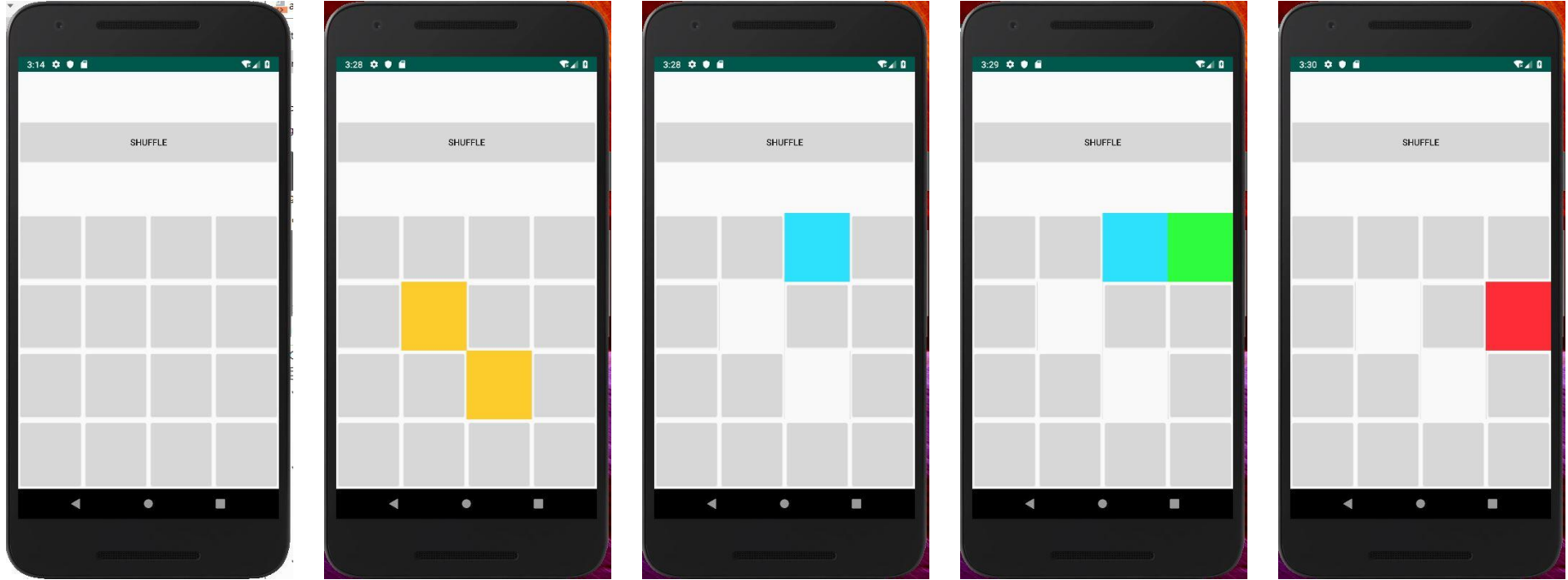


Color Match

An app to kill time

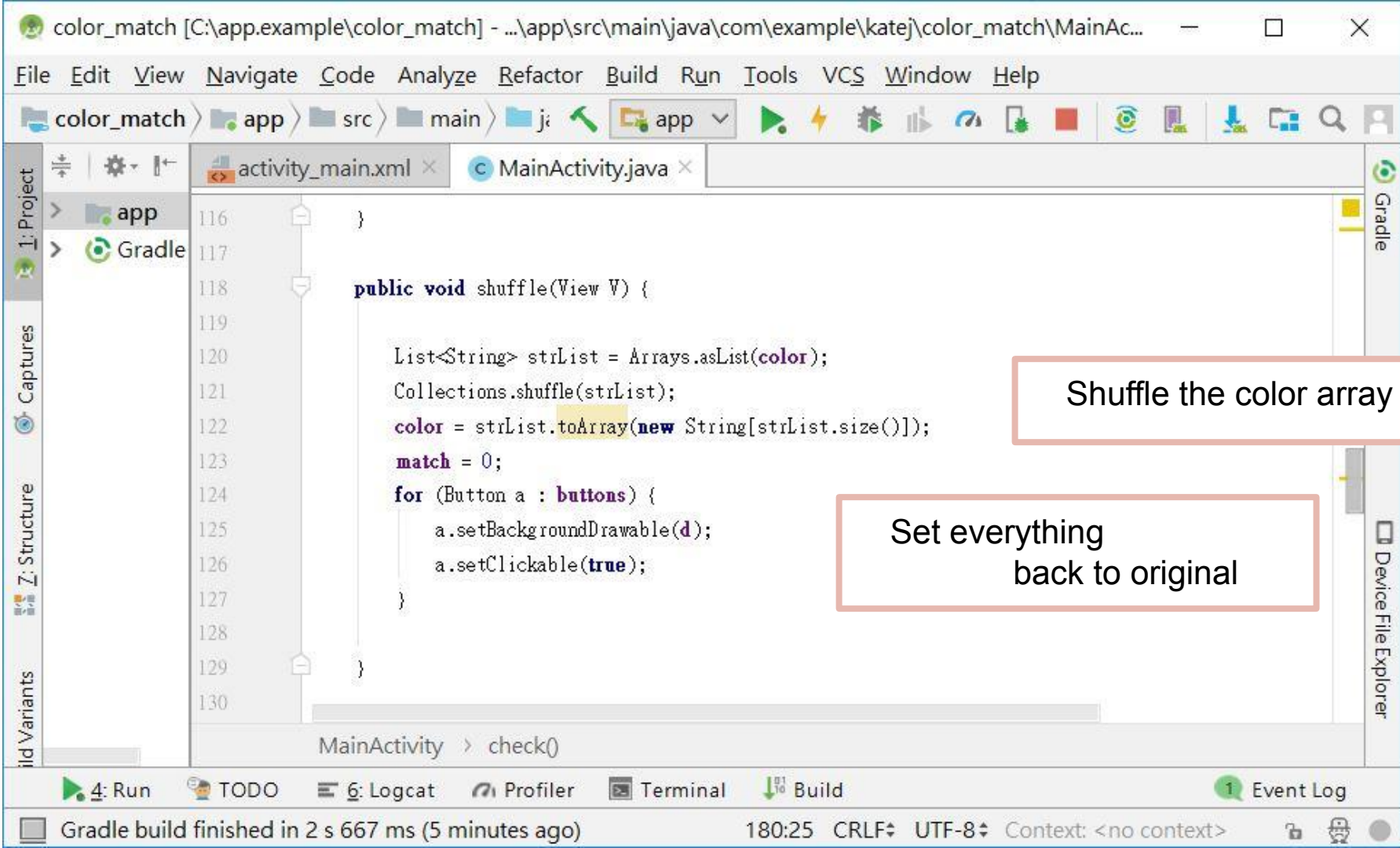
1.shuffle 2.select 3.check(match?) 4. hide or whiten

[illegible]

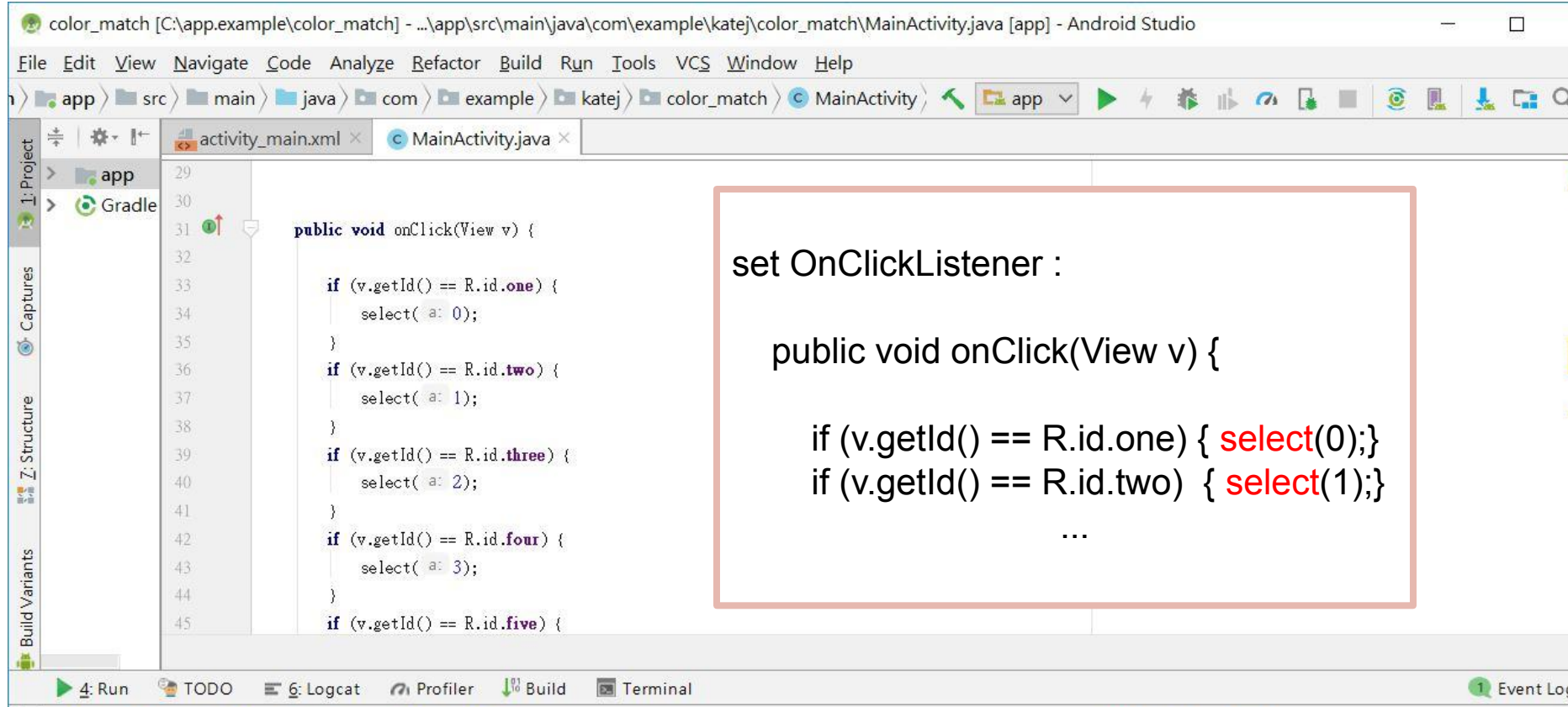
step 1 : shuffle

```
String[] color =
```

```
    new String[]{ "R", "R", "R", "R",  
                  "G", "G", "G", "G",  
                  "B", "B", "B", "B",  
                  "O", "O", "O", "O"};
```

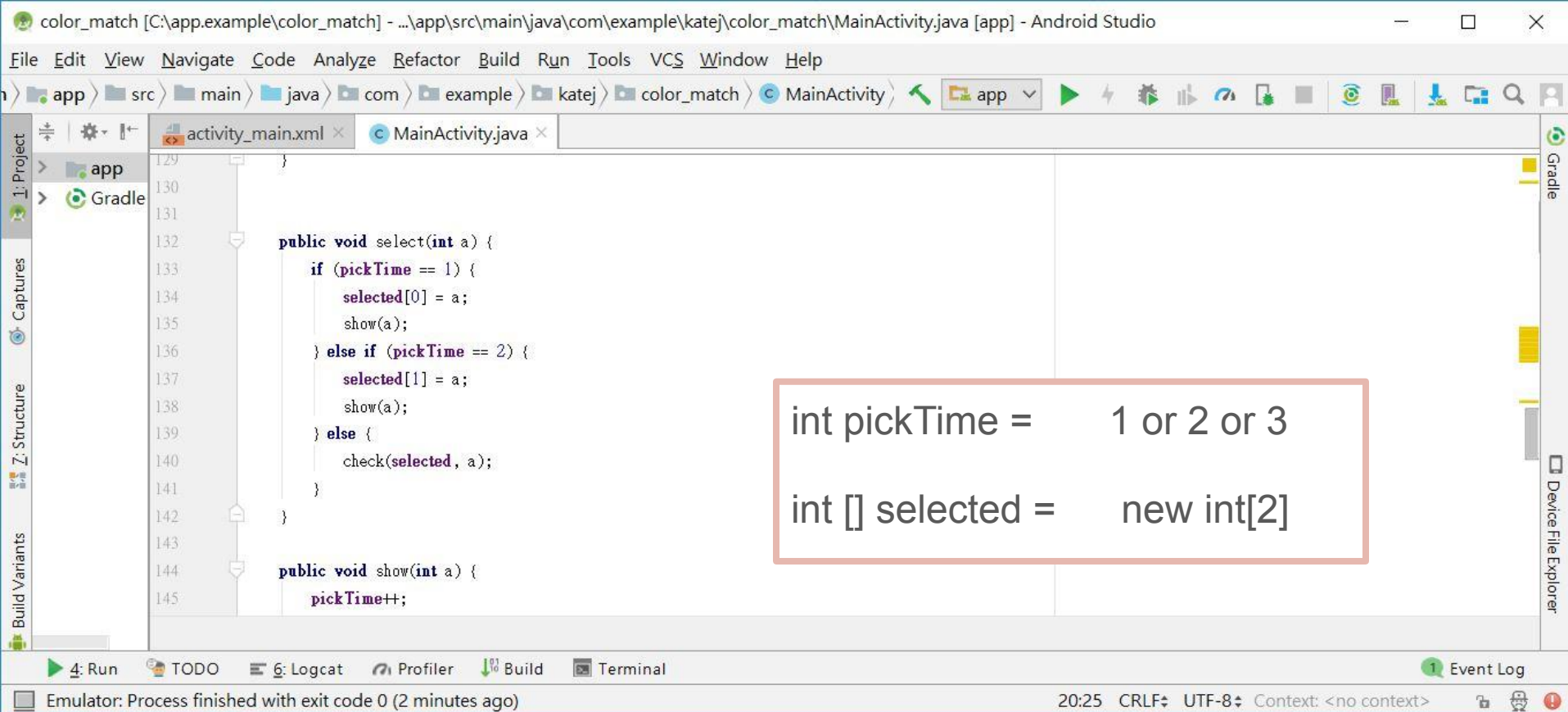


step 1.5 : on click to select



step 2 : select to show / check

```
public void select(int a) {  
    if (pickTime == 1) { selected[0] = a; show(a);  
    } else if (pickTime == 2) { selected[1] = a; show(a);  
    } else { check(selected, a);  
    }  
}
```



step 2.5 : show

```
public void show(int a) {
```

```
    pickTime++;
```

```
    switch (color[a]) { case "R":
```

```
        buttons[a].setBackgroundColor(Color.rgb(252, 45, 55)); break;
```

```
        case "G" { ...} ; case "B" { ...} ; case "O" { ...} ;
```

```
    } //end of switch
```

```
} //end of show
```

When pick time != 3

step 3 : check

When pick time = 3

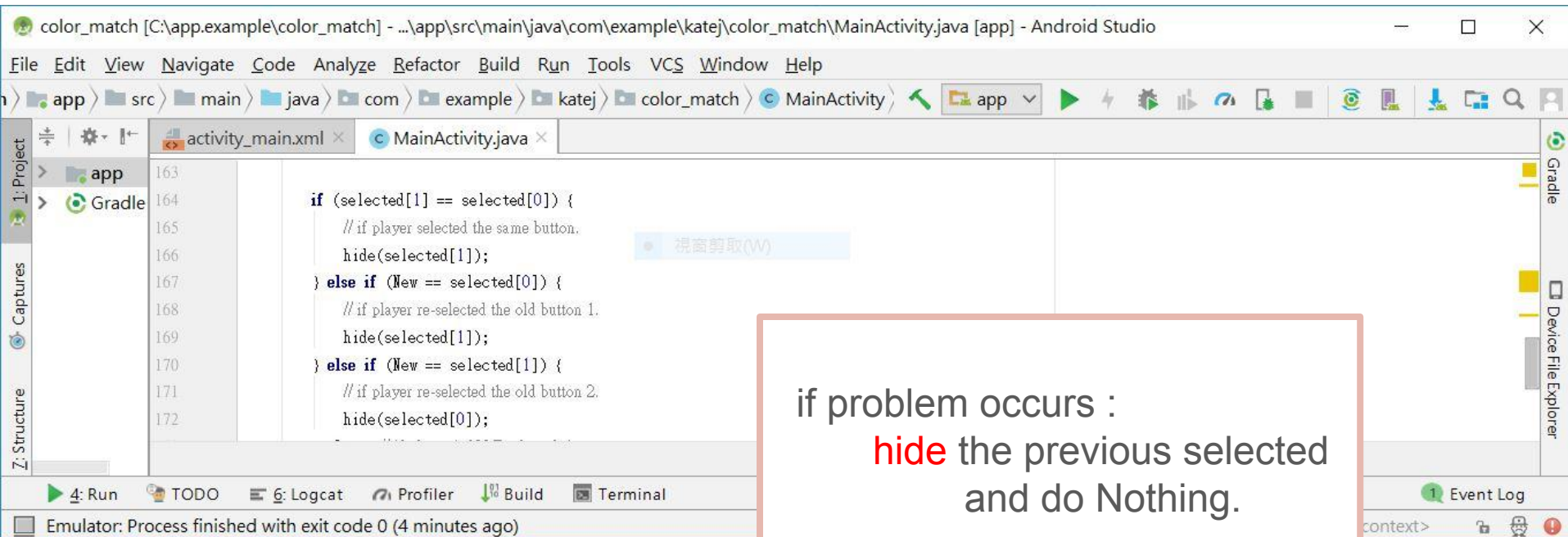
problem 1 : player select the same button

> color match but it **whiten itself**

problem 2 : player select the previous button

> color match but it whiten **only one of selected** button

Solution :



step 3 to 4 :

check to hide / whiten

Action 1 : colour mach with no problem

> whiten and set clickable False

Action 2 : colour not mach with no problem

> hide previous two buttons

color_match [C:\app.example\color_match] - ...\app\src\main\java\com\example\katej\color_match\MainActivity.java [app] - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

h > app > src > main > java > com > example > katej > color_match > MainActivity > app >

activity_main.xml x MainActivity.java x

1: Project
app
Gradle
Captures
Z: Structure
Build Variants

```
173     } else { // if player is NOT selected the same
174         if (color[selected[0]].equals(color[selected[1]])) {
175             // and the colors matched
176             buttons[selected[0]].setBackgroundColor(Color.rgb( red: 249, green: 249, blue: 249));
177             buttons[selected[1]].setBackgroundColor(Color.rgb( red: 249, green: 249, blue: 249));
178             buttons[selected[0]].setClickable(false);
179             buttons[selected[1]].setClickable(false);
180             match++;
181             if (match == 7) {
182                 Toast.makeText( context: this, text: "All Clear, click <SHUFFLE> to restart", Toast.LENGTH_LONG).show();
183             }
184         } else {
185             // but the color don't match
186             hide(selected[0]);
187             hide(selected[1]);
188         }
189     }
190 }
```

Action 1

Action 2

4: Run TODO 6: Logcat Profiler Build Terminal

Emulator: Process finished with exit code 0 (5 minutes ago)

21:1 CRLF UTF-8 Context: <no context>

Event Log

Device File Explorer

step 3 to 1.5 :

from check back to select

no matter what Action resulted :

> **reset** “New” into select 0

pickTime = 3
back to pickTime =1

if the game is clear, no more pickTime = 3:

> Action 1 will pop out a **toast** to remind player

Toast : Click <Shuffle> to Restart

```
// reset New into select 0
```

```
pickTime = 1;
```

```
select(New);
```

```
//enf of check
```

Reset

Device File Explorer

MainActivity.java [app] - Android Studio

Activity

app

Logcat

Profiler

Build

Terminal

Event Log

or

shed with... (a minute ago) 21:1 CRLF UTF-8 Context: <no con

Captures

Structure

variants

175

176

177

178

179

180

181

182

183

184

185

186

187

188

```
// and the colors matched
```

```
buttons[selected[0]].setBackgroundColor(Color.rgb( red: 249, green: 249, blue: 249));
```

```
buttons[selected[1]].setBackgroundColor(Color.rgb( red: 249, gre
```

```
buttons[selected[0]].setClickable(false);
```

```
buttons[selected[1]].setClickable(false);
```

```
match++;
```

```
if (match == 7) {
```

```
    Toast.makeText( context: this, text: "All Clear, click <SHUFFLE> to restart", Toast.LENGTH_LONG).show();
```

```
}
```

```
} else {
```

```
    // but the color don't match
```

```
    hide(selected[0]);
```

```
    hide(selected[1]);
```

Toast

step 4 : **hide** or whiten

```
public void hide(int a) {
```

```
    buttons[a].setBackgroundDrawable(d);
```

```
    buttons[a].setClickable(true);
```

```
}
```

```
Drawable d;
```

```
> onCreate{ d =  
    buttons[0].getBackground(); }
```

```
Button buttons[] = new Button[16];
```

```
> onCreate{ buttons[0] =  
    (Button) findViewById(R.id.one);  
    buttons[1] ..[2]....[3] ..... }
```

Hide

On Create (Part 1) :

```
public void hide(int a) {  
    buttons[a].setBackgroundDrawable(d);  
    buttons[a].setClickable(true);  
}
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    buttons[0] = (Button) findViewById(R.id.one);
```

```
    buttons[1] = (Button) findViewById(R.id.two);
```

```
    buttons[2] = (Button) findViewById(R.id.three);
```

```
    buttons[3] = (Button) findViewById(R.id.four);
```

```
    buttons[4] = (Button) findViewById(R.id.five);
```

```
    buttons[5] = (Button) findViewById(R.id.six);
```

```
    buttons[6] = (Button) findViewById(R.id.seven);
```

```
    buttons[7] = (Button) findViewById(R.id.eight);
```

```
    buttons[8] = (Button) findViewById(R.id.nine);
```

```
    buttons[9] = (Button) findViewById(R.id.ten);
```

```
    buttons[10] = (Button) findViewById(R.id.eleven);
```

```
    buttons[11] = (Button) findViewById(R.id.twelve);
```

```
    buttons[12] = (Button) findViewById(R.id.thirteen);
```

```
    buttons[13] = (Button) findViewById(R.id.fourteen);
```

```
    buttons[14] = (Button) findViewById(R.id.fifteen);
```



```
98 buttons[10] = (Button) findViewById(R.id.eleven);
99 buttons[11] = (Button) findViewById(R.id.twelve);
100 buttons[12] = (Button) findViewById(R.id.thirteen);
101 buttons[13] = (Button) findViewById(R.id.fourteen);
102 buttons[14] = (Button) findViewById(R.id.fifteen);
103 buttons[15] = (Button) findViewById(R.id.sixteen);
104
105 for (Button b : buttons) {
106     b.setOnClickListener(this);
107 }
108
109 d = buttons[0].getBackground();
110
111 color = new String[]{"R", "R", "R", "R", "G", "G", "G", "G", "B", "B", "B", "B", "O", "O", "O", "O"};
112 pickTime = 1;
113 selected = new int[]{16, 17};
114
115 match = 0;
116 // end of on create
117
118 public void shuffle(View V) {
```

On Create (Part 2) :

step 4 : hide or **whiten**

```
buttons[selected[0]].setBackgroundColor(Color.rgb(249, 249, 249));
```

```
buttons[selected[1]].setBackgroundColor(Color.rgb(249, 249, 249));
```

```
buttons[selected[0]].setClickable(false);
```

```
buttons[selected[1]].setClickable(false);
```

Check result:
Action 1

END