

Санкт-Петербургский политехнический университет

Институт компьютерных наук и технологий

Кафедра «Компьютерные системы и программные технологии»

КУРСОВОЙ ПРОЕКТ

Разработка игры "2048"

по дисциплине «Технологии программирования»

Выполнил студент

гр. 3530901/20002

Захаров И.

Преподаватель

Степанов Д.С.

Санкт-Петербург

2023

Санкт-Петербургский политехнический университет

ЗАДАНИЕ

НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА

студенту группы 3530901/20002 Захарову Ивану

1. Тема проекта: создание игры 2048 с графическим интерфейсом для игры одного человека.
2. Срок сдачи законченного проекта: 31 мая.
3. Содержание пояснительной записки: введение с описанием правил игры, описание предложенного решения, тестирование приложения, заключение, список использованных источников.

Дата получения задания: «21» апреля 2023 г.

Руководитель:

Степанов Д.С.

Задание принял к исполнению:

Захаров И.

«21» апреля 2023 г.

СОДЕРЖАНИЕ

| | |
|-------------------------------------|---|
| ВВЕДЕНИЕ..... | 4 |
| ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ..... | 5 |
| ТЕСТИРОВАНИЕ ПРОГРАММЫ | 6 |
| ЗАКЛЮЧЕНИЕ | 7 |
| СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ..... | 8 |

Введение

Цель работы: создать и протестировать однопользовательскую игру 2048 с графическим интерфейсом.

Правила игры:

Игровое поле имеет форму квадрата 4x4. Целью игры является получение плитки номинала «2048»

- В каждом раунде появляется плитка номинала «2» (с вероятностью 90 %) или «4» (с вероятностью 10 %)
- Нажатием стрелки игрок может скинуть все плитки игрового поля в одну из 4 сторон. Если при сбрасывании две плитки одного номинала «налетают» одна на другую, то они превращаются в одну, номинал которой равен сумме соединившихся плиток. После каждого хода на свободной секции поля появляется новая плитка номиналом «2» или «4». Если при нажатии кнопки местоположение плиток или их номинал не изменится, то ход не совершается.
- Если в одной строчке или в одном столбце находится более двух плиток одного номинала, то при сбрасывании они начинают соединяться с той стороны, в которую были направлены. Например, находящиеся в одной строке плитки (4, 4, 4) после хода влево превратятся в (8, 4), а после хода вправо — в (4, 8). Данная обработка неоднозначности позволяет более точно формировать стратегию игры.
- За каждое соединение игровые очки увеличиваются на номинал получившейся плитки.
- Игра заканчивается поражением, если после очередного хода невозможно совершить действие.

Описание предложенного решения

Для создания графического пользовательского интерфейса (GUI) использовалось приложение Android Studio, которое обладает большим числом заготовок элементов интерфейса а также встроенных библиотек.

В частности были использованы следующие элементы: кнопка (Button), Grid Layout (специальный модуль CSS3, позволяющий позиционировать элементы в виде таблицы либо сетки), компонент отображения текста (TextView), а также компоненты , предназначенные для отображения изображений (ImageView).

Взаимодействие пользователя с графическим элементом описывается в событиях для этого элемента. Существует множество различных событий. Например, на нажатие или смахивание.

Программа была написана с использованием концепции MVC (model - view - controller) для отделения бизнес-логики от визуализации, поэтому проект содержит классы, выполняющие разные функции.

В классах MainGame и MainMenu описывается визуальная часть программы, прописывается расположение и отображаются компоненты, отвечающие за View, их параметры. К большинству элементов привязаны поля и методы, прописанные в классе GameActivity.

Классы GameActivity и MainMenuActivity содержат объявление всех полей и методов, относящихся к элементам интерфейса, описанным в представлении.

Класс Board содержит бизнес-логику работы приложения. В данном классе прописаны методы для правильной работы логики игры , такие как Merge() , Cover_up(), Transpose(), Reverse(), Add_tile().

В соответствии с выбранным шаблоном разработки, пользователь взаимодействует с view, все команды от пользователя обрабатывает controller, который в свою очередь обращается к model и, если это необходимо, перерисовывает view

Тестирование программы

Тестирование программы предстоит провести разработчикам.

Заключение

Было создано приложение с графическим интерфейсом, предназначенное для однопользовательской игры в 2048. Также были разработаны автоматические тесты для проверки работоспособности кода. В ходе выполнения этого задания мною были изучены тонкости работы в Android Studio и шаблон MVC.

В дальнейшем к этому приложению можно добавить возможность выбора размера поля , например 5 X 5 или 8 X 8. При желании есть возможность изменить дизайн приложения, сменив цветовую гамму, расположение элементов и добавив новые функции. Таким образом, приложение имеет потенциал для дальнейшего развития.

Исходный код приложения расположен в репозитории на GitHub:

<https://github.com/IvanZaharov2004/2048>

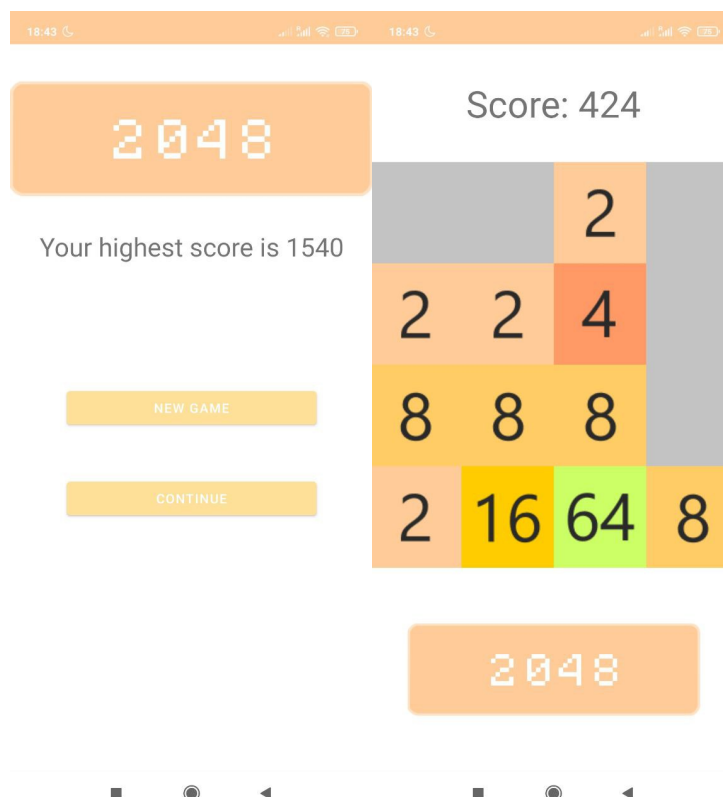


Рис. 1. Скриншоты приложения

Список использованных источников

1. <https://metanit.com/java/tutorial/> - Руководство по языку программирования Java.
2. <https://developer.alexanderklimov.ru/android/studio/> - основы создания графических приложения в Android Studio.
3. <https://2048game.com/ru/> - онлайн игра 2048.
4. <https://developer.android.com/training/data-storage/shared-preferences> - Shared preferences в Android Studio.
5. <https://developer.android.com/training/data-storage/shared-preferences> - обработка жестов в Android Studio.
6. <https://stfalcon.com/ru/blog/post/learning-android-gestures> - овладение жестами в Android Studio.
7. <https://blog.skillfactory.ru/glossary/mvc/> - всё о паттерне MVC.
8. <https://habr.com/ru/articles/524288/> - что такое XML в Android.