Nguyễn Minh Khang\_Lab6\_1,2

**Tạo thư mục anim để chứa các các animation để thực hiện bài lab6**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

**Trong thư mục Values tạo thêm file xml dimens để quản lý và thay đổi các giá trị kích thước của ứng dụng**

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Trong string.xml chứa các chuỗi văn bản mà sẽ sử dụng trong ứng dụng.

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

**Trong MainActivity khai báo các biến và viết logic animation cho từng nút tương ứng để thực hiện các chức năng của animation.**

package com.example.nguyenminhkhang\_lab6;

import android.content.Intent;

import android.os.Bundle;

import android.view.GestureDetector;

import android.view.MotionEvent;

import android.view.View;

import android.view.animation.AlphaAnimation;

import android.view.animation.Animation;

import android.view.animation.AnimationSet;

import android.view.animation.AnimationUtils;

import android.view.animation.RotateAnimation;

import android.view.animation.ScaleAnimation;

import android.view.animation.TranslateAnimation;

import android.widget.Button;

import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity implements GestureDetector.OnGestureListener {

// Khai báo các biến

private Button btnFadeInXml, btnFadeInCode, btnFadeOutXml, btnFadeOutCode,

btnBlinkXml, btnBlinkCode, btnZoomInXml, btnZoomInCode,

btnZoomOutXml, btnZoomOutCode, btnRotateXml, btnRotateCode,

btnMoveXml, btnMoveCode, btnSlideUpXml, btnSlideUpCode,

btnBounceXml, btnBounceCode, btnCombineXml, btnCombineCode;

private ImageView ivUitLogo;

private GestureDetector gestureDetector; // Khai báo GestureDetector

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// Khởi tạo GestureDetector để phát hiện các cử chỉ

gestureDetector = new GestureDetector(this, this);

// Khởi tạo các views và xử lý sự kiện

findViewsByIds();

// Thêm sự kiện click cho ImageView để mở NewActivity

ivUitLogo.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent iNewActivity = new Intent(MainActivity.this, NewActivity.class);

startActivity(iNewActivity);

overridePendingTransition(R.anim.slide\_in\_right, R.anim.slide\_out\_left);

}

});

// Định nghĩa hành động khi người dùng nhấn vào từng nút animation

setButtonListeners();

}

// Hàm tìm các view

private void findViewsByIds() {

ivUitLogo = findViewById(R.id.iv\_uit\_logo);

btnFadeInXml = findViewById(R.id.btn\_fade\_in\_xml);

btnFadeInCode = findViewById(R.id.btn\_fade\_in\_code);

btnFadeOutXml = findViewById(R.id.btn\_fade\_out\_xml);

btnFadeOutCode = findViewById(R.id.btn\_fade\_out\_code);

btnBlinkXml = findViewById(R.id.btn\_blink\_xml);

btnBlinkCode = findViewById(R.id.btn\_blink\_code);

btnZoomInXml = findViewById(R.id.btn\_zoom\_in\_xml);

btnZoomInCode = findViewById(R.id.btn\_zoom\_in\_code);

btnZoomOutXml = findViewById(R.id.btn\_zoom\_out\_xml);

btnZoomOutCode = findViewById(R.id.btn\_zoom\_out\_code);

btnRotateXml = findViewById(R.id.btn\_rotate\_xml);

btnRotateCode = findViewById(R.id.btn\_rotate\_code);

btnMoveXml = findViewById(R.id.btn\_move\_xml);

btnMoveCode = findViewById(R.id.btn\_move\_code);

btnSlideUpXml = findViewById(R.id.btn\_slide\_up\_xml);

btnSlideUpCode = findViewById(R.id.btn\_slide\_up\_code);

btnBounceXml = findViewById(R.id.btn\_bounce\_xml);

btnBounceCode = findViewById(R.id.btn\_bounce\_code);

btnCombineXml = findViewById(R.id.btn\_combine\_xml);

btnCombineCode = findViewById(R.id.btn\_combine\_code);

}

// Hàm thiết lập các sự kiện click cho nút

private void setButtonListeners() {

btnFadeInXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_fade\_in));

btnFadeInCode.setOnClickListener(v -> startAnimationFromCode(new AlphaAnimation(0.0f, 1.0f)));

btnFadeOutXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_fade\_out));

btnFadeOutCode.setOnClickListener(v -> startAnimationFromCode(new AlphaAnimation(1.0f, 0.0f)));

btnBlinkXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_blink));

btnBlinkCode.setOnClickListener(v -> startAnimationFromCode(createBlinkAnimation()));

btnZoomInXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_zoom\_in));

btnZoomInCode.setOnClickListener(v -> startAnimationFromCode(createZoomInAnimation()));

btnZoomOutXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_zoom\_out));

btnZoomOutCode.setOnClickListener(v -> startAnimationFromCode(createZoomOutAnimation()));

btnRotateXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_rotate));

btnRotateCode.setOnClickListener(v -> startAnimationFromCode(createRotateAnimation()));

btnMoveXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_move));

btnMoveCode.setOnClickListener(v -> startAnimationFromCode(createMoveAnimation()));

btnSlideUpXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_slide\_up));

btnSlideUpCode.setOnClickListener(v -> startAnimationFromCode(createSlideUpAnimation()));

btnBounceXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_bounce));

btnBounceCode.setOnClickListener(v -> startAnimationFromCode(createBounceAnimation()));

btnCombineXml.setOnClickListener(v -> startAnimationFromXml(R.anim.anim\_combine));

btnCombineCode.setOnClickListener(v -> startAnimationFromCode(createCombineAnimation()));

}

// Hàm xử lý animation từ XML

private void startAnimationFromXml(int animationResource) {

Animation animation = AnimationUtils.loadAnimation(this, animationResource);

ivUitLogo.startAnimation(animation);

}

// Hàm xử lý animation từ code

private void startAnimationFromCode(Animation animation) {

ivUitLogo.startAnimation(animation);

}

// Các hàm tạo animation từ code (ví dụ animation blink, zoom in...)

private Animation createBlinkAnimation() {

AlphaAnimation blink = new AlphaAnimation(0.0f, 1.0f);

blink.setDuration(300);

blink.setRepeatMode(Animation.REVERSE);

blink.setRepeatCount(3);

return blink;

}

private Animation createZoomInAnimation() {

ScaleAnimation zoomIn = new ScaleAnimation(1.0f, 3.0f, 1.0f, 3.0f);

zoomIn.setDuration(1000);

zoomIn.setFillAfter(true);

return zoomIn;

}

private Animation createZoomOutAnimation() {

ScaleAnimation zoomOut = new ScaleAnimation(1.0f, 0.5f, 1.0f, 0.5f);

zoomOut.setDuration(1000);

zoomOut.setFillAfter(true);

return zoomOut;

}

private Animation createRotateAnimation() {

RotateAnimation rotate = new RotateAnimation(0, 360, Animation.RELATIVE\_TO\_SELF, 0.5f, Animation.RELATIVE\_TO\_SELF, 0.5f);

rotate.setDuration(600);

rotate.setRepeatMode(Animation.RESTART);

rotate.setRepeatCount(2);

return rotate;

}

private Animation createMoveAnimation() {

TranslateAnimation move = new TranslateAnimation(0, 200, 0, 0);

move.setDuration(800);

move.setFillAfter(true);

return move;

}

private Animation createSlideUpAnimation() {

ScaleAnimation slideUp = new ScaleAnimation(1.0f, 1.0f, 1.0f, 0.0f);

slideUp.setDuration(500);

slideUp.setFillAfter(true);

return slideUp;

}

private Animation createBounceAnimation() {

ScaleAnimation bounce = new ScaleAnimation(1.0f, 1.0f, 0.0f, 1.0f);

bounce.setDuration(500);

bounce.setInterpolator(this, android.R.anim.bounce\_interpolator);

bounce.setFillAfter(true);

return bounce;

}

private Animation createCombineAnimation() {

AnimationSet combine = new AnimationSet(true);

ScaleAnimation scale = new ScaleAnimation(1.0f, 3.0f, 1.0f, 3.0f);

scale.setDuration(4000);

RotateAnimation rotate = new RotateAnimation(0, 360);

rotate.setDuration(500);

rotate.setRepeatCount(2);

combine.addAnimation(scale);

combine.addAnimation(rotate);

return combine;

}

@Override

public boolean onTouchEvent(MotionEvent event) {

// Phân tích cử chỉ khi người dùng vuốt

return gestureDetector.onTouchEvent(event) || super.onTouchEvent(event);

}

@Override

public boolean onDown(MotionEvent e) {

return true;

}

@Override

public void onShowPress(MotionEvent e) {}

@Override

public boolean onSingleTapUp(MotionEvent e) {

return false;

}

@Override

public void onLongPress(MotionEvent e) {}

@Override

public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {

if (distanceX < 0) { // Vuốt từ trái qua phải

finish(); // Quay lại MainActivity từ NewActivity

overridePendingTransition(R.anim.slide\_in\_left, R.anim.slide\_out\_right);

} else if (distanceX > 0) { // Vuốt từ phải qua trái

Intent intent = new Intent(MainActivity.this, NewActivity.class);

startActivity(intent);

overridePendingTransition(R.anim.slide\_in\_right, R.anim.slide\_out\_left);

}

return true;

}

@Override

public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {

return false;

}

}

**Trong activity\_main.xml tạo các nút và gán id cho từng nút**

package com.example.nguyenminhkhang\_lab6;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.view.GestureDetector;

import android.view.MotionEvent;

import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class NewActivity extends AppCompatActivity implements GestureDetector.OnGestureListener {

private Button btn\_back;

private GestureDetector gestureDetector; // Khai báo GestureDetector

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_new);

// Đặt animation cho hoạt động chuyển sang từ phải qua trái

overridePendingTransition(R.anim.slide\_in\_right, R.anim.slide\_out\_left);

btn\_back = findViewById(R.id.btn\_back);

btn\_back.setOnClickListener(v -> {

Intent intent = new Intent(NewActivity.this, MainActivity.class);

startActivity(intent);

overridePendingTransition(R.anim.slide\_in\_left, R.anim.slide\_out\_right);

});

// Khởi tạo GestureDetector để phát hiện các cử chỉ

gestureDetector = new GestureDetector(this, this);

}

@SuppressLint("GestureBackNavigation")

@Override

public void onBackPressed() {

super.onBackPressed();

// Đặt animation khi quay lại MainActivity

overridePendingTransition(R.anim.slide\_in\_left, R.anim.slide\_out\_right);

}

@Override

public boolean onTouchEvent(MotionEvent event) {

// Phân tích cử chỉ khi người dùng vuốt

return gestureDetector.onTouchEvent(event) || super.onTouchEvent(event);

}

@Override

public boolean onDown(MotionEvent e) {

return true;

}

@Override

public void onShowPress(MotionEvent e) {}

@Override

public boolean onSingleTapUp(MotionEvent e) {

return false;

}

@Override

public void onLongPress(MotionEvent e) {}

@Override

public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {

// Vuốt từ phải qua trái để quay lại MainActivity

if (distanceX < 0) {

Intent intent = new Intent(NewActivity.this, MainActivity.class);

startActivity(intent);

overridePendingTransition(R.anim.slide\_in\_left, R.anim.slide\_out\_right);

}

return true;

}

@Override

public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {

return false;

}

}

**Lab6\_3**

**Tạo thêm các file xml trong anim**

**Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, màu trắng

Nội dung do AI tạo ra có thể không chính xác.**

**Tạo 1 button Back trong activity\_main**

**Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.**

**Khai báo và tạo logic cho button**

**Ảnh có chứa văn bản, Phông chữ, màu trắng, thiết kế

Nội dung do AI tạo ra có thể không chính xác.**

**Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.**

**Tạo thêm NewActivity để có thể chuyển hướng và áp dụng các animation vừa tạo**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Nội dung do AI tạo ra có thể không chính xác.**

**Trong activity\_new.xml**

package com.example.nguyenminhkhang\_lab6;

import android.annotation.SuppressLint;

import android.os.Bundle;

import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.graphics.Insets;

import androidx.core.view.ViewCompat;

import androidx.core.view.WindowInsetsCompat;

public class NewActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_new);

// Đặt animation cho hoạt động chuyển sang từ phải qua trái

overridePendingTransition(R.anim.slide\_in\_right, R.anim.slide\_out\_left);

}

@SuppressLint("GestureBackNavigation")

@Override

public void onBackPressed() {

super.onBackPressed();

// Đặt animation khi quay lại MainActivity

overridePendingTransition(R.anim.slide\_in\_left, R.anim.slide\_out\_right);

}

}

**Trong MainActivity thêm logic xử lý các animation để qua giao diện NewActivity**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

**Thành phẩm**

**Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Nội dung do AI tạo ra có thể không chính xác.Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Nội dung do AI tạo ra có thể không chính xác.**