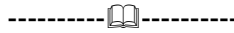


Lab 8: CAMERA VÀ MEDIA



MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

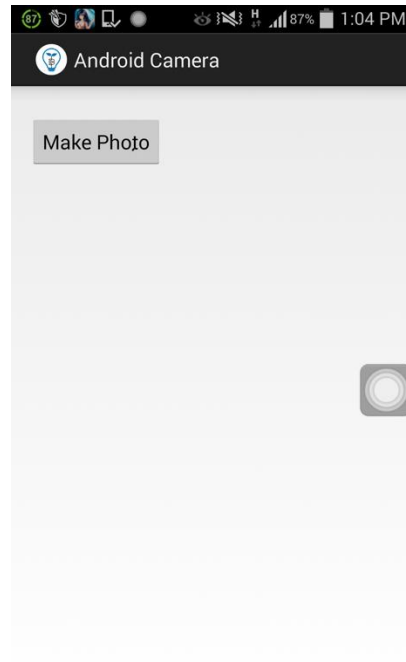
- ✓ Thao tác với Camera để chụp ảnh
- ✓ Sử dụng MediaPlayer để nghe nhạc

PHẦN I

BÀI 1 (4ĐIỂM)

Tạo ứng dụng thao tác với Camera

Sinh viên thực hiện xây dựng ứng dụng **AndroidCamera** hoạt động như bên dưới:



- Sinh viên thực hiện thiết kế giao diện cho ứng dụng:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin">

    <Button
        android:id="@+id/btnPhoto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Make Photo" />

    <ImageView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:visibility="visible"
        android:id="@+id/imgPhoto" />

</LinearLayout>
```

- Sinh viên thực hiện thay đổi **MainActivity**:

```
public class MainActivity extends ActionBarActivity {

    // Declare variables
    static final int REQUEST_IMAGE_CAPTURE = 1;
    static final int REQUEST_TAKE_PHOTO = 1;
    private static final String JPEG_FILE_PREFIX = "IMG_";
    private static final String JPEG_FILE_SUFFIX = ".jpg";

    private AlbumStorageDirFactory mAlbumStorageDirFactory = null;

    Button btnPhoto;
    ImageView imgPhoto;
    // Store Photo Path
    String mCurrentPhotoPath;

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

        // Get layout control
        btnPhoto = (Button) findViewById(R.id.btnPhoto);
        imgPhoto = (ImageView) findViewById(R.id.imgPhoto);

        btnPhoto.setOnClickListener(new Button.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

                File f = null;

                try {
                    // Create image file
                    f = setUpPhotoFile();
                    mCurrentPhotoPath = f.getAbsolutePath();
                    takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(f));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```
        } catch (IOException e) {
            e.printStackTrace();
            f = null;
            mCurrentPhotoPath = null;
        }

        startActivityForResult(takePictureIntent, 1);
    }
});

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.FROYO) {
    mAlbumStorageDirFactory = new FroyoAlbumDirFactory();
} else {
    mAlbumStorageDirFactory = new BaseAlbumDirFactory();
}
}

/* Photo album for this application */
private String getAlbumName() {
    return "CameraSample";
}
```

```
private File getAlbumDir() {
    File storageDir = null;

    if (Environment.MEDIA_MOUNTED.equals(Environment.getExternalStorageState())) {

        storageDir = mAlbumStorageDirFactory.getAlbumStorageDir(getAlbumName());

        if (storageDir != null) {
            if (! storageDir.mkdirs()) {
                if (! storageDir.exists()){
                    Log.d("CameraSample", "failed to create directory");
                    return null;
                }
            }
        }

    } else {
        Log.v(getString(R.string.app_name), "External storage is not mounted READ/WRITE.");
    }

    return storageDir;
}

private File createImageFile() throws IOException {
    // Create an image file name
    String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
    String imageFileName = JPEG_FILE_PREFIX + timeStamp + "_";
    File albumF = getAlbumDir();
    File imageF = File.createTempFile(imageFileName, JPEG_FILE_SUFFIX, albumF);
    return imageF;
}

private File setUpPhotoFile() throws IOException {
    File f = createImageFile();
    mCurrentPhotoPath = f.getAbsolutePath();
    return f;
}
```

```
private void setPic() {

    /* There isn't enough memory to open up more than a couple camera photos */
    /* So pre-scale the target bitmap into which the file is decoded */

    /* Get the size of the ImageView */
    int targetW = imgPhoto.getWidth();
    int targetH = imgPhoto.getHeight();

    /* Get the size of the image */
    BitmapFactory.Options bmOptions = new BitmapFactory.Options();
    bmOptions.inJustDecodeBounds = true;
    BitmapFactory.decodeFile(mCurrentPhotoPath, bmOptions);
    int photoW = bmOptions.outWidth;
    int photoH = bmOptions.outHeight;

    /* Figure out which way needs to be reduced less */
    int scaleFactor = 1;
    if ((targetW > 0) || (targetH > 0)) {
        scaleFactor = Math.min(photoW/targetW, photoH/targetH);
    }

    /* Set bitmap options to scale the image decode target */
    bmOptions.inJustDecodeBounds = false;
    bmOptions.inSampleSize = scaleFactor;
    bmOptions.inPurgeable = true;

    /* Decode the JPEG file into a Bitmap */
    Bitmap bitmap = BitmapFactory.decodeFile(mCurrentPhotoPath, bmOptions);

    /* Associate the Bitmap to the ImageView */
    imgPhoto.setImageBitmap(bitmap);
    imgPhoto.setVisibility(View.VISIBLE);
}
```

FPT POLYTECHNIC

```
private void galleryAddPic() {
    Intent mediaScanIntent = new Intent("android.intent.action.MEDIA_SCANNER_SCAN_FILE");
    File f = new File(mCurrentPhotoPath);
    Uri contentUri = Uri.fromFile(f);
    mediaScanIntent.setData(contentUri);
    this.sendBroadcast(mediaScanIntent);
}

private void handleBigCameraPhoto() {
    if (mCurrentPhotoPath != null) {
        setPic();
        galleryAddPic();
        mCurrentPhotoPath = null;
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == RESULT_OK) {
        handleBigCameraPhoto();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}
```

- Sinh viên đăng ký **Permission** trong **AndroidManifest**:

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
```

- Sinh viên thực hiện cài đặt ứng dụng trên thiết bị và sử dụng

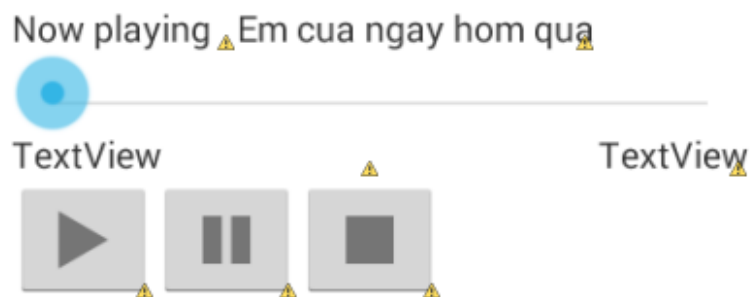
* Sử dụng 3 lớp có sẵn trong thư mục TaiNguyen

PHẦN II

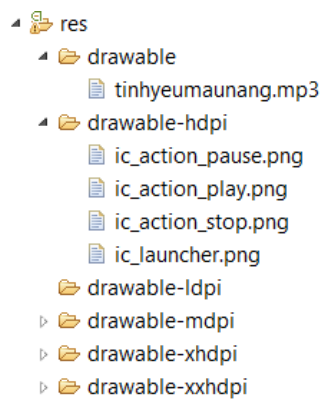
BÀI 2 (4ĐIỂM)

Thực hiện nghe nhạc với MediaPlayer

- Sinh viên thực hiện thiết kế giao diện như bên dưới



- Lưu ý copy một file nhạc mp3 và các icon vào thư mục res:



- Sinh viên thay đổi nội dung **Class MainActivity** :


```
// Declare variables
public TextView startTimeField, endTimeField;
private MediaPlayer mediaPlayer;
private double startTime = 0;
private double finalTime = 0;
private Handler myHandler = new Handler();
private int forwardTime = 5000;
private int backwardTime = 5000;
private SeekBar seekbar;
private ImageButton playButton, pauseButton;
public static int oneTimeOnly = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    startTimeField = (TextView) findViewById(R.id.lblStart);
    endTimeField = (TextView) findViewById(R.id.lblEnd);
    seekbar = (SeekBar) findViewById(R.id.seekBar1);
    playButton = (ImageButton) findViewById(R.id.imageButton1);
    pauseButton = (ImageButton) findViewById(R.id.imageButton2);
    seekbar.setClickable(false);
    pauseButton.setEnabled(false);
    mediaPlayer = MediaPlayer.create(MainActivity.this,
        R.drawable.tinhyeumaunang);
}
```

FPT POLYTECHNIC

```
public void play(View view) {
    Toast.makeText(getApplicationContext(), "Playing sound",
        Toast.LENGTH_SHORT).show();
    mediaPlayer.start();
    finalTime = mediaPlayer.getDuration();
    startTime = mediaPlayer.getCurrentPosition();
    if (oneTimeOnly == 0) {
        seekbar.setMax((int) finalTime);
        oneTimeOnly = 1;
    }

    endTimeField.setText(String.format(
        "%d min, %d sec",
        TimeUnit.MILLISECONDS.toMinutes((long) finalTime),
        TimeUnit.MILLISECONDS.toSeconds((long) finalTime)
            - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS
                .toMinutes((long) finalTime))));
    startTimeField.setText(String.format(
        "%d min, %d sec",
        TimeUnit.MILLISECONDS.toMinutes((long) startTime),
        TimeUnit.MILLISECONDS.toSeconds((long) startTime)
            - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS
                .toMinutes((long) startTime))));
    seekbar.setProgress((int) startTime);
    myHandler.postDelayed(UpdateSongTime, 100);
    pauseButton.setEnabled(true);
    playButton.setEnabled(false);
}
```

FPT POLYTECHNIC

```

private Runnable UpdateSongTime = new Runnable() {
    public void run() {
        startTime = mediaPlayer.getCurrentPosition();
        startTimeField.setText(String.format(
            "%d min, %d sec",
            TimeUnit.MILLISECONDS.toMinutes((long) startTime),
            TimeUnit.MILLISECONDS.toSeconds((long) startTime)
                - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS
                    .toMinutes((long) startTime))));
        seekbar.setProgress((int) startTime);
        myHandler.postDelayed(this, 100);
    }
};

public void pause(View view) {
    Toast.makeText(getApplicationContext(), "Pausing sound",
        Toast.LENGTH_SHORT).show();
    mediaPlayer.pause();
    pauseButton.setEnabled(false);
    playButton.setEnabled(true);
}

public void forward(View view) {
    int temp = (int) startTime;
    if ((temp + forwardTime) <= finalTime) {
        startTime = startTime + forwardTime;
        mediaPlayer.seekTo((int) startTime);
    } else {
        Toast.makeText(getApplicationContext(),
            "Cannot jump forward 5 seconds", Toast.LENGTH_SHORT).show();
    }
}

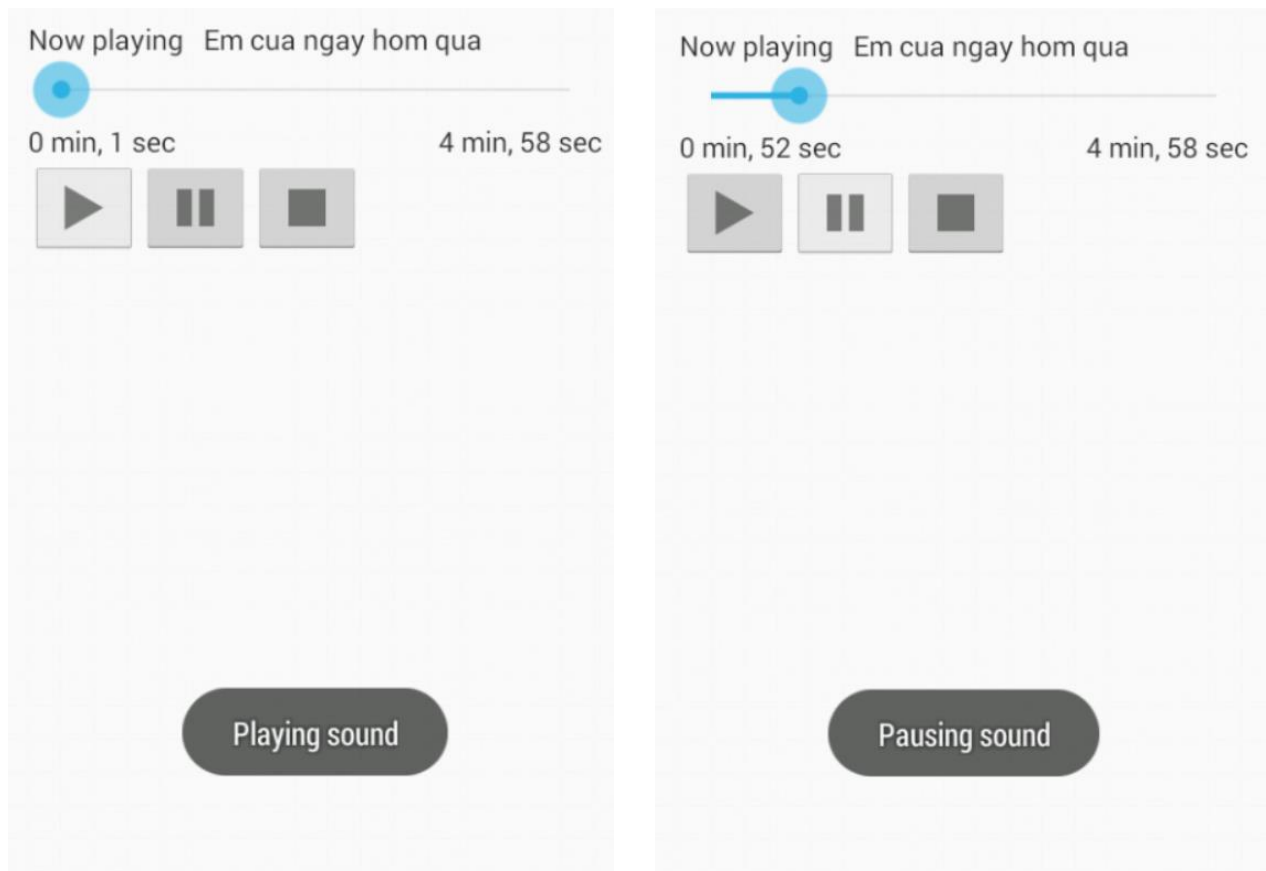
@Override
public boolean onCreateOptionsMenu(Menu menu) {

    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

- Chạy ứng dụng và xem kết quả trên màn hình



BÀI 3 (2ĐIỂM)

Xây dựng ứng dụng nghe nhạc

Sinh viên sử dụng bài 2 và xây dựng thêm một Activity hiển thị danh sách các bài nhạc và người dùng có thể chọn để nghe