TASK GUIDE (B2.04)

A. Objectives.

Student will declare fields needed by application and define method to check validity to enter the game.

B. Requirements.

Hardware:

- 2 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality

Software:

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- JDK 8
- Android Studio IDE (Minimum 3.5)

C. Resources.

Documents:

- Guide

Supplement files:

- -

Test code:

- TestB2AdvancedWidgetsKT041.java

D. Task Description.

Student start to define fields and make method for validation.

E. Specification.

1. Open task B2.03 (ColorGame project) that already test passed.
2. Open “MyActivity.kt” file, to start programming in activity.

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

}

fun openGame(v: View?) {

}

fun startGame(v: View?) {

}

fun submitColor(v: View) {

}
```

3. Declare some fields in MyActivity class, with this description.

Name	Data type	Modifiers access
timer	TextView	-
clrText	TextView	-
scoreText	TextView	-
passwd	EditText	-
submit	Button	-
start	Button	-
accessbox	ViewGroup	-
colorbox	ViewGroup	-
buttonbox1	ViewGroup	-
buttonbox2	ViewGroup	-
scorebox	ViewGroup	-
progressbox	ViewGroup	-
progress	ProgressBar	-
isMinus	Switch	-

How to declare?

```
class MyActivity : AppCompatActivity() {
    <Modifier> <Field name>: <Data type>? = null;

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

    fun openGame(v: View?) {
    }

    fun startGame(v: View?) {
    }

    fun submitColor(v: View) {
    }
}
```

Put here

4. In the "onCreate" method, assign each declared field with related view resource. Refer on below

Name	Data type	Resource name
timer	TextView	timerText
clrText	TextView	clrText
scoreText	TextView	scoreText
passwd	EditText	appCode
submit	Button	submitBtn
start	Button	startBtn
accessbox	ViewGroup	accessBox
colorbox	ViewGroup	colorBox
buttonbox1	ViewGroup	buttonBox1
buttonbox2	ViewGroup	buttonBox2
scorebox	ViewGroup	scorebox
progressbox	ViewGroup	progressBox
progress	ProgressBar	progressScore
isMinus	Switch	isMinus

How to declare?

```
override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_layout)
    }
    <Field name> = findViewById<View>(R.id.<Resource name>) as <Data type>
    .
}
```

Put here

5. In “openGame” method and make validation with compare String “keyword” (in string resource) with String from text in passwd (in EditText).

   String “keyword” can access with this way.

   ```
   val keyword = getString(R.string.keyword)
   ```

   To access string in EditText use this way.

   ```
   val pass = passwd!!.text.toString()
   ```

   Compare both of them, if not same show the wrong Toast like this.

   ```
   Toast.makeText(applicationContext, "Password is wrong",
   Toast.LENGTH_LONG).show()
   ```

   If both string is same (valid), do this ways:
   - Set “passwd” and “submit” visibility become INVISIBLE, like below

     ```
     <fieldname>!!.visibility = View.VISIBLE
     ```

   - Set “accessbox”, “colorbox”, “buttonbox1”, “buttonbox2”, “scorebox”, and “progressbox” visibility become VISIBLE, like below

     ```
     <fieldname>!!.visibility = View.VISIBLE
     ```

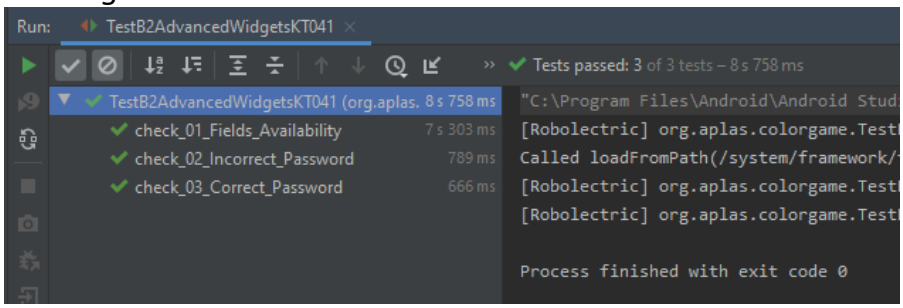
   - Show correct Toast

     ```
     Toast.makeText(applicationContext, "Login Success",
     Toast.LENGTH_LONG).show()
     ```

F. Testing.

1. Copy “TestB2AdvancedWidgetsKT041.java” file to “org.aplas.colorgame (test)” folder.
2. Right click on the “TestB2AdvancedWidgetsKT041.java” file then choose Run ‘TestB2AdvancedWidgetsKT041’ and click it. It may take long time to execute.

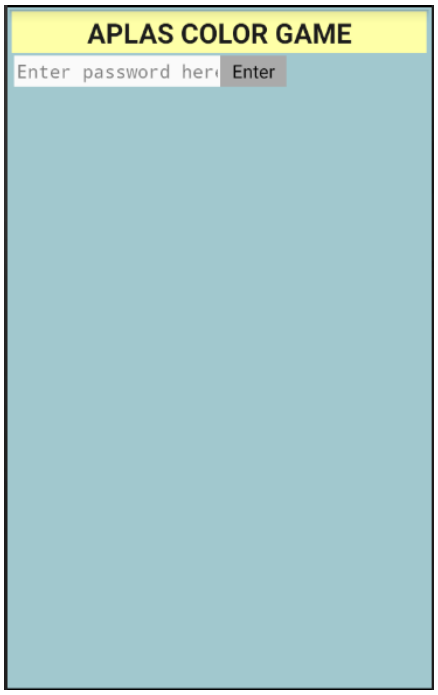
3. Get the result of your task. If passed you will get green check like below. If the test failed, you will get orange check get the messages and you must check your work again.



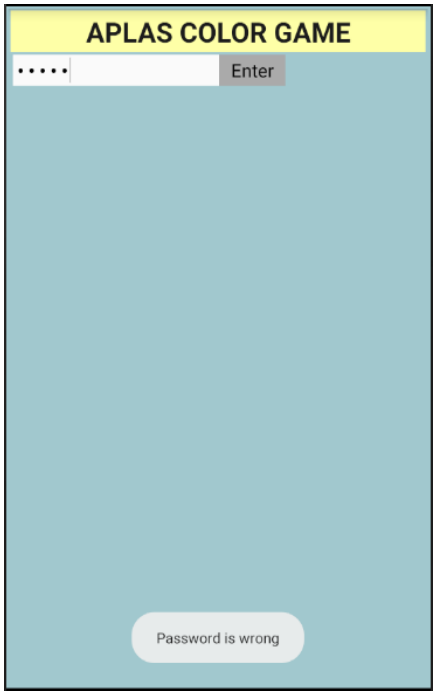
You have to try until get all green checkes and continue to the next task.

Run the App

If you have passed the test, you can run your application with this result.



First, enter password with “123” or “abc”, then press the “Enter” button.



Then, enter the correct password with “quiz@123”, then press the “Enter” button. This display will be shown.