

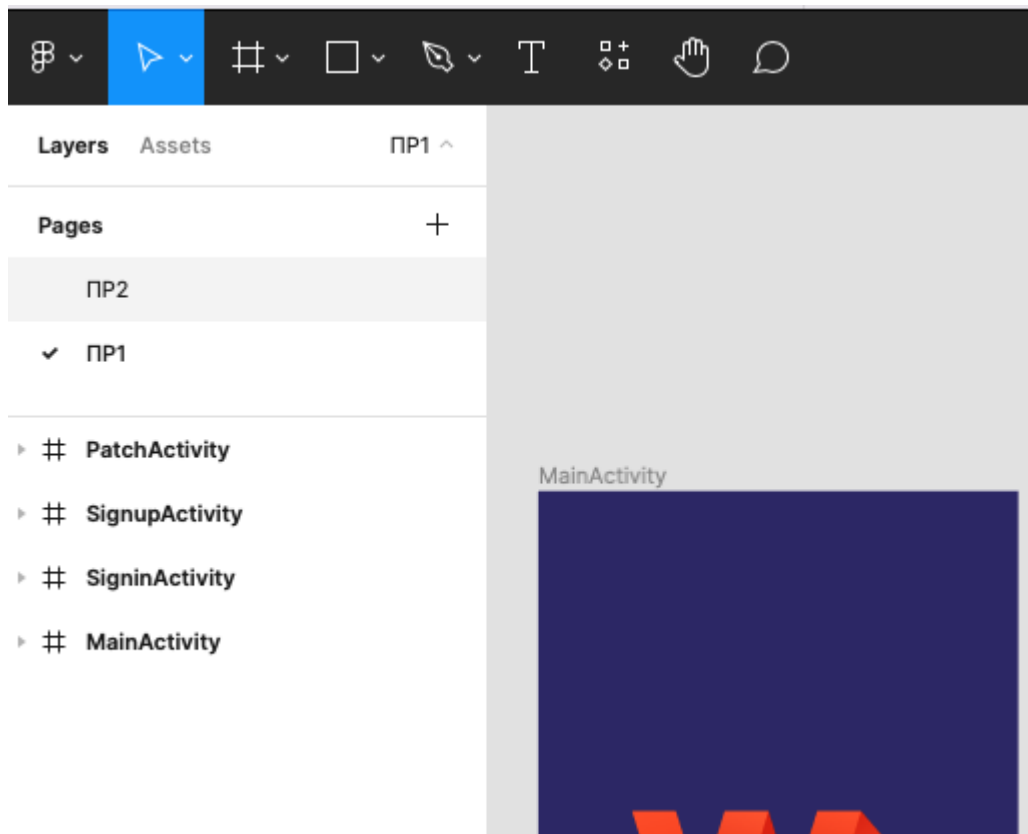
Практическая работа 4

“Создание проекта. Верстка загрузочного экрана по макету”

если числовые значения на макете не совпадают с числовыми значениями в тексте или на изображениях, приоритетными считать значения на макете

Откройте макет дизайна приложения по ссылке:
<https://www.figma.com/file/DkXzGaO7kAGdrZ987ItlzC/%D0%9F%D0%A01?node-id=0%3A1>

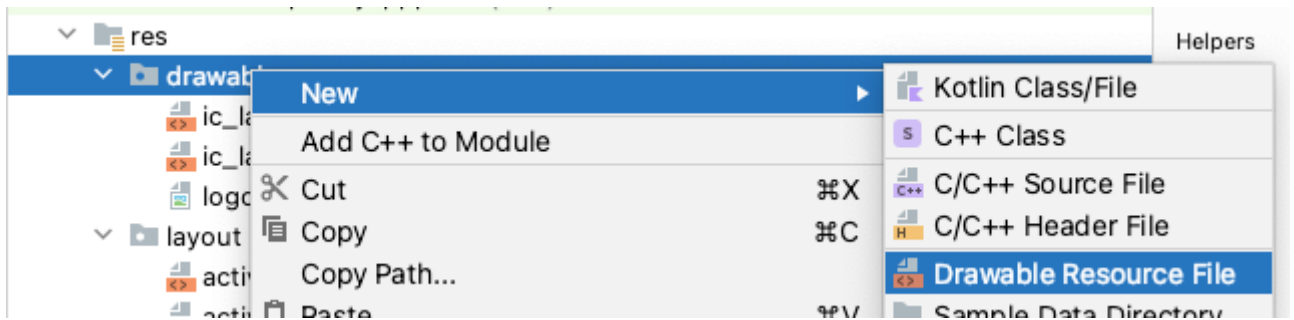
Переключитесь на вторую страницу макета ПР4:



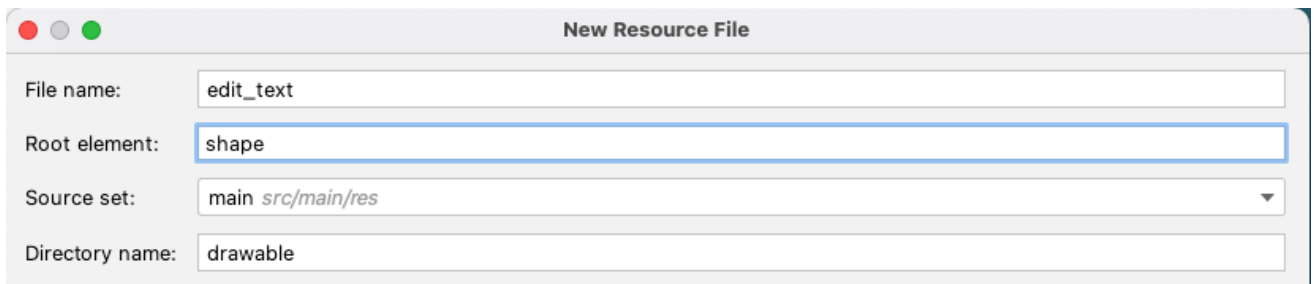
Для оформления внешнего вида элементов интерфейса необходимо использовать специальные файлы ресурсов, позволяющие настраивать внешнее оформление элементов интерфейса, а для размещения - организовать вертикальную прокрутку экрана.

Откройте созданный ранее проект Cinema

Для оформления текстовых полей и кнопок создайте специальные файлы - Drawable Resource File. Все ресурсы проекта должны быть размещены в папке res - drawable. Для создания файла ресурсов для оформления текстового поля щелкните левой кнопкой мыши по папке drawable и в появившемся меню выберите команду New - Drawable Resource File:



О появившемся окне настроек укажите имя файла, в поле Root element пропишите shape (позволяет использовать настройку простых геометрических фигур для настройки внешнего вида элементов интерфейса) и нажмите OK:



Текстовое поле на макете выглядит как контурный прямоугольник, серого цвета со скругленными углами. В созданном файле пропишите следующий код для отрисовки такой фигуры:



3 строка кода - задание формы геометрической фигуры

4 - задание радиуса скругления углов

5 - задание толщины и цвета линии контура (все значения берете с макета)



Вы настроили файл оформления. Для применения файла ресурсов в качестве шаблона оформления внешнего вида, необходимо в файле разметки для настраиваемого элемента интерфейса прописать следующую строку, где `edit_text` - имя созданного ресурсного файла:

```
9      <EditText
10          android:id="@+id/editTextTextPersonName"
11          android:background="@drawable/edit_text"
```

До применения файла ресурсов:

Name

После:

Name

Теперь текстовое поле выглядит как на макете, все остальные настройки для этого компонента выполните самостоятельно (обратите внимание, на макете представлен образец оформления пустого и заполненного текстового поля).

Создайте файл ресурсов для оформления красной кнопки `red_but.xml`. Код файла:

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <shape xmlns:android="http://schemas.android.com/apk/res/android"
3      android:shape="rectangle">
4      <solid android:color="@color/red"/>
5      <corners android:radius="4dp"/>
6
7  </shape>

```

4 строка кода - залить фигуру сплошным цветом. Цвет - красный
Примените файл к кнопке.

```

<Button
    android:id="@+id/button3"
    android:background="@drawable/red_but"

```

Настройка применяется не полностью.



Если необходимо через ресурсный файл поменять не только закругление и обводку для элемента Button, но и выполнить настройку цвета (для простого Button через ресурсный файл цвет не задать), необходимо изменить тип кнопки с Button на `androidx.appcompat.widget.AppCompatButton` (в коде на месте button начните набирать AppCom и сразу в подсказке появится нужный тип виджета)

```

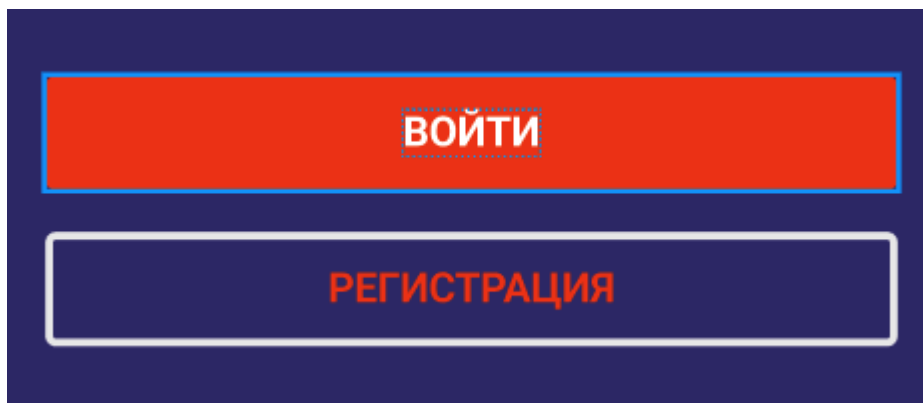
<app
    androidx.appcompat.widget.AppCompatAutoCompleteTextView
    androidx.appcompat.widget.AppCompatButton

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/button3"
    android:background="@drawable/red_but"

```



Закончите верстку второго экрана, самостоятельно настроив и применив ресурсный файл для второй кнопки.



Верстка третьего экрана будет содержать настройку вертикальной прокрутки. В Android Studio существует специальный компонент (контейнер) для настройки прокрутки экран `ScrollView`. Его можно найти в разделе `Containers`. Однако, намного проще создавать и настраивать данный компонент не в режиме `Design`, а в `Code`.

В файле разметки пропишите следующие строки кода:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8      <ScrollView
9          android:layout_width="match_parent"
10         android:layout_height="wrap_content"
11         tools:ignore="MissingConstraints" >
12         <LinearLayout
13             android:layout_width="match_parent"
14             android:layout_height="wrap_content"
15             android:orientation="vertical"
16             tools:ignore="MissingConstraints" >|
17         </LinearLayout>
18     </ScrollView>
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

8 строка кода - добавление контейнера `ScrollView` (внутри может располагаться только один компонент)

11 - игнорирование привязок

12 - добавление контейнера `LinearLayout` - группирует и упорядочивает элементы интерфейса друг за другом по вертикали или горизонтали в порядке создания

15 - тип расположения

16 - игнорирование привязок (внутри `LinearLayout` привязки и отступы справа и слева не работают, размеры элементов задаются через параметры ширины и высоты)

все элементы интерфейса, которые необходимо "прокручивать" следует прописать до закрывающего тега

17 - закрывающий тег для LinearLayout (обязателен)

18 - закрывающий тег для контейнера ScrollView

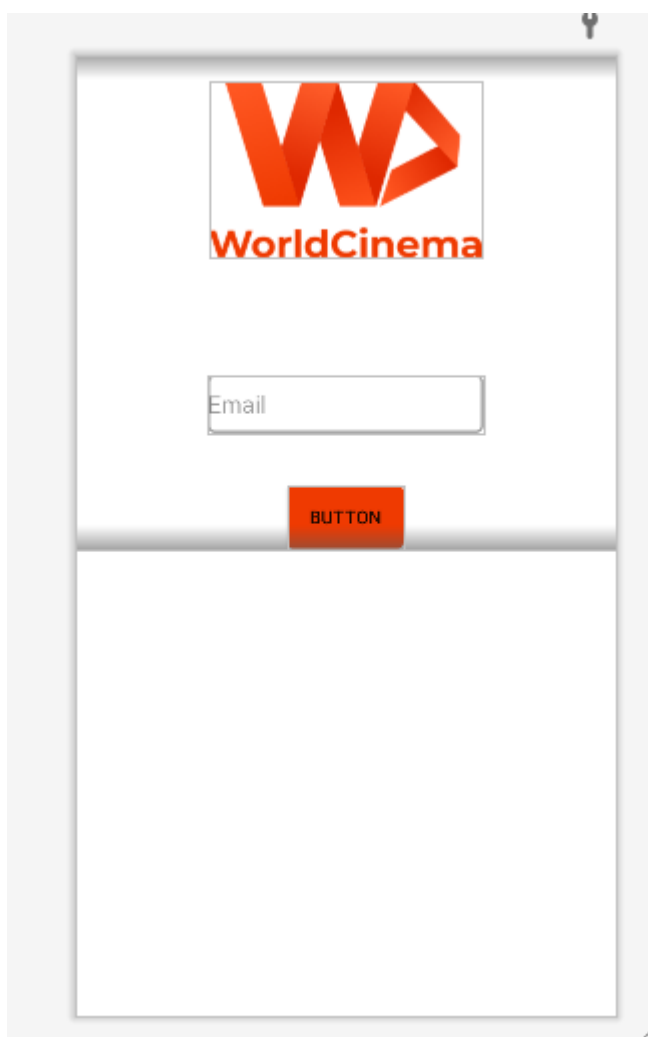
Все, что прописано внутри LinearLayout и помещено внутрь ScrollView, будет вертикально прокручиваться на экране. Размер области прокрутки задает размер контейнера ScrollView.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8     <ScrollView
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        tools:ignore="MissingConstraints" >
12        <LinearLayout
13            android:layout_width="match_parent"
14            android:layout_height="wrap_content"
15            android:orientation="vertical"
16            tools:ignore="MissingConstraints" >
17            <ImageView
18                android:id="@+id/imageView"
19                android:layout_width="207dp"
20                android:layout_height="134dp"
21                android:scaleType="centerCrop"
22                app:srcCompat="@drawable/logo"
23                android:layout_marginTop="20dp"
24                android:layout_gravity="center"
25                tools:layout_editor_absoluteX="102dp"
26                tools:layout_editor_absoluteY="97dp" />
27            <EditText
28                android:id="@+id/editText2"
29                android:layout_width="wrap_content"
30                android:layout_height="44dp"
31                android:layout_marginTop="89dp"
32                android:ems="10"
33                android:hint="Email"
34                android:background="@drawable/edit_text"
```

```

35         android:layout_gravity="center"/>
36     <androidx.appcompat.widget.AppCompatButton
37         android:id="@+id/button2"
38         android:layout_marginTop="40dp"
39         android:layout_gravity="center"
40         android:layout_width="wrap_content"
41         android:layout_height="wrap_content"
42         android:background="@drawable/red_but"
43         android:text="Button"
44         tools:layout_editor_absoluteX="204dp"
45         tools:layout_editor_absoluteY="548dp" />
46 </LinearLayout>
47 </ScrollView>
48 </androidx.constraintlayout.widget.ConstraintLayout>

```



`android:layout_gravity="center"` - разместить по центру контейнера `LinearLayout`

`android:ems="10"` - минимальный размер текстового поля - будут отображаться 20 символов

Разместите все элементы интерфейса 3 экрана в области прокрутки. Закончите верстку экрана (ориентируйтесь на макет).