МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федерально автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационных систем

Куркчи Ариф Эрнестович

Институт информационных технологий и управления в технических системах

курс 3 группа ИС/б-31-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Мобильные ИТ»

на тему «Исследование принципов разработки приложений,

распознающих жесты, вводимые пользователем.»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

cт.преподаватель Шишкевич В.Е.

(должность) (подпись) (инициалы, фамилия)

Севастополь 2017

1. ЦЕЛЬ РАБОТЫ

Изучить принципы разработки приложений для работы с жестами вводимыми пользователями.

1. ИСХОДНЫЙ КОД

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | package ru.justnero.sevsu.s3.mit.lab04; | | 2 |  | | 3 | import *android.gesture.Gesture*; | | 4 | import *android.gesture.GestureLibraries*; | | 5 | import *android.gesture.GestureLibrary*; | | 6 | import *android.gesture.GestureOverlayView*; | | 7 | import *android.gesture.Prediction*; | | 8 | import *android.support.v7.app.AppCompatActivity*; | | 9 | import *android.os.Bundle*; | | 10 | import *android.util.Log*; | | 11 | import *android.view.View*; | | 12 | import *android.widget.Button*; | | 13 | import *android.widget.EditText*; | | 14 | import *android.widget.TextView*; | | 15 |  | | 16 | import *java.util.ArrayList*; | | 17 |  | | 18 | public class MainActivity extends *AppCompatActivity* implements *GestureOverlayView*.*OnGesturePerformedListener*, *View*.*OnClickListener* { | | 19 |  | | 20 | *GestureLibrary* gLib; | | 21 | *GestureOverlayView* gestures; | | 22 | *TextView* tvResult; | | 23 | *EditText* etResult; | | 24 | *Button* btnResult; | | 25 |  | | 26 |  | | 27 | *@Override* | | 28 | protected *void* onCreate(*Bundle* *savedInstanceState*) { | | 29 | super.onCreate(savedInstanceState); | | 30 | setContentView(*R*.layout.activity\_main); | | 31 |  | | 32 | tvResult = (*TextView*) findViewById(*R*.id.tvResult); | | 33 | etResult = (*EditText*) findViewById(*R*.id.etResult); | | 34 | btnResult = (*Button*) findViewById(*R*.id.btnResult); | | 35 |  | | 36 | btnResult.setOnClickListener(this); | | 37 |  | | 38 | gLib = *GestureLibraries*.fromRawResource(this, *R*.raw.gestures); | | 39 | if (!gLib.load()) { | | 40 | finish(); | | 41 | } | | 42 |  | | 43 | gestures = (*GestureOverlayView*) findViewById(*R*.id.gestureOverlayView1); | | 44 | gestures.addOnGesturePerformedListener(this); | | 45 | } | | 46 |  | | 47 | *@Override* | | 48 | public *void* onGesturePerformed(*GestureOverlayView* *overlay*, *Gesture* *gesture*) { | | 49 | //Создаёт ArrayList c загруженными из gestures жестами | | 50 | *ArrayList<Prediction>* predictions = gLib.recognize(gesture); | | 51 | if (predictions.size() > 0) { | | 52 | *Log*.d("Lab.04", predictions.toString()); | | 53 | //если загружен хотябы один жест из gestures | | 54 | *Prediction* prediction = predictions.get(0); | | 55 | if (prediction.score > 1.0) { | | 56 | switch (prediction.name) { | | 57 | case "0": { | | 58 | etResult.setText(etResult.getText().toString() + 0); | | 59 | break; | | 60 | } | | 61 | case "1": { | | 62 | etResult.setText(etResult.getText().toString() + 1); | | 63 | break; | | 64 | } | | 65 | case "2": { | | 66 | etResult.setText(etResult.getText().toString() + 2); | | 67 | break; | | 68 | } | | 69 | case "3": { | | 70 | etResult.setText(etResult.getText().toString() + 3); | | 71 | break; | | 72 | } | | 73 | case "4": { | | 74 | etResult.setText(etResult.getText().toString() + 4); | | 75 | break; | | 76 | } | | 77 | case "5": { | | 78 | etResult.setText(etResult.getText().toString() + 5); | | 79 | break; | | 80 | } | | 81 | case "6": { | | 82 | etResult.setText(etResult.getText().toString() + 6); | | 83 | break; | | 84 | } | | 85 | case "7": { | | 86 | etResult.setText(etResult.getText().toString() + 7); | | 87 | break; | | 88 | } | | 89 | case "8": { | | 90 | etResult.setText(etResult.getText().toString() + 8); | | 91 | break; | | 92 | } | | 93 | case "9": { | | 94 | etResult.setText(etResult.getText().toString() + 9); | | 95 | break; | | 96 | } | | 97 | case "+": { | | 98 | etResult.setText(etResult.getText().toString() + "+"); | | 99 | break; | | 100 | } | | 101 | case "-": { | | 102 | etResult.setText(etResult.getText().toString() + "-"); | | 103 | break; | | 104 | } | | 105 | } | | 106 | } | | 107 | } | | 108 | } | | 109 |  | | 110 | *@Override* | | 111 | public *void* onClick(*View* *v*) { | | 112 | switch (v.getId()) { | | 113 | case *R*.id.btnResult: { | | 114 | *String* res = ""; | | 115 | *String* s = etResult.getText().toString(); | | 116 | try { | | 117 | etResult.setText(etResult.getText().toString() + " = " + *String*.valueOf(eval(s))); | | 118 | } catch (*Exception* ex) { | | 119 | etResult.setText(ex.getMessage()); | | 120 | } | | 121 |  | | 122 | break; | | 123 | } | | 124 | } | | 125 | } | | 126 |  | | 127 | public static *double* eval(final *String* *str*) { | | 128 | return new *Object*() { | | 129 | *int* pos = -1, ch; | | 130 |  | | 131 | *void* nextChar() { | | 132 | ch = (++pos < str.length()) ? str.charAt(pos) : -1; | | 133 | } | | 134 |  | | 135 | *boolean* eat(*int* *charToEat*) { | | 136 | while (ch == ' ') nextChar(); | | 137 | if (ch == charToEat) { | | 138 | nextChar(); | | 139 | return true; | | 140 | } | | 141 | return false; | | 142 | } | | 143 |  | | 144 | *double* parse() { | | 145 | nextChar(); | | 146 | *double* x = parseExpression(); | | 147 | if (pos < str.length()) throw new *RuntimeException*("Unexpected: " + (*char*) ch); | | 148 | return x; | | 149 | } | | 150 |  | | 151 | // Grammar: | | 152 | // expression = term | expression `+` term | expression `-` term | | 153 | // term = factor | term `\*` factor | term `/` factor | | 154 | // factor = `+` factor | `-` factor | `(` expression `)` | | 155 | //        | number | functionName factor | factor `^` factor | | 156 |  | | 157 | *double* parseExpression() { | | 158 | *double* x = parseTerm(); | | 159 | for (; ; ) { | | 160 | if (eat('+')) x += parseTerm(); // addition | | 161 | else if (eat('-')) x -= parseTerm(); // subtraction | | 162 | else return x; | | 163 | } | | 164 | } | | 165 |  | | 166 | *double* parseTerm() { | | 167 | *double* x = parseFactor(); | | 168 | for (; ; ) { | | 169 | if (eat('\*')) x \*= parseFactor(); // multiplication | | 170 | else if (eat('/')) x /= parseFactor(); // division | | 171 | else return x; | | 172 | } | | 173 | } | | 174 |  | | 175 | *double* parseFactor() { | | 176 | if (eat('+')) return parseFactor(); // unary plus | | 177 | if (eat('-')) return -parseFactor(); // unary minus | | 178 |  | | 179 | *double* x; | | 180 | *int* startPos = this.pos; | | 181 | if (eat('(')) { // parentheses | | 182 | x = parseExpression(); | | 183 | eat(')'); | | 184 | } else if ((ch >= '0' && ch <= '9') || ch == '.') { // numbers | | 185 | while ((ch >= '0' && ch <= '9') || ch == '.') nextChar(); | | 186 | x = *Double*.parseDouble(str.substring(startPos, this.pos)); | | 187 | } else if (ch >= 'a' && ch <= 'z') { // functions | | 188 | while (ch >= 'a' && ch <= 'z') nextChar(); | | 189 | *String* func = str.substring(startPos, this.pos); | | 190 | x = parseFactor(); | | 191 | if (func.equals("sqrt")) x = *Math*.sqrt(x); | | 192 | else if (func.equals("sin")) x = *Math*.sin(*Math*.toRadians(x)); | | 193 | else if (func.equals("cos")) x = *Math*.cos(*Math*.toRadians(x)); | | 194 | else if (func.equals("tan")) x = *Math*.tan(*Math*.toRadians(x)); | | 195 | else throw new *RuntimeException*("Unknown function: " + func); | | 196 | } else { | | 197 | throw new *RuntimeException*("Unexpected: " + (*char*) ch); | | 198 | } | | 199 |  | | 200 | if (eat('^')) x = *Math*.pow(x, parseFactor()); // exponentiation | | 201 |  | | 202 | return x; | | 203 | } | | 204 | }.parse(); | | 205 | } | | 206 | } | |  |

ВЫВОДЫ

В ходе лабораторной работы были изучены принципы разработки приложений, распознающих жесты, вводимые пользователем, создано приложение, демонстрирующее ввод жестами.