**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 17**

по дисциплине «Разработка мобильных приложений»

**Тема: Unit 2 Пользовательский интерфейс “Адаптивные карты и цвета”**

Студент группы РИЗ-300028у: А.С. Севостьянов

Преподаватель: С.И.Тимошенко,

доц., к.т.н.

**Екатеринбург 2022**

содержание

[1 Постановка задачи 3](#_Toc118041527)

[2 Результаты проделанной работы 4](#_Toc118041528)

[3 Анализ полученных результатов 5](#_Toc118041529)

## 1 Постановка задачи

Модифицировать приложение в соответствии с рекомендациями Material Desing. Добавить изображения и стиль в Recycler View список. Реализовать ItemTochHelper, чтобы добавить функциональность перетаскивания в приложение. Добавить альтернативный макет для планшетов. Локализировать содержимое приложения.

.

## Результаты проделанной работы

Программный код проекта, файл проекта MainActivity.

package com.example.android.materialme;  
  
import android.content.res.TypedArray;  
import android.os.Bundle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.GridLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.support.v7.widget.helper.ItemTouchHelper;  
import android.view.View;  
  
import java.util.ArrayList;  
import java.util.Collections;  
  
*/\*\*\*  
 Основное действие приложения Material Me, имитирующего приложение для спортивных новостей.  
 \*/*public class MainActivity extends AppCompatActivity {  
  
 // Переменные-члены.  
 private RecyclerView mRecyclerView;  
 private ArrayList<Sport> mSportsData;  
 private SportsAdapter mAdapter;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Инициализируйте RecyclerView.  
 mRecyclerView = findViewById(R.id.*recyclerView*);  
  
 // Получить соответствующее количество столбцов.  
 int gridColumnCount = getResources().getInteger(R.integer.*grid\_column\_count*);  
  
// Установите менеджер компоновки.  
 mRecyclerView.setLayoutManager(new GridLayoutManager(  
 this, gridColumnCount));  
  
// Инициализируйте ArrayList, который будет содержать данные.  
 mSportsData = new ArrayList<>();  
  
 // Инициализируйте адаптер и установите его в RecyclerView.  
 mAdapter = new SportsAdapter(this, mSportsData);  
 mRecyclerView.setAdapter(mAdapter);  
  
// Получить данные.  
 initializeData();  
  
// Если имеется более одного столбца, отключите свайп, чтобы отклонить  
 int swipeDirs;  
 if(gridColumnCount > 1){  
 swipeDirs = 0;  
 } else {  
 swipeDirs = ItemTouchHelper.*LEFT* | ItemTouchHelper.*RIGHT*;  
 }  
  
 // Вспомогательный класс для создания салфетки для удаления и перетаскивания  
 // функциональность  
 ItemTouchHelper helper = new ItemTouchHelper(new ItemTouchHelper  
 .SimpleCallback(  
 ItemTouchHelper.*LEFT* | ItemTouchHelper.*RIGHT* |   
 ItemTouchHelper.*DOWN* | ItemTouchHelper.*UP*,  
 swipeDirs) {  
 */\*\*  
 Определяет функциональность перетаскивания.  
 \*/* @Override  
 public boolean onMove(RecyclerView recyclerView,  
 RecyclerView.ViewHolder viewHolder,  
 RecyclerView.ViewHolder target) {  
 // Получить позиции "от" и "до".  
 int from = viewHolder.getAdapterPosition();  
 int to = target.getAdapterPosition();  
  
 // Поменяйте местами элементы и уведомите адаптер.  
 Collections.*swap*(mSportsData, from, to);  
 mAdapter.notifyItemMoved(from, to);  
 return true;  
 }  
  
 */\*\*  
 Определяет свайп для отключения функциональности.  
 \*/* @Override  
 public void onSwiped(RecyclerView.ViewHolder viewHolder,  
 int direction) {  
// Удалить элемент из набора данных.  
 mSportsData.remove(viewHolder.getAdapterPosition());  
 // Уведомить адаптер.  
 mAdapter.notifyItemRemoved(viewHolder.getAdapterPosition());  
 }  
 });  
  
 // Прикрепите помощник к RecyclerView.  
 helper.attachToRecyclerView(mRecyclerView);  
 }  
  
 */\*\*  
 Инициализируйте спортивные данные из ресурсов.  
 \*/* private void initializeData() {  
 // Получите ресурсы из XML-файла.  
 String[] sportsList = getResources()  
 .getStringArray(R.array.*sports\_titles*);  
 String[] sportsInfo = getResources()  
 .getStringArray(R.array.*sports\_info*);  
 TypedArray sportsImageResources =  
 getResources().obtainTypedArray(R.array.*sports\_images*);  
  
// Очистите существующие данные (чтобы избежать дублирования).  
 mSportsData.clear();  
  
// Создайте ArrayList спортивных объектов с названиями и  
// информацией о каждом виде спорта.  
 for (int i = 0; i < sportsList.length; i++) {  
 mSportsData.add(new Sport(sportsList[i], sportsInfo[i],  
 sportsImageResources.getResourceId(i, 0)));  
 }  
  
// Переработать введенный массив.  
 sportsImageResources.recycle();  
  
// Уведомить адаптер об изменении.  
 mAdapter.notifyDataSetChanged();  
 }  
  
 */\*\*  
 Метод onClick для th FAB, который сбрасывает данные.  
 \*/* public void resetSports(View view) {  
 initializeData();  
 }  
}

Програмнный код файл, DetailActivity.

package com.example.android.materialme;  
  
import android.os.Bundle;  
import android.support.v7.app.AppCompatActivity;  
import android.widget.ImageView;  
import android.widget.TextView;  
import com.bumptech.glide.Glide;  
  
*/\*\*  
 Подробное действие, которое запускается при нажатии на элемент списка.  
 Он показывает больше информации о виде спорта.  
 \*/*public class DetailActivity extends AppCompatActivity {  
  
 */\*\*  
 Инициализирует действие, заполняя данные из намерения.  
 \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_detail*);  
  
 // Инициализируйте представления.  
 TextView sportsTitle = findViewById(R.id.*titleDetail*);  
 ImageView sportsImage = findViewById(R.id.*sportsImageDetail*);  
  
 // Установите текст из списка намерений дополнительно.  
 sportsTitle.setText(getIntent().getStringExtra("title"));  
  
// Загрузите изображение, используя библиотеку Glide и Intent extra.  
 Glide.*with*(this)  
 .load(getIntent()  
 .getIntExtra("image\_resource",0))  
 .into(sportsImage);  
 }  
}

Программный код, файл Sport.

package com.example.android.materialme;  
  
*/\*\*  
 \* Data model for each row of the RecyclerView  
 \*/*class Sport {  
  
 // Переменные-члены, представляющие название и информацию о виде спорта.  
 private String title;  
 private String info;  
 private final int imageResource;  
  
 */\*\*  
 \*Конструктор для модели спортивных данных.  
 \*/* public Sport(String title, String info, int imageResource) {  
 this.title = title;  
 this.info = info;  
 this.imageResource = imageResource;  
 }  
  
 */\*\*  
 Получает звание чемпиона по этому виду спорта.  
 \*/* String getTitle() {  
 return title;  
 }  
  
 */\*\*  
 Получает информацию о виде спорта.  
 \*/* String getInfo() {  
 return info;  
 }  
  
 public int getImageResource() {  
 return imageResource;  
 }  
}

Программный код, файл SportsAdapter.

package com.example.android.materialme;  
  
import android.content.Context;  
import android.content.Intent;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.bumptech.glide.Glide;  
  
import java.util.ArrayList;  
  
*/\*\*\*  
 \* Класс адаптера для RecyclerView содержит спортивные данные.  
 \*/*class SportsAdapter extends RecyclerView.Adapter<SportsAdapter.ViewHolder> {  
  
  
 private ArrayList<Sport> mSportsData;  
 private Context mContext;  
  
 */\*\*  
 \* Конструктор, который передает спортивные данные и контекст..  
 \*/* SportsAdapter(Context context, ArrayList<Sport> sportsData) {  
 this.mSportsData = sportsData;  
 this.mContext = context;  
 }  
  
  
 */\*\*  
 \* Требуемый метод для создания объектов viewholder.  
 \*/* @Override  
 public SportsAdapter.ViewHolder onCreateViewHolder(  
 ViewGroup parent, int viewType) {  
 return new ViewHolder(LayoutInflater.*from*(mContext)  
 .inflate(R.layout.*list\_item*, parent, false));  
 }  
  
 */\*\*  
 Требуемый метод, который привязывает данные к viewholder.  
 \*/* @Override  
 public void onBindViewHolder(SportsAdapter.ViewHolder holder,  
 int position) {  
 // Получить текущий вид спорта.  
 Sport currentSport = mSportsData.get(position);  
  
 // Заполните текстовые представления данными.  
 holder.bindTo(currentSport);  
 }  
  
  
 */\*\*  
 Необходимый метод для определения размера набора данных.  
 \*/* @Override  
 public int getItemCount() {  
 return mSportsData.size();  
 }  
  
  
 */\*\*  
 Класс ViewHolder, который представляет каждую строку данных в RecyclerView.  
 \*/* class ViewHolder extends RecyclerView.ViewHolder  
 implements View.OnClickListener{  
  
 // Переменные-члены для текстовых представлений  
 private TextView mTitleText;  
 private TextView mInfoText;  
 private ImageView mSportsImage;  
  
 */\*\*  
 Конструктор для ViewHolder, используемый в onCreateViewHolder().  
 \*/* ViewHolder(View itemView) {  
 super(itemView);  
  
 //Инициализируйте представления.  
 mTitleText = itemView.findViewById(R.id.*title*);  
 mInfoText = itemView.findViewById(R.id.*subTitle*);  
 mSportsImage = itemView.findViewById(R.id.*sportsImage*);  
  
 // Установите OnClickListener на весь вид.  
 itemView.setOnClickListener(this);  
 }  
  
 void bindTo(Sport currentSport){  
 // Заполните текстовые представления данными.  
 mTitleText.setText(currentSport.getTitle());  
 mInfoText.setText(currentSport.getInfo());  
  
 //Загрузите изображения в ImageView с помощью библиотеки Glide.  
 Glide.*with*(mContext).load(  
 currentSport.getImageResource()).into(mSportsImage);  
  
 }  
  
 */\*\*  
 \* Обработайте щелчок, чтобы показать подробную активность.  
 \*/* @Override  
 public void onClick(View view) {  
 Sport currentSport = mSportsData.get(getAdapterPosition());  
 Intent detailIntent = new Intent(mContext, DetailActivity.class);  
 detailIntent.putExtra("title", currentSport.getTitle());  
 detailIntent.putExtra("image\_resource",  
 currentSport.getImageResource());  
 mContext.startActivity(detailIntent);  
 }  
 }  
}

На рисунке 1 показано изначальное состояния программы.

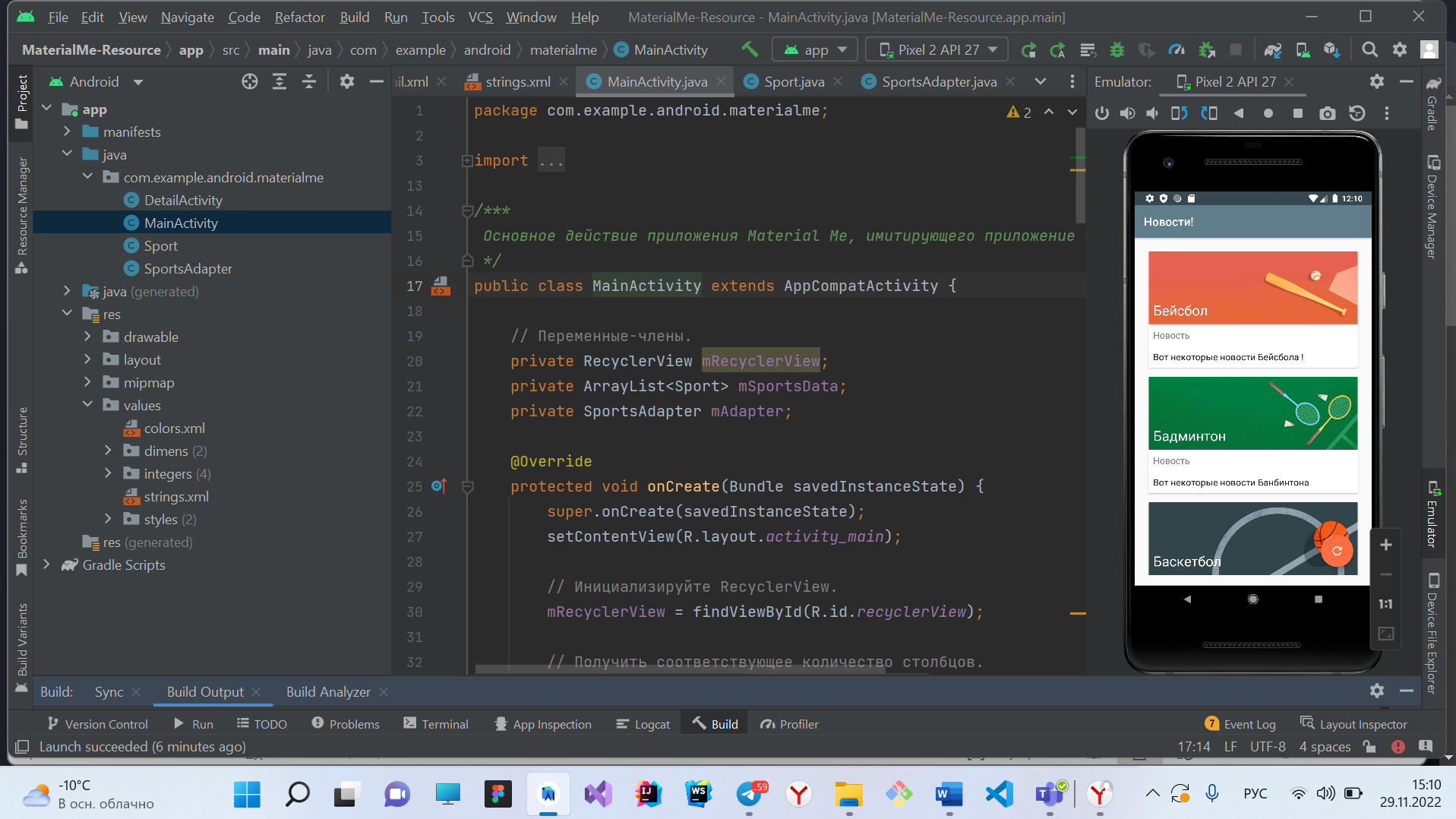


Рисунок 1 – Изначальное состояние программы

На рисунке 2 показано состояние программы при клике на поле новости.

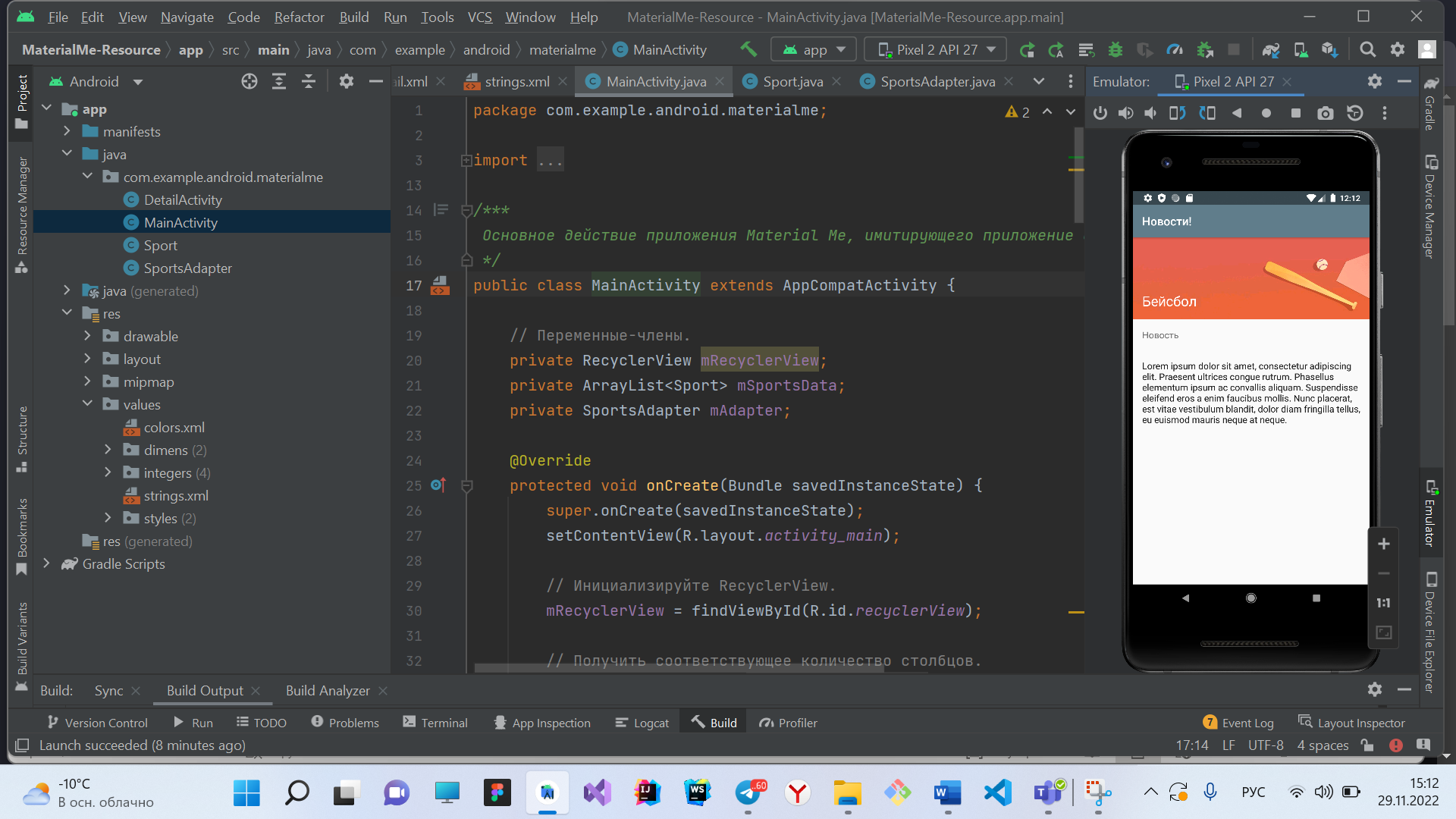
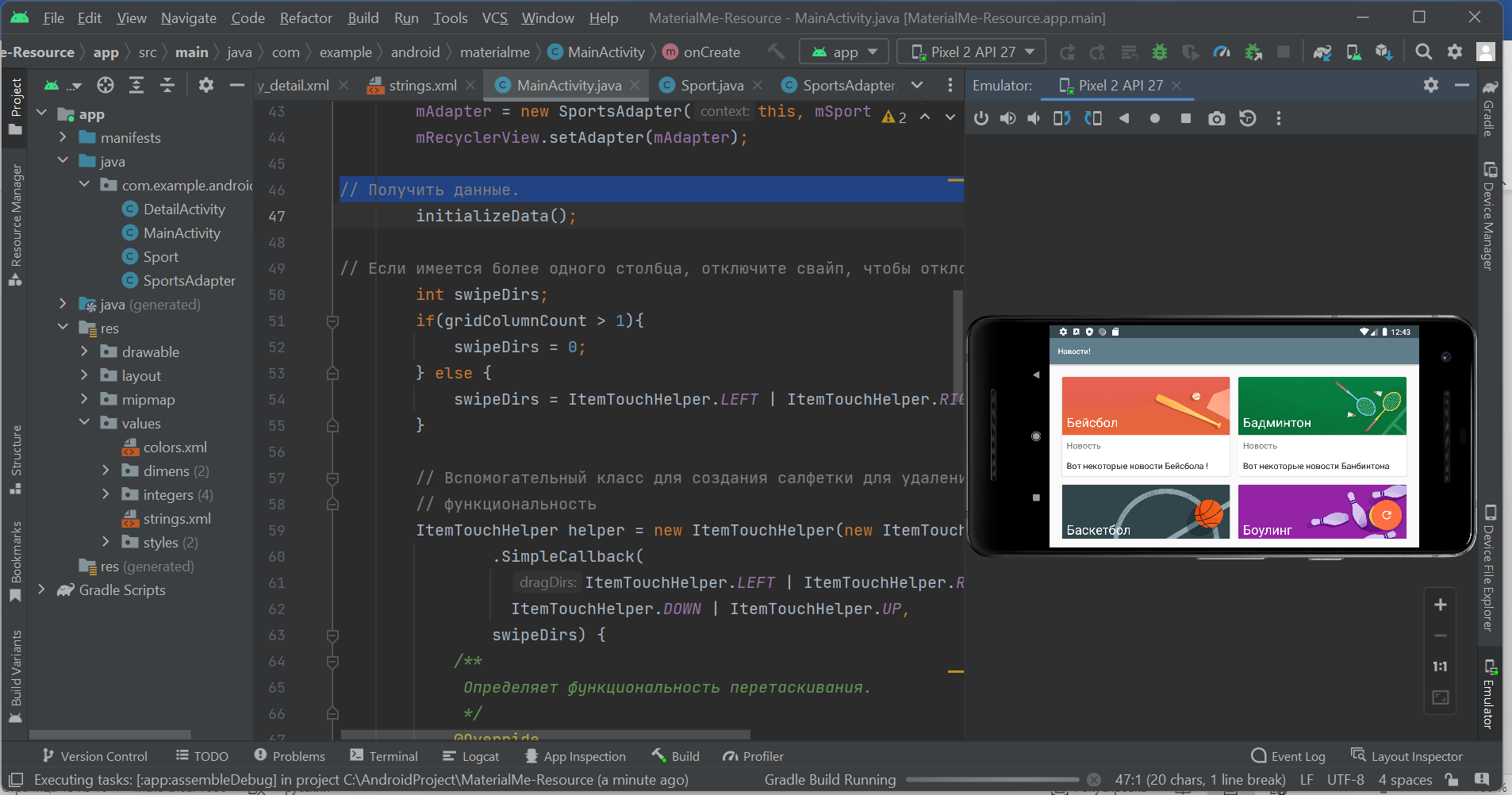


Рисунок 2 – Состояние программы при клике на поле новости

На рисунке 3 показно результат работы, макет стал отзывчивый и гибкий.

Рисунок 3 – Гибкий и Отзывчивый макет

## **Анализ полученных результатов**

В результате выполнения данной лабораторной работы мной был разработан проект, в котором я научился использовать виджеты Material Design, такие как FloatingActionButton и CardView. Как эффективно использовать изображения в приложении. Рекомендуемые рекомендации по разработке интуитивно понятных макетов с использованием жирных цветов. Как создавать альтернативные ресурсы для планшетов. Как создать альтернативные ресурсы для разных локалей.

.