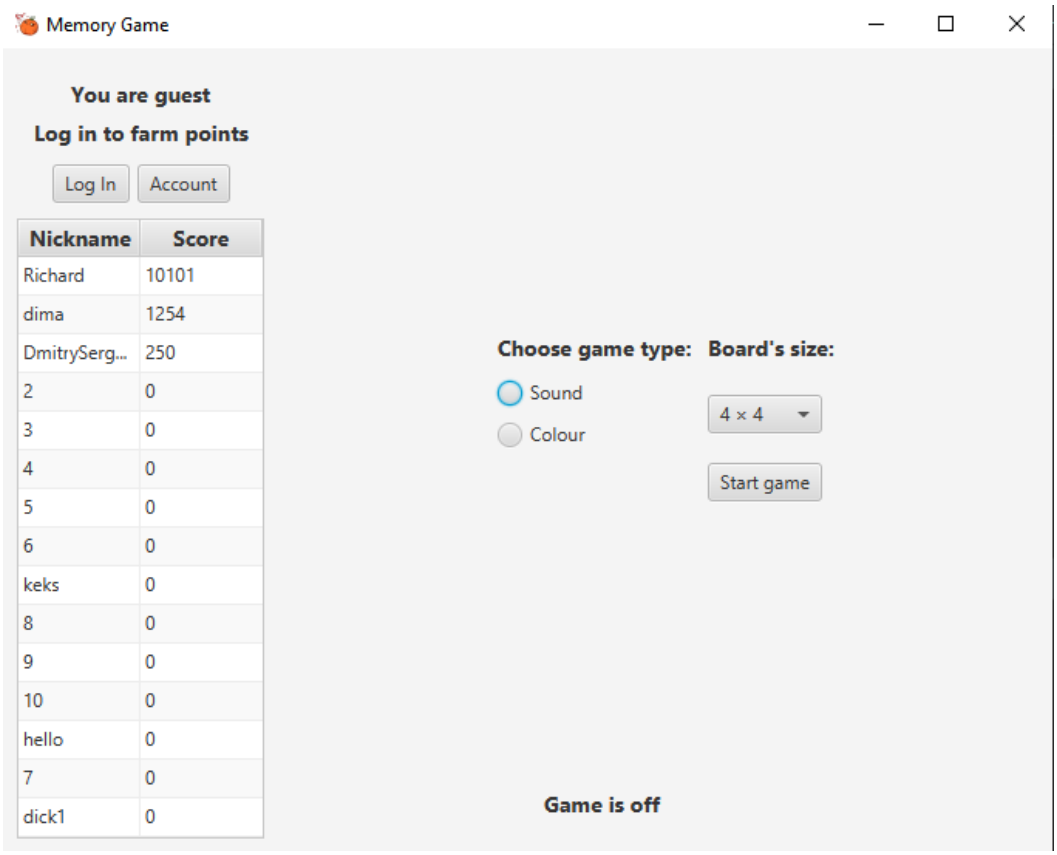
ТЕСТИРОВАНИЕ ПРОГРАММЫ

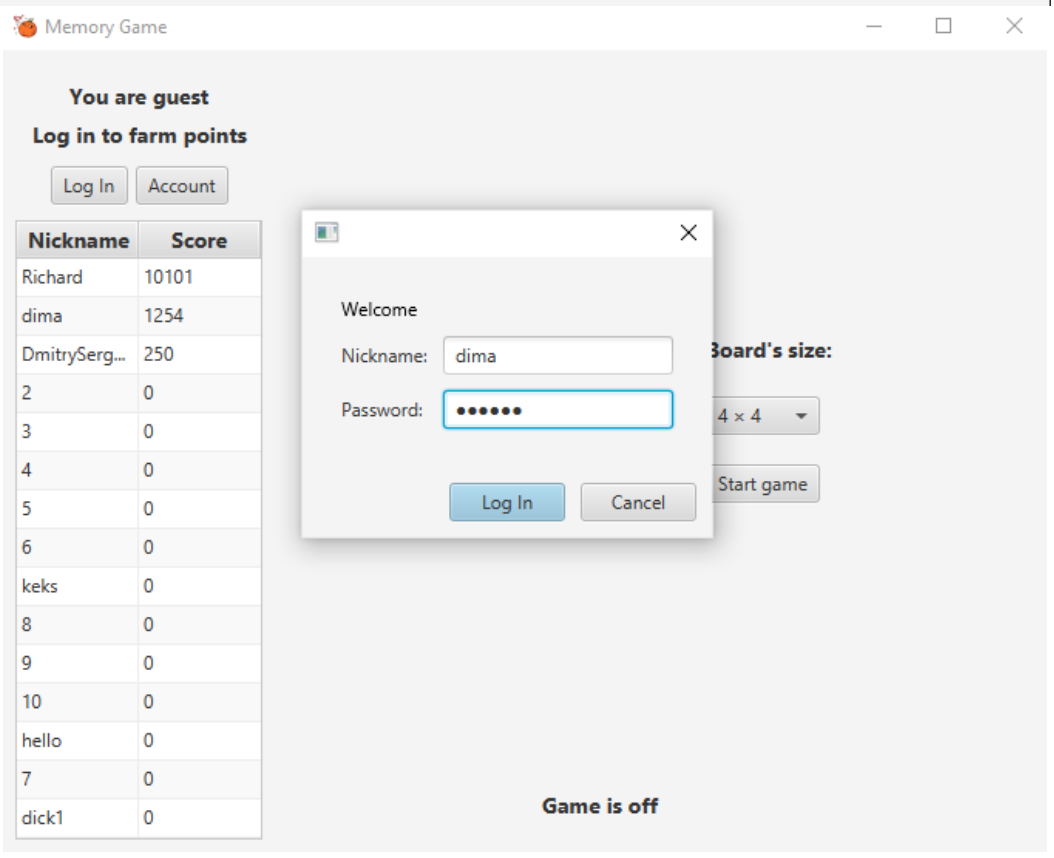
Для тестирования бизнес-логики было написано 3 автоматических теста, использующих возможности библиотеки JUnit.

Первый тест создавал модель игры и проверял правильность создания игрового поля, а именно соответствия числа созданных клеток с ожидаемым, а также то, что у каждой клетки ровно по одной паре. Второй тест проверял аналогичные свойства, но для большего поля и для другого подтипа игры, что позволяет быть уверенным в правильности создания любой игровой доски. Третий тест проверял понимание модели игры о окончании/не окончании игры. Путём идеального решения самой игры, с постоянным вызовом метода isGameIsOn().

ЗАКЛЮЧЕНИЕ

Было создано и протестировано приложение с графическим интерфейсом, предназначенное для игры Memory. В ходе выполнения этого задания мною были изучены библиотека JavaFX и паттерн MVC.





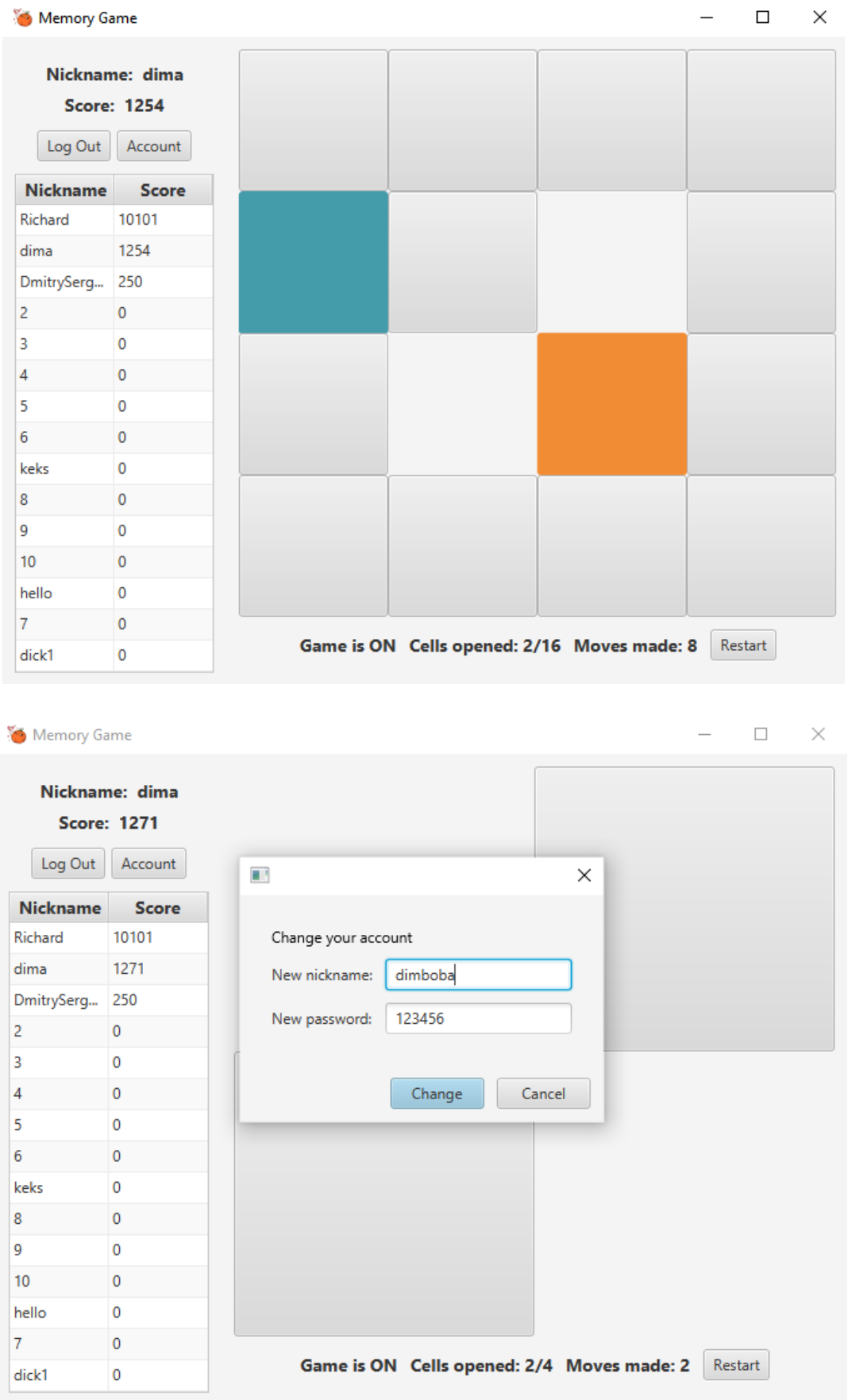


Рис.1-4. Скриншоты приложения

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. https://metanit.com/java/javafx/ – описание и руководство по JavaFX
2. https://docs.oracle.com/javase/8/javafx/api/ – документация по JavaFX