Министерство образования Республики Беларусь
Учреждение образования
Брестский государственный технический университет
Кафедра ИИТ

Лабораторная работа №4
За 6 семестр
По дисциплине «Разработка ПО для мобильных систем»
Тема: «Игра “Память” на Android»

Выполнил:
студент 3 курса
Группы ПО-4(2)
Яковчик И.А.
Проверил:
Козинский А.П..

Брест 2022

Лабораторная работа №3

Цель: необходимо реализовать интерфейс приложения для последовательного открытия пар карточек для запоминания.

Приложение должно:
отображать игровое поле минимального размера 4x4
в случае если пользователь открыл 2 несовпадающие картинки, они должны скрываться предоставлять возможность перезапустить игру

Приложение будет называться «Memory game». Интерфейс приложения состоит из заголовка TextView, игрового поля 4x4 и кнопки Button. На игровом поле располагаются 16 попарно одинаковых изображений ImageButton.
Логика приложения следующая: игрок наживает по очереди на две карточки, которые в свою очередь переворачиваются изображением вверх. В случае если изображения не совпадают, карточки обратно переворачиваются, в ином случае данные карточки становятся неактивными. По завершению игры можно нажать на кнопку «restrart», которая перетасует карточки, и игра начнётся заново.

Код программы:

```
package com.example.memory

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.ImageButton
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*
import com.example.memory.R.drawable.*

private const val TAG = "MainActivity"

class MainActivity : AppCompatActivity() {

    private lateinit var buttons: List<ImageButton>
    private lateinit var cards: List<MemoryCard>
    private var indexOfSingleSelectedCard: Int? = null
    private var matchResult: Boolean = true

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val images = mutableListOf(ic_cake, ic_chocolate, ic_fries,
ic_pizza, ic_ramen, ic_salad, ic_soup, ic_steak)
        images.addAll(images)
        images.shuffle()
```

```
        buttons = listOf(imageButton1, imageButton2, imageButton3,
imageButton4, imageButton5, imageButton6,
            imageButton7, imageButton8, imageButton10,
imageButton11, imageButton12, imageButton13,
            imageButton14, imageButton15, imageButton16,
imageButton18)

        cards = buttons.indices.map { index ->
            MemoryCard(images[index])
        }

        button.setOnClickListener {
            indexOfSingleSelectedCard = null
            cards.forEachIndexed { index, memoryCard ->
                val button = buttons[index]
                button.alpha = 1.0f
                memoryCard.isFaceUp = false
                memoryCard.isMatched = false
                memoryCard.isSelected = false
                button.setImageResource(ic_creative_food)
            }
            images.shuffle()
            cards = buttons.indices.map { index ->
                MemoryCard(images[index])
            }
            matchResult = true
        }

        buttons.forEachIndexed { index, imageButton ->
            imageButton.setOnClickListener {
                Log.i(TAG, "button clicked")

                updateModels(index, imageButton)

                updateViews()

                updateView()
            }
        }
    }

    private fun updateView() {
        if (!matchResult) {
            Thread {
                Thread.sleep(300)
                runOnUiThread {
                    cards.forEachIndexed { index, memoryCard ->
                        val button = buttons[index]
```

```
                        if (memoryCard.isSelected &&
memoryCard.isFaceUp) {

button.setImageResource(ic_creative_food)
                            memoryCard.isSelected = false
                            memoryCard.isFaceUp = false
                        }
                    }
                }
            }.start()

        }
        matchResult = true
    }

    private fun updateViews() {
        cards.forEachIndexed { index, memoryCard ->
            val button = buttons[index]
            if (memoryCard.isMatched)
                button.alpha = 0.1f
            button.setImageResource(if (memoryCard.isFaceUp)
memoryCard.identifier else ic_creative_food)
        }

    }

    private fun updateModels(index: Int, imageButton: ImageButton)
{
        val card = cards[index]
        //Error checking
        if (card.isFaceUp){
            return
        }

        if(indexOfSingleSelectedCard == null) {
            //0 or 2 selected cards
            restoreCards()
            indexOfSingleSelectedCard = index
            card.isSelected = true
        }
        else {
            if (indexOfSingleSelectedCard != null) {
                //1 card
                matchResult =
checkForMatch(indexOfSingleSelectedCard!!, index)
                indexOfSingleSelectedCard = null
                card.isSelected = true
            }
```

```
        }

        card.isFaceUp = !card.isFaceUp
    }

    private fun restoreCards() {
        for (card in cards) {
            if (!card.isMatched) {
                card.isFaceUp = false
            }
            //card.isSelected = false
        }
    }

    private fun checkForMatch(index1: Int, index2: Int) : Boolean {
        if(cards[index1].identifier == cards[index2].identifier) {
            Toast.makeText(this, "Match found",
Toast.LENGTH_SHORT).show()
            cards[index1].isMatched = true
            cards[index2].isMatched = true
            return true
        }
        return false
    }
}
```

Вывод: в ходе данной лабораторной работы была разработана игра «Память». При создании проекта был использованы дополнительные материалы. Использование дополнительных материалов объясняется работой с классами, потоками и коллекциями, с которыми раньше не доводилось работать:
класс MemoryCard нужен был для хранения состояния отдельной карточки
коллекция List для хранения карточек и классов их состояний
поток Thread был использован для переворачивания карточки рубашкой вверх, если две выбранные карточки не совпадают

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- Buil the Memory Game: Intro to Android Workshop[Видеоматериал]. – Режим доступа: https://www.youtube.com/watch?v=U4Wtjewy7EY. – Дата доступа: 25.02.2022
- Metanit[Электронный ресурс]. – Режим доступа: https://metanit.com/kotlin/tutorial/. – Дата доступа: 25.02.2022