

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждениевысшего образования"МИРЭА - Российский технологический университет"РТУ МИРЭА |

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

**Лабораторная работа №2**

**по дисциплине**

**«Разработка программных приложений»**

**Тема**

**«Реализация приложения»**

Студент группы: ИКБО-16-18 Осадчук Г. М. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(Подпись студента)*

Руководитель работы Зорина Н.В.

*(Подпись преподавателя)*

Москва 2020

**Цель работы**

Цель данной лабораторной работы – получение практических навыков разработки приложения, изучение разработки под Android на языке Java.

**Выполнения лабораторной работы**

Задание:

Реализовать приложение, состоящее из двух активити:

Первое активити — сплеш скрин из первой лабораторной, только показывается не больше чем время загрузки.

Второе активити представляет собой список и ViewPager сформированный из json файла: <https://raw.githubusercontent.com/wesleywerner/ancient-tech/02decf875616dd9692b31658d92e64a20d99f816/src/data/techs.ruleset.json>

Этот JSON файл — массив «технологий» в игре Цивилизация. В каждой технологии вас должны интересовать поля:

●graphic – картинка (относительно

https://raw.githubusercontent.com/wesleywerner/ancient-tech/02decf875616dd9692b31658d92e64a20d99f816/src/images/tech/​ )

●name – название

●helptext - дополнительная информация (может не быть)

Второе активити должно начинаться с фрагмента со списком, в элементе списка маленькая картинка 64dp и название.

Список кликабельный. При клике на технологии открывается ViewPager c фрагментом, в котором соответствующая картинка большого размера (не больше шириныэкрана-20dp) и описанием технологии. При свайпе вправо описание должно меняться на описание предмета выше в списке, соответственно влево - на описание ниже в списке.

Опционально приложение может предоставлять Content Provider с данными технологий.

Загрузка JSON должна происходить в момент показа сплеш скрина. Загрузка картинок, в процессе показа списка (естественно не в UI потоке), имена технологий не должны прыгать в процессе загрузки. Необходимо минимизировать использование трафика.

**Решение**

Весь код можно найти по [ссылке](https://github.com/gosharmot/RPP_lub_2).

**MainActivity**

|  |
| --- |
| import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent; import android.os.AsyncTask; import android.os.Bundle; import android.widget.TextView;  import com.fasterxml.jackson.core.type.TypeReference; import com.fasterxml.jackson.databind.DeserializationConfig; import com.fasterxml.jackson.databind.DeserializationFeature; import com.fasterxml.jackson.databind.ObjectMapper;  import java.util.List;  import okhttp3.HttpUrl; import okhttp3.OkHttpClient; import okhttp3.Request; import okhttp3.Response;  public class MainActivity extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);  HttpHandler handler = new HttpHandler();  handler.execute("https://raw.githubusercontent.com/wesleywerner/ancient-tech/02decf875616dd9692b31658d92e64a20d99f816/src/data/techs.ruleset.json");  }  public class HttpHandler extends AsyncTask<String , Void, Void > {   @Override  protected Void doInBackground(String[] objects) {  String url = objects[0];  final OkHttpClient client = new OkHttpClient();  Request request = new Request.Builder().url(url)  .build();  try (Response response = client.newCall(request).execute()) {  String res = response.body().string();  ObjectMapper mapper = new ObjectMapper();  mapper.disable(DeserializationFeature.*FAIL\_ON\_UNKNOWN\_PROPERTIES*);  List<Item> items = mapper.readValue(res, new TypeReference<List<Item>>(){});  ItemHandler.*createInstance*(items);  Intent intent = new Intent(MainActivity.this, ListActivity.class);  startActivity(intent);  finish();  } catch (Exception e ) {e.printStackTrace();}  return null;  }  } } |

**Item**

|  |
| --- |
| import androidx.annotation.NonNull;  import com.fasterxml.jackson.annotation.JsonIgnoreProperties;  import com.fasterxml.jackson.annotation.JsonInclude;  import com.fasterxml.jackson.annotation.JsonProperty;  import com.fasterxml.jackson.annotation.JsonPropertyOrder;  @JsonIgnoreProperties(ignoreUnknown = true)  @JsonInclude(JsonInclude.Include.USE\_DEFAULTS)  @JsonPropertyOrder(  {  "graphic",  "name",  "helptext"  }  )  public class Item {  @JsonProperty("graphic")  private String graphic;  @JsonProperty("name")  private String name;  @JsonProperty("helptext")  private String helptext = "";  public Item(String graphic, String name, String helptext) {  this.graphic = graphic;  this.name = name;  this.helptext = helptext;  }  public Item() {  this.graphic = null;  this.name = null;  this.helptext = "";  }  @JsonProperty("graphic")  public String getGraphic() {  return graphic;  }  @JsonProperty("graphic")  public void setGraphic( String graphic) {  this.graphic = graphic;  }  @JsonProperty("name")  public String getName() {  return name;  }  @JsonProperty("name")  public void setName(String name) {  this.name = name;  }  @JsonProperty("helptext")  public String getHelptext() {  return helptext;  }  @JsonProperty("helptext")  public void setHelptext(String helptext) {  this.helptext = helptext;  }  } |

**ImageHolder**

|  |
| --- |
| import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.graphics.Color;  import android.os.AsyncTask;  import androidx.fragment.app.Fragment;  import java.util.Arrays;  import java.util.HashMap;  import java.util.List;  import java.util.Map;  public class ImageHolder {  private static ImageHolder instance;  public Map<String, Bitmap> images = new HashMap<>();  ListActivityFragment adapter;  public static ImageHolder createInstance(List<Item> items, ListActivityFragment adapter) {  if(instance == null) {  instance = new ImageHolder(items, adapter);  }  return instance;  }  public static ImageHolder getInstance() {  return instance;  }  private ImageHolder(List<Item> items,ListActivityFragment adapter) {  this.adapter = adapter;  int[] colors = new int[300\*300];  Arrays.fill(colors, 0, 300\*300, Color.GRAY);  for(Item item : items) {  images.put(item.getGraphic(), Bitmap.createBitmap(colors, 300, 300, Bitmap.Config.ARGB\_8888));  }  }  public Bitmap getImage(String graphics) {  return images.get(graphics);  }  } |

**ViewPagerAdapter**

|  |
| --- |
| import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.ImageView;  import android.widget.TextView;  import androidx.annotation.NonNull;  import androidx.recyclerview.widget.RecyclerView;  import java.util.List;  public class ViewPagerAdapter extends RecyclerView.Adapter<ViewPagerAdapter.Holder> {  List<Item> items = ItemHandler.getInstance().getItems();  ImageHolder imageHolder = ImageHolder.getInstance();  @NonNull  @Override  public Holder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.view\_pager\_fragment, parent, false);  return new Holder(view);  }  @Override  public void onBindViewHolder(@NonNull Holder holder, int position) {  holder.name.setText(items.get(position).getName());  holder.helptext.setText(items.get(position).getHelptext());  holder.img.setImageBitmap(imageHolder.getImage(items.get(position).getGraphic()));  }  @Override  public int getItemCount() {  return items.size();  }  public class Holder extends RecyclerView.ViewHolder {  ImageView img;  TextView name;  TextView helptext;  public Holder(@NonNull View itemView) {  super(itemView);  img = itemView.findViewById(R.id.img\_pager);  name = itemView.findViewById(R.id.name\_pager);  helptext = itemView.findViewById(R.id.help\_pager);  }  }  } |

**ViewPagerHolderFragment**

|  |
| --- |
| import android.os.Bundle;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.Adapter;  import androidx.annotation.NonNull;  import androidx.annotation.Nullable;  import androidx.fragment.app.Fragment;  import androidx.viewpager.widget.PagerAdapter;  import androidx.viewpager2.widget.ViewPager2;  public class ViewPagerHolderFragment extends Fragment {  int position;  public ViewPagerHolderFragment(int position) {  super();  this.position = position;  }  @Nullable  @Override  public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  View v = inflater.inflate(R.layout.view\_pager\_holder, container, false);  ViewPager2 viewPager2 = v.findViewById(R.id.view\_pager);  viewPager2.setOrientation(ViewPager2.ORIENTATION\_HORIZONTAL);  ViewPagerAdapter adapter = new ViewPagerAdapter();  viewPager2.setAdapter(adapter);  viewPager2.setCurrentItem(position);  return v;  }  } |

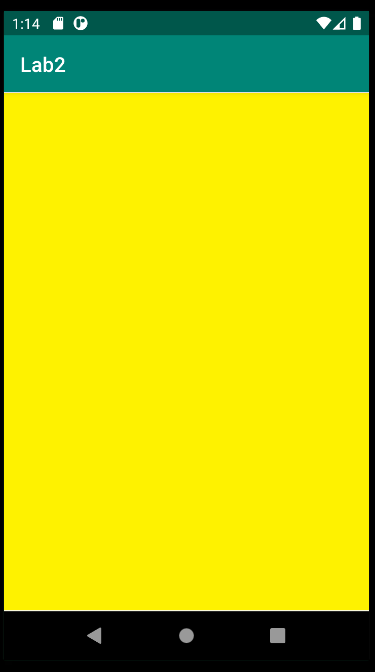
**ListActivity**

|  |
| --- |
| import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.os.AsyncTask;  import android.os.Bundle;  import com.google.android.material.floatingactionbutton.FloatingActionButton;  import com.google.android.material.snackbar.Snackbar;  import androidx.appcompat.app.AppCompatActivity;  import androidx.appcompat.widget.Toolbar;  import androidx.fragment.app.Fragment;  import androidx.fragment.app.FragmentActivity;  import androidx.fragment.app.FragmentManager;  import androidx.fragment.app.FragmentTransaction;  import android.view.View;  import android.widget.TextView;  import java.util.Arrays;  import java.util.List;  import okhttp3.OkHttpClient;  import okhttp3.Request;  import okhttp3.Response;  public class ListActivity extends FragmentActivity {  ImageHolder holder;  ListActivityFragment recyclerFragment;  ViewPagerHolderFragment pagerHolderFragment;  final static String TAG\_1 = "RECYCLER\_FRAGMENT";  final static String TAG\_2 = "VIEW\_PAGER\_FRAGMENT";  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_list);  Click click = new Click() {  @Override  public void click(int position) {  pagerHolderFragment = new ViewPagerHolderFragment(position);  FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();  transaction.replace(R.id.frame, pagerHolderFragment, TAG\_2);  transaction.commit();  }  };  recyclerFragment = new ListActivityFragment(click);  FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();  transaction.add(R.id.frame, recyclerFragment, TAG\_1);  transaction.commit();  List<Item> items = ItemHandler.getInstance().getItems();  holder = ImageHolder.createInstance(items, recyclerFragment);  String[] keys = holder.images.keySet().toArray(new String[0]);  Arrays.sort(keys);  for (String key : keys) {  HttpHandler httpHandler = new HttpHandler();  httpHandler.execute(key);  }  }  public class HttpHandler extends AsyncTask<String, Void, Bitmap> {  @Override  protected Bitmap doInBackground(String... strings) {  String key = strings[0];  final OkHttpClient client = new OkHttpClient();  Request request = new Request.Builder().url("https://raw.githubusercontent.com/wesleywerner/ancient-tech/02decf875616dd9692b31658d92e64a20d99f816/src/images/tech/" + key)  .build();  try (Response response = client.newCall(request).execute()) {  Bitmap bitmap = BitmapFactory.decodeStream(response.body().byteStream());  holder.images.put(key, bitmap);  System.out.println(key);  return bitmap;  } catch (Exception e) {  e.printStackTrace();  }  return null;  }  @Override  protected void onPostExecute(Bitmap bitmap) {  super.onPostExecute(bitmap);  recyclerFragment.adapter.notifyDataSetChanged();  }  }  @Override  public void onBackPressed() {  if(getSupportFragmentManager().findFragmentByTag(TAG\_2) != null) {  getSupportFragmentManager().beginTransaction()  .replace(R.id.frame, recyclerFragment, TAG\_1)  .commit();  } else {  finish();  }  }  } |

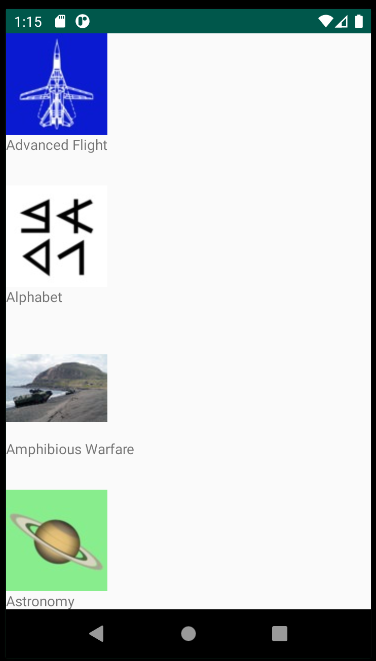
**ListAdapter**

|  |
| --- |
| import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.ImageView;  import android.widget.LinearLayout;  import android.widget.TextView;  import androidx.annotation.NonNull;  import androidx.recyclerview.widget.RecyclerView;  import java.util.List;  public class ListAdapter extends RecyclerView.Adapter<ListAdapter.ListViewHolder> {  List<Item> items;  ImageHolder imageHolder;  Click click;  public ListAdapter(Click click) {  super();  this.click = click;  this.items = ItemHandler.getInstance().getItems();  }  @NonNull  @Override  public ListViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.list\_item, parent, false);  return new ListViewHolder(view);  }  @Override  public void onBindViewHolder(@NonNull ListViewHolder holder, final int position) {  holder.helptext.setText(items.get(position).getHelptext());  holder.name.setText(items.get(position).getName());  holder.graphics.setImageBitmap(imageHolder.getImage(items.get(position).getGraphic()));  holder.layout.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  click.click(position);  }  });  }  @Override  public int getItemCount() {  return items.size();  }  public class ListViewHolder extends RecyclerView.ViewHolder {  ImageView graphics;  TextView name;  TextView helptext;  LinearLayout layout;  public ListViewHolder(@NonNull View itemView) {  super(itemView);  graphics = itemView.findViewById(R.id.graphics);  name = itemView.findViewById(R.id.name);  helptext = itemView.findViewById(R.id.helptext);  layout = itemView.findViewById(R.id.item\_layout);  }  }  public void setImageHolder(ImageHolder imageHolder) {  this.imageHolder = imageHolder;  }  } |

**Результат работы программы:**



**Рис 1. Активити 1**



**Рис 2. Активити 2**

**Выводы по работе:**

В течение данной лабораторной работы был изучен процесс разработки под Android.