

Lab 12. Ứng dụng nghe nhạc (4 tiết)

I. Yêu cầu

- Sinh viên sử dụng công cụ Android Studio để làm bài. Kết quả bài làm cần được chụp lại và giữ lại toàn bộ dự án để sử dụng khi cần thiết.
- Mỗi người làm trên một dự án khác nhau. Mỗi sinh viên chỉ làm trên một dự án
- Khi có yêu cầu, sinh viên nộp qua email giáo viên hoặc một kênh khác.

II. Luyện tập

Tạo giao diện

Phạm vi kiến thức vận dụng

- Các kiến thức căn bản về Design UI
 - Drawable, Color Schemes, String constants, Style & themes
 - Layout constraint rules (padding, margin...)
- Các đối tượng View System
 - TextView, ImageView
 - Các ViewGroup gồm: LinearLayout, FrameLayout, TableRow
- Các đối tượng xử lý logic
 - Activity, MediaPlayer, RecyclerView, Adapter; ContentResolver giúp truy cập tài nguyên của điện thoại để lấy danh sách nhạc

Nội dung bài thực hành

1. Yêu cầu: Tạo 1 ứng dụng My Music và xây dựng giao diện layout sau:



Mô tả: Đây là giao diện hiển thị danh sách các bài nhạc có trong điện thoại của người sử dụng. Khi click vào bài hát, nó sẽ thực hiện phát nhạc.

Yêu cầu:

- Thiết kế đúng như giao diện hiển thị bên trái
- Sử dụng RecyclerView để hiển thị danh sách bài hát
- Sử dụng bộ icon trong nguồn Vector Assets của AndroidStudio có sẵn.
- Khai báo quyền **READ_EXTERNAL_STORAGE** để có thể truy cập bộ nhớ của điện thoại
- ContentResolver giúp truy cập tài nguyên của điện thoại để lấy danh sách nhạc

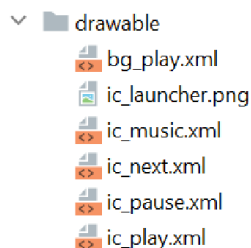
2. Gợi ý: Các kiến thức sử dụng

- Tạo 1 layout **xml** trong thư mục res/layout
- Sử dụng các ViewSystem:
 - o TextView, ImageView
- Sử dụng các ViewGroup:
 - o LinearLayout
 - o Thẻ include
- Sử dụng các đối tượng giao diện
 - o RecyclerView, Adapter
- Sử dụng các đối tượng logic
 - o Activity
- Lấy ảnh icon trong kho ảnh vector assets của AndroidStudio

3. Hướng dẫn

A> Xây dựng giao diện

Bước 1: Chuẩn bị tài nguyên **drawable**, **AndroidManifest** như sau:



Nội dung file **res/values/strings.xml**

```
<resources>
  <string name="app_name">My Music</string>
  <string name="txt_alert">Hãy cấp quyền truy cập để lấy ds bài hát</string>
</resources>
```

Nội dung file **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  package="com.techja.mymusic">
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
  <application
    android:requestLegacyExternalStorage="true"
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/Theme.MyMusic"
    tools:ignore="AllowBackup">
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Bước 2: Tạo file **res/layout/item_song.xml** có nội dung sau:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:orientation="vertical">
  <TableRow
    android:id="@+id/tr_song"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:padding="10dp">
    <ImageView
      android:layout_width="50dp"
      android:layout_height="50dp"
```

```

        android:padding="10dp"
        android:src="@drawable/ic_music"
        app:tint="@color/design_default_color_primary_dark"
        tools:ignore="ContentDescription" />
    <TextView
        android:id="@+id/tv_song"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fontFamily="sans-serif-light"
        android:gravity="center_vertical"
        android:text="Câu hẹn câu thề"
        android:textColor="@color/design_default_color_primary_dark"
        android:textSize="18sp" />
</TableRow>
<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"
    android:background="#E5E5E5" />
</LinearLayout>

```

Bước 3: Tạo file **res/layout/item_controller.xml** có nội dung sau:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/design_default_color_primary_dark"
    android:orientation="vertical"
    android:padding="10dp">
    <TextView
        android:id="@+id/tv_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif"
        android:text="Câu hẹn câu thề"
        android:textColor="@color/white"
        android:textSize="18sp" />
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical">
        <TextView
            android:id="@+id/tv_album"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="sans-serif-light"
            android:text="Vol.4 - Đình Dũng"
            android:textColor="@color/white"

```

```
        android:textSize="12sp" />
    <TextView
        android:id="@+id/tv_time"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-light"
        android:gravity="end"
        android:text="00:00/04:35"
        android:textColor="@color/white"
        android:textSize="14sp" />
</TableRow>
<SeekBar
    android:id="@+id/seekbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:foregroundTint="@color/white"
    android:max="100"
    android:padding="0dp"
    android:progress="50"
    android:progressBackgroundTint="@color/white"
    android:progressTint="@color/white"
    android:thumbTint="@color/white" />
<TableRow
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginTop="10dp">
    <ImageView
        android:id="@+id/iv_back"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:padding="10dp"
        android:rotation="180"
        android:src="@drawable/ic_next"
        app:tint="@color/white"
        tools:ignore="ContentDescription" />
    <ImageView
        android:id="@+id/iv_play"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:src="@drawable/bg_play"
        app:tint="@color/yellow"
        tools:ignore="ContentDescription" />
    <ImageView
        android:id="@+id/iv_next"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
```

```

        android:padding="10dp"
        android:src="@drawable/ic_next"
        app:tint="@color/white"
        tools:ignore="ContentDescription" />
    </TableRow>
</LinearLayout>

```

Bước 4: Tạo file `res/layout/activity_main.xml` có nội dung sau:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv_song"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        tools:listitem="@layout/item_song" />
    <include layout="@layout/item_controller" />
</LinearLayout>

```

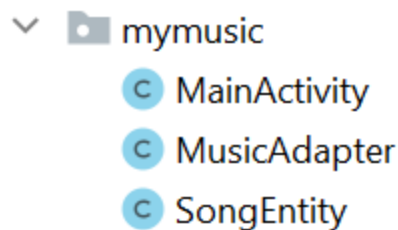
Xây dựng controller nghe nhạc bằng MediaPlayer

Phạm vi kiến thức vận dụng

- Tiếp lab 6.1

Nội dung bài thực hành

B> Xử lý logic



Bước 1: Xây dựng đối tượng `SongEntity` để định nghĩa các thông tin của bài hát

```
package com.techja.mymusic;
```

```
public class SongEntity {  
    private String name, path, album;  
  
    public SongEntity(String name, String path, String album) {  
        this.name = name;  
        this.path = path;  
        this.album = album;  
    }  
    public String getName() {  
        return name;  
    }  
    public String getPath() {  
        return path;  
    }  
    public String getAlbum() {  
        return album;  
    }  
}
```

Bước 2: Xây dựng đối tượng MusicAdapter để quản lý và hiển thị danh sách bài hát lên màn hình

```
package com.techja.mymusic;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.animation.AnimationUtils;  
import android.widget.TextView;  
  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
  
public class MusicAdapter extends RecyclerView.Adapter<MusicAdapter.MusicHolder> {  
    private final ArrayList<SongEntity> listSong;  
    private final Context mContext;  
    public MusicAdapter(ArrayList<SongEntity> listSong, Context mContext) {  
        this.listSong = listSong;  
        this.mContext = mContext;  
    }  
    @Override  
    public MusicHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
        View view = LayoutInflater.from(mContext).inflate(R.layout.item_song, parent, false);  
        return new MusicHolder(view);  
    }  
    @Override  
    public void onBindViewHolder(MusicAdapter.MusicHolder holder, int position) {  
        SongEntity item = listSong.get(position);
```

```

        holder.tvName.setText(item.getName());
        holder.tvName.setTag(item);
    }
    @Override
    public int getItemCount() {
        return listSong.size();
    }
    public class MusicHolder extends RecyclerView.ViewHolder {
        TextView tvName;
        public MusicHolder(View itemView) {
            super(itemView);
            tvName = itemView.findViewById(R.id.tv_song);
            itemView.setOnClickListener(v -> {
                v.startAnimation(AnimationUtils.loadAnimation(mContext,
androidx.appcompat.R.anim.abc_fade_in));
                ((MainActivity) mContext).playSong((SongEntity) tvName.getTag());
            });
        }
    }
}

```

Bước 3: Xây dựng đối tượng MainActivity để hiển thị giao diện activity_main và xử lý các vấn đề sau:

- Load danh sách bài hát từ bộ nhớ của điện thoại
- Xử lý phát nhạc
- Xử lý tương tác của người dùng trên giao diện

```

package com.techja.mymusic;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class MainActivity extends AppCompatActivity implements View.OnClickListener,

```



```

SeekBar.OnSeekBarChangeListener {
    private static final int LEVEL_PAUSE = 0;
    private static final int LEVEL_PLAY = 1;
    private static final MediaPlayer player = new MediaPlayer();
    private static final int STATE_IDE = 1;
    private static final int STATE_PLAYING = 2;
    private static final int STATE_PAUSED = 3;
    private final ArrayList<SongEntity> listSong = new ArrayList<>();
    private TextView tvName, tvAlbum, tvTime;
    private SeekBar seekBar;
    private ImageView ivPlay;
    private int index;
    private SongEntity songEntity;
    private Thread thread;
    private int state = STATE_IDE;
    private String totalTime;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initView();
    }
    private void initView() {
        ivPlay = findViewById(R.id.iv_play);
        ivPlay.setOnClickListener(this);
        findViewById(R.id.iv_back).setOnClickListener(this);
        findViewById(R.id.iv_next).setOnClickListener(this);
        tvName = findViewById(R.id.tv_name);
        tvAlbum = findViewById(R.id.tv_album);
        tvTime = findViewById(R.id.tv_time);
        seekBar = findViewById(R.id.seekbar);
        seekBar.setOnSeekBarChangeListener(this);
        if (checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, 101);
            return;
        }
        loadingListSongOffline();
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            loadingListSongOffline();
        } else {
            Toast.makeText(this, R.string.txt_alert, Toast.LENGTH_SHORT).show();
            finish();
        }
    }
    private void loadingListSongOffline() {

```

```

Cursor c = getResolver().query(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
    null, null, null, null);
if (c != null) {
    c.moveToFirst();
    listSong.clear();
    while (!c.isAfterLast()) {
        String name = c.getString(c.getColumnIndex(MediaStore.Audio.Media.TITLE));
        String path = c.getString(c.getColumnIndex(MediaStore.Audio.Media.DATA));
        String album = "N/A";
        if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.R) {
            album = c.getString(c.getColumnIndex(MediaStore.Audio.Media.ALBUM_ARTIST));
        }
        listSong.add(new SongEntity(name, path, album));
        c.moveToNext();
    }
    c.close();
}
RecyclerView rv = findViewById(R.id.rv_song);
rv.setLayoutManager(new LinearLayoutManager(this));
rv.setAdapter(new MusicAdapter(listSong, this));
play();
playPause();
}

public void playSong(SongEntity songEntity) {
    index = listSong.indexOf(songEntity);
    this.songEntity = songEntity;
    play();
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.iv_play) {
        playPause();
    } else if (v.getId() == R.id.iv_next) {
        next();
    } else if (v.getId() == R.id.iv_back) {
        back();
    }
}

private void back() {
    if (index == 0) {
        index = listSong.size() - 1;
    } else {
        index--;
    }
    play();
}

```

```

private void next() {
    if (index >= listSong.size()) {
        index = 0;
    } else {
        index++;
    }
    play();
}

private void playPause() {
    if (state == STATE_PLAYING && player.isPlaying()) {
        player.pause();
        ivPlay.setImageLevel(LEVEL_PAUSE);
        state = STATE_PAUSED;
    } else if (state == STATE_PAUSED) {
        player.start();
        state = STATE_PLAYING;
        ivPlay.setImageLevel(LEVEL_PLAY);
    } else {
        play();
    }
}

private void play() {
    if(listSong.size() != 0){
        songEntity = listSong.get(index);
        tvName.setText(songEntity.getName());
        tvAlbum.setText(songEntity.getAlbum());
        player.reset();
        try {
            player.setDataSource(songEntity.getPath());
            player.prepare();
            player.start();
            ivPlay.setImageLevel(LEVEL_PLAY);
            state = STATE_PLAYING;
            totalTime = getTime(player.getDuration());
            seekBar.setMax(player.getDuration());
            if (thread == null) {
                startLooping();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

private void startLooping() {
    thread = new Thread() {
        @Override

```

```

    public void run() {
        while (true) {
            try {
                Thread.sleep(200);
            } catch (Exception e) {
                return;
            }
            runOnUiThread() -> updateTime();
        }
    }
};
thread.start();
}

private void updateTime() {
    if (state == STATE_PLAYING || state == STATE_PAUSED) {
        int time = player.getCurrentPosition();
        tvTime.setText(String.format("%s/%s", getTime(time), totalTime));
        seekBar.setProgress(time);
    }
}

@SuppressWarnings("SimpleDateFormat")
private String getTime(int time) {
    return new SimpleDateFormat("mm:ss").format(new Date(time));
}

@Override
protected void onDestroy() {
    super.onDestroy();
    thread.interrupt();
}

@Override
public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
}

@Override
public void onStartTrackingTouch(SeekBar seekBar) {
}

@Override
public void onStopTrackingTouch(SeekBar seekBar) {
    if (state == STATE_PLAYING || state == STATE_PAUSED) {
        player.seekTo(seekBar.getProgress());
    }
}
}

```

4. Lưu ý:

- Sử dụng thẻ: **android:requestLegacyExternalStorage="true"** để cho phép truy cập bộ nhớ của điện thoại đối với các thiết bị có version Android SDK ≥ 10

III. Bài tập

Sinh viên đọc, hiểu code và comment lại các đoạn code tương ứng.

--Hết--