Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

«Принципы работы с жестами вводимыми пользователями»

Выполнил:

Студент 4 курса

Группы АС-56

Волк Н.А.

Проверил:

Козинский А.А.

Брест 2022

Лабораторная работа №3

**Цель работы:** Разработка приложения, предоставляющего возможность использования жестов.

**Аннотация:** Разработка приложения, помогающего понять принципы работы c жестами вводимыми пользователями.

**Задачи лабораторной работы:**

• создать набор жестов

• использовать созданные жесты в приложении

**Бонусы (то, что способствует оценке выше 4)**

* Разработка собственного СВЯЗНОГО набора жестов
* Представление в приложении дополнительной справочной информации (текст о назначении приложения)
* Вывод различных информационных сообщений после соответствующих жестов ПОЛЬЗОВАТЕЛЯ. Подготовка сценария управления приложением с помощью жестов пользователя
* Возможность навигации по приложению с помощью набора жестов пользователя

**Код программы:**

MainActivity.java

package com.example.lab7gestures;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.gesture.GestureLibrary;  
import android.gesture.GestureOverlayView;  
import android.gesture.GestureOverlayView.OnGestureListener;  
import android.os.Bundle;  
import java.util.ArrayList;  
import android.app.Activity;  
import android.gesture.Gesture;  
import android.gesture.GestureLibraries;  
import android.gesture.GestureLibrary;  
import android.gesture.GestureOverlayView;  
import android.gesture.GestureOverlayView.OnGesturePerformedListener;  
import android.gesture.Prediction;  
import android.view.Menu;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity implements OnGesturePerformedListener{  
  
 GestureLibrary gLib;  
 GestureOverlayView gestures;  
 TextView tvOut;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 tvOut=(TextView)findViewById(R.id.*startTextContext*);  
 gLib = GestureLibraries.*fromRawResource*(this, R.raw.*gestures*);  
 if (!gLib.load()) {  
 finish();  
 }  
 gestures = (GestureOverlayView) findViewById  
 (R.id.*gestureView*);  
 gestures.addOnGesturePerformedListener(this);  
 }  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 return true;  
 }  
 public void onGesturePerformed(GestureOverlayView overlay,  
 Gesture gesture) {  
 ArrayList<Prediction> predictions = gLib.recognize(gesture);  
 if (predictions.size() > 0) {  
 Prediction prediction = predictions.get(0);  
 if (prediction.score > 3.0) {  
 if (prediction.name.equals("exitGesture"))  
 {  
 System.*exit*(0);  
 }  
 else if (prediction.name.equals("helpGesture"))  
 {  
 Intent intent = new Intent(this, HelpWindowActivity.class);  
 startActivity(intent);  
 tvOut.setText("Помощь");  
 }  
 else if (prediction.name.equals("swipeLeft"))  
 {  
 Intent intent = new Intent(this, SimpleNotepad.class);  
 startActivity(intent);  
 tvOut.setText("Свайп влево.");  
 }  
 else if (prediction.name.equals("swipeRight"))  
 {  
 tvOut.setText("Свайп вправо.");  
 Intent intent = new Intent(this, SimpleIntuitGame.class);  
 startActivity(intent);  
 }  
 else if (prediction.name.equals("calculator"))  
 {  
 Intent intent = new Intent(this, SimpleCalculator.class);  
 startActivity(intent);  
 tvOut.setText("Калькулятор.");  
 }  
  
 }  
 else{  
 Toast.*makeText*( this,"Жест неизвестен",Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
}

Gestures.java

package com.example.lab7gestures;  
  
import java.util.ArrayList;  
import android.os.Bundle;  
import android.app.Activity;  
import android.gesture.Gesture;  
import android.gesture.GestureLibraries;  
import android.gesture.GestureLibrary;  
import android.gesture.GestureOverlayView;  
import android.gesture.GestureOverlayView.OnGesturePerformedListener;  
import android.gesture.Prediction;  
import android.view.Menu;  
import android.widget.TextView;  
public class Gestures extends Activity implements  
 OnGesturePerformedListener  
{  
 GestureLibrary gLib;  
 GestureOverlayView gestures;  
 TextView tvOut;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 tvOut=(TextView)findViewById(R.id.*startTextContext*);  
//Загрузка жестов (gestures) из res/raw/gestures  
 gLib = GestureLibraries.*fromRawResource*(this, R.raw.*gestures*);  
 if (!gLib.load()) {  
//Если жесты не загружены, то выход из приложения  
 finish();  
 }  
 gestures = (GestureOverlayView) findViewById  
 (R.id.*gestureView*);  
 gestures.addOnGesturePerformedListener(this);  
 }  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
// Inflate the menu; this adds items to the action bar if it is present.  
 //getMenuInflater().inflate(R.menu.gestures, menu);  
 return true;  
 }  
 public void onGesturePerformed(GestureOverlayView overlay,  
 Gesture gesture) {  
//Создаёт ArrayList c загруженными из gestures жестами  
 ArrayList<Prediction> predictions = gLib.recognize(gesture);  
 if (predictions.size() > 0) {  
//если загружен хотя бы один жест из gestures  
 Prediction prediction = predictions.get(0);  
 if (prediction.score > 1.0) {  
 if (prediction.name.equals("exitGesture"))  
 {  
 tvOut.setText("1");  
 }  
 else if (prediction.name.equals("swipeHorizontal"))  
 {  
 tvOut.setText("stop");  
 }  
 else if (prediction.name.equals("two"))  
 {  
 tvOut.setText("2");  
 }  
 }  
 else{  
 tvOut.setText("Жест неизвестен");  
 }  
 }  
 }  
}

HelpWindowActivity.java

package com.example.lab7gestures;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.gesture.Gesture;  
import android.gesture.GestureLibraries;  
import android.gesture.GestureLibrary;  
import android.gesture.GestureOverlayView;  
import android.gesture.Prediction;  
import android.os.Bundle;  
import android.view.Menu;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
  
public class HelpWindowActivity extends AppCompatActivity implements  
 GestureOverlayView.OnGesturePerformedListener {  
 GestureLibrary gLib;  
 GestureOverlayView gestures;  
 TextView textHelp;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_help\_window*);  
 textHelp=(TextView)findViewById(R.id.*textHelp*);  
 textHelp.setText("С - калькулятор\n > - игра-угадайка\n < - блокнот\n o - выход\n /\\ - помощь");  
 gLib = GestureLibraries.*fromRawResource*(this, R.raw.*gestures*);  
 if (!gLib.load()) {  
 finish();  
 }  
 gestures = (GestureOverlayView) findViewById  
 (R.id.*gestureHelp*);  
 gestures.addOnGesturePerformedListener(this);  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 //getMenuInflater().inflate(R.menu.gestures, menu);  
 return true;  
 }  
  
 public void onGesturePerformed(GestureOverlayView overlay,  
 Gesture gesture) {  
 ArrayList<Prediction> predictions = gLib.recognize(gesture);  
 if (predictions.size() > 0) {  
 Prediction prediction = predictions.get(0);  
 if (prediction.score > 3.0) {  
 if (prediction.name.equals("exitGesture"))  
 {  
 this.finish();  
 }  
 }  
 else{  
 Toast.*makeText*(this,"Жест неизвестен",Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
}

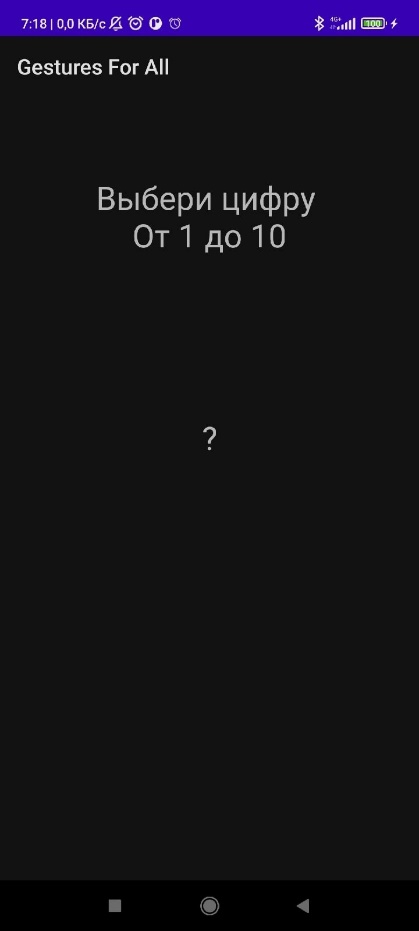
SimpleIntuitGame.java

package com.example.lab7gestures;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.gesture.Gesture;  
import android.gesture.GestureLibraries;  
import android.gesture.GestureLibrary;  
import android.gesture.GestureOverlayView;  
import android.gesture.Prediction;  
import android.os.Bundle;  
import android.os.SystemClock;  
import android.widget.TextView;  
import android.widget.Toast;  
import java.util.ArrayList;  
  
public class SimpleIntuitGame extends AppCompatActivity implements GestureOverlayView.OnGesturePerformedListener {  
 GestureLibrary gLib;  
 GestureOverlayView gestures;  
 String message="";  
 TextView text,infoHeader;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_simple\_intuit\_game*);  
 text=(TextView)findViewById(R.id.*msgTmr*);  
 infoHeader=(TextView)findViewById(R.id.*infoHead*);  
 text.setText("?");  
 //Загрузка жестов (gestures) из res/raw/gestures  
 gLib = GestureLibraries.*fromRawResource*(this, R.raw.*gestures\_numbers*);  
 if (!gLib.load()) {  
 finish();  
 }  
 gestures = (GestureOverlayView) findViewById  
 (R.id.*gestureIntuit*);  
 gestures.addOnGesturePerformedListener(this);  
 Integer temp = (int)(Math.*random*()\*10);  
 message+= temp.toString();  
 }  
 public void onGesturePerformed(GestureOverlayView overlay,  
 Gesture gesture) {  
 ArrayList<Prediction> predictions = gLib.recognize(gesture);  
 if (predictions.size() > 0) {  
 Prediction prediction = predictions.get(0);  
 if (prediction.score > 3.0) {  
 if (prediction.name.equals(message))  
 {  
 infoHeader.setText("Win");  
 text.setText(message);  
 this.finish();  
 }else{  
 infoHeader.setText("Lose");  
 text.setText(message);  
 this.finish();  
 }  
 }  
 else{  
 Toast.*makeText*( this,"Жест неизвестен",Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 SystemClock.*sleep*(1000);  
 }  
}

SimpleNotepad.java

package com.example.lab7gestures;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.gesture.Gesture;  
import android.gesture.GestureLibraries;  
import android.gesture.GestureLibrary;  
import android.gesture.GestureOverlayView;  
import android.gesture.Prediction;  
import android.os.Bundle;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
  
public class SimpleNotepad extends AppCompatActivity implements GestureOverlayView.OnGesturePerformedListener {  
 GestureLibrary gLib;  
 GestureOverlayView gestures;  
 String message="";  
 TextView text;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_simple\_notepad*);  
 text=(TextView)findViewById(R.id.*notepadText*);  
 //Загрузка жестов (gestures) из res/raw/gestures  
 gLib = GestureLibraries.*fromRawResource*(this, R.raw.*gestures\_alphabet*);  
 if (!gLib.load()) {  
//Если жесты не загружены, то выход из приложения  
 finish();  
 }  
 gestures = (GestureOverlayView) findViewById  
 (R.id.*notepadGesture*);  
 gestures.addOnGesturePerformedListener(this);  
 }  
 public void onGesturePerformed(GestureOverlayView overlay,  
 Gesture gesture) {  
//Создаёт ArrayList c загруженными из gestures жестами  
 ArrayList<Prediction> predictions = gLib.recognize(gesture);  
 if (predictions.size() > 0) {  
//если загружен хотя бы один жест из gestures  
 Prediction prediction = predictions.get(0);  
 if (prediction.score > 3.0) {  
 if (prediction.name.equals("save"))  
 {  
 this.finish();  
 }  
 else if (prediction.name.equals("a")) message+="а";  
 else if (prediction.name.equals("b")) message+="б";  
 else if (prediction.name.equals("v")) message+="в";  
 else if (prediction.name.equals("g")) message+="г";  
 else if (prediction.name.equals("d")) message+="д";  
 else if (prediction.name.equals("e")) message+="е";  
 else if (prediction.name.equals("z")) message+="з";  
 else if (prediction.name.equals("zh")) message+="ж";  
 else if (prediction.name.equals("i")) message+="и";  
 else if (prediction.name.equals("k")) message+="к";  
 else if (prediction.name.equals("l")) message+="л";  
 else if (prediction.name.equals("m")) message+="м";  
 else if (prediction.name.equals("n")) message+="н";  
 else if (prediction.name.equals("o")) message+="о";  
 else if (prediction.name.equals("p")) message+="п";  
 else if (prediction.name.equals("r")) message+="р";  
 else if (prediction.name.equals("s")) message+="с";  
 else if (prediction.name.equals("t")) message+="т";  
 else if (prediction.name.equals("u")) message+="у";  
 else if (prediction.name.equals("f")) message+="ф";  
 else if (prediction.name.equals("h")) message+="х";  
 else if (prediction.name.equals("c")) message+="ц";  
 else if (prediction.name.equals("ch")) message+="ч";  
 else if (prediction.name.equals("sh")) message+="ш";  
 else if (prediction.name.equals("sch")) message+="щ";  
 else if (prediction.name.equals("tv")) message+="ъ";  
 else if (prediction.name.equals("hu")) message+="ы";  
 else if (prediction.name.equals("mg")) message+="ь";  
 else if (prediction.name.equals("ea")) message+="э";  
 else if (prediction.name.equals("yu")) message+="ю";  
 else if (prediction.name.equals("ya")) message+="я";  
 else if (prediction.name.equals("space")) message+=" ";  
 else if (prediction.name.equals("erase")) message = message.substring(0,message.length()-1);  
  
 text.setText(message);  
 }  
 else{  
 Toast.*makeText*( this,"Жест неизвестен",Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 }  
}

Пример выполнения программы:

***Вывод:*** разработал приложение, предоставляющее возможность использования жестов.