Молдавский Государственный Университет
Факультет Математики и Информатики
Департамент Информатики

Лабораторная работа Nr.2
Тема: “ Создание программ с несколькими активностями. Взаимодействие действий.”

Проверил; Геогрий Латул

Выполнил: Константин Рунтов

Группа: i 1802

Кишинев 2021

Содержание

Условие задачи

Создать приложение с несколькими активностями, которые общаются между собой.

Создаем 1 активности, первая является главной и вызывает вторую для выбора пользователем изображения из предложенного списка. При выборе картинки, активность уничтожается, а на главной активности отображается выбранное пользователем изображение. Под элементом отображения выбранного изображения, расположен счетчик вызова второй активности.

Выполнение задачи

1. Создание приложения

1.1 Создание скелета приложения

Первым делом созданием скелет нашего приложения на основе макета с пустой активностью. На основе данного макета будем выстраивать вид и логику нашего приложения.

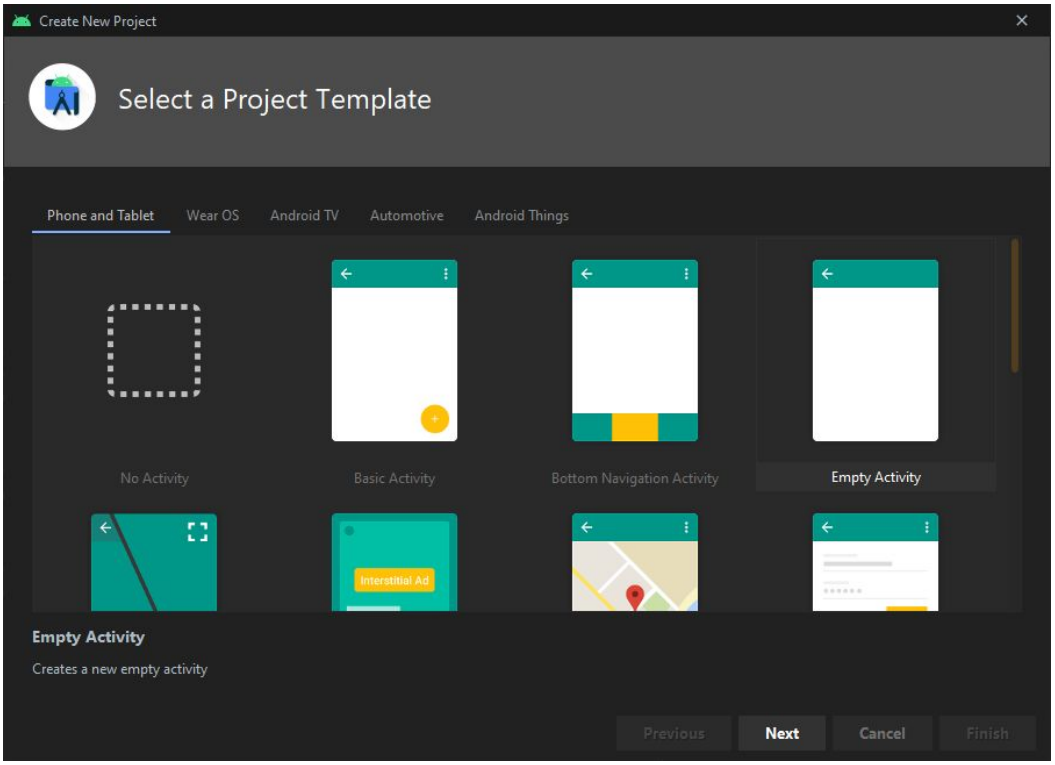

Рис. 1.1 Окно выбора шаблона приложения

1.2 Добавление активных элементов

Редактируем основное окно приложения путём добавления следующих элементов в файл *activity_main.xml* :

- Текстовое поле с названием приложения, а также счётчик вызова второй активности — *TextView*
- Кнопки, расположенных по центру окна приложения — *Button*
- Элемент отображения изображения — *ImageView*

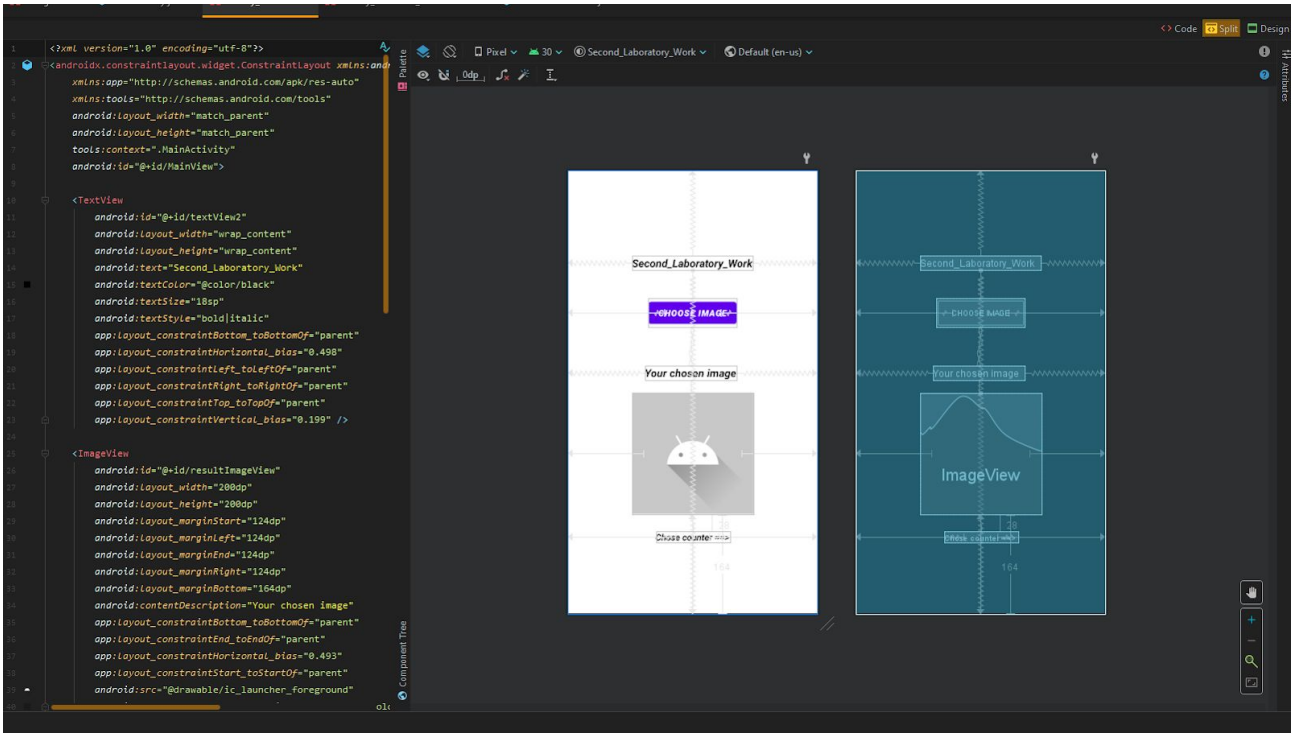

Рис. 1.2 Конструктор главного окна

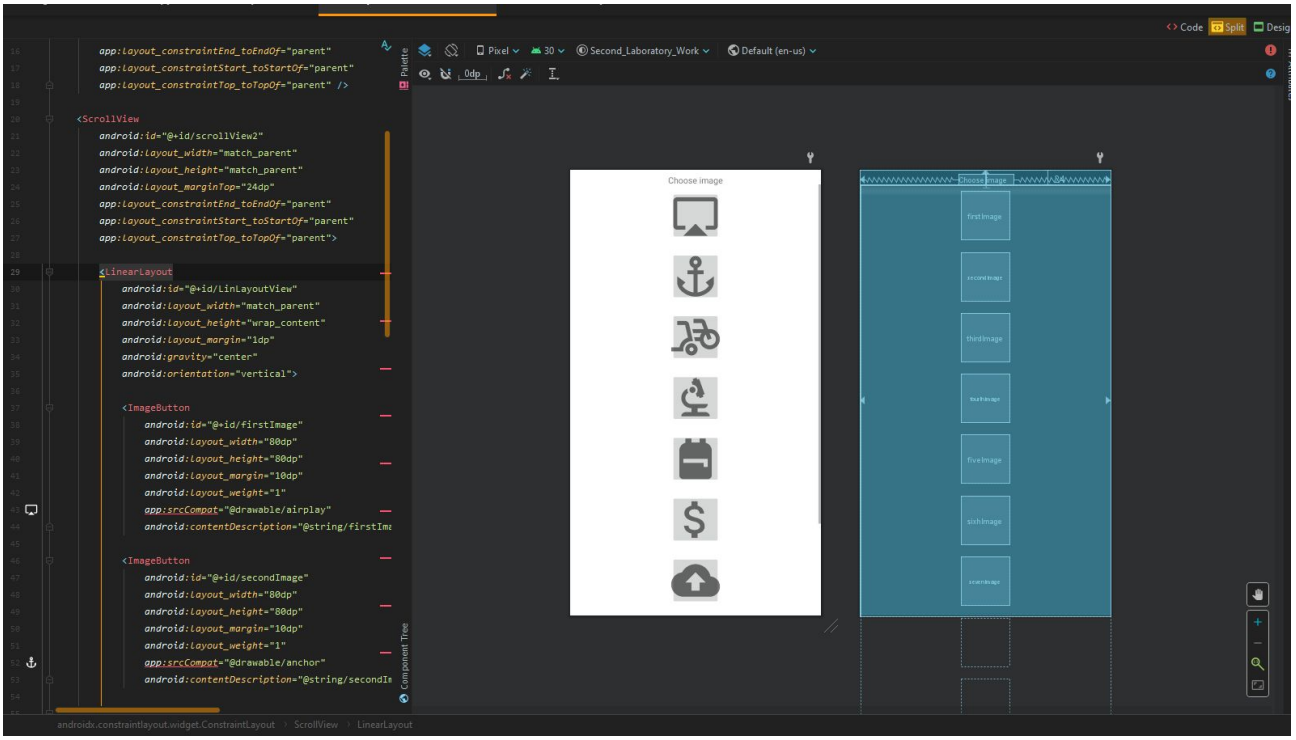

Рис. 1.3 Конструктор окна выбора изображения

Для корректного объявления отображаемого текста на кнопках и текстовом поле, добавляем соответствующие элементы в файл *strings.xml*, который содержит

все ресурсы данного типа. Использование данного подхода облегчает создание и редактирование текстов, используемых приложением, так как при необходимости изменить какую-нибудь часть текста не нужно будет искать каждый элемент, который содержит данный текст, а просто изменить данный объект и изменения автоматически будет применены ко всем использующим объектам.

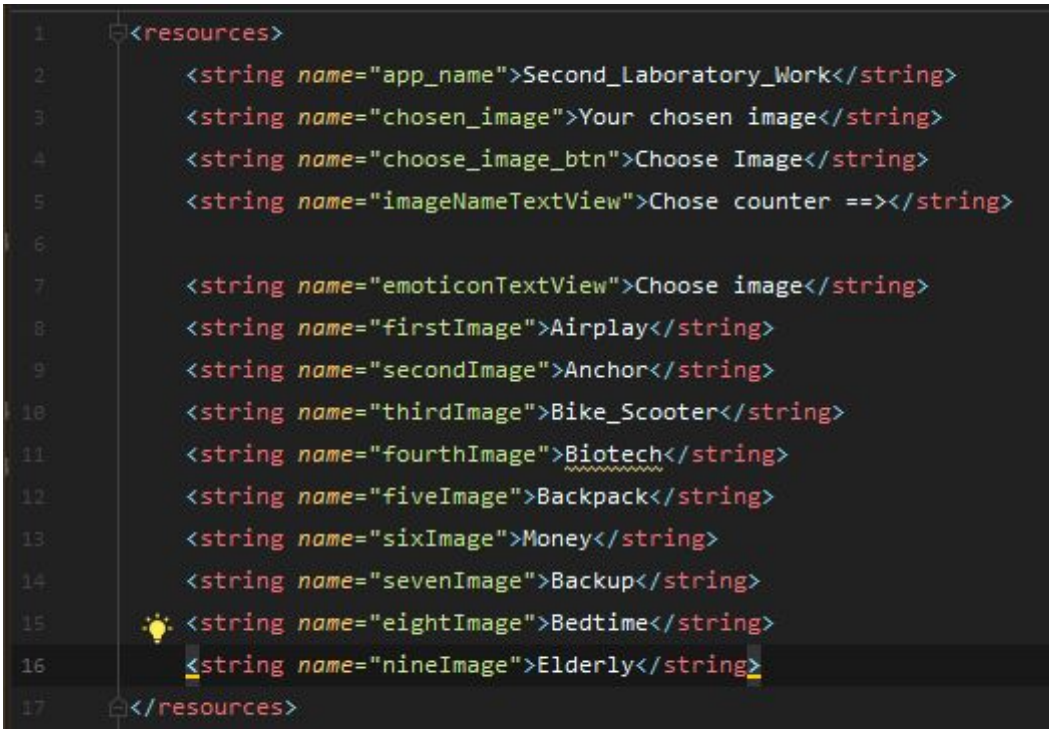

Рис. 1.4 Содержимое strings.xml

Получение текста осуществляется путём вызова *“@string/<name_value>”* в качестве значения для параметра объекта содержащего текстовое значение.

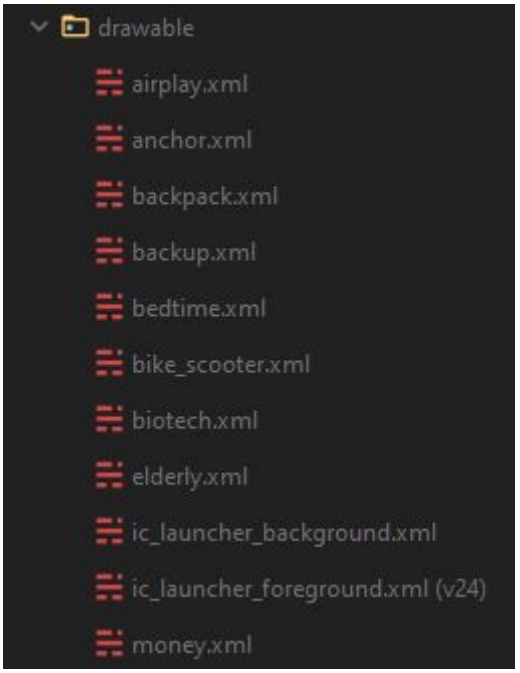

Рис. 1.5 Содержание папки *drawable*

Данная папка содержит объекты, которые могут быть используованы в качестве ресурсов изображения

1.3 Написание кода приложения

После создания вида нашего основного окна приложения следует написать код, для корректного поведения наших добавленных элементов.Поведение наших объектов будем описывать в файле *MainActivity.java*, который отвечает за основное окно нашего приложения.

Первым делом добавим пару параметров в метод *onCreate.* Данные параметры будут отвечать за вызов метода инициализации слушателей на кнопку и установление текста на элемент отвечающий за отображение счетчика вызова второй активности.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
    /// Initialize button listener
    this.init_listeners();
    /// Setting TextView counter
    this.setChoseCounterText();
}
```

```
/**
* Method which initializes button listener
* */
private void init_listeners(){
    findViewById(R.id.ImageChooseBtn).setOnClickListener(view -> {
        Intent intent = new Intent(MainActivity.this,
EmoticonSelection.class);
        startActivityForResult(intent, 1);
    });
}
```

Данный метод отвечает за поиск кнопки по её *id* и инициализацию слушателя для неё. Действие, исполняемое данным слушателем задаётся путем использования лямбда выражения, так как в нашем случае выполняется лишь одно действие.

Слушатель создаёт объект *Intent* в который мы передаем наш текущий класс как контекст и класс компонента который будет вызван. После запускаем нашу активность для её выполнения и получения результата, передавая ей код запроса.

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (resultCode == 1) {
        this.choseCounter++ ;
```

```
        this.init_image(data);
        this.setChoseCounterText();
    }
}
```

Данный метод вызывается при завершении вызванной активности. Выполняется проверка ответ-кода и при прохождении данной проверки выполняются следующие действия:

1. Счётчик вызова увеличивается
2. Вызывается метод установки выбранного изображения в который передается полученные данные от отработавшей активности
3. Вызов метода для обновления текста счётчика вызовов

```
private void init_image(Intent savedState){
    if (savedState == null) { return;}
    ImageView imageView = findViewById(R.id.resultImageView);
    imageView.setImageResource(
            this.drawable_Choose(
                    savedState.getIntExtra("ImageChosen",
R.id.resultImageView)
            )
    );
}

private int drawable_Choose(int btn){
    switch(btn){
        case R.id.firstImage:
            return R.drawable.airplay;
        case R.id.secondImage:
            return R.drawable.anchor;
        case R.id.thirdImage:
            return R.drawable.bike_scooter;
        case R.id.fourthImage:
```

```
            return R.drawable.biotech;
        case R.id.fiveImage:
            return R.drawable.backpack;
        case R.id.sixhImage:
            return R.drawable.money;
        case R.id.sevenImage:
            return R.drawable.backup;
        case R.id.eightImage:
            return R.drawable.bedtime;
        case R.id.nineImage:
            return R.drawable.elderly;
        default:
            return R.drawable.ic_launcher_foreground;
    }
}
```

Метод обновления элемента ImageView сперва проверяет не является ли входной параметр *null* и в случает, если он не пустой, то выполняется создание объекта ImageView которому присваивается найденный по id элемент ImageView.

Следующим шагом устанавливаем ресурс нашего изображения путём указания id желаемого ресурса. Данный id мы получаем путём передачи полученного id выбранной пользователем кнопки из вызванной активности в метод *drawable_Choose* который на его основе возвращает id ресурса использованного в выбранной кнопке.

```
private void setChoseCounterText(){
    TextView chooseCounterShow = findViewById(R.id.imageNameTextView);
    chooseCounterShow.setText(
            String.format(
                    "%s %s",
                    getString(R.string.imageNameTextView),
                    (this.choseCounter == 0) ? "First Launch" :
Integer.toString(this.choseCounter)
            )
```

```
    );
}
```

Данный метод выполняет обновление текста показываемого элементом TextView по такой же логике как и вышеописанный метод, только сейчас вызывается метод в который мы передаем форматированную строку состоящую из двух элементов:

1. Объект полученный из файла *strings.xml* по указанному id
2. Результат выполнения тернарного оператора:
    a. Строка “*First Launch”* при значении счётчика равном 0
    b. Значение счётчика конвертированное в String

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_emoticon_selection);

    /// ImageButtons initialization
    this.initButtons();
}

private void initButtons(){
    buttons = new ImageButton[]{
            findViewById(R.id.firstImage),
            findViewById(R.id.secondImage),
            findViewById(R.id.thirdImage),
            findViewById(R.id.fourthImage),
            findViewById(R.id.fiveImage),
            findViewById(R.id.sixhImage),
            findViewById(R.id.sevenImage),
            findViewById(R.id.eightImage),
            findViewById(R.id.nineImage)
    };
    this.initListeners();
```

```
}

private void initListeners(){
   for(ImageButton btn : this.buttons){
       btn.setOnClickListener(item -> {
           this.intent = new Intent(EmoticonSelection.this,
MainActivity.class);
           intent.putExtra("ImageChosen", btn.getId());
           setResult(1, intent);
           finish();
       });
   }
}
```

Код второй активности, которая вызывается главной для выбора пользователем изображения из списка доступных.

В методе *onCreate* вызывается метод инициализации массива *ImageButton*, в котором также вызывается метод инициализации слушателей на каждую из кнопок. Данные слушатели выполняют следующие действия:

1. Создание объекта *Intent*
2. Передача в него id кнопки под ключом с названием *“ImageChosen”*
3. Вызов метода *setResult* для установки результата выполнения активности под кодом результата *1*
4. Завершение данной активности

2. Работа приложения

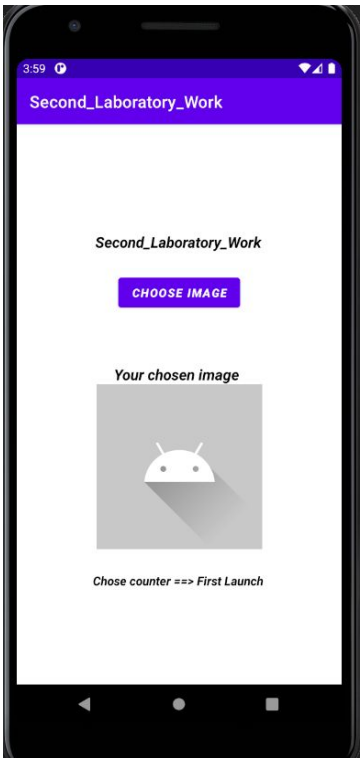

Рис. 1.4 Начальное состояние

Рис. 1.5 Вызванная активность при нажатии кнопки

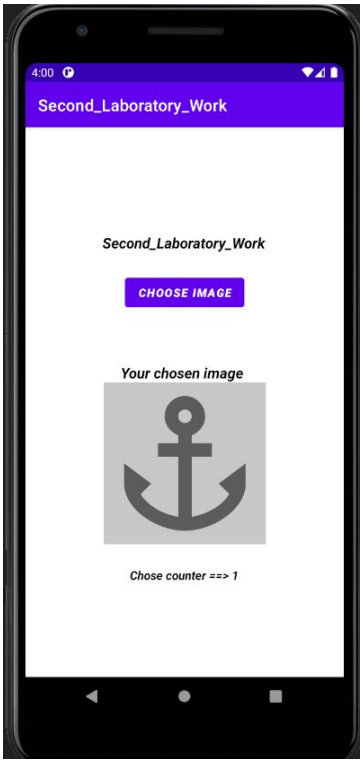

Рис. 1.6 Показ выбранного изображения и изменения значения счётчика

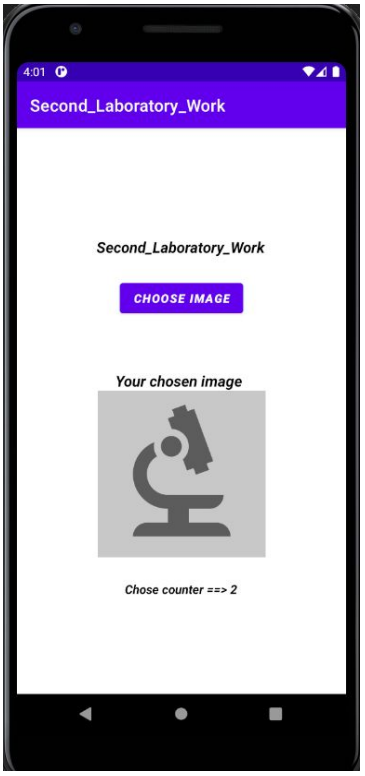

Рис. 1.7 Выбрано новое изображение и счётчик увеличился

Вывод

Выполнение данной работы дало понимание того, как взаимодействуют активности приложения между собой при помощи методов *startActivityForResult* и *onActivityResult.*