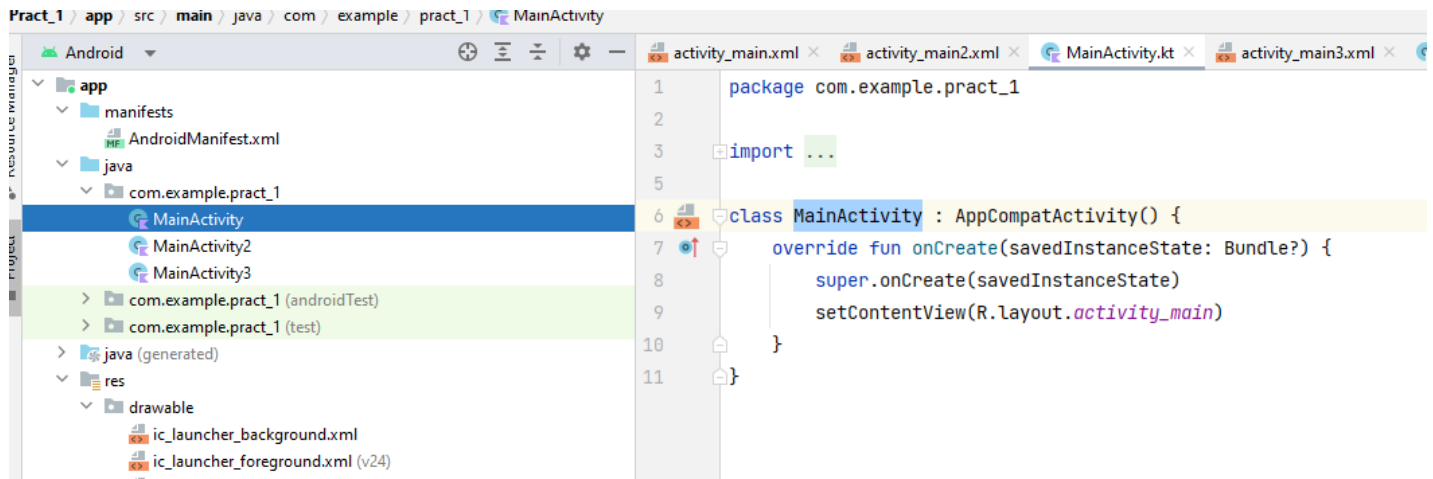


Практическая работа № 2 “Верстка экрана входа. Настройка смены экранов”

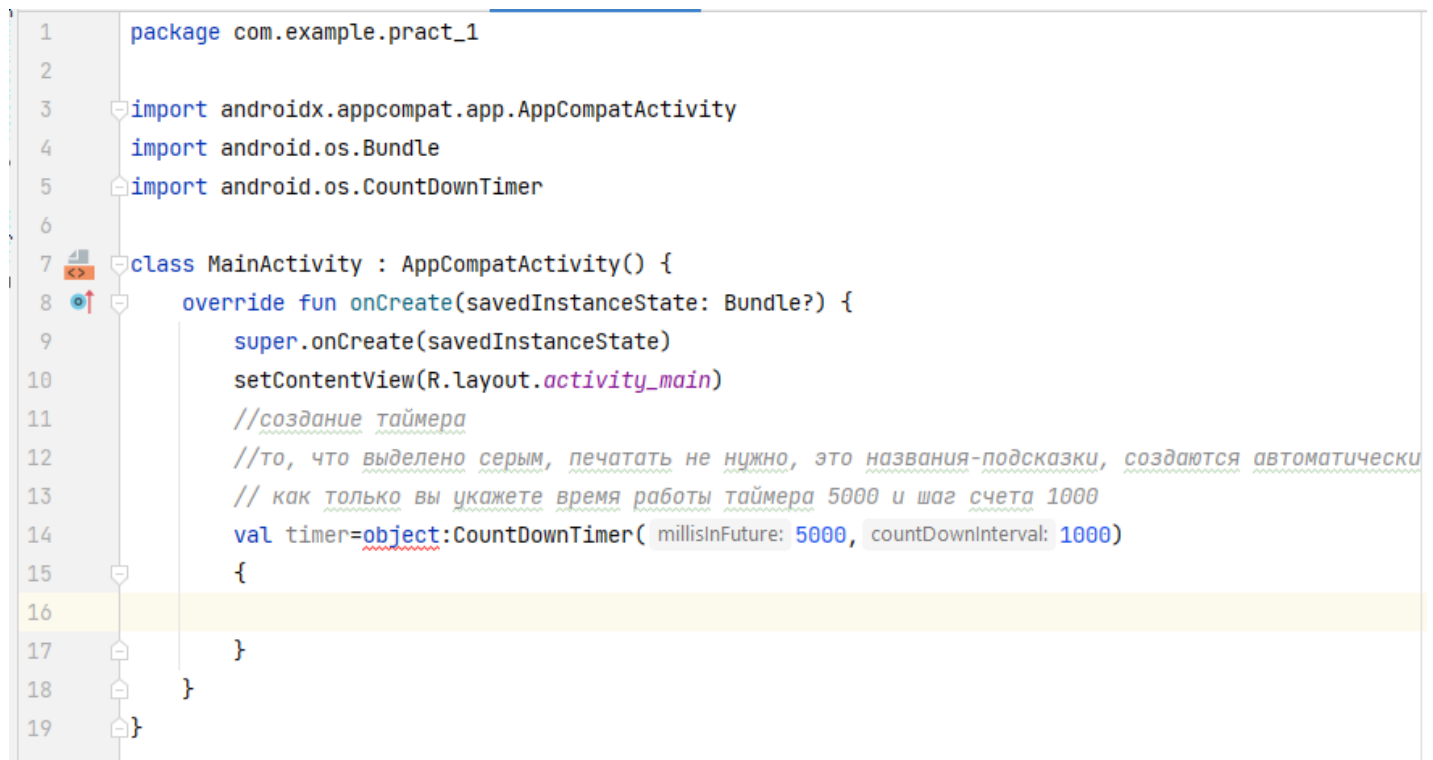
код содержит пояснения в виде комментариев, комментарии перепечатывать не нужно

Откройте проект, созданный в 1 практической работе. Настройте смену экранов приложения по таймеру и по нажатию на кнопку

Откройте файл логики первой активности MainActivity.kt



Напишите следующие строки кода для настройки таймера.



В 16 строке начните печатать override, а появившемся окне автозавершения команды выберите метод onTick - что делать пока идет отсчет времени по таймеру. Пока таймер считает, ничего происходить не будет, удалите 17 строку кода:

```
activity_main.xml x activity_main2.xml x MainActivity.kt x activity_main3.xml x MainActivity3.kt x AndroidManifest.xml
1 package com.example.pract_1
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.os.CountDownTimer
6
7 class MainActivity : AppCompatActivity() {
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11        //создание таймера
12        //то, что выделено серым, печатать не нужно, это названия-подсказки, создаются автомат
13        // как только вы укажете время работы таймера 5000 и шаг счета 1000
14        val timer=object:CountDownTimer( millisInFuture: 5000, countDownInterval: 1000)
15        {
16            ove
17            override fun onFinish() {...} CountDownTimer
18            override
19            override fun onTick(millisUntilFinished: Long) {...} CountDownTimer
20        }
21    }
22}
```

```
7 class MainActivity : AppCompatActivity() {
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11        //создание таймера
12        //то, что выделено серым, печатать не нужно, это названия-подсказки, создаются автоматически
13        // как только вы укажете время работы таймера 5000 и шаг счета 1000
14        val timer=object:CountDownTimer( millisInFuture: 5000, countDownInterval: 1000)
15        {
16            override fun onTick(millisUntilFinished: Long) {
17                TODO( reason: "Not yet implemented")
18            }
19        }
20    }
21}
```

После закрывающей скобки (18 строка) начните печатать overr и в качестве завершения команды выберите метод onFinish - что делать, когда таймер закончит отсчет времени:

```

7  class MainActivity : AppCompatActivity() {
8      override fun onCreate(savedInstanceState: Bundle?) {
9          super.onCreate(savedInstanceState)
10         setContentView(R.layout.activity_main)
11         //создание таймера
12         //то, что выделено серым, печатать не нужно, это названия-подсказки, создаются авт
13         // как только вы укажете время работы таймера 5000 и шаг счета 1000
14         override fun onFinish() {...} CountDownTimer(interval: 1000)
15         override
16         override fun hashCode(): Int {...} Any
17         override fun toString(): String {...} Any
18         override fun equals(other: Any?): Boolean {...} Any
19         Ctrl+Down and Ctrl+Up will move caret down and up in the editor Next Tip
20         over
21     }
22 }
23

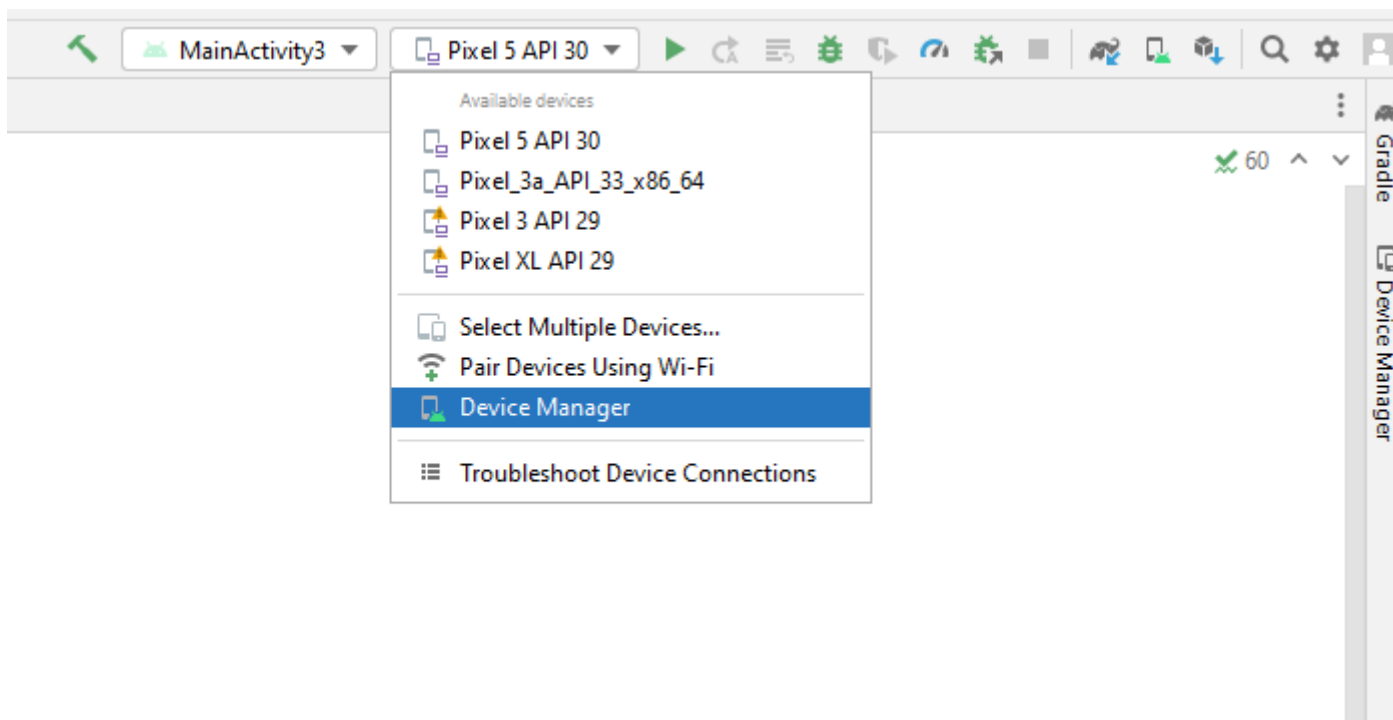
```

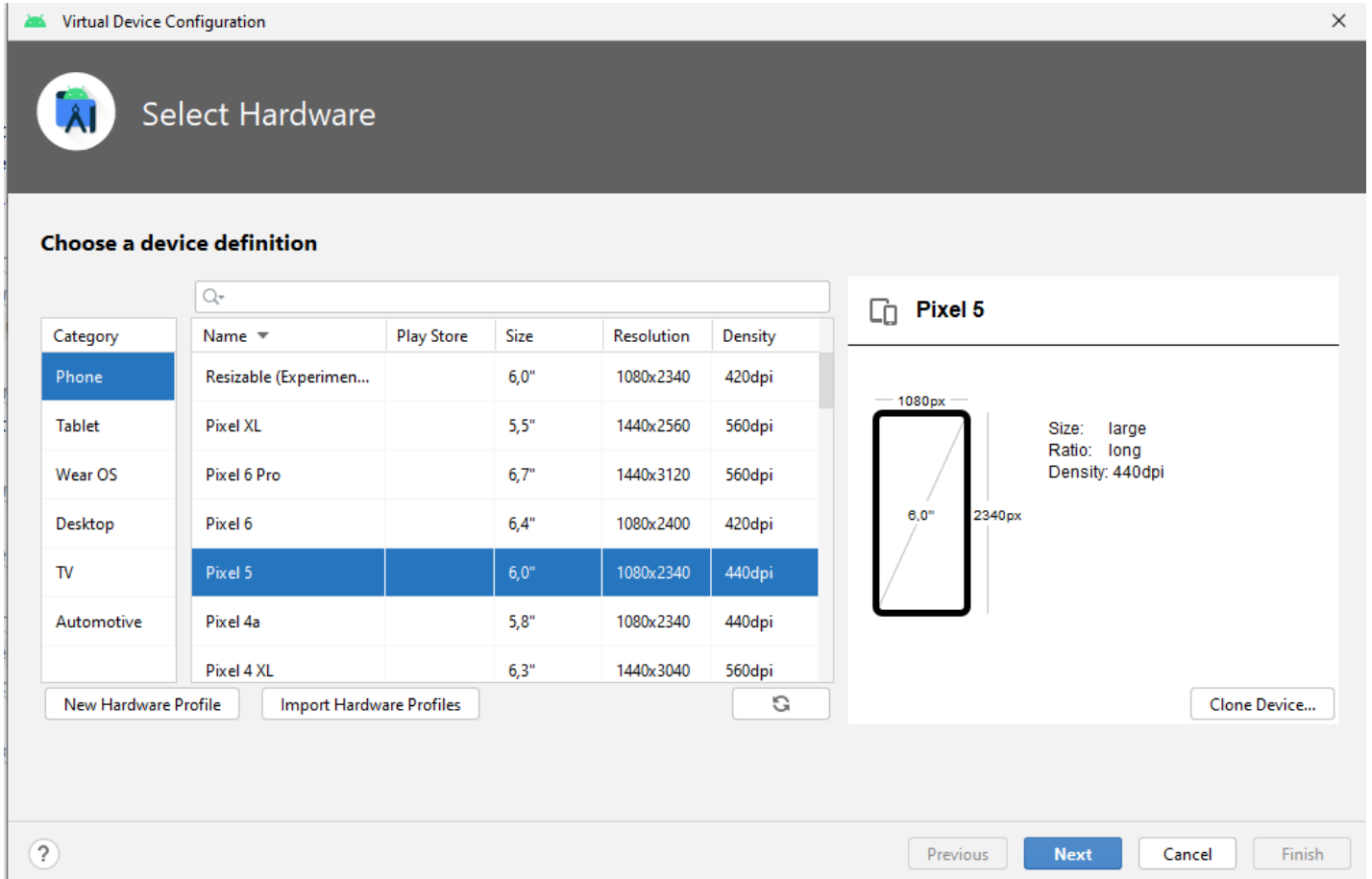
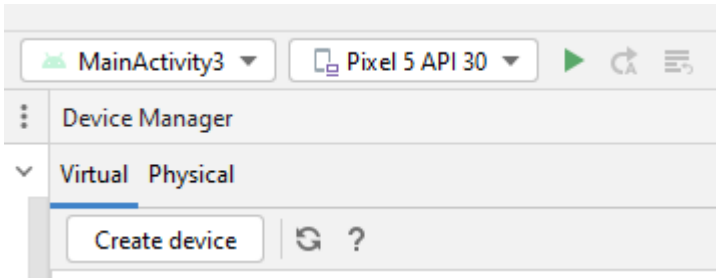
Настройте переключение экранов, прописав следующие строки кода:

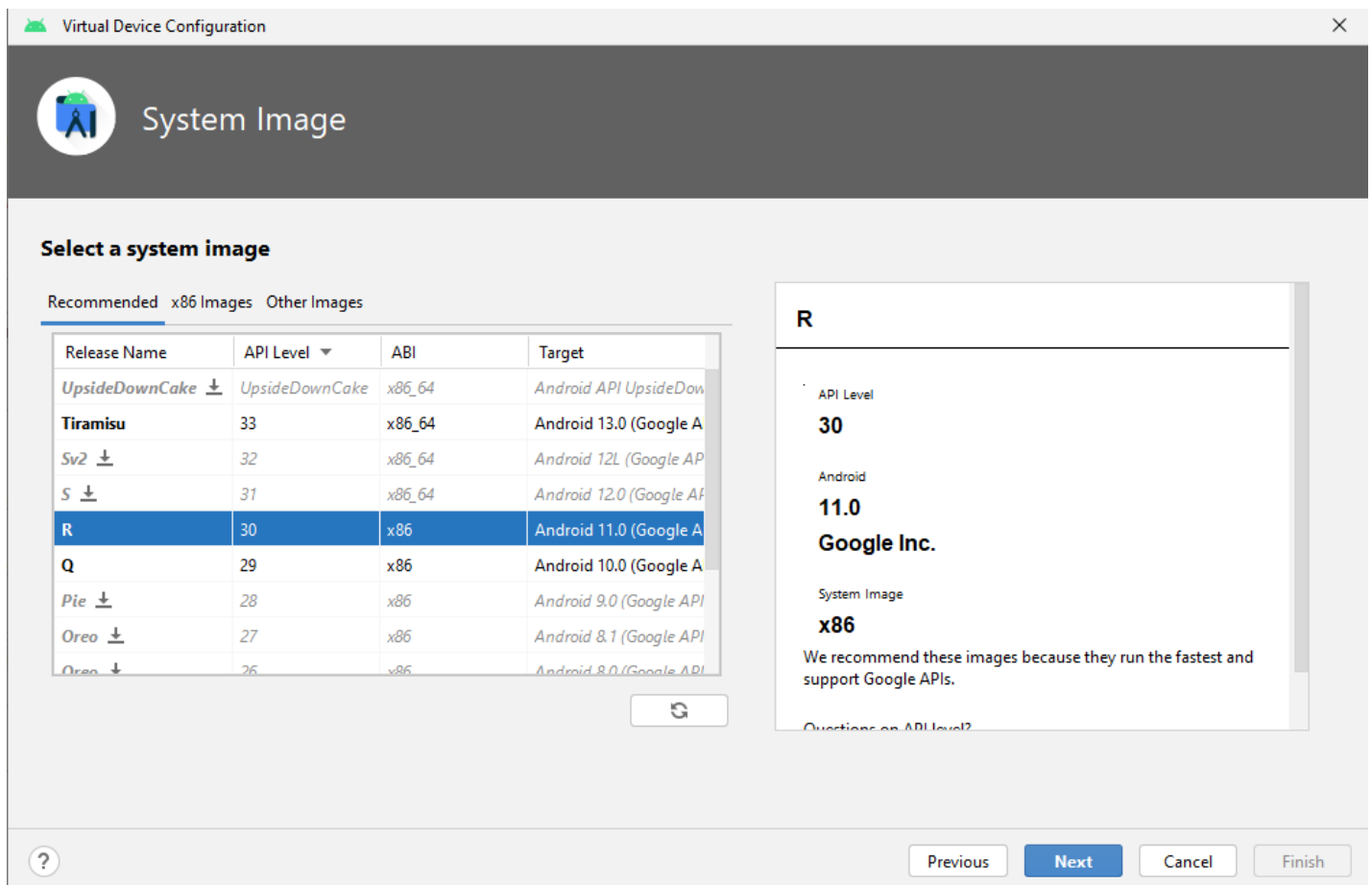
```
Pract_1 > app > src > main > java > com > example > pract_1 > MainActivity
activity_main.xml x activity_main2.xml x MainActivity.kt x activity_main3.xml x MainActivity3.kt x AndroidManifest.xml x MainActivity2.kt x image.png x

1 package com.example.pract_1
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.os.CountDownTimer
7
8 class MainActivity : AppCompatActivity() {
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12         //создание таймера
13         //то, что выделено серым, печатать не нужно, это названия-подсказки, создаются автоматически
14         // как только вы укажете время работы таймера 5000 и шаг счета 1000
15         val timer=object:CountDownTimer( millisInFuture: 5000, countDownInterval: 1000)
16         {
17             //что делать, пока время таймера не закончилось
18             override fun onTick(millisUntilFinished: Long) {
19             }
20             //что делать, когда отсчет завершился
21             override fun onFinish() {
22                 //настройка перехода с текущей активити MainActivity - имя вашей активити, с которой нужно совершить переход
23                 //на экран MainActivity2 - имя вашей активити, на которую следует перейти, какой экран отобразить
24                 //packageContext: печатать не надо
25                 val intent=Intent( packageContext: this@MainActivity, MainActivity2::class.java)
26                 //реализовать переход, указать имя интента, в данном примере это intent, но оно может быть любым
27                 startActivity(intent)
28                 //закрыть-завершить активити
29                 finish()
30             }
31         }
32         //запустите таймер
33         timer.start()
34     }
35 }
36
37
```

Настройте виртуальный эмулятор для запуска приложений:

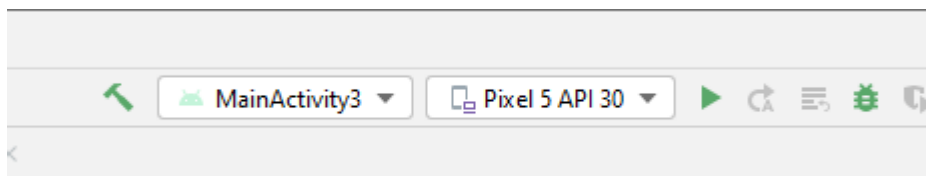




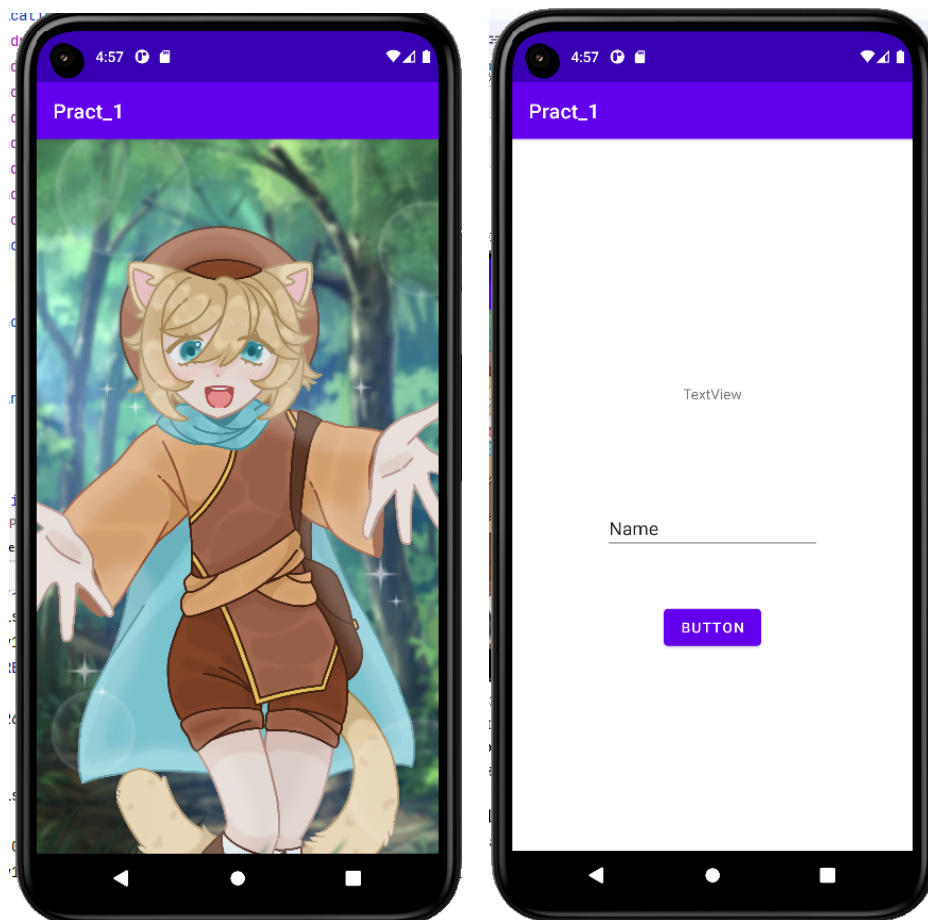


На этом этапе нажмите на значок скачать и подождите, пока установка будет завершена, затем нажмите Finish.

Убедитесь, что в окне выбранного устройства отображается только что установленный эмулятор Pixel 5



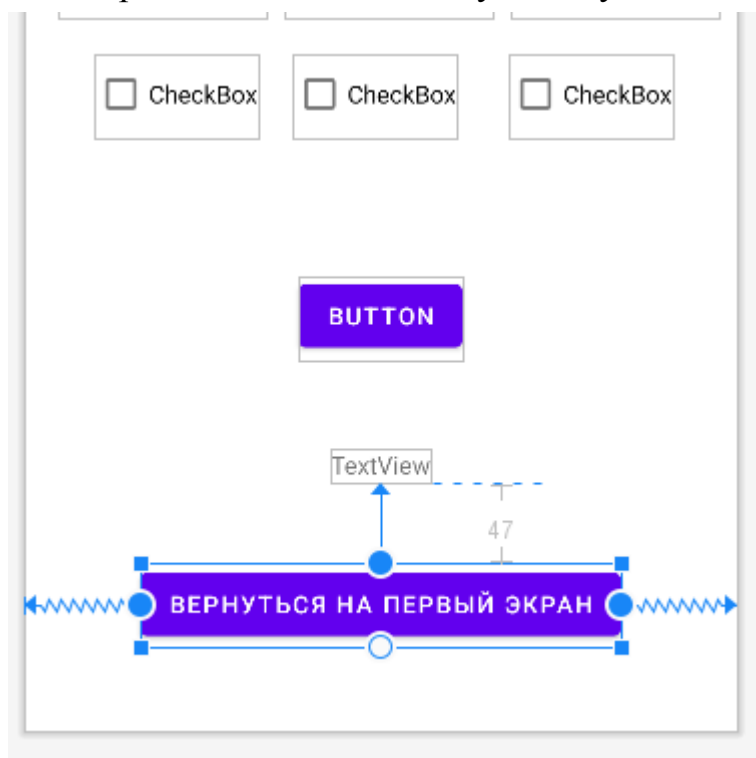
Нажмите на значок Run - запустить (зеленый треугольник). Если вы все сделали правильно, у вас на экране отобразится телефон с загрузочным экраном вашего приложения, который сменится на 2 экран.



Самостоятельно настройте автоматический переход со 2 экрана на 3 экран через 10000 ms с шагом 2000.

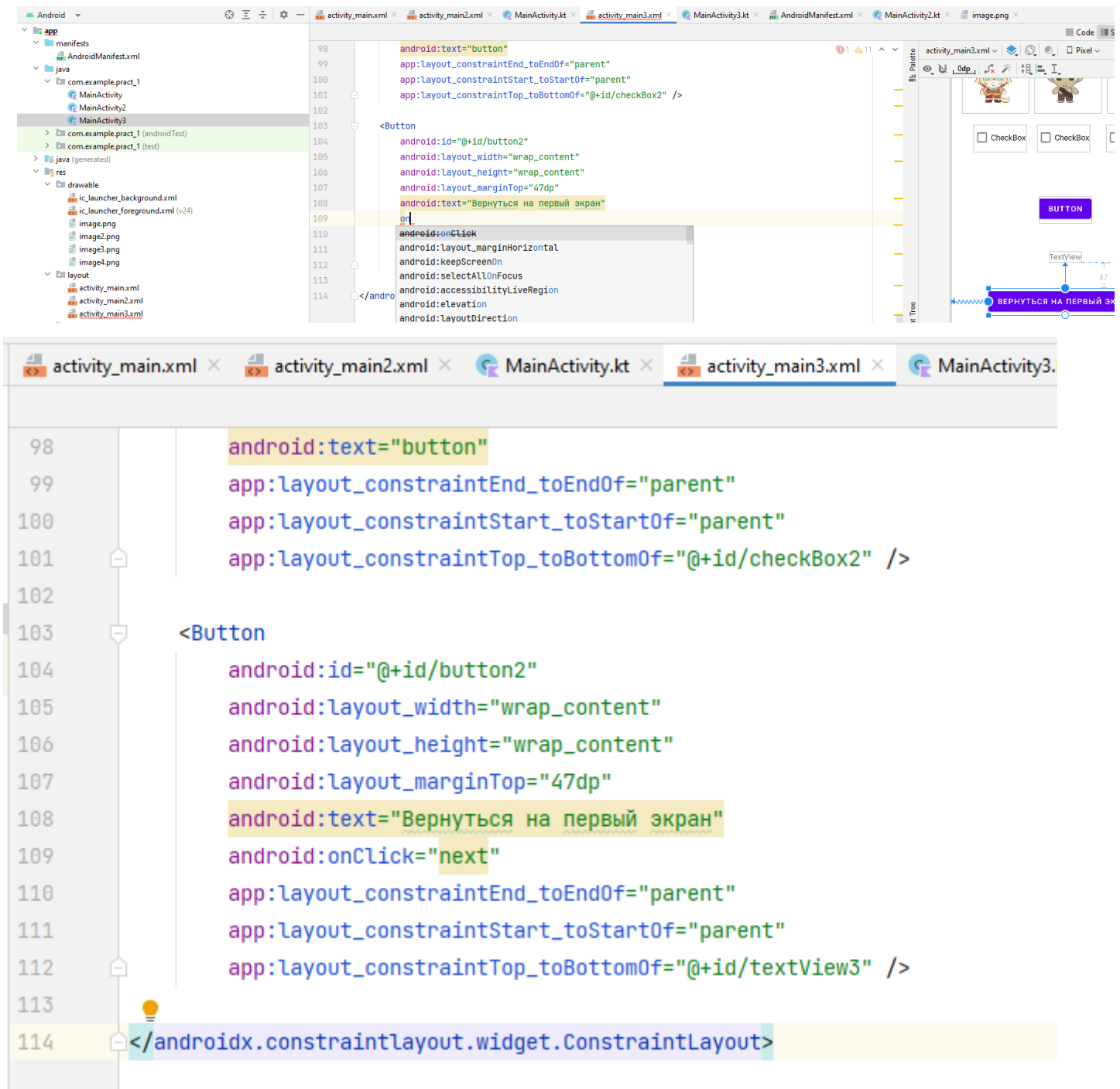
Запустите выполнение приложения на эмуляторе.

На 3 экране добавьте еще одну кнопку:



Смена экранов может происходить не только автоматически, но и по нажатию на кнопку (или иное действие). В файле разметки `activity_main3.xml` для созданной кнопки добавьте параметр

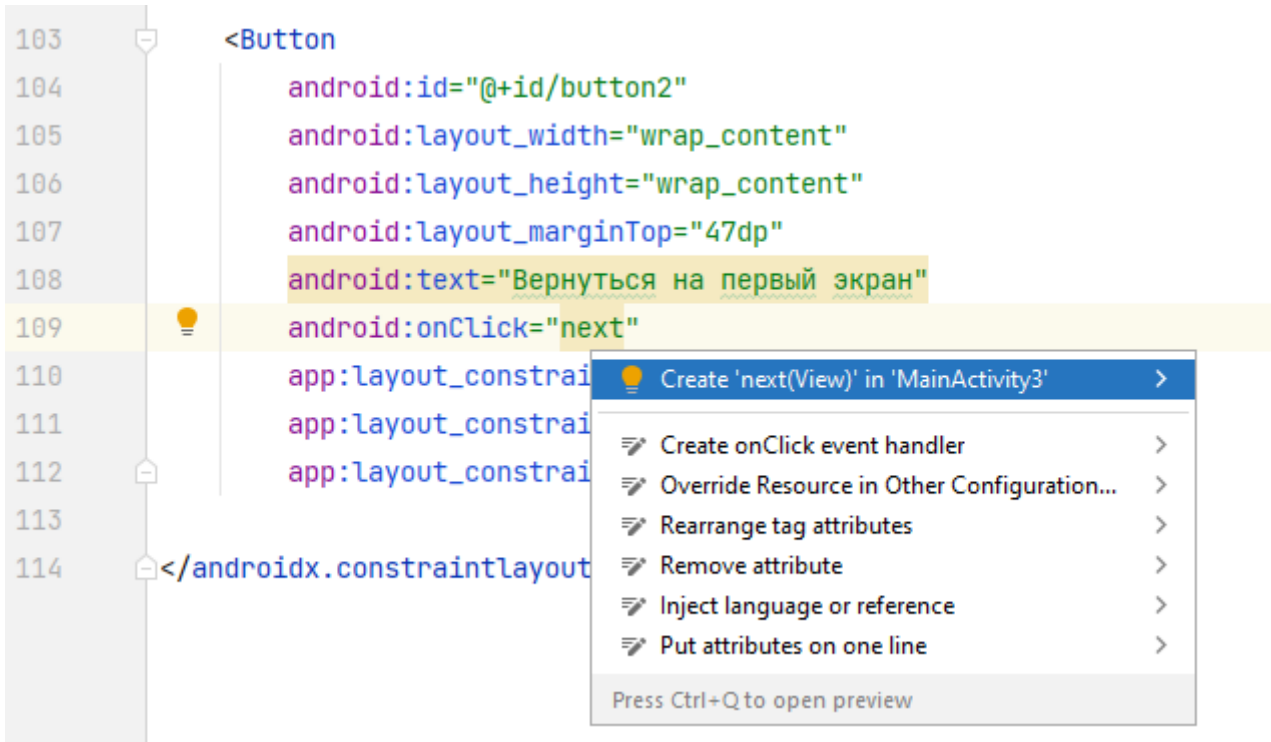
`android:onClick="next"` - обработка события нажатия на элемент, при нажатии на элемент будет вызвана и выполнена функция "next" (имя функции может быть любым)



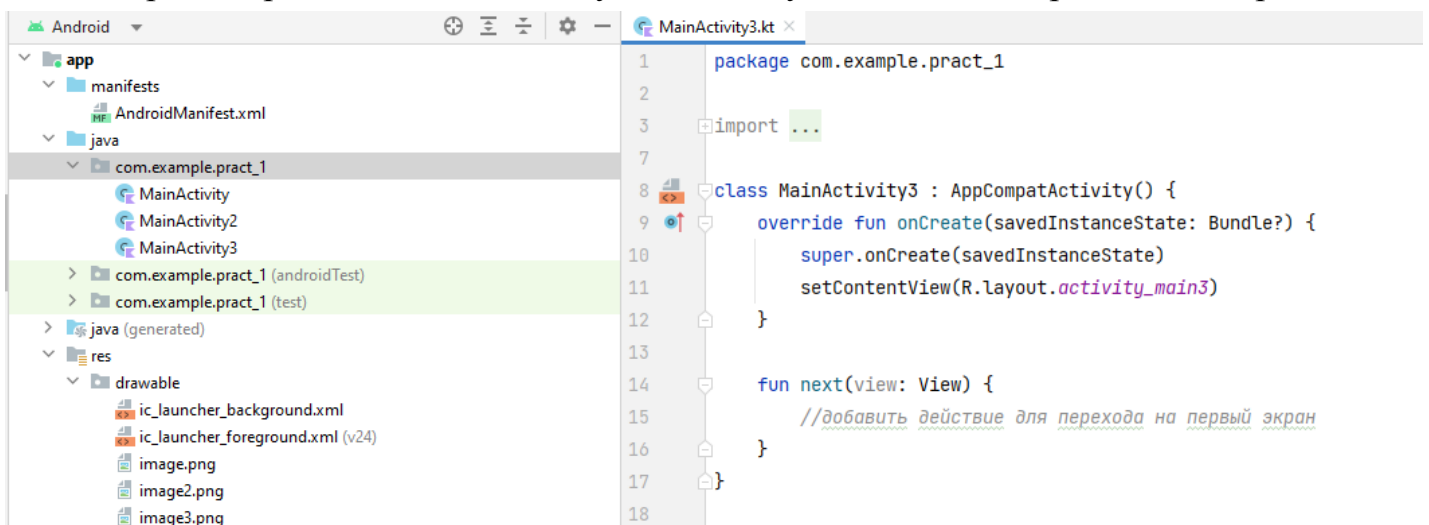
Однако, среда разработки не знает что такое next - Эту функцию необходимо создать в файле логики активности:

- установите курсор внутри слова next
- нажмите комбинацию клавиш Alt+Enter

- выберите вариант Create 'next(View)' in 'MainActivity3'



Откройте файл логики 3 экрана. В теле функции next напишите действие для смены экрана. При нажатии на кнопку должен осуществляться переход на 1 экран.



Запустите приложение и проверьте как работает смена экранов.