

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

Отчёт по:

ОТЧЕТ ПО ПРОИЗВОДСТВЕННО-ЭКСПЛУАТАЦИОННОЙ ПРАКТИКЕ

Выполнил студент 595 группы:

_____ Д. В. Осипенко

«___» _____ 2022 г.

Проверил: ст. преп. каф. ВТиЭ.

_____ В. В. Белозерских

«___» _____ 2022 г.

Барнаул 2022 г.

СОДЕРЖАНИЕ

Введение	3
1 Глава. Обязанности и деятельность производимые на предприятии .	5
1.1 Диагностика компьютеров. Устранение найденных проблем. . .	5
1.2 Установка Linux Ubuntu. Установка необходимого и удаление предустановленного/ненужного программного обеспечения . . .	6
1.3 Руководство приставленными студентами и первокурсниками . .	6
2 Глава. Индивидуальное задание. Приложение для тренировки уст- ных вычислений	7
2.1 Проектирование	8
2.2 Реализация	10
Заключение	13
Список использованной литературы	14
Приложение А	14
Приложение В	20

ВВЕДЕНИЕ

Осуществлено прохождение производственно-эксплуатационной практике (далее - практика) направления подготовки «Информатика и вычислительная техника» на базе кафедры Вычислительной техники и электроники (ВТиЭ) Алтайского государственного университета (АлтГУ) [1]. Период прохождения практики 4 недели: с 16 мая, по 11 июня 2022 года. За это время мной произведено:

1. Ознакомление с нормативно-правовой базой, регламентирующей деятельность программиста на месте практики:
 - 1.1. Инструкция по охране труда для программиста [2]
 - 1.2. Вводный инструктаж
 - 1.3. Инструктаж по технике безопасности
2. Диагностика компьютеров на работоспособность, выявление технических проблем.
3. Установка и настройка программного обеспечения для операционных систем Windows XP, 10 [3] и Linux Ubuntu [4].
4. Разработка скриптов для powershell [5]
5. Разработка мобильного приложения на тему «Устные вычисления» [6]
6. Руководство группой людей

Цель практики

1. Получить знания и навыки их практического применения в сфере системного администрирования и программирования
2. Ознакомится со спецификой деятельности системного администратора
3. Развитие лидерских качеств

Задачи практики

1. Поиск и изучение руководств по установке, настройке, наладке, использованию программно-аппаратного обеспечения вычислительной техники, информационных и автоматизированных систем;
2. Освоение методик использования необходимого программного обеспечения;

3. Проверка работоспособности типовых узлов и устройств;
4. Использование программного обеспечения для решения практических задач, составление схем приема-передачи данных.

1. ГЛАВА. ОБЯЗОНОСТИ И ДЕЯТЕЛЬНОСТЬ ПРОИЗВОДИМЫЕ НА ПРЕДПРИЯТИИ

1.1. Диагностика компьютеров. Устранение найденных проблем.

Диагностика компьютера направлена на выявление проблем в его работе. В нашем случае план тестирования следующий:

1. Попытка запуска
2. Проверка подключенных устройств в BIOS
3. Проверка настроек BIOS
4. Проверка работоспособности RAM с помощью программы Memory Test 86 (memtest86) [7] с загрузочной флешки
5. Проверка HDD на наличие битых секторов с помощью mhdh [8] с загрузочной флешки

В общем случае работа была проведена с 10тью компьютерами, которые в последующем были установлены в 206 аудиторию:

- Для двух ПК была произведена замена/установка блока питания
- У 4 ПК была заменена батарейка BIOS
- Для 5 ПК устранена проблема с ошибкой объема памяти для дискетоприемника путем изменения значения в BIOS
- У 1 ПК устранена проблема с S.M.A.R.T. (self-monitoring, analysis and reporting technology) [9] при загрузке компьютера путем отключения параметра в BIOS
- У 5 ПК установлено верное значение даты и времени
- Для каждого компьютера установлены RAM примерно на 1гб (в сумме)
- Успешно протестировано 9 ПК на наличие ошибок RAM, 10 на проблем с HDD
- У 1 ПК выявлены проблемы с разъемом оперативной памяти материнской платы
- Обнаружены две неисправные плашки оперативной памяти

1.2. Установка Linux Ubuntu. Установка необходимого и удаление предустановленного/ненужного программного обеспечения

Необходимо установить Linux на один диск с Windows для запуска с помощью Dual Boot. Логично предположить, что производить установку для каждого ПК по отдельности долго и утомительно, поэтому был выбран один ПК, на нем установлена Linux, произведена настройка, установка всего необходимого программного обеспечения и после этого дополнительно подключается hdd от другого ПК и производится клонирование данных с помощью программы CloneZilla [10].

Установка Ubuntu linux производилась с помощью стандартного GUI установщика, добавлены дистрибутивы АлтГУ, осуществлено разбиение свободной части диска на три раздела: загрузка (boot), виртуальная память (swap), домашний раздел (root/home). После завершения установки был отредактирован и запущен bash скрипт для скачивания необходимых программных пакетов.

Для Windows XP была произведена активация с помощью ключа. Установлены недостающие драйвера и необходимы программы в 202, 206, 208, 210 аудиториях.

Для Windows 10 был написан скрипт на языке оболочки PowerShell для отключения некоторых служб, удаление предустановленных программ (Xbox, Cortana, ...), увеличения виртуальной памяти. Скрипт опробован и использован в 208 и 210 аудиториях.

1.3. Руководство приставленными студентами и первокурсниками

Со второй недели прохождения практики ко мне были приставлены студенты Колледжа АлтГУ, проходящие практику на базе кафедры ВТиЭ. Мной была осуществлена помощь и обучение в проделывание перечисленных выше действиях, распределение обязанностей и раздача указаний при выполнении некоторых действий, например: установка компьютеров на рабочие места; контроль над первокурсниками.

С четвертой недели к подопечным присоединились первокурсники, проходящие ознакомительную практику. В отношении их выполнялись надзор над выполнением поставленных задач руководством, помощь в разрешении трудностей, контроль посещаемости.

2. ГЛАВА. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ. ПРИЛОЖЕНИЕ ДЛЯ ТРЕНИРОВКИ УСТНЫХ ВЫЧИСЛЕНИЙ

В качестве индивидуального задания было необходимо написать мобильное приложение для тренировки устных вычислений. Разработку мобильного приложения можно произвести с помощью двух способов:

1. Нативная разработка [11]
2. Кроссплатформенная разработка [12]

Нативная разработка

Нативная разработка делится на основе двух платформ: Android с языком Kotlin/Java и IOS на Swift/Object-c.

К достоинствам данного подхода можно отнести возможность написания нативного кода для каждой операционной системы, что позволяет иметь значительное преимущество в производительности и потенциальной оптимизированности кода. Так же к преимуществам можно отнести добротную документацию и поддержку со стороны вендеров: Google и Apple.

К недостаткам данного способа относятся скорость разработки и невозможность использовать одну кодовую базу для двух видов ОС, что при желании выйти на два рынка может возникнуть необходимость писать одно и то же приложения для каждой платформы, что выливается в дополнительные затраты на разработку, тестирование.

Данный метод наиболее подходит для приложений, требующих производить множество вычислений и возможному прямому доступу к возможностям платформы. Примеры таких приложений: Игры, нативные приложения (основанные на функционале платформы).

Кроссплатформенная разработка

Лидерами кроссплатформенного рынка являются React Nativ [13] - расширение популярного web-based framework, разработанного компанией Facebook (запрещенной на территории Российской Федерации), призванного использовать написанный ранее код для web-приложения на мобилке. Главным преимуществом является я.п. JavaScript и ОГРОМНОЕ количество frontend-разработчиков, разного уровня, освоивших данный framework. К

недостаткам относится малая производительность скомпилированного приложения.

Другим вариантом, стремительно набирающем популярность является Flutter Framework [14], разработанный компанией Google, направленный для создание приложений на мобильных устройствах (iOS, Android), ПК (Linux, Windows) и веб страниц с использованием одной базы кода. Отличается относительно высокой производительностью, схожим способом построения UI с React, быстрорастущим количеством сторонних библиотек и сообщества. Недостатками можно назвать саму компанию Google, которая своеобразно относится к своим проектам (может спокойно тянуть неудачные и не востребованные проекты, забить на потенциально интересные и успешные идеи и пустить все на "свалку проектов гугл") и сложность в отделении бизнес логики от кода ui.

В итоге мной выбор пал в сторону Flutter из-за высокой производительности (по сравнению с React native) и желанием в будущем выпустить приложение на Android и IOS.

2.1. Проектирование

Процесс разработки я начал с проектирования приложения. Первым делом я сделал наброски приложения, чтобы определиться с функционалом и внешним видом

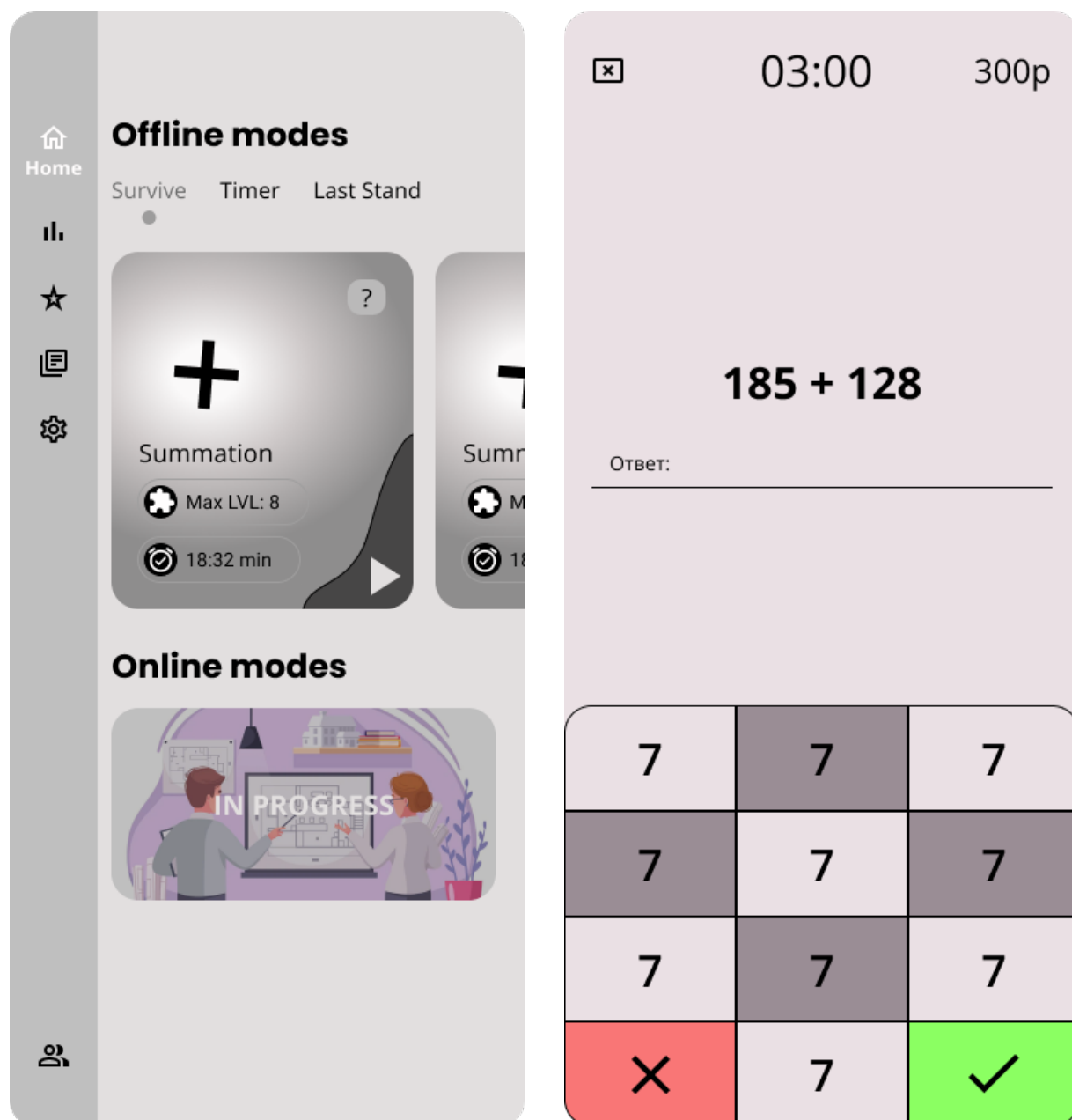


Рис 1. Экраны домашней страницы и процесса игры

Далее было решено использовать библиотеку контроля состояний Riverpod [15], для последующий реализации архитектуры MVC(P) (model view controller (provider)) [16] для отделения бизнес логики от графического интерфейса, но из-за возникших трудностей с dependency injection (передача состояния и значений через дочерние (вложенные) элементы(виджеты)) и трудности создание новых объектов provider при переходе к игровму экрану, основанные на глобальном состоянии providers, окончательный выбор был сделан на библиотеку BLoC [17], использующие принципы bloc, cubit, state(состояние), event(событие)

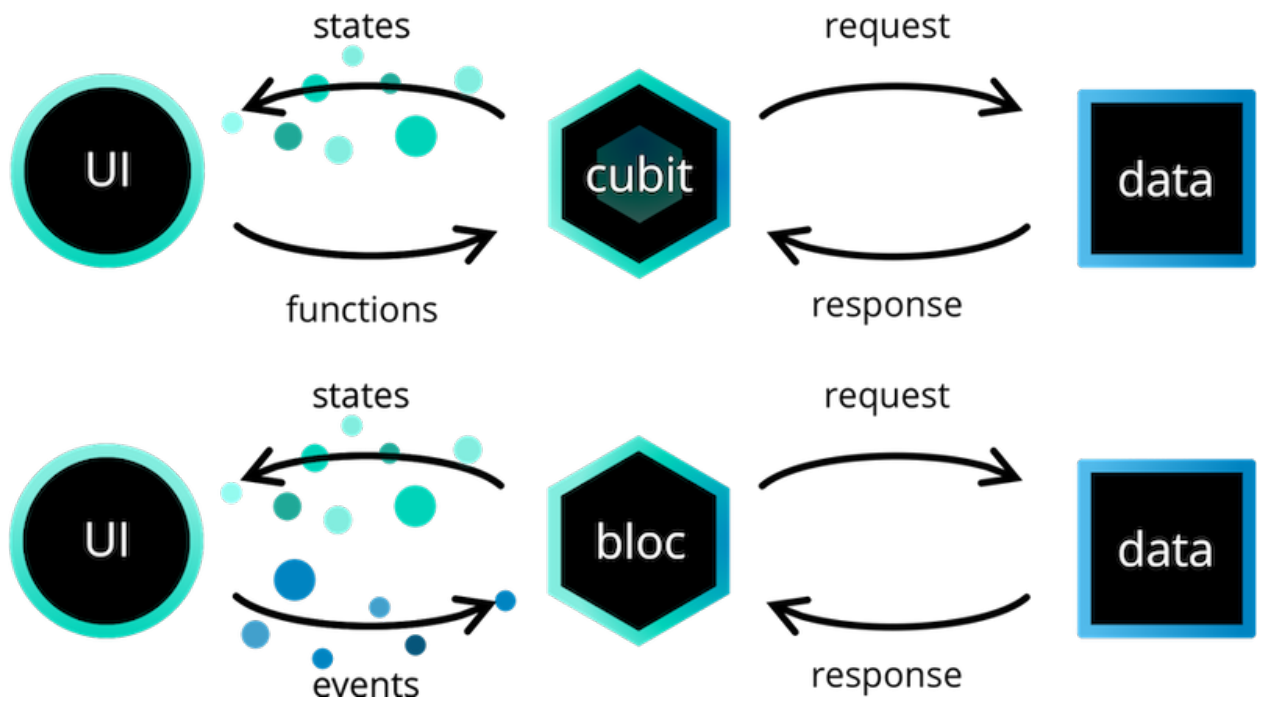


Рис. 2 Пример BLoC архитектуры с использованием bloc и cubit

2.2. Реализация

Проект представляет из себя 2 экрана: домашний и игровой. Для каждого экрана осуществлено извлечение бизнес логики путем использования Bloc элементов, состояний и событий.

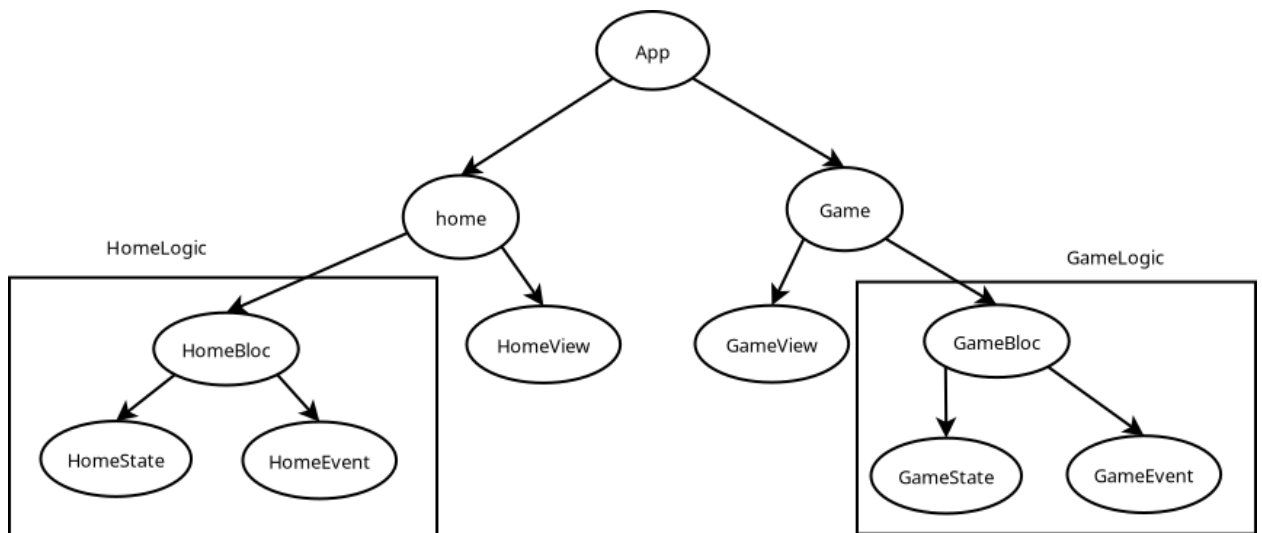


Рис. 3 Структура проекта.

Home

Состояния Homestate:

- HomeInit - начальное состояние, определяющее выбранным первый (0) экран

- HomeCurrentView - состояние, отображающее выбранный экран на данный момент

События HomeEvent:

- HomeChangeView - информирует HomeBloc, что необходимо изменить выбранный экран

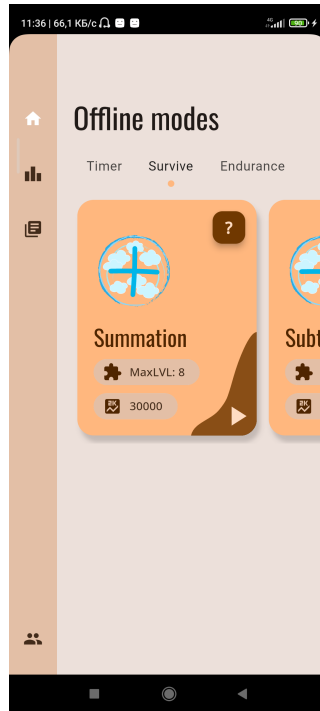


Рис. 4 HomeInit состояние

Game

Состояния Game:

- GameInit - начальное состояние, определяющее базовые настройки игры
- GameStateRunning - состояние, отображающее процесс игры
- GameStateFinish - состояние, отображающее окончание игры

События GameEvent:

- GameEventStart - информирует GameBloc, что необходимо начать игровой процесс
- GameEventSubmit - информирует GameBloc, что игрок отправил свой вариант ответа
- GameEventEnd - информирует GameBloc, что игра закончена
- GameEventRestart - информирует GameBloc, что пользователь желает перезапустить игру

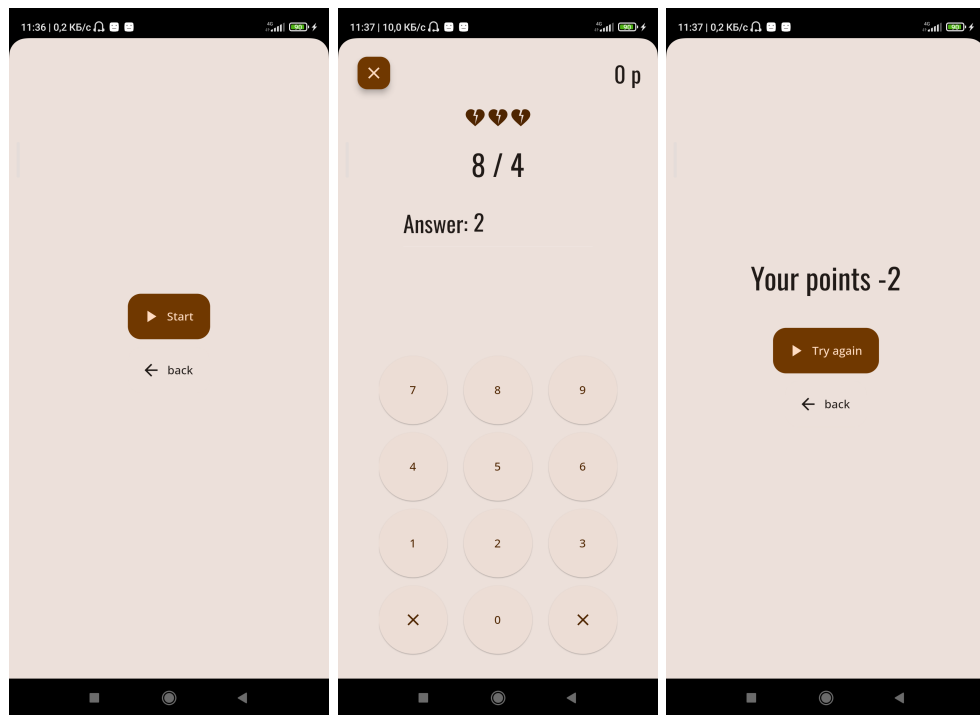


Рис. 5 GameInit, GameStateRunning, GameStateFinish состояния

ЗАКЛЮЧЕНИЕ

В ходе данной практики мной были изучены основы работы системного администратора, получены навыки диагностики и ремонта компьютеров, администрирования операционных систем Linux и Windows.

Был приобретен опыт работы и руководства на группой людей, произведено развитие лидерских качеств.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. [Электронный ресурс] АлтГУ - Алтайский Государственный Университет. — URL: <https://www.asu.ru> (дата обр. 09.06.2022).
2. [Электронный ресурс] АлтГУ - Инструкция по охране труда для программиста. — URL: <https://www.asu.ru/search/documents/25538/> (дата обр. 09.06.2022).
3. [Электронный ресурс] Microsoft Windows - Wikipedia. — URL: https://en.wikipedia.org/wiki/Microsoft_Windows (дата обр. 09.06.2022).
4. [Электронный ресурс] Enterprise Open Source and Linux | Ubuntu. — URL: <https://ubuntu.com/> (дата обр. 09.06.2022).
5. [Электронный ресурс] PowerShell Documentation - PowerShell | Microsoft Docs. — URL: <https://docs.microsoft.com/en-us/powershell/> (дата обр. 09.06.2022).
6. *A. Benjamin M. S.* Secrets of Mental Math: The Mathemagician's Guide to Lightning Calculation and Amazing Math Tricks. — Crown, 0206. — ISBN 9780307338402. — URL: https://www.amazon.com/Secrets-Mental-Math-Mathemagicians-Calculation/dp/0307338401/ref=tmm_pap_swatch_0?_encoding=UTF8&qid=&sr=.
7. [Электронный ресурс] MemTest86 - Official Site of the x86 Memory Testing Tool. — URL: <https://www.memtest86.com/> (дата обр. 09.06.2022).
8. [Электронный ресурс] MHDD. — URL: <http://hddguru.com/software/2005.10.02-MHDD/> (дата обр. 09.06.2022).
9. [Электронный ресурс] S.M.A.R.T. - Wikipedia. — URL: <https://en.wikipedia.org/wiki/S.M.A.R.T.> (дата обр. 09.06.2022).
10. [Электронный ресурс] Clonezilla - About. — URL: <https://clonezilla.org> (дата обр. 09.06.2022).
11. [Электронный ресурс] Native Mobile App Development - What is it? Pros and Cons. — URL: <https://mdevelopers.com/blog/what-is-a-native-mobile-app-development-> (дата обр. 09.06.2022).

12. [Электронный ресурс] Flutter vs React Native 2022: A Detailed Comparison - Hackr.io. — URL: <https://hackr.io/blog/react-native-vs-flutter> (дата обр. 09.06.2022).
13. [Электронный ресурс] React Native · Learn once, write anywhere. — URL: <https://reactnative.dev> (дата обр. 09.06.2022).
14. [Электронный ресурс] Flutter documentation | Flutter. — URL: <https://docs.flutter.dev/> (дата обр. 09.06.2022).
15. [Электронный ресурс] Riverpod. — URL: <https://riverpod.dev/> (дата обр. 09.06.2022).
16. [Электронный ресурс] Overview of ASP.NET Core MVC. — URL: https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?WT.mc_id=dotnet-35129-website&view=aspnetcore-6.0 (дата обр. 09.06.2022).
17. [Электронный ресурс] Bloc State Management Library. — URL: <https://bloclibrary.dev/> (дата обр. 09.06.2022).

ПРИЛОЖЕНИЕ А

Листинг 2.1 Код powershell скрипта

```

1 Write-Host "Change Virtual Memory"
2 $physicalmem=get-wmiobject Win32_ComputerSystem | %
   → {$_TotalPhysicalMemory}
3 $physicalmem1=[math]::Round($physicalmem / 1048576)
4 $computersys = Get-WmiObject Win32_ComputerSystem
   → -EnableAllPrivileges;
5 $computersys AutomaticManagedPagefile = $False;
6 $computersys.Put();
7 $pagefile = Get-WmiObject -Query "Select * From
   → Win32_PageFileSetting Where Name like
   → '%pagefile.sys'";
8 $newphysicalmem = $physicalmem1*2
9 $pagefile.InitialSize = $newphysicalmem;
10 $pagefile.MaximumSize = $newphysicalmem;
11 $newpagefile=$pagefile.Put();
12
13 Write-Host "'$physicalmem1' old, '$newphysicalmem'
   → new"
14
15 # Set-ExecutionPolicy RemoteSigned
16 Write-Host "Disable Hyper-V Service"
17 # Интерфейс гостевой службы Hyper-V
18 Set-Service vmicguestinterface -StartupType Disabled
19 # Служба пульса (Hyper-V)
20 Set-Service vmicheartbeat -StartupType Disabled
21 # Служба обмена данными (Hyper-V)
22 Set-Service vmickvpxexchange -StartupType Disabled
23 # Служба виртуализации удаленных рабо...
24 Set-Service vmicrdv -StartupType Disabled
25 # Служба завершения работы в качестве...
```



```

26 Set-Service vmicshutdown -StartupType Disabled
27 # Служба синхронизации времени Hyper-V
28 Set-Service vmictimesync -StartupType Disabled
29 # Служба Hyper-V PowerShell Direct
30 Set-Service vmicvmsession -StartupType Disabled
31 # Служба запросов на теневое копирова...
32 Set-Service vmicvss -StartupType Disabled
33
34 Write-Host "Disable Bluetooth Service"
35 Set-Service bthserv -StartupType Disabled
36 Set-Service BluetoothUserService -StartupType Disable
37
38
39 Write-Host "Disable Live Tiles"
40 function Pin-Remove-App {      param(
41     [string]$apptilename
42 )
43     try{
44         ((New-Object -Com
            ↳ Shell.Application).Namespace('shell:::{
            ↳ 0245-4df3-b780-
            ↳ 3893943456e1}').Items() |
            ↳ ?{$_ .Name -eq
            ↳ $apptilename}).Verbs() |
            ↳ ?{$_ .Name.replace('&','') -match
            ↳ 'Von "Start" lösen|Unpin from
            ↳ Start'} | %{$_.DoIt()}
45         Write-Host "App '$apptilename'
            ↳ unpinned from Start"
            ↳ -ForegroundColor Green
46     } catch{
47         #Write-Error "Error Pinning/Unpinning App!
            ↳ (App-Name correct)"

```

```

48         Write-Host "App '$apptilename' not
           ↳ found" -ForegroundColor Red
49     }
50 }
51
52 #----- List of apps -----
53 $tiles = "Mail" ,
54     "Microsoft Store" ,
55     "Calendar" ,
56     "Microsoft Edge" ,
57     "Photos" ,
58     "Cortana" ,
59     "Weather" ,
60     "Phone Companion" ,
61     "Twitter" ,
62     "Skype Video" ,
63     "Skype" ,
64     "Candy Crush Soda Saga" ,
65     "Xbox" ,
66     "Groove music" ,
67     "Wallet" ,
68     "Get Office" ,
69     "Onenote" ,
70     "News" ,
71     "Camera" ,
72     "3D Viewer" ,
73     "Alarms & Clock" ,
74     "Calendar" ,
75     "Candy Crush Saga" ,
76     "Candy Crush Friends Saga" ,
77     "Feedback Hub" ,
78     "Get Help" ,
79     "Microsoft Solitaire Collection" ,
80     "Minecraft for Windows 10" ,

```

```

81  "Mixed Reality Portal" ,
82  "Money" ,
83  "Movies & TV" ,
84  "Netflix" ,
85  "People" ,
86  "Scan" ,
87  "Snip & Sketch" ,
88  "Sports" ,
89  "Sticky Notes" ,
90  "Tips" ,
91  "Voice Recorder" ,
92  "Xbox Game Bar" ,
93  "Your Phone"
94  #----- End of list of apps -----
95
96  foreach($tile in $tiles) {
97      Pin-Remove-App $tile
98  }
99
100 Write-Host "Remove apps"
101
102 $apps = "3dbuilder",
103         "windowsalarms",
104         "windowscalculator",
105         "windowscommunicaitonsapps",
106         "windowscamera",
107         "officehub",
108         "skypeapp",
109         "getstarted",
110         "zunemisic",
111         "windowsmaps",
112         "solitairecollection",
113         "bingfinance",
114         "zunevideo",

```

```

115     "bingnews",
116     "onenote",
117     "people",
118     "windowsphone",
119     "photos",
120     "windowsstore",
121     "bingsports",
122     "soundrecorder",
123     "bingweather",
124     "xboxapp",
125     "549981c3f5f10",
126     "XboxGamingOverlay"
127
128 foreach($app in $apps) {
129     try{
130         Get-AppxPackage -AllUsers
131         ↪ -PackageTypeFilter Bundle -name
132         ↪ *$app* | Remove-AppxPackage
133         Write-Host "App '$app' removed"
134         ↪ -ForegroundColor Green
135     } catch{
136         Write-Host "App '$app' not found or
137         ↪ cant remove" -ForegroundColor Red
138     }
139 }
140
141 $confirmation=Read-host "Do you want to Restart the PC
142 ↪ (Y/N) "
143 if($confirmation -eq 'N'){
144     Write-host "The system restart is skipped by
145     ↪ the user" -BackgroundColor DarkRed
146 }else{
147     Write-Host "The system will be restarting
148     ↪ ...." -BackgroundColor DarkGreen

```

```
142         Restart-Computer -Force
143     }
```

ПРИЛОЖЕНИЕ В

Листинг 2.2 Код приложения

```

1  import 'package:flutter/material.dart';
2  import 'package:flutter_bloc/flutter_bloc.dart';
3  import 'package:google_fonts/google_fonts.dart';
4  import 'package:smct/gameselect/gameselect.dart';
5  import 'package:smct/home/home.dart';
6
7  void main() {
8    runApp(MultiBlocProvider(providers: [
9      BlocProvider<HomeBloc>(create: (_) => HomeBloc()),
10     BlocProvider<GameSelectBloc>(create: (_) =>
11     ↪ GameSelectBloc()),
12   ], child: const MentalCalc()));
13
14  class MentalCalc extends StatelessWidget {
15     const MentalCalc({Key? key}) : super(key: key);
16
17     @override
18     Widget build(BuildContext context) {
19       return MaterialApp(
20         debugShowCheckedModeBanner: false,
21         home: HomeView(),
22         theme: ThemeData.from(
23           colorScheme: const ColorScheme(
24             brightness: Brightness.dark,
25             // Primary
26             primary: Color(0xffffFB77E),
27             onPrimary: Color(0xff4F2500),
28             primaryContainer: Color(0xff703800),
29             onPrimaryContainer: Color(0xffffFDCC2),

```

```

30      // Secondary
31      secondary: Color(0xffE3BFA6),
32      onSecondary: Color(0xff422B1A),
33      secondaryContainer: Color(0xff5A412F),
34      onSecondaryContainer: Color(0xffFFDCC2),
35      // Tertiary
36      tertiary: Color(0xffC6CA95),
37      onTertiary: Color(0xff2F330B),
38      tertiaryContainer: Color(0xff464A21),
39      onTertiaryContainer: Color(0xffE3E7AF),
40      // Error
41      error: Color(0xffFFB4A9),
42      onError: Color(0xff680003),
43      errorContainer: Color(0xff930006),
44      onErrorContainer: Color(0xffFFDAD4),
45      // Background
46      background: Color(0xffECE0DA),
47      onBackground: Color(0xff201A17),
48      surface: Color(0xffECE0DA),
49      onSurface: Color(0xff201A17),
50    ),
51    useMaterial3: true,
52    textTheme: TextTheme(
53      displaySmall: GoogleFonts.oswald(
54        fontSize: 36,
55        fontWeight: FontWeight.w400,
56      ),
57      headlineMedium: GoogleFonts.oswald(
58        fontSize: 28,
59        fontWeight: FontWeight.w400,
60      ),
61      labelLarge: GoogleFonts.openSans(
62        fontSize: 14,
63        fontWeight: FontWeight.w500,

```

```

64        )),
65     ),
66 );
67 }
68 }

```

```

1  import 'package:equatable/equatable.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_bloc/flutter_bloc.dart';
4  import 'package:smct/gameselect/gameselect.dart';
5  import 'package:smct/widgets/widgets.dart';
6
7  part 'homestate.dart';
8  part 'homeevent.dart';
9  part 'homebloc.dart';
10 part 'homeview.dart';

```

```

1  part of 'home.dart';
2
3  class HomeView extends StatelessWidget {
4    HomeView({Key? key}) : super(key: key);
5    final hub_pages = [
6      GameSelectView(),
7      Center(
8        child: Text('leaderboard'),
9      ),
10     Center(
11       child: Text('tutorials'),
12     ),
13     Center(
14       child: Text('Profile'),
15     ),
16   ];

```



```

17
18  @override
19  Widget build(BuildContext context) {
20    return Scaffold(
21      backgroundColor:
↪ Theme.of(context).colorScheme.background,
22      body: SafeArea(
23        child: Container(
24          height: MediaQuery.of(context).size.height,
25          width: MediaQuery.of(context).size.width,
26          child: Row(children: [
27            SideBar(),
28            SizedBox(
29              width: MediaQuery.of(context).size.width
↪      * .85,
30              height:
↪      MediaQuery.of(context).size.height,
31              child:
32                BlocBuilder<HomeBloc,
↪      HomeState>(builder: (context, state) {
33                  return
↪      this.hub_pages[state.current_view];
34                })),
35            ),
36          ]),
37        ),
38      ),
39    );
40  }
41 }

```

```

1  part of 'home.dart';
2

```

```

3  class HomeBloc extends Bloc<HomeEvent, HomeState> {

```

```

4   HomeBloc() : super(const HomeInit()) {
5       on<HomeChangeView>(_changeView);
6   }
7
8   void _changeView(HomeChangeView event,
↪   Emitter<HomeState> emit) {
9       final view_id = event.view_id;
10      if (view_id >= 0) {
11          emit(HomeCurrentView(view_id));
12      }
13  }
14 }

```

```

1  part of 'home.dart';
2
3  abstract class HomeState extends Equatable {
4      final int current_view;
5
6      const HomeState(this.current_view);
7
8      @override
9      List<Object> get props => [this.current_view];
10 }
11
12 class HomeInit extends HomeState {
13     const HomeInit() : super(0);
14 }
15
16 class HomeCurrentView extends HomeState {
17     const HomeCurrentView(int view_id) : super(view_id);
18 }

```

```

1  part of 'home.dart';
2
3  abstract class HomeEvent extends Equatable {
4      const HomeEvent();
5
6      List<Object> get props => [];
7  }
8
9  class HomeChangeView extends HomeEvent {
10     final int view_id;
11
12     const HomeChangeView({required this.view_id});
13 }

```

```

1  import 'package:equatable/equatable.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_bloc/flutter_bloc.dart';
4  import 'package:smct/game/game.dart';
5  import 'package:smct/widgets/widgets.dart';
6
7  part 'gameselectview.dart';
8  part 'gameselectstate.dart';
9  part 'gameselectbloc.dart';
10
11 enum OfflineType { Timer, Survive, Endurance }

```

```

1  part of 'gameselect.dart';
2
3  class GameSelectBloc extends Cubit<GameSelectState> {
4      GameSelectBloc() : super(GameSelectInit()) {}
5
6      void changeType(int index) {
7          var type = OfflineType.values[index];

```

```

8      if (type.name != state.type.name) {
9          emit(GameSelectCurrent(type));
10     }
11 }
12 }

```

```

1 part of 'gameselect.dart';
2
3 abstract class GameSelectState extends Equatable {
4     final OfflineType type;
5
6     GameSelectState(this.type);
7
8     @override
9     List<Object> get props => [type];
10 }
11
12 class GameSelectInit extends GameSelectState {
13     GameSelectInit() : super(OfflineType.Timer);
14 }
15
16 class GameSelectCurrent extends GameSelectState {
17     GameSelectCurrent(OfflineType type) : super(type);
18 }

```

```

1 part of 'gameselect.dart';
2
3 class GameSelectView extends StatefulWidget {
4     const GameSelectView({Key? key}) : super(key: key);
5
6     @override
7     State<StatefulWidget> createState() =>
8     ↪ _GameSelectViewState();

```

```

8   }
9
10  class _GameSelectViewState extends
    ↪ State<GameSelectView>
11      with SingleTickerProviderStateMixin {
12      late TabController _tab_controller;
13      final animations_list = [
14          "assets/anim/plus.json",
15          "assets/anim/minus.json",
16          "assets/anim/plus.json",
17          "assets/anim/div.json",
18      ];
19      @override
20      void initState() {
21          super.initState();
22          this._tab_controller = TabController(length: 3,
    ↪ vsync: this);
23      }
24
25      @override
26      Widget build(BuildContext context) {
27          return Scaffold(
28              body: Container(
29                  width: MediaQuery.of(context).size.width,
30                  height: MediaQuery.of(context).size.height,
31                  padding: EdgeInsets.only(top: 80, left: 20),
32                  child: Column(
33                      crossAxisAlignment: CrossAxisAlignment.start,
34                      children: [
35                          Text(
36                              "Offline modes",
37                              style:
    ↪ Theme.of(context).textTheme.displaySmall!.copyWith(

```

```

38         color:
↪ Theme.of(context).colorScheme.onBackground,
39     ),
40 ),
41 Container(
42     height: 80,
43     child: Align(
44         alignment: Alignment.centerLeft,
45         child: TabBar(
46             isScrollable: true,
47             controller: this._tab_controller
48                 ..addListener(() {
49                 context
50                     .read<GameSelectBloc>()
51
↪ .changeType(this._tab_controller.index);
52             })),
53             indicator: DotIndicator(
54                 color:
↪ Theme.of(context).colorScheme.primary,
55                 distanceFromCenter: 20,
56                 radius: 4,
57             ),
58             tabs: [
59                 Tab(
60                     text: "Timer",
61                 ),
62                 Tab(
63                     text: "Survive",
64                 ),
65                 Tab(
66                     text: "Endurance",
67                 )
68             ],

```

```

69         ),
70     ),
71 ),
72 Container(
73     width: MediaQuery.of(context).size.width
↪     - 20,
74     height: 300,
75     child: ListView.builder(
76         scrollDirection: Axis.horizontal,
77         itemBuilder: (context, index) {
78             return ModeCard(
79                 mode: Mode.values[index],
80                 mode_anim:
↪     this.animations_list[index],
81                 lvl: 8,
82                 points: 30000,
83             );
84         },
85         itemCount: Mode.values.length,
86     ),
87 )
88 ],
89 ),
90 ),
91 );
92 }
93 }

```

```

1  import 'dart:async';
2  import 'dart:math';
3
4  import 'package:equatable/equatable.dart';
5  import 'package:flutter/material.dart';
6  import 'package:flutter_bloc/flutter_bloc.dart';

```

```

7 import 'package:smct/gameselect/gameselect.dart';
8
9 part 'gamemodel.dart';
10 part 'gameview.dart';
11 part 'gamestate.dart';
12 part 'gameevent.dart';
13 part 'gamebloc.dart';
14 part 'gamepage.dart';
15
16 enum Mode {
17     Summation,
18     Subtraction,
19     Multiplication,
20     Division,
21 }

```

```

1 part of 'game.dart';
2
3 class GameView extends StatelessWidget {
4     GameView({Key? key}) : super(key: key);
5     final TextEditingController txt_edit_controller =
6         TextEditingController(text: '');
7
8     final num_pad = ['7', '8', '9', '4', '5', '6', '1',
9         ↪ '2', '3'];
10     @override
11     Widget build(BuildContext context) {
12         var state = context.select((GameBloc bloc) =>
13         ↪ bloc.state);
14         var game = context.select((GameBloc bloc) =>
15         ↪ bloc.state.game);
16         var points = game.points;
17         var left_value = game.left_value;
18         var right_value = game.right_value;

```



```

16     var sign;
17     switch (game.mode) {
18         case Mode.Summation:
19             sign = '+';
20             break;
21         case Mode.Subtraction:
22             sign = '-';
23             break;
24         case Mode.Multiplication:
25             sign = '*';
26             break;
27         case Mode.Division:
28             sign = '/';
29             break;
30     }
31     var buttons_text_style = Theme.of(context)
32         .textTheme
33         .labelLarge!
34         .copyWith(color:
↪ Theme.of(context).colorScheme.onPrimary);
35     return Scaffold(
36         body: Column(
37             children: [
38                 if (state is GameStateRunning) ...[
39                     Container(
40                         width: MediaQuery.of(context).size.width,
41                         height: 80,
42                         margin: EdgeInsets.only(top: 40, left:
↪ 20, right: 20),
43                         child: Row(
44                             mainAxisAlignment:
↪ MainAxisAlignment.spaceBetween,
45                             crossAxisAlignment:
↪ CrossAxisAlignment.center,

```

```

46         children: [
47             FloatingActionButton.small(
48                 onPressed: () {
49                     Navigator.pop(context);
50                 },
51                 child: Icon(Icons.clear),
52             ),
53             Container(
54                 child: Text(
55                     "$points p",
56                     style: Theme.of(context)
57                         .textTheme
58                         .headlineMedium!
59                         .copyWith(
60                             color:
↪ Theme.of(context).colorScheme.onBackground,
61                         ),
62                 ),
63             ),
64         ],
65     ),
66     Container(
67         width: 120,
68         height: 30,
69         margin: EdgeInsets.only(bottom: 20),
70         child: Row(
71             mainAxisAlignment:
↪ MainAxisAlignment.center,
72             crossAxisAlignment:
↪ CrossAxisAlignment.center,
73             children: [
74                 for (int i = 0; i < game.health; i++)
75                     Icon(
76

```

```

77         Icons.heart_broken,
78         size: 28,
79         color:
↪ Theme.of(context).colorScheme.onPrimary,
80     )
81     ],
82     ),
83 ),
84 Text(
85     "$left_value $sign $right_value",
86     style:
↪ Theme.of(context).textTheme.displaySmall!.copyWith(
87         color:
↪ Theme.of(context).colorScheme.onBackground,
88     ),
89 ),
90 Container(
91     height: 120,
92     width: MediaQuery.of(context).size.width,
93     padding: EdgeInsets.symmetric(
94         horizontal: 80,
95     ),
96     margin: EdgeInsets.only(top: 20),
97     child: TextFormField(
98         controller: this.txt_edit_controller,
99         enabled: false,
100        style:
↪ Theme.of(context).textTheme.headlineMedium!.copyWith(
101            color:
↪ Theme.of(context).colorScheme.onBackground,
102        ),
103        decoration: InputDecoration(
104            prefixIcon: Row(
105                mainAxisAlignment: MainAxisAlignment.min,

```

```

106         mainAxisAlignment:
↪     MainAxisAlignment.center,
107         children: [
108             Text(
109                 'Answer: ',
110                 style: Theme.of(context)
111                     .textTheme
112                     .headlineMedium!
113                     .copyWith(
114                         color:
↪     Theme.of(context).colorScheme.onBackground,
115                     ),
116             ),
117         ],
118     ),
119 ),
120 ),
121 ),
122 Expanded(child: SizedBox()),
123 Container(
124     width: MediaQuery.of(context).size.width,
125     height:
↪     MediaQuery.of(context).size.height / 1.9,
126     padding: const EdgeInsets.symmetric(
127         horizontal: 50,
128     ),
129     child: GridView.count(
130         physics: NeverScrollableScrollPhysics(),
131         crossAxisCount: 3,
132         crossAxisSpacing: 20,
133         mainAxisSpacing: 10,
134         children: [
135             for (var i in this.num_pad)
136                 NumPadBtn(

```

```

137         txt_edit_controller:
→   txt_edit_controller,
138         buttons_text_style:
→   buttons_text_style,
139         text: i,
140     ),
141     ElevatedButton(
142         onPressed: () {
143             this.txt_edit_controller.clear();
144         },
145         child: Icon(
146             Icons.clear,
147             color:
→   Theme.of(context).colorScheme.onPrimary,
148         )),
149     ElevatedButton(
150         onPressed: () {
151             this.txt_edit_controller.text
→   += '0';
152         },
153         child: Text(
154             '0',
155             style: buttons_text_style,
156         )),
157     ElevatedButton(
158         onPressed: () {
159             if (this.txt_edit_controller.text
→   != '') {
160
→   context.read<GameBloc>().add(GameEventSubmit(
161
→   int.parse(this.txt_edit_controller.text)));
162             this.txt_edit_controller.clear();
163         }

```

```

164         },
165         child: Icon(
166             Icons.clear,
167             color:
→ Theme.of(context).colorScheme.onPrimary,
168         ),
169     ),
170 ],
171 ),
172 ),
173 ],
174 if (state is GameStateFinish) ...[
175     Container(
176         width: MediaQuery.of(context).size.width,
177         height:
→ MediaQuery.of(context).size.height,
178         child: Column(
179             mainAxisAlignment:
→ MainAxisAlignment.center,
180             crossAxisAlignment:
→ CrossAxisAlignment.center,
181             children: [
182                 Text(
183                     'Your points ${game.points}',
184                     style:
→ Theme.of(context).textTheme.displaySmall!.copyWith(
185                         color:
→ Theme.of(context).colorScheme.onBackground),
186                 ),
187                 SizedBox(
188                     height: 40,
189                 ),
190                 FloatingActionButton.extended(
191                     heroTag: null,

```

```

192         onPressed: () {
193
194             ↪ context.read<GameBloc>().add(GameEventStart());
195
196             },
197             elevation: 0,
198             label: Text('Try again'),
199             icon: Icon(Icons.play_arrow),
200         ),
201         SizedBox(
202             height: 10,
203         ),
204         FloatingActionButton.extended(
205             heroTag: null,
206             onPressed: () {
207                 Navigator.pop(context);
208             },
209             label: Text('back'),
210             icon: Icon(Icons.arrow_back),
211             elevation: 0,
212             backgroundColor:
213             ↪ Theme.of(context).colorScheme.background,
214             foregroundColor:
215             ↪ Theme.of(context).colorScheme.onBackground,
216         ),
217     ],
218     ),
219     ],
220     if (state is GameStateInit) ...[
221         Container(
222             width: MediaQuery.of(context).size.width,
223             height:
224             ↪ MediaQuery.of(context).size.height,
225             child: Column(

```

```

222         mainAxisAlignment:
↪     MainAxisAlignment.center,
223         crossAxisAlignment:
↪     CrossAxisAlignment.center,
224         children: [
225             FloatingActionButton.extended(
226                 heroTag: null,
227                 onPressed: () {
228
↪     context.read<GameBloc>().add(GameEventStart());
229                 },
230                 elevation: 0,
231                 label: Text('Start'),
232                 icon: Icon(Icons.play_arrow),
233             ),
234             SizedBox(
235                 height: 10,
236             ),
237             FloatingActionButton.extended(
238                 heroTag: null,
239                 onPressed: () {
240                     Navigator.pop(context);
241                 },
242                 label: Text('back'),
243                 icon: Icon(Icons.arrow_back),
244                 elevation: 0,
245                 backgroundColor:
↪     Theme.of(context).colorScheme.background,
246                 foregroundColor:
↪     Theme.of(context).colorScheme.onBackground,
247             ),
248         ],
249     ),
250 )

```



```

251         ],
252     ],
253 ),
254 );
255 }
256 }
257
258 class NumPadBtn extends StatelessWidget {
259     const NumPadBtn({
260         Key? key,
261         required this.txt_edit_controller,
262         required this.buttons_text_style,
263         required this.text,
264     }) : super(key: key);
265
266     final TextEditingController txt_edit_controller;
267     final TextStyle buttons_text_style;
268     final String text;
269
270     @override
271     Widget build(BuildContext context) {
272         return ElevatedButton(
273             onPressed: () {
274                 this.txt_edit_controller.text += text;
275             },
276             child: Text(
277                 text,
278                 style: buttons_text_style,
279             ));
280     }
281 }

```

```

1 part of 'game.dart';

```

```

2

```

```

3  class GameBloc extends Bloc<GameEvent, GameState> {
4      late final Game _game;
5      GameBloc(Game game) : super(GameStateInit(game)) {
6          this._game = game;
7          on<GameEventStart>(_onStart);
8          on<GameEventSubmit>(_onSubmit);
9      }
10
11     void _onSubmit(GameEventSubmit event,
↪     Emitter<GameState> emit) {
12         var game = state.game;
13         if (event.answer == game.result) {
14             game = game.incrementPoints();
15             if (game.points % (5 * game.lvl) == 0) {
16                 game = game.changeLvl();
17                 game = game.changeMinMax();
18             }
19         } else {
20             game = game.decrementPoints();
21             game = game.decrementHealth();
22         }
23         if (game.health == 0) {
24             emit(GameStateFinish(game));
25         } else {
26             game = game.generateValues();
27             emit(GameStateRunning(game));
28         }
29     }
30
31     void _onStart(GameEventStart event,
↪     Emitter<GameState> emit) {
32         Game game = this._game.copyWith();
33         game = game.generateValues();
34         emit(GameStateRunning(game));

```

```

35     }
36 }

```

```

1  part of 'game.dart';
2
3  abstract class GameState extends Equatable {
4      final Game game;
5
6      GameState(this.game);
7
8      @override
9      List<Object> get props => [game];
10 }
11
12 class GameStateInit extends GameState {
13     GameStateInit(Game? game) : super(game ?? Game());
14 }
15
16 class GameStateRunning extends GameState {
17     GameStateRunning(Game game) : super(game);
18 }
19
20 class GameStateFinish extends GameState {
21     GameStateFinish(Game game) : super(game);
22 }

```

```

1  part of 'game.dart';
2
3  abstract class GameEvent extends Equatable {
4      const GameEvent();
5
6      @override
7      List<Object> get props => [];

```

```
8  }
9
10 class GameEventStart extends GameEvent {}
11
12 class GameEventSubmit extends GameEvent {
13     final int answer;
14
15     GameEventSubmit(this.answer);
16 }
17
18 class GameEventEnd extends GameEvent {}
19
20 class GameEventRestart extends GameEvent {}
```
