

A. XML Layout (Code)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="236dp"
        android:layout_height="220dp"
        android:layout_marginBottom="8dp"
        android:scaleType="centerInside"
        app:layout_constraintBottom_toTopOf="@+id/linearLayout2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:srcCompat="@drawable/empty" />

    <Switch
        android:id="@+id/switchMode"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="4dp"
        android:minHeight="48dp"
        android:text="@string/start_game"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="394dp"
        android:layout_height="57dp"
        android:layout_marginBottom="16dp"
        android:orientation="horizontal"
        app:layout_constraintBottom_toTopOf="@+id/switchMode"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.47"
        app:layout_constraintStart_toStartOf="parent">

    <Button
        android:id="@+id/btnApple"
        android:layout_width="102dp"
        android:layout_height="48dp"
        android:layout_marginLeft="5dp"
        android:text="@string/show_apple"
        android:textAppearance="@style/TextAppearance.AppCompat.Small"
        android:textSize="12sp"
        app:icon="@drawable/ic_apple" />
```

```

<Button
    android:id="@+id/btnGrape"
    android:layout_width="102dp"
    android:layout_height="48dp"
    android:layout_marginLeft="35dp"
    android:text="@string/show_grape"
    android:textAppearance="@style/TextAppearance.AppCompat.Small"
    android:textSize="12sp"
    app:icon="@drawable/ic_grape" />

<Button
    android:id="@+id/btnRandom"
    android:layout_width="102dp"
    android:layout_height="48dp"
    android:layout_marginLeft="30dp"
    android:text="@string/random_image"
    android:textAppearance="@style/TextAppearance.AppCompat.Small"
    android:textSize="12sp" />
</LinearLayout>
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="253dp"
    android:layout_height="210dp"
    android:layout_marginBottom="36dp"
    app:layout_constraintBottom_toTopOf="@+id/txtguess"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:srcCompat="@drawable/empty" />

<TextView
    android:id="@+id/txtguess"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:text="Your Guess Image"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/imageView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
<TextView
    android:id="@+id/txtrandom"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Random Image"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/imageView2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.526" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

B. Main Activity File Source Code (KOTLIN)

```
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private var fruits = mutableListOf(R.drawable.apple, R.drawable.grapes,
R.drawable.orange)
    private var mode: Boolean = true
    private var userschoice: Int = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        modeOff()
        binding.switchMode.setOnCheckedChangeListener { _, isChecked ->
            ToggleGameModed(isChecked)
        }
    }

    private fun ToggleGameModed(isChecked: Boolean) {
        if (isChecked) {
            binding.switchMode.text = getString(R.string.stop_game)
            AppleClick()
            GrapeClick()
            RandomImage()

        } else {
            binding.switchMode.text = getString(R.string.start_game)
            binding.imageView2.setImageResource(R.drawable.empty)
            binding.imageView.setImageResource(R.drawable.empty)

            modeOff()
        }
    }

    private fun GrapeClick() {
        binding.btnGrape.setOnClickListener {
            userschoice = R.drawable.grapes
            binding.imageView.setImageResource(userschoice)
        }
    }

    private fun AppleClick() {
        binding.btnApple.setOnClickListener {
            userschoice = R.drawable.apple
            binding.imageView.setImageResource(userschoice)
        }
    }
}
```

```

private fun RandomImage() {
    binding.btnRandom.setOnClickListener {
        val random = (fruits).shuffled().first()
        binding.imageView2.setImageResource(random)

        if(userschoice == (random)){
            Snackbar.make(it, "Congratulations, You Win!!!",
Snackbar.LENGTH_LONG).show()
        }
        else{
            Snackbar.make(it, "Sorry, You Lose...",
Snackbar.LENGTH_LONG).show()
        }
    }
}

private fun modeOff() {
    if (mode) {
        binding.btnApple.setOnClickListener {
            Snackbar.make(it, "Start the game", Snackbar.LENGTH_LONG).show()
        }
        binding.btnGrape.setOnClickListener {
            Snackbar.make(it, "Start the game", Snackbar.LENGTH_LONG).show()
        }
        binding.btnRandom.setOnClickListener {
            Snackbar.make(it, "Start the game", Snackbar.LENGTH_LONG).show()
        }
    }
}
}

```

B.1. Include a short discussion on how you were able to accomplished the task

In my short discussion on how I was able to accomplished the task is, first of all make an Boolean “mode” that whenever the app is off and when you try to press any of the buttons it will only show Snack Bar message “start the game” and it will only be remove when you turn the game on, and for the comparing of both images what I did was, based on the hint that was given to us, I make a variable to store the two R.ID of apple and grapes in that variable depending on what button was press, then compare it to the variable random that was provided if it is the same value.

- C. Screen Record a short video demo (10-15 seconds) of your working app showing the accomplished functionalities ..

<https://www.youtube.com/watch?v=djhOW1msbQc>

- D. your self-assessment (what have you learned?).

In my self-assessment, what I have learned is how to manipulate and use images in Android Kotlin and how you even compare it, also using List to randomize elements with combination of string interpolation to view the random element that was shown. The topic was very fun to actually learn, especially doing it on an actual application of android.