

Санкт-Петербургский политехнический университет
Институт компьютерных наук и технологий
Кафедра «Информатики и вычислительной техники»

КУРСОВОЙ ПРОЕКТ
Разработка игры "2048"
по дисциплине «Технологии программирования»

Выполнил студент гр. 3530901/20002

Копейкин Д.В.

Преподаватель

Степанов Д.С.

31 мая 2023 г.

Санкт-Петербург

2023

Санкт-Петербургский политехнический университет

ЗАДАНИЕ

НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА

студенту группы 3530901/20002 Копейкину Даниилу Валерьевичу

1. Тема проекта: создание игры 2048 с графическим интерфейсом для одиночной игры.
2. Срок сдачи законченного проекта: 31 мая
3. Исходные данные к проекту: требования к реализуемому проекту
4. Содержание пояснительной записки: введение с описанием правил игры, основная часть (технологии android studio, MVC и их применение в приложении), описание автоматических тестов для бизнес-логики, заключение, список используемых источников.

Дата получения задания: «28» апреля 2023 г.

Руководитель

Степанов Д.С.

Задание принял к исполнению

Копейкин Д.В.

28 апреля 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4

ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ..... 5

ТЕСТИРОВАНИЕ ПРОГРАММЫ 6

ЗАКЛЮЧЕНИЕ 7

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ8

ВВЕДЕНИЕ

Цель работы: создать и протестировать игру 2048 с графическим интерфейсом для одиночной игры.

Правила игры:

2048 - Игровое поле имеет форму квадрата 4x4. Целью игры является получение плитки номинала «2048» (при желании можно продолжить дальше).

1. В каждом раунде появляется плитка номинала «2» (с вероятностью 90 %) или «4» (с вероятностью 10 %)
2. 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.
3. 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.
4. За каждое соединение игровые очки увеличиваются на номинал получившейся плитки.
5. Игра заканчивается поражением, если после очередного хода невозможно совершить действие.

ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ

Для создания графического пользовательского интерфейса (GUI) использовалась библиотека android studio, которая обладает большим числом заготовок элементов интерфейса. В частности были использованы следующие элементы: кнопка (Button), текст (TextView), сетка/таблица (GridLayout), а также вертикальные ряды (LinearLayout). Взаимодействие пользователя с графическим элементом описывается в событиях для этого элемента. Существует множество различных событий. Например, на нажатие, или на свайп пальца.

Программа была написана с использованием концепция MVC (model-view-controller) для отделения бизнес-логики от визуализации, поэтому весь код разбит на три файла MainActivity, Board и BoardView.

В классе BoardView содержится main функция программы, задание параметров окна и расположения внутри него всех элементов интерфейса, а также прописаны события на действия мыши.

Класс MainActivity содержит объявление всех объектов графического интерфейса, а также методы для работы с ними.

Класс Board содержит переменные и методы, предназначенные для работы бизнес-логики. Все переменные здесь объявлены как приватные. 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

Для тестирования бизнес-логики было написано 5 автоматических тестов, использующих возможности библиотеки JUnit.

Первый тест проверяет добавляет ли плитки. Второй тест проверяет заканчивается ли игра при полностью заполненном поле и наоборот. Третий, четвёртый, пятый и шестой тесты проверяют правильность перемещения во все стороны. Седьмой тест проверяет добавляется ли блок на сетку, если ни один блок не сливается и не перемещается.

ЗАКЛЮЧЕНИЕ

Было создано приложение с графическим интерфейсом, предназначенное для одиночной игры в 2048. Также были разработаны автоматические тесты для проверки работоспособности кода. В ходе выполнения этого задания мною были изучены библиотека android studio и шаблон MVC.

Исходные файлы приложения лежат в репозитории на GitHub:
<https://github.com/DANIK0P/App2048>.

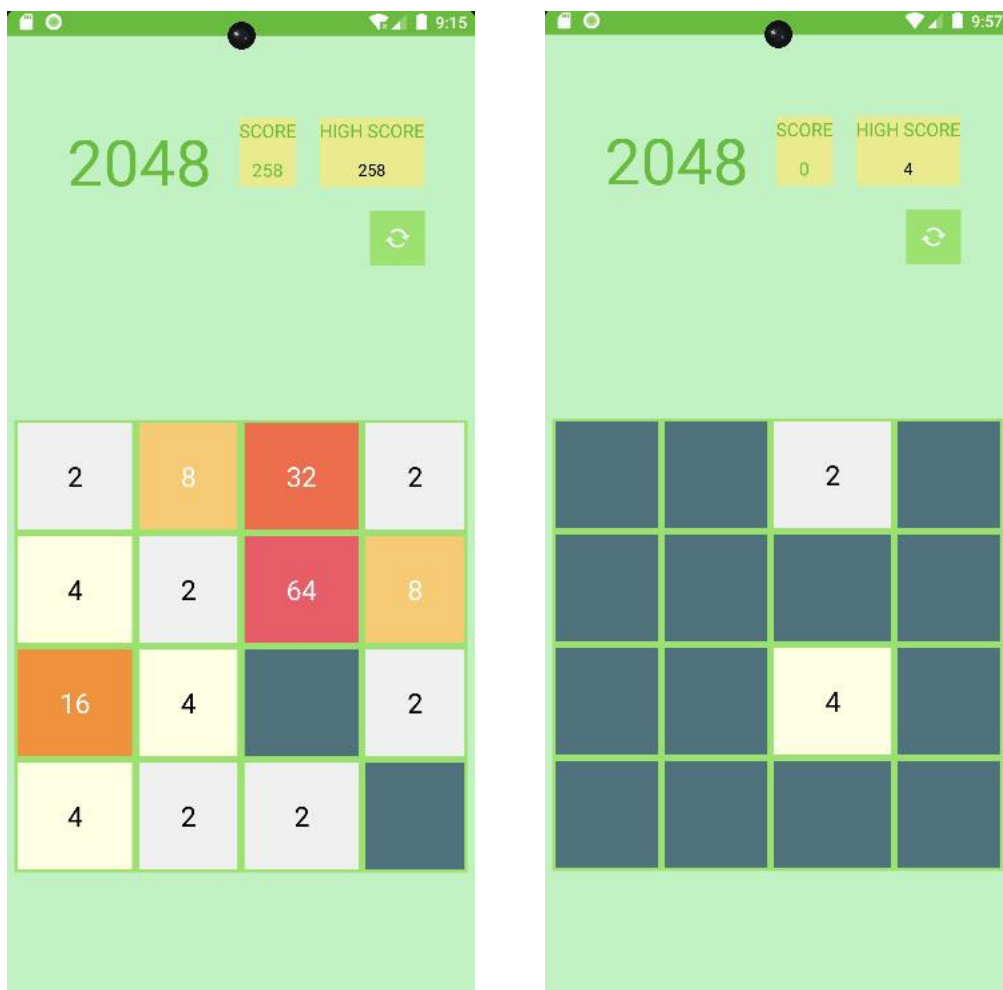


Рис 1. Скриншоты приложения

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. <https://kotlinlang.org> – описание языка Java
2. <https://developer.android.com/reference> – описание Android Studio
3. <https://stackoverflow.com> – описание различных ситуаций и ошибок
4. <https://www.youtube.com> – описание Android Studio
5. <https://play2048.co> – онлайн игра 2048