5COSC023W - Tutorial 5 Exercises - Sample Solutions

1 The Lost Dog Application

The solution is in the lecture slides. For the extension, you just need to add a `TextField`, extract its contents and append to the email content.

2 Identify the Dog Breed App

Before attempting the solution, copy the relevant images (the ones you use in your application) inside the `drawable` directory.

The first activity:

```
package com.example.dogbreedscomposable

import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.material3.TextField
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import kotlin.random.Random

// Pick up the images from: http://vision.stanford.edu/aditya86/ImageNetDogs/
val drawable_ids = listOf(R.drawable.n02106030_16171, R.drawable.n02106030_16173,
                          R.drawable.n02096585_10380, R.drawable.n02096585_10452,
                          R.drawable.n02091467_1110, R.drawable.n02091467_1144,
                          R.drawable.n02085620_10074, R.drawable.n02085620_10131)

val breeds = listOf("Collie", "Collie", "Boston_Bull", "Boston_Bull",
                    "Elkhound", "Elkhound", "Chihuahua", "Chihuahua")

var score = 0
var total_questions = -1

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            GUI()
        }
    }
}

@Preview
@Composable
fun GUI() {
    var drawable1 by remember{mutableStateOf(0) }
    var drawable2 by remember{mutableStateOf(0) }
    var drawable3 by remember{mutableStateOf(0) }

    var clicked by remember{ mutableStateOf(true) }
    var answer by remember{ mutableStateOf("") }
    var breed_choice by remember{ mutableStateOf(0) }

    // initialise with some images the first time this composable is called
    if (clicked) {
        ++total_questions  // how many questions have been presented to the player

        val new_drawables = chooseNewImages()
        drawable1 = new_drawables[0]
        drawable2 = new_drawables[1]
        drawable3 = new_drawables[2]
        clicked = false

        // choose one of the 3 images as the breed to identify
        breed_choice = Random.nextInt(3)
    }
```

```
Column (modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally){
    var name by remember{ mutableStateOf("") }
    Row {
        Image(painterResource(id = drawable1),
            contentDescription = "image 1",
            modifier = Modifier.clickable {
                clicked = true
                if (breed_choice == 0) {
                    answer = "Correct"
                    ++score
                }
                else
                    answer = "Incorrect"
            })

        Image(painterResource(id = drawable2),
            contentDescription = "image 2",
            modifier = Modifier.clickable {
                clicked = true
                if (breed_choice == 1) {
                    answer = "Correct"
                    ++score
                }
                else
                    answer = "Incorrect"
            })

        Image(painterResource(id = drawable3),
            contentDescription = "image 3",
            modifier = Modifier.clickable {
                clicked = true
                if (breed_choice == 2) {
                    answer = "Correct"
                    ++score
                }
                else
                    answer = "Incorrect"
            })
    }

    var breed: String = ""
    var idx = 0
    if (breed_choice == 0)
        idx = drawable_ids.indexOf(drawable1)
    else if (breed_choice == 1)
        idx = drawable_ids.indexOf(drawable2)
    else
        idx = drawable_ids.indexOf(drawable3)
        breed = breeds[idx]

        // display the breed
        Text(text = breed,
            modifier = Modifier
                .padding(vertical = 10.dp),
            fontSize = 20.sp,
            textAlign = TextAlign.Center)

        // display correct or incorrect when clicked
        Text(text = answer, modifier = Modifier.fillMaxWidth(),
            textAlign = TextAlign.Center)

        // start the result activity when button pressed
        val context = LocalContext.current
        Button(modifier = Modifier.padding(40.dp),
            onClick = {
            var i = Intent(context, ResultsActivity::class.java)
            i.putExtra("score", score)
            i.putExtra("total", total_questions)
            i.putExtra("name", name)
            context.startActivity(i)
        })
        {
             Text("Finish")
        }

        Row (verticalAlignment = Alignment.CenterVertically) {
            Text("Name: ")
            TextField(value = name, onValueChange = {
                new_name -> name = new_name
            })
        }
    }
}


// return the ids of drawable for 3 new images
fun chooseNewImages(): List<Int> {
    var drawables_list = mutableListOf<Int>()

    var index1 = Random.nextInt(breeds.size)
    var breed1 = breeds[index1]

    var index2 = Random.nextInt(breeds.size)
    var breed2 = breeds[index2]
    while (breed1 == breed2) {
        index2 = Random.nextInt(breeds.size)
        breed2 = breeds[index2]
    }
    var index3 = Random.nextInt(breeds.size)
    var breed3 = breeds[index3]
    while (breed1 == breed3 || breed2 == breed3) {
        index3 = Random.nextInt(breeds.size)
        breed3 = breeds[index3]
    }

    // add the 3 drawables in the list
    drawables_list.add(drawable_ids[index1])
    drawables_list.add(drawable_ids[index2])
    drawables_list.add(drawable_ids[index3])

    return drawables_list
}
```

The second activity:

```
package com.example.dogbreedscomposable

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.material3.Text

class ResultsActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Column {
                val score = intent.getIntExtra("score", 0)
                val total = intent.getIntExtra("total", 0)
                val name = intent.getStringExtra("name")

                Text(text=name + " Score")
                Text(text = "Correct guesses: " + score)
                Text(text = "Incorrect guesses: " + (total - score))
            }
        }
    }
}
```