

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук  
Образовательная программа «Прикладная математика и информатика»

Отчет о программном проекте на тему:  
Валидное sudoku (Valid Sudoku)

**Выполнил:**

студент группы БПМИ217  
Сахаров Роман Максимович

**Принял руководитель проекта:**

Сосновский Григорий Михайлович  
Преподаватель  
Факультета компьютерных наук НИУ ВШЭ

# Содержание

<b>Аннотация</b>	<b>3</b>
<b>1 Введение</b>	<b>4</b>
1.1 Описание предметной области . . . . .	4
1.2 Актуальность . . . . .	4
1.3 Цель . . . . .	4
1.4 Задачи . . . . .	5
<b>2 Обзор и сравнительный анализ аналогов</b>	<b>6</b>
2.1 «Судoku – классическое судoku» . . . . .	6
2.2 «Sudoku – The Clean One» . . . . .	6
2.3 «Судoku Мир» . . . . .	6
2.4 Обобщение . . . . .	7
2.5 «Valid Sudoku» . . . . .	7
<b>3 Алгоритм</b>	<b>8</b>
3.1 Мотивация . . . . .	8
3.2 Идея алгоритма . . . . .	8
3.3 Подробное описание алгоритма . . . . .	8
<b>4 Выбор стека и технологий</b>	<b>11</b>
4.1 Flutter . . . . .	11
4.2 Hive . . . . .	12
<b>5 Архитектура</b>	<b>13</b>
5.1 Backend . . . . .	13
5.2 Frontend . . . . .	13
<b>6 Обзор разработанного приложения</b>	<b>14</b>
<b>7 Заключение</b>	<b>15</b>
<b>Список литературы</b>	<b>16</b>
<b>Приложения</b>	<b>17</b>

## Аннотация

Идея проекта заключается в создании мобильного приложения для игры в sudoku. Для разработки был выбран фреймворк Flutter, благодаря чему приложение получилось кроссплатформенным.

Файлы программного кода можно условно разделить на backend- и frontend-часть. На уровне бэкэнда в программе реализован алгоритм генерации валидных sudoku разного размера и сложности, представление sudoku в программе, а также механизмы взаимодействия с базой данных, предназначенной для хранения информации о предыдущих играх.

Frontend был реализован в виде нескольких классов, описывающих отдельные окна приложения и их взаимодействие и обращение к интерфейсам, предоставленным классами, описанными в бэкэнде.

Помимо непосредственно игрового интерфейса для решения генерируемых программой sudoku пользователю предоставляется доступ к его статистике по играм, а также выбор темы оформления.

Продукт монетизируется за счёт баннерной рекламы от Яндекса.

Итоговое приложение доступно в магазине RuStore [10] по ссылке [https://apps.rustore.ru/app/com.ramses44.valid\\_sudoku](https://apps.rustore.ru/app/com.ramses44.valid_sudoku).

## Ключевые слова

Мобильная разработка, разработка игр, генерация и решение sudoku, Flutter

# 1 Введение

## 1.1 Описание предметной области

За последнее время мобильные гаджеты успели стать важной частью нашей жизни. Теперь телефон – это и средство общения, и рабочая станция, и фотоаппарат, и медиаплеер, и средство для развлечения. В связи с этим все более востребованными становятся мобильные приложения, призванные расширить привычный функционал устройства, дать пользователю новый опыт. В частности, особой популярностью пользуются различные развлекательные приложения. К ним относятся игры, сервисы для просмотра видео-контента и прослушивания музыки, социальные сети. Но наиболее «классическим» и неустаревающим представителем в этой области являются всё-таки игры.

Существует множество жанров мобильных игр [5]: шутеры, стратегии, MMORPG, головоломки, симуляторы и т.д. Остановим среди них внимание на головоломках. Этот жанр пользуется большой популярностью среди игроков [12]. Вот некоторые широко известные головоломки [11]:

- Сканворды
- Кроссворды
- Ребусы
- Судоку
- Японские кроссворды
- «Три в ряд»

## 1.2 Актуальность

Игры-головоломки относительно просты в разработке и при этом пользуются популярностью. Совокупность всех вышеописанных факторов делает создание мобильного приложения с головоломкой **актуальной** задачей.

## 1.3 Цель

Из вышеперечисленных головоломок особенно удобными для реализации являются японские кроссворды и судоку, т.к. не требуют сложной разработки дизайна или предварительного создания большой базы вопросов и загадок с ответами. Для генерации полей для

этих игр можно использовать программные алгоритмы, что также даёт большее разнообразие игрового опыта для пользователя.

Но более популярной<sup>1</sup> головоломкой, и, следовательно, более перспективной для разработки является sudoku.

Таким образом, **цель проекта** – реализовать мобильное приложение для игры в sudoku.

## 1.4 Задачи

Для достижения поставленной цели потребовалось выполнить ряд задач:

1. Провести сравнительный анализ существующих решений и выдвинуть требования к собственному приложению
2. Выбрать инструментарий для создания программного продукта
3. Продумать архитектуру будущего приложения
4. Изучить алгоритмы решения и генерации sudoku, реализовать их (backend)
5. Реализовать пользовательский интерфейс программы (frontend)
6. Оформить готовое приложение
7. Встроить монетизацию за счет платных оформлений или рекламы
8. Запустить продукт в общее пользование (загрузить в магазин приложений)

---

<sup>1</sup>На основании скачиваний и рейтинга в Google Play [2], Apple Store [6] и RuStore [10]

## 2 Обзор и сравнительный анализ аналогов

В настоящий момент уже существуют успешные решения. Рассмотрим некоторые из них.

### 2.1 «Судоку – классическое судоку»

Этот продукт от компании *Easybrain* является одним из самых популярных среди аналогов (*50+ млн скачиваний в Google Play [2]*).

#### Плюсы:

1. Выбор нескольких вариантов размера и сложности судоку
2. Возможность сбора статистики по играм

#### Минусы:

- Отсутствие возможности поменять визуал (тему)

### 2.2 «Sudoku – The Clean One»

Это приложение от разработчика *Dustland Design* тоже достаточно популярно (*5+ млн скачиваний в Google Play [2]*).

#### Плюсы:

1. Современный дизайн с разными темами
2. Хорошая производительность

#### Минусы:

1. Минимальный функционал
2. Отсутствие поддержки русского языка

### 2.3 «Судоку Мир»

Это уже намного менее популярный представитель (*50+ тыс. скачиваний в Google Play [2]*). Разрабатывается программистом *Мнемопето Tree*.

#### Плюсы:

1. Сбор статистики по играм
2. Оригинальные звуковые эффекты

#### Минусы:

1. Нестабильная работа
2. Только один размер судоку

## 2.4 Обобщение

На рынке присутствуют достойные продукты, но и они имеют недостатки. Главные удобства и удачные особенности, которые можно выделить:

1. Выбор разных уровней сложности
2. Возможность брать подсказки
3. Автоматическая валидация<sup>2</sup> не полностью решенного sudoku
4. Возможность делать «заметки карандашом»<sup>3</sup>
5. Статистика по играм
6. Возможность менять темы оформления под вкус пользователя
7. Поддержка разных языков

Таким образом, при разработке необходимо было учесть минусы существующих решений и по возможности перенять их удачный опыт.

## 2.5 «Valid Sudoku»

На основании выводов, сделанных из сравнительного анализа уже имеющихся на рынке продуктов, к разрабатываемому приложению были выдвинуты следующие требования:

### Функциональные

1. Возможность генерировать разные по сложности sudoku
2. Наличие механики подсказок
3. Наличие механики «карандашных пометок»
4. Возможность сбора статистики по играм
5. Автоматическая валидация по ходу решения sudoku пользователем
6. Возможность выбора тем [1]
7. Поддержка нескольких языков (как минимум русского и английского)

### Нефункциональные

1. Кроссплатформенность (для большего охвата аудитории)
2. Монетизация за счёт рекламы или платных тем

При разработке были учтены эти требования. В то же время, ни один из проанализированных аналогов не обладает ими всеми одновременно.

---

<sup>2</sup>В данном случае имеется в виду проверка sudoku на валидность с учётом заполненных пользователем ячеек

<sup>3</sup>Имеется в виду возможность вписать в ячейку несколько предполагаемых пользователем вариантов чисел. При этом при валидации программой эти заметки учитываться не будут

## 3 Алгоритм

### 3.1 Мотивация

Цель приложения – предоставить пользователю возможность разгадывать sudoku. Для этого есть три возможности:

1. Хранить заранее подготовленный набор готовых головоломок
2. Генерировать sudoku на сервере и отправлять по запросу с пользовательского устройства
3. Генерировать их прямо на устройстве пользователя

Главный недостаток первого подхода заключается в отсутствии разнообразия, а также увеличенном пользовательского приложения. Во втором случае существенным минусом является необходимость доступа к серверу (т.е. в случае отсутствия сети или неполадок на сервере приложение перестанет работать). Таким образом остаётся третий вариант.

Итак, приложение должно генерировать sudoku по запросу от пользователя, притом делать это достаточно быстро, чтобы пользователю не приходилось ждать.

### 3.2 Идея алгоритма

В основе большинства алгоритмов для генерации sudoku [3] лежит следующий принцип. Сначала создаётся заполненное по всем правилам поле требуемого размера. Затем из случайных ячеек последовательно удаляются числа и производится проверка на валидность. Так продолжается пока sudoku не окажется нужной сложности. Если на очередной итерации sudoku перестало быть валидным (т.е. появился вариант решения альтернативный исходному), то последнее действие отменяется и повторяется уже с другим числом на поле.

### 3.3 Подробное описание алгоритма

#### 1 Этап (генерация решенного sudoku)

Для начала возьмём некое “базовое” sudoku: просто заполним строки числами от 1 до N (где N – размер поля) с циклическим сдвигом равным для k-й (начиная с 0) строки:

$$\delta_k = \lfloor k/\sqrt{N} \rfloor + (k \bmod \sqrt{N}) * \sqrt{N}$$

Т.е. для размера 4x4 результатом будет следующее поле (рис.3.1):



1	2	3	4
3	4	1	2
2	3	4	1
4	1	2	3

Рис. 3.1: Базовое поле 4 x 4.

Далее выполним несколько итераций, на каждой из которых будем производить одно из следующих действий:

1. Транспонирование всей таблицы (строки становятся столбцами, а столбцы – строками)
2. Отражение таблицы по вертикали/горизонтали
3. Обмен местами двух строк (или столбцов) в пределах одного блока (последовательных строк/столбцов, для каждого  $S_i$  и  $S_j$  из которых:  $i \bmod \sqrt{N} == j \bmod \sqrt{N}$ )
4. Обмен местами двух блоков строк или столбцов
5. Обмен двух чисел местами (все числа А становятся числами В и наоборот)

Важно, что каждое из этих преобразований поддерживает инвариант корректности sudoku. Таким образом получаем заполненное поле.

## 2 Этап (генерация нерешенного sudoku заданной сложности)

Для каждого вида сложности (возьмём 3: лёгкий, средний, сложный) определим количество (или процент) изначально заполненных ячеек [7] (37-43%, 31-37% и 21-29% соответственно). Будем удалять из случайных ячеек таблицы числа до тех пор, пока не достигнем нужного кол-ва. При этом на каждой итерации будем проверять решается ли sudoku методом выбранной сложности (для этого заранее определим алгоритмы решения для каждого уровня сложности). Если на очередном шаге получили невалидное/нерешаемое sudoku, то возвращаем в таблицу удалённое число и пробуем удалить другое.

Для разных уровней сложности были реализованы следующие алгоритмы [7]:

Лёгкий: наивный алгоритм, на каждой итерации проверяющий есть ли пустые ячейки, числа в которые вписывается однозначно (т.е. есть только одно число, подстановка которого не приведет к коллизии на той же итерации). Если таких ячеек нет, то алгоритм признаёт sudoku нерешаемым.

Средний: на каждой итерации сначала заполняет все ячейки, которые получается, наивным алгоритмом. Далее действует методом перебора: в ячейку с наименьшим количеством

вариантов на данном шаге по очереди подставляет все возможные числа, для каждой подстановки запуская новую итерацию. Если за  $\sqrt{N}$  итераций решение не найдено, алгоритм признаёт sudoku нерешаемым.

Сложный: действует так же, как и предыдущий за исключением того, что имеет больший лимит итераций в переборе решений. Если за  $N^2$  итераций (уровней перебора) не находится решение, то алгоритм признаёт sudoku нерешаемым. Таким алгоритмом решается любое валидное sudoku.

Важно отметить, что для решения sudoku алгоритм для средней и сложной сложности запускается два раза: в первый раз перебор идёт в прямом порядке (т.е. последовательно подставляются возможные числа от 1 до N), а во второй – в обратном (от N до 1). Таким образом, если у sudoku больше одного решения, то при двух запусках алгоритм найдёт разные решения. И наоборот: если алгоритм в обоих случаях получил одно и то же решение, то оно единственно.<sup>4</sup>

После реализации данного алгоритма были проведены тесты производительности (в однопоточном режиме из-за особенностей языка Dart, на котором и разрабатывалось приложение). Результаты показали, что sudoku размера 4x4 генерируются почти мгновенно, размера 9x9 – в зависимости от сложности, но не более 1 секунды. С sudoku 16x16 ситуация оказалась сложнее: головоломка лёгкой сложности генерировалась также не более секунды, на создание sudoku 16x16 средней сложности в среднем уходило несколько минут, для третьего уровня сложности тесты провести не удалось, т.к. программа работала больше нескольких часов. Возможность создания и решения sudoku больших размеров было решено исключить из приложения.<sup>5</sup>

Таким образом был реализован и впоследствии использован в приложении алгоритм генерации sudoku на языке Dart.

---

<sup>4</sup>Доказательство этого факта изложено в статье [13]

<sup>5</sup>Все тесты проводились на ПК с процессором Ryzen 5 5600H. После, во время тестирования готового приложения на смартфоне Google Pixel 5 подтвердилось, что все виды sudoku кроме 16x16 средней и сложной сложности генерируются меньше, чем за секунду

## 4 Выбор стека и технологий

### 4.1 Flutter

Основным инструментом для разработки был выбран Flutter [9]. Flutter – это фреймворк для языка Dart [8] с открытым исходным кодом от Google для создания красивых, нативно скомпилированных, многоплатформенных приложений из единой кодовой базы.

Рассмотрим плюсы и минусы Flutter по сравнению с конкурентами.

#### Плюсы:

##### 1. Относительно простой и продуманный язык

*Dart – язык, созданный компанией Google в 2011 году, как альтернатива JavaScript, который по мнению разработчиков “имеет фундаментальные изъяны”. Опыт других языков позволил программистам из Google собрать положительные черты, а также учесть и исправить некоторые их недостатки.*

##### 2. Кроссплатформенность

*Приложение запускается на виртуальной машине Dart с собственным движком. Это позволяет создавать приложения, одновременно работающие на Android, IOS и Fuchsia.*

##### 3. Модульное программирование интерфейса

*Flutter позволяет разбивать интерфейс на виджеты, представленные в виде классов и работать с ними.*

##### 4. Перспективность и активное развитие

*Flutter – молодой фреймворк, разрабатываемый и активно поддерживаемый Google. С учётом вышеперечисленных плюсов, Flutter, вероятно, только укрепится на рынке и продолжит своё развитие.*

#### Минусы:

##### 1. Мало библиотек

*Т.к. Dart – достаточно молодой язык, то для него пока что написано не так много библиотек, хотя все самые необходимые уже есть.*

##### 2. Небольшое комьюнити

*Но активно развивающееся.*

##### 3. Худшая производительность по сравнению с нативными приложениями

*Т.к. Flutter-приложения запускаются на виртуальной машине, то это создаёт дополнительную нагрузку на производительность.*

#### 4. Большой вес конечного пакета приложения

*Т.к. в пакет необходимо включать ещё виртуальную машину Dart, а не только само приложение*

На основании всех вышеперечисленных плюсов и минусов и был выбран Flutter.

## 4.2 Hive

Для хранения данных об играх потребовалось использование базы данных. В качестве БД была выбрана Hive.

Hive — это легкая и молниеносно быстрая база данных «ключ-значение», написанная на чистом Dart. [4]

Основные преимущества Hive:

1. Кроссплатформенность
2. Отличная производительность (рис.4.1):
3. Отсутствие нативных зависимостей
4. Встроенное шифрование
5. Удобный API для работы из Flutter



Рис. 4.1: Производительность Hive БД.

Таким образом для разработки приложения были выбраны фреймворк Flutter и база данных Hive для языка Dart.

## 5 Архитектура

### 5.1 Backend

В программе реализован класс `Sudoku`, задачей которого является хранение всей информации о конкретном sudoku и предоставление интерфейса для взаимодействия других элементов программы с sudoku. Он содержит размер игрового поля и матрицу из экземпляров класса `Cell`, в которых хранятся данные о находящемся в клетке числе (или его отсутствии), числе, которое должно стоять в клетке в решенном sudoku, «заметках» в этой клетке, а также поставлено ли число в клетку программой-генератором или пользователем.

В том же модуле представлены классы `SudokuSolver` и `SudokuBuilder`, отвечающие соответственно за решение sudoku и генерацию нового sudoku по размеру и сложности.

Для взаимодействия с БД необходимы классы моделей (в данном случае классы `SudokuGameModel` и `Settings`), отвечающие за хранение в БД информации об играх sudoku и пользовательских настройках.

Также в программе представлен класс `SudokuGame`, предоставляющий программный интерфейс к игре (набор методов, вызываемых непосредственно frontend-частью) и обеспечивающий синхронизацию игры с базой данных.

### 5.2 Frontend

Пользовательский интерфейс программы представлен несколькими окнами: `MainScreen` (главное окно с кнопками перехода к другим окнам и выбором параметров для генерации sudoku), `GameScreen` (окно с непосредственно с игровым интерфейсом), `SettingsScreen` (окно настроек), `InfoScreen` (окно с информацией о разработчиков/правилами игры) и `StatScreen` (окно со статистикой). Реализация каждого из окон вынесена в отдельный файл. `MainScreen` — домашнее окно, из которого вызываются все остальные окна.

Также к frontend-части можно отнести файлы `labels.dart` и `themes.dart` с переводами надписей для кнопок и других элементов интерфейса и описанием тем оформления приложения соответственно.

## 6 Обзор разработанного приложения

При входе в приложение пользователь попадает на главный экран (*прил. 7.1*). На нём располагаются выпадающие списки с выбором размера и сложности sudoku для новой игры, кнопки создания новой игры, продолжения старой и просмотра статистики. Также в верхней панели приложения находятся кнопки получения информации по правилам игры в sudoku, настроек и получения информации о приложении и разработчике.

В игровом окне (*прил. 7.2*) пользователь непосредственно получает возможность разгадывания sudoku. Помимо игрового поля в окне присутствует «клавиатура» с доступными для данного sudoku цифрами/буквами, а также кнопки «Ластик», «Заметки карандашом» и «Взять подсказку». В верхней части находится таймер, показывающий время с начала решения головоломки. При нажатии кнопки «Пауза» в правом верхнем углу таймер ставится на паузу, к окну применяется blur-эффект, а поверх него появляется диалоговое окно для снятия с паузы (*прил. 7.3*). По завершении игры (когда всё поле верно заполнено пользователем) элементы управления исчезают, а вместо них появляется статистика по прошедшей игре (время, кол-во ошибок и взятых подсказок), далее пользователь может вернуться на главный экран (*прил. 7.4*).

В окне настроек (*прил. 7.5*) есть всего три выпадающих меню непосредственно с настройками: язык приложения, режим игры (указывать ли пользователю на ошибки сразу во время решения) и выбор темы.

В окне со статистикой (*прил. 7.6*) можно выбрать в выпадающих списках размер и сложность sudoku-игр, по которым пользователь хочет получить статистику (можно также выбрать «Все» и получить статистику по всем играм).

## 7 Заключение

В ходе работы последовательно выполнялись поставленные задачи. Были проанализированы аналоги, выделены их основные плюсы и минусы, на основе которых сформулированы требования к готовому продукту. Для разработки был выбран и использован фреймворк Flutter, исходя из его особенностей намечена архитектура будущего приложения. Была реализована backend-часть программы, а затем интерфейс. В приложение были добавлены настройки, темы, сбор статистики, реклама. После успешного тестирования продукт был загружен в магазин Android-приложений RuStore.

Особенности разработанного приложения:

1. Позволяет пользователю выбрать сложность и размер sudoku
2. Позволяет пользователю брать подсказки
3. Предоставляет возможность делать заметки в пустых клетках sudoku (записывать туда несколько вариантов чисел без проверки со стороны программы)
4. Сбор статистики по играм
5. Возможность указания на ошибки во время решения
6. Несколько тем оформления приложения
7. Поддержка Русского и Английского языков
8. Кроссплатформенность
9. Монетизация за счёт рекламы

Готовый продукт соответствует предъявленным требованиям (2.5).

Все поставленные задачи были выполнены, цель проекта – реализация мобильного приложения для игры в sudoku – была достигнута. В будущем проект планируется развивать, обновляя приложение. Можно добавить больше тем оформления, дополнительные языки, специальные анимации игрового процесса, звуковые эффекты.

## Список литературы

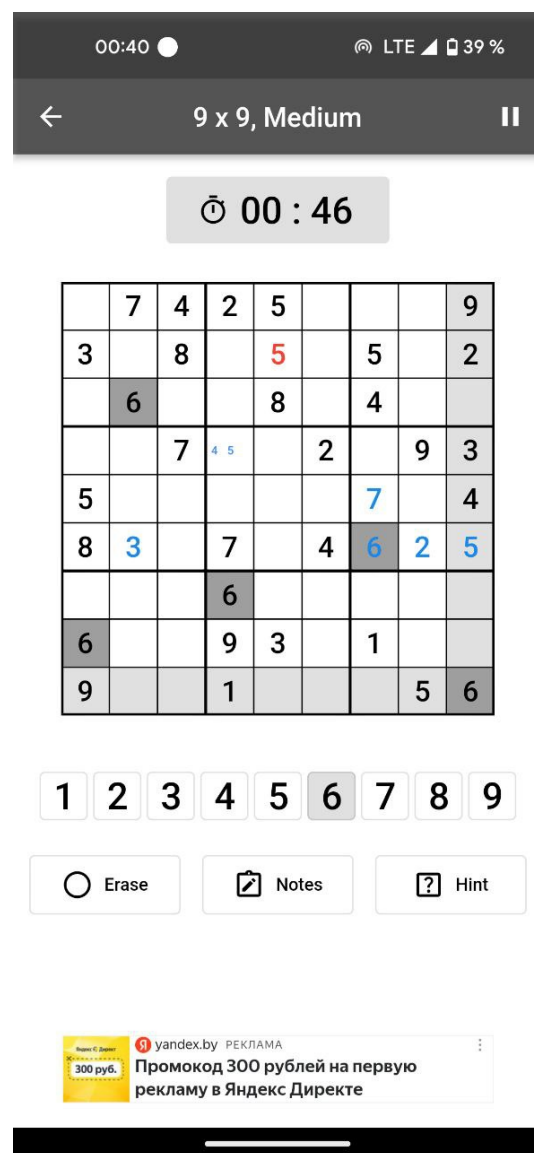
- [1] Выбор темы приложения [Электронный ресурс] / ESRI. URL: <https://doc.arcgis.com/ru/experience-builder/latest/build-apps/change-app-theme.htm> (дата обр. 12.02.2023).
- [2] Google play [Электронный ресурс] / Google Inc. URL: <https://play.google.com/store> (дата обр. 12.02.2023).
- [3] Алгоритм генерации sudoku [Электронный ресурс] / HashSpark. URL: <https://habr.com/ru/articles/192102/> (дата обр. 12.02.2023).
- [4] Hive dart plugin [Электронный ресурс] / HiveDB Dev. URL: <https://pub.dev/packages/hive> (дата обр. 11.05.2023).
- [5] Жанры мобильных игр [Электронный ресурс] / Monster-book. URL: <https://monster-book.com/blog/zhanry-mobilnyh-igr> (дата обр. 12.02.2023).
- [6] Apple Store [Электронный ресурс] / Apple Inc. URL: <https://www.apple.com/ru/app-store/> (дата обр. 12.02.2023).
- [7] Sudoku [Электронный ресурс] / Easybrain. URL: <https://sudoku.com/ru/pravila-sudoku> (дата обр. 12.02.2023).
- [8] Dart [Электронный ресурс] / Google Inc. URL: <https://dart.dev/> (дата обр. 12.02.2023).
- [9] Flutter [Электронный ресурс] / Google Inc. URL: <https://flutter.dev/> (дата обр. 12.02.2023).
- [10] RuStore [Электронный ресурс] / VK. URL: <https://www.rustore.ru/> (дата обр. 22.05.2023).
- [11] Головоломка [Электронный ресурс] / Wikipedia. URL: <https://ru.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%BE%D0%B2%D0%BE%D0%BB%D0%BE%D0%BC%D0%BA%D0%B0> (дата обр. 12.02.2023).
- [12] Развитие индустрии GameDev: анализ популярности жанров мобильных игр и тренды 2021 [Электронный ресурс] / 2023 VOKI GAMES. URL: <https://vokigames.com/razvitie-industrii-gamedev-analiz-populyarnosti-zhanrov-mobilnyh-igr-i-trendy-2021> (дата обр. 12.02.2023).
- [13] В. В. Евстратов. “Рекурсивный алгоритм решения sudoku с проверкой найденного решения на единственность”. В: *Молодой учёный* 51.341 (2020), с. 8—10.



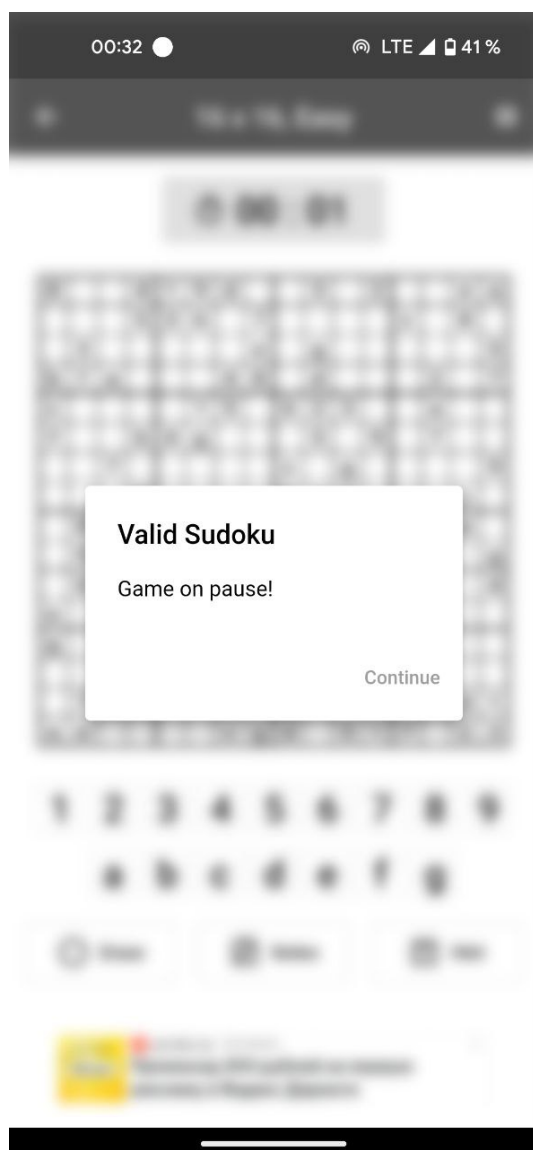
# Приложения



Прил. 7.1: Главное меню



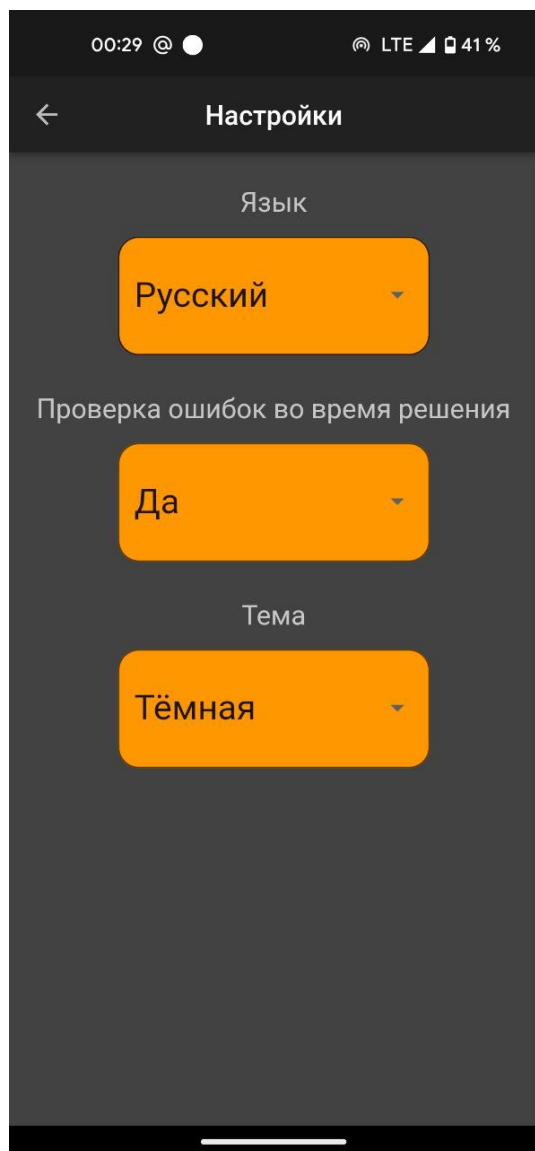
Прил. 7.2: Процесс игры



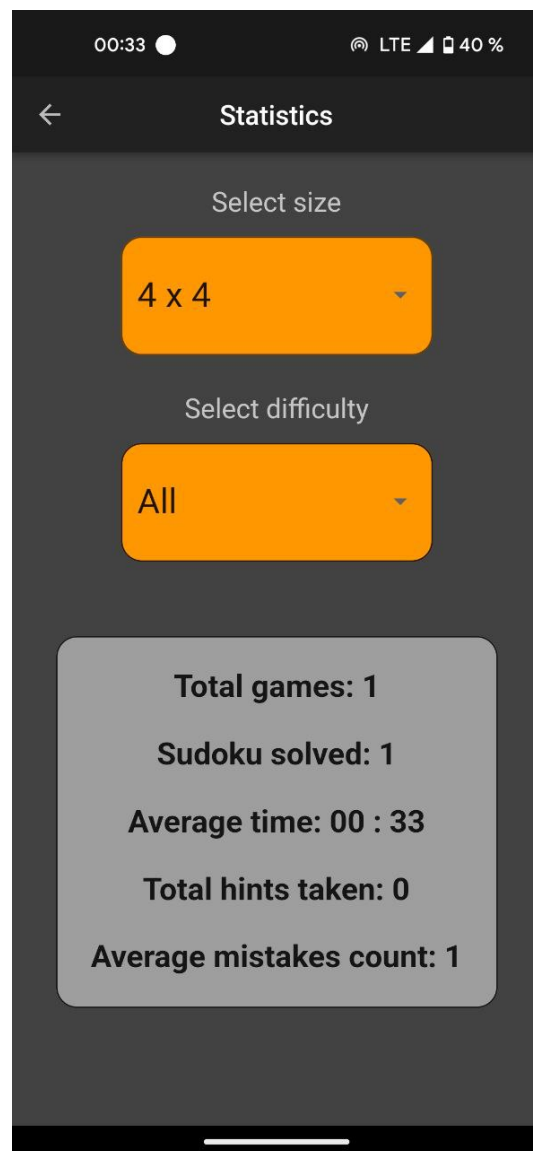
Прил. 7.3: Игра на паузе



Прил. 7.4: Игра окончена



Прил. 7.5: Настройки



Прил. 7.6: Статистика