

Разработка приложения «Blockudoku»

Отчет о проектной работе по курсу «Разработка приложений для мобильных ОС»

Костин Данила

17 января 2022

Цель проекта - реализовать игру «Blockudoku» - используя среду мобильной разработки Android Studio и язык Java.

Задачи проекта:

1. Разработать графический интерфейс приложения с использованием Canvas
2. Реализовать функции обработки нажатия на экран, установки и обработки блоков, случайной генерации блоков.

Описание игрового процесса

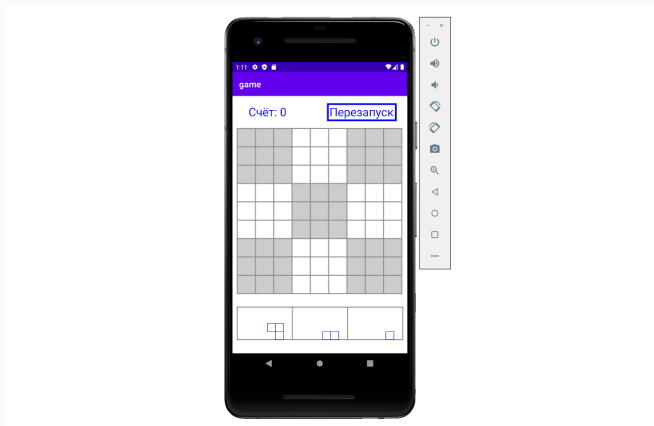


Рис. 1: Игровое окно

Описание игрового процесса

Название игры «Blockudoku» - состоит из слов «судоку» и «блок». Игроку предоставлен выбор одной из трёх фигур, составленных из блоков. Цель игрока - заполнить блоками любой квадрат 3 на 3 (квадраты отличны цветами). После заполнения квадрата, он пустеет, игроку добавляются очки. Сложность в том, что блоки могут генерироваться различного размера, и игроку просто не хватит места для установки блока. В этом случае произойдёт проигрыш.

1. Игрок выбирает 1 из 3 фигур
2. Игрок нажимает на клетку, куда хочет разместить фигуру
3. В клетку нажатия размещается правый нижний элемент фигуры, остальные блоки фигуры достраиваются вслед
4. Генерируются новые блоки

Было создано 3 модуля

1. MainActivity - основной модуль активности
2. GameView - модуль, содержащий функцию отрисовки, а также все функции взаимодействия с интерфейсом
3. GameThread - модуль нового потока, производящего отрисовку Canvas

Данный модуль - почти ничем не отличается от автоматически сгенерированного модуля MainActivity. Отличие состоит в том, что функция `setContentView()` принимает экземпляр объекта `SurfaceView` - реализованный на Canvas интерфейс игры.

В данном модуле реализована работа второго потока, который производит отрисовку интерфейса игры. Модуль представляет собой класс, унаследованный от Thread. В данном классе определены методы `setRunning()` - устанавливающий флаг того, что процесс может работать, а также `run()` - функция, создающая пустой элемент Canvas, и передающая его в функцию `onDraw()`, определённую в модуле `GameView`, рисующую интерфейс игры.

Данный модуль выполняет все основные функции управления игрой. Он содержит:

1. Класс GameView, реализующий интерфейс SurfaceHolder.Callback
2. Класс представления ячеек поля - GameLogic, и класс Destinations - содержит данные о направлениях для сгенерированных фигур.
3. Различные функции работы приложения, например onTouchEvent(), onDraw(), generateBlocks(), и тд.

Это класс, реализующий интерфейс `SurfaceHolder.Callback`. Этот интерфейс реализуется тремя методами: `surfaceCreated()`, `surfaceChanged()` и `surfaceDestroyed()`, вызываемые соответственно при создании области для рисования, ее изменении и разрушении.

Метод `surfaceCreated()` создаёт новый поток, который обновляет `Canvas`, вызывая функцию `onDraw()`.

Метод `surfaceDestroyed()` останавливает поток, например, при выходе из приложения на начальный экран.

Классы GameLogic и Destinations

Класс GameLogic содержит такие поля, как `int Left, Right, Top, Bottom`, соответственно означающие координаты сторон, используемые функцией рисования прямоугольников на canvas - `drawRect()`, а также поле `colored` - если оно равно 0, то клетка не окрашена, 1 - окрашена.

Класс Destinations содержит массив `int`, в котором содержатся числа - либо 1, либо 2, генерируемые псевдослучайно, 1 означает направление вверх, 2 - влево. Эти числа используются для построения фигуры из блоков на поле и их визуального уменьшенного представления для игрока.

Функция onDraw()

Данная функция использует данные из класса GameLogic: клетки поля представлены в виде двухмерного массива GameLogic[9][9], функция проходится по всем элементам данного массива классов, и, используя данные о координатах сторон клеток, отрисовывает на canvas с помощью функции drawRect() квадраты поля. При этом, если поле colored равно 1, то при рисовании прямоугольника используется Paint с заливкой в синий цвет.

Окрашенные клетки



Функция onTouchEvent()

Данная функция обрабатывает нажатие на экран. Короткое описание происходящих в функции действий:

- Получает координаты нажатия
- Проверяют попали ли координаты в конкретную ячейку
- Если попадают, то происходит проверка возможности поставить выбранную фигуру
- Если фигура помещается, то происходит перебор всех подходящих под фигуру клеток, в соответствующем элементе класса GameLogic , colored = 1.
- Вызывается функция проверки заполненность квадратов 3 на 3
- Вызывается функция генерации новых фигур
- Происходит проверка, что новые фигуры можно разместить в пустых ячейках. Если не возможно - то вывод сообщения о проигрыше.

Функция `inside()`

Данная функция проверяет, что координаты нажатия находятся в пределах какой-либо клетки по следующему правилу: x , y - координаты места нажатия. Если

$$Right - x > 0 \ \&\& \ Left - x < 0 \ \&\& \ Top - y < 0 \ \&\& \ Bottom - y > 0$$

тогда место клика принадлежит текущей клетке.

Функция `suitable()`

Данная функция получает на вход индексы i, j клетки из двумерного и проверяет для текущей клетки, что возможно размещение выбранной игроком фигуры с текущей клетки. (Размещение фигуры происходит из правого нижнего блока фигуры в клетку клика, далее фигура достраивается от данного начального блока фигуры).

Функция берёт данные о направлении построения фигуры из класса `Destinations`, перебирает элементы его массива. Если элемент равен 1, то осуществляется переход к `GameLogic [i-1][j]` - то есть к клетке выше, если 2 - то к `GameLogic [i][j-1]` (левой клетке). Если её `colored = 0`, то функция продолжает работу.

Она возвращает `true`, если фигуру можно построить от данной клетки.

Красная точка - место нажатия



Попытка поставить фигуру в невозможно место (красная точка)



Функция generateBlocks()

Функция, которая генерирует псевдослучайное число от 1 до 4 - это число - размер фигуры в блоках, которая будет отображена в панели выбора. Также это число означает длину массива в классе Destinations.

Затем, для каждого элемента массива, функция генерирует псевдослучайное число - 1 или 2 - означающее направление, в котором будет построен блок с номером k.

Для генерации чисел используется функция Math.random().

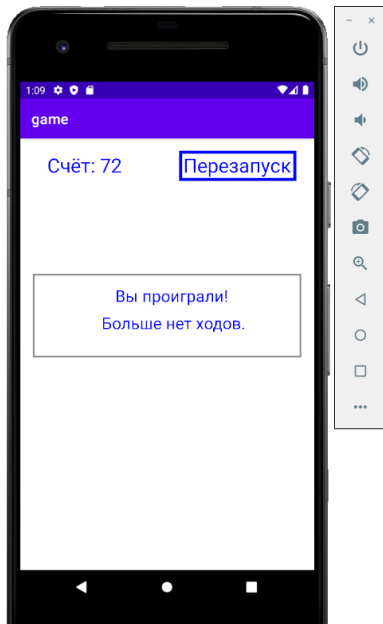
Функция checkKub()

Функция, которая проверяет каждый квадрат клеток 3 на 3 (из поля 9 на 9), если все элементы в классе GameLogic имеют colored = 1, то происходит увеличение счёта и установка для всех этих клеток colored = 0 - то есть очистка.

Функция `loss()`

Функция, которая определяет пришёл ли игрок к такому состоянию поля, при котором невозможна установка любой из 3 сгенерированных фигур. Реализация функции проста - происходит проход всех клеток, а также для любой из доступных для выбора фигур проверяется возвращаемое значение функции `suitable()`. (То есть проверяется возможность успешной установки фигуры в какую-нибудь клетку). Если таковой клетки нет, происходит вывод сообщения о проигрыше.

Сообщение о проигрыше



В заключение можно сказать, что было создано мобильное приложение (аналог игры Blockdoku из Playmarket), на языке Java. Данное приложение получилось относительно небольшим, количество взаимодействующих функций было невелико, так как в основном функции вызываются после нажатия на экран.

