package com.example.matchingimage  
  
import android.os.Bundle  
import android.widget.ImageButton  
import androidx.appcompat.app.AlertDialog  
import androidx.appcompat.app.AppCompatActivity  
  
class MainActivity : AppCompatActivity() {  
  
 private var clicked = 0  
 private var turnOver = false  
 private var lastClicked = -1  
 private lateinit var images: MutableList<Int>  
 private lateinit var buttons: Array<ImageButton>  
 private lateinit var cardFlipped: BooleanArray  
 private val cardBack = R.drawable.*teemo\_\_2\_* override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 initGame()  
  
 val resetButton = findViewById<ImageButton>(R.id.*resetButton*)  
 resetButton.setOnClickListener **{** resetGame()  
 **}** }  
  
 private fun initGame() {  
 images = *mutableListOf*(  
 R.drawable.*fiddle\_\_2\_*, R.drawable.*garen\_\_2\_*, R.drawable.*kata\_\_2\_*,  
 R.drawable.*kayn\_\_2\_*, R.drawable.*nami\_\_2\_*, R.drawable.*nunu\_\_2\_*,  
 R.drawable.*fiddle\_\_2\_*, R.drawable.*garen\_\_2\_*, R.drawable.*kata\_\_2\_*,  
 R.drawable.*kayn\_\_2\_*, R.drawable.*nami\_\_2\_*, R.drawable.*nunu\_\_2\_* ).*apply* **{** *shuffle*() **}** buttons = *arrayOf*(  
 findViewById(R.id.*imageButton*),  
 findViewById(R.id.*imageButton2*),  
 findViewById(R.id.*imageButton3*),  
 findViewById(R.id.*imageButton4*),  
 findViewById(R.id.*imageButton5*),  
 findViewById(R.id.*imageButton6*),  
 findViewById(R.id.*imageButton7*),  
 findViewById(R.id.*imageButton8*),  
 findViewById(R.id.*imageButton9*),  
 findViewById(R.id.*imageButton10*),  
 findViewById(R.id.*imageButton11*),  
 findViewById(R.id.*imageButton12*)  
 )  
  
 cardFlipped = BooleanArray(buttons.size) **{** false **}** for (i in buttons.*indices*) {  
 buttons[i].setImageResource(cardBack)  
 buttons[i].*isClickable* = true  
 buttons[i].setOnClickListener **{** onCardClicked(i) **}** }  
  
 clicked = 0  
 turnOver = false  
 lastClicked = -1  
 }  
  
 private fun onCardClicked(index: Int) {  
 if (cardFlipped[index] || turnOver) return  
  
 buttons[index].setImageResource(images[index])  
 cardFlipped[index] = true  
  
 if (clicked == 0) {  
 lastClicked = index  
 clicked = 1  
 } else if (clicked == 1) {  
 if (images[index] == images[lastClicked]) {  
 buttons[index].*isClickable* = false  
 buttons[lastClicked].*isClickable* = false  
  
 // 🏆 Check if player wins  
 if (cardFlipped.*all* **{ it }**) {  
 showWinDialog()  
 }  
  
 } else {  
 turnOver = true  
 buttons[index].postDelayed(**{** buttons[index].setImageResource(cardBack)  
 buttons[lastClicked].setImageResource(cardBack)  
 cardFlipped[index] = false  
 cardFlipped[lastClicked] = false  
 turnOver = false  
 **}**, 500)  
 }  
 clicked = 0  
 }  
 }  
  
 private fun resetGame() {  
 images.*shuffle*()  
 for (i in buttons.*indices*) {  
 buttons[i].setImageResource(cardBack)  
 buttons[i].*isClickable* = true  
 cardFlipped[i] = false  
 }  
 clicked = 0  
 turnOver = false  
 lastClicked = -1  
 }  
  
 private fun showWinDialog() {  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Congratulations!")  
 builder.setMessage("You've matched all the cards! 🎉")  
 builder.setPositiveButton("Play Again") **{** dialog, \_ **->** resetGame()  
 dialog.dismiss()  
 **}** builder.setCancelable(false)  
 builder.show()  
 }  
}