

# 임베디드 시스템 실습 lab12

2016310936 우승민

이번 exercise 는 Android studio 의 notification 을 활용하는 것입니다.

우선 layout 의 code 를 보면

```
activity_main.xml x MainActivity.java x AndroidManifest.xml x
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical">
6
7   <EditText
8     android:id="@+id/title"
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    android:ems="10"
12    android:inputType="text" />
13
14   <EditText
15     android:id="@+id/contents"
16     android:layout_width="match_parent"
17     android:layout_height="wrap_content"
18     android:ems="10"
19     android:inputType="text" />
20
```

처음에 title 과 content 의 내용을 입력 받기  
위해 EditText 를 2 번 사용하였고,

```
activity_main.xml x MainActivity.java x AndroidManifest.xml x
22
23   <Button
24     android:id="@+id/Notify"
25     android:layout_width="wrap_content"
26     android:layout_height="wrap_content"
27     android:text="Notify"
28     tools:layout_editor_absoluteX="138dp"
29     tools:layout_editor_absoluteY="445dp" />
30
31   <Button
32     android:id="@+id/Update"
33     android:layout_width="wrap_content"
34     android:layout_height="wrap_content"
35     android:text="Update"
36     tools:layout_editor_absoluteX="135dp"
37     tools:layout_editor_absoluteY="536dp" />
38
39   <Button
40     android:id="@+id/RemoveAll"
41     android:layout_width="wrap_content"
42     android:layout_height="wrap_content"
43     android:text="RemoveAll"
44     tools:layout_editor_absoluteX="135dp"
45     tools:layout_editor_absoluteY="629dp" />
46
47
48 </LinearLayout>
```

그 후 notify, update, remove all 의 동작을 수행하는  
Button 3 개를 추가해주었습니다.

다음으로 main activity code 를 보면, 우선 notification 에서 사용될 title, content 및 각 notification 의 ID 를 check 할 channel\_num 을 선언해주었습니다.



```
1 package com.example.week12;
2
3 import ...
16
17 public class MainActivity extends AppCompatActivity {
18     private final static String CHANNEL_ID = "channel_id";
19     int channel_num;
20     EditText title1;
21     EditText content1;
22     String title;
23     String content;
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_main);
29
30         Button b1 = (Button) findViewById(R.id.Notify);
31         Button b2 = (Button) findViewById(R.id.Update);
32         Button b3 = (Button) findViewById(R.id.RemoveAll);
33
34
35
36         channel_num = -1;
37
38         createNotificationChannel();
```

여기서 title1 과 content1 은 layout 에서 입력 받은 EditText 를 그대로 전달 받는 역할이고 후에 title 과 content 로 String 형태로 변환하여 옮겨줍니다.

처음 어플이 실행될 때 channel\_num 을 -1 로 설정하고 NotificationChannel 을 만들어줍니다.

```
activity_main.xml x MainActivity.java x AndroidManifest.xml x
40 b1.setOnClickListener(new View.OnClickListener(){
41     @Override
42     public void onClick(View v) {
43         channel_num = channel_num + 1;
44         title1 = (EditText) findViewById(R.id.title);
45         content1 = (EditText) findViewById(R.id.contents);
46         title = title1.getText().toString();
47         content = content1.getText().toString();
48         addNotification();
49     }
50 });
51
52 b2.setOnClickListener(new View.OnClickListener(){
53     @Override
54     public void onClick(View v) {
55         title1 = (EditText) findViewById(R.id.title);
56         content1 = (EditText) findViewById(R.id.contents);
57         title = title1.getText().toString();
58         content = content1.getText().toString();
59         addNotification();
60     }
61 });
62
63
64 b3.setOnClickListener(new View.OnClickListener(){
65     @Override
66     public void onClick(View v) {
67         removeNotification();
68     }
69 });
70
71 }
```

이후 button 에 따라 동작을 수행하였습니다. 1 번, 2 번 button 은 title 과 content 를 입력 받은 내용으로 변경해준 후, notification 을 추가하는 동작을 수행합니다.

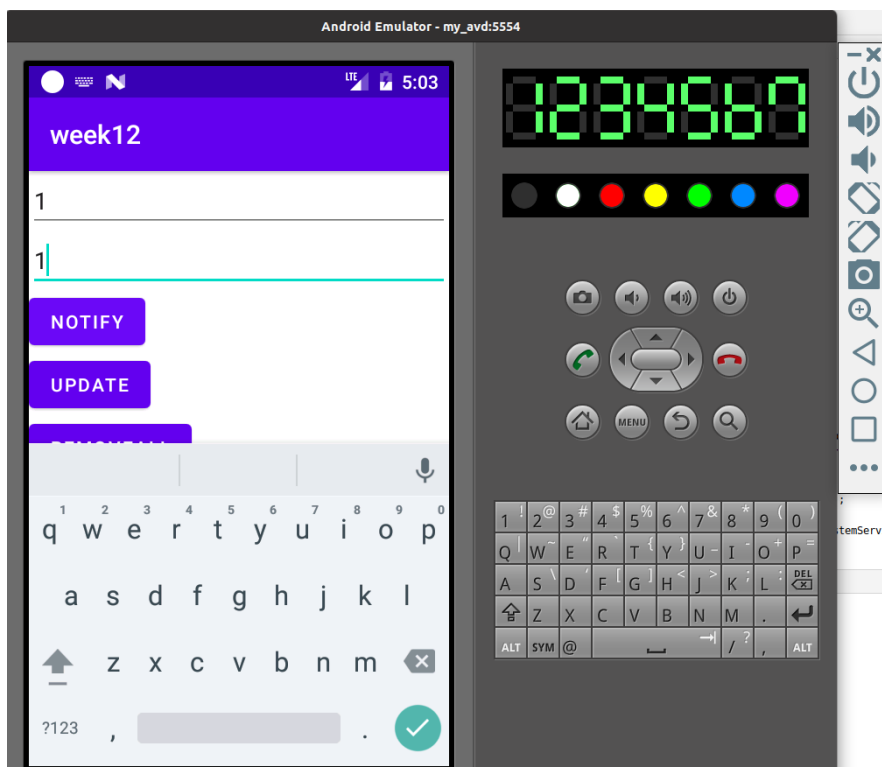
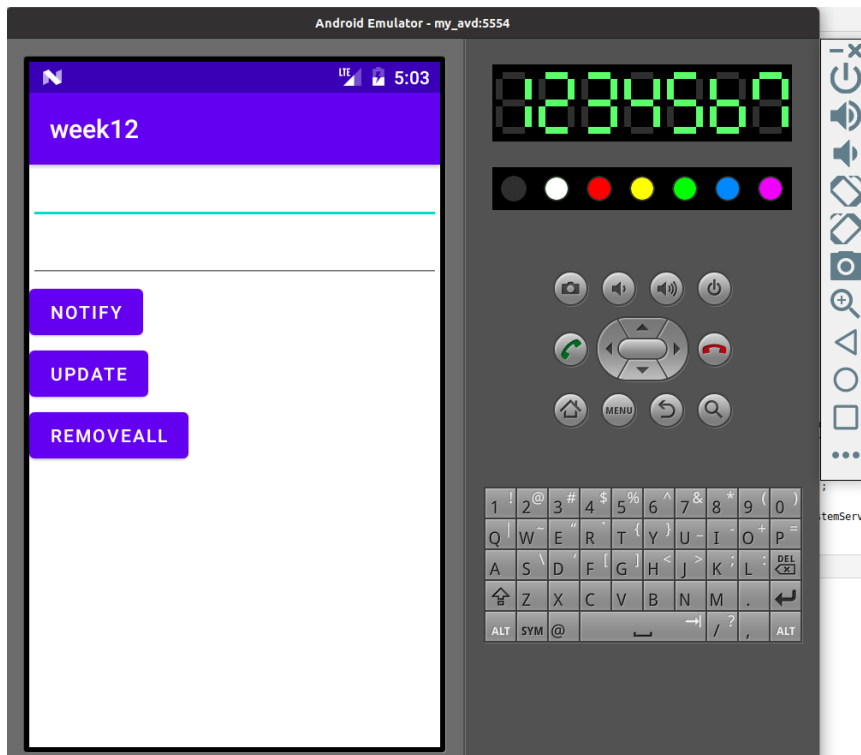
다만 차이점은 1 번은 새로 notification 을 추가하는 것이기 때문에 channel\_num 을 하나씩 증가해주고 2 번 button 은 최근 notification 을 수정하는 것이므로 channel\_num 을 변경하지 않습니다.

3 번 button 은 모든 notification 을 지우는 동작을 수행합니다.

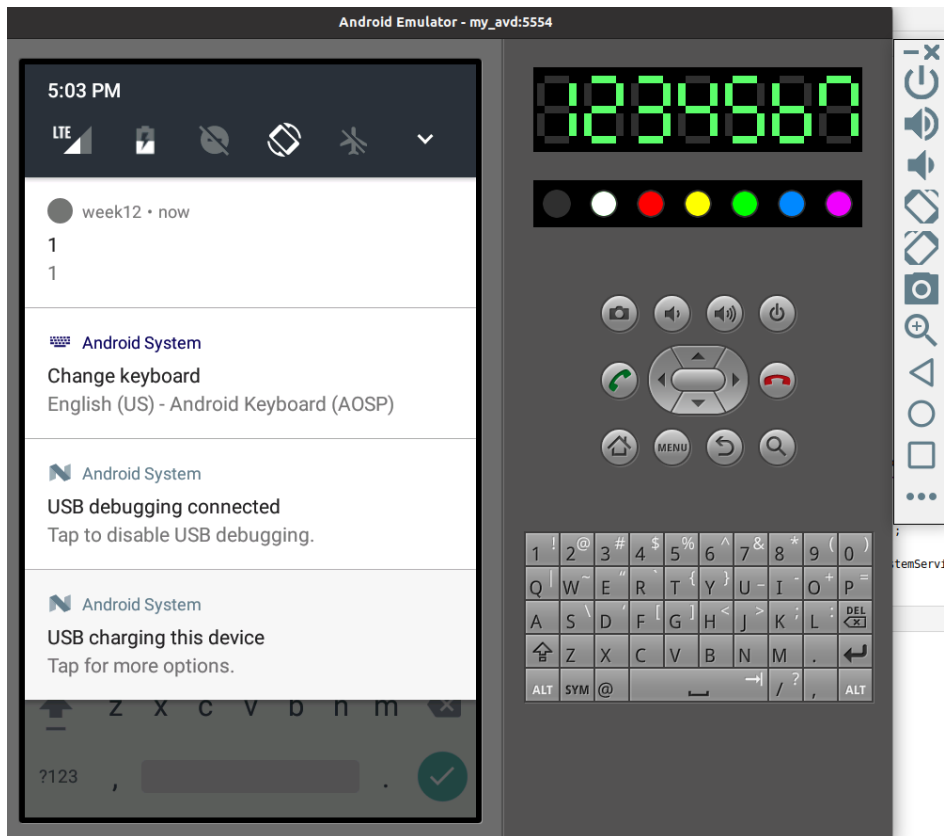
```
activity_main.xml x MainActivity.java x AndroidManifest.xml x
73 private void addNotification(){
74
75     NotificationCompat.Builder builder = new NotificationCompat.Builder( context: this, CHANNEL_ID)
76         .setSmallIcon(R.drawable.notification_icon_background)
77         .setContentTitle(title)
78         .setContentText(content)
79         .setPriority(NotificationCompat.PRIORITY_DEFAULT);
80
81     NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
82     manager.notify(channel_num, builder.build());
83
84 }
85
86 private void removeNotification(){
87     NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
88     manager.cancelAll();
89     channel_num = - 1;
90 }
91
92 private void createNotificationChannel(){
93     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
94         CharSequence name = title;
95         String description = content;
96         int importance = NotificationManager.IMPORTANCE_DEFAULT;
97         NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
98         channel.setDescription(description);
99         NotificationManager notificationManager = getSystemService(NotificationManager.class);
100         notificationManager.createNotificationChannel(channel);
101     }
102 }
103 }
```

각 함수들은 다음과 같이 정의 되어있습니다.

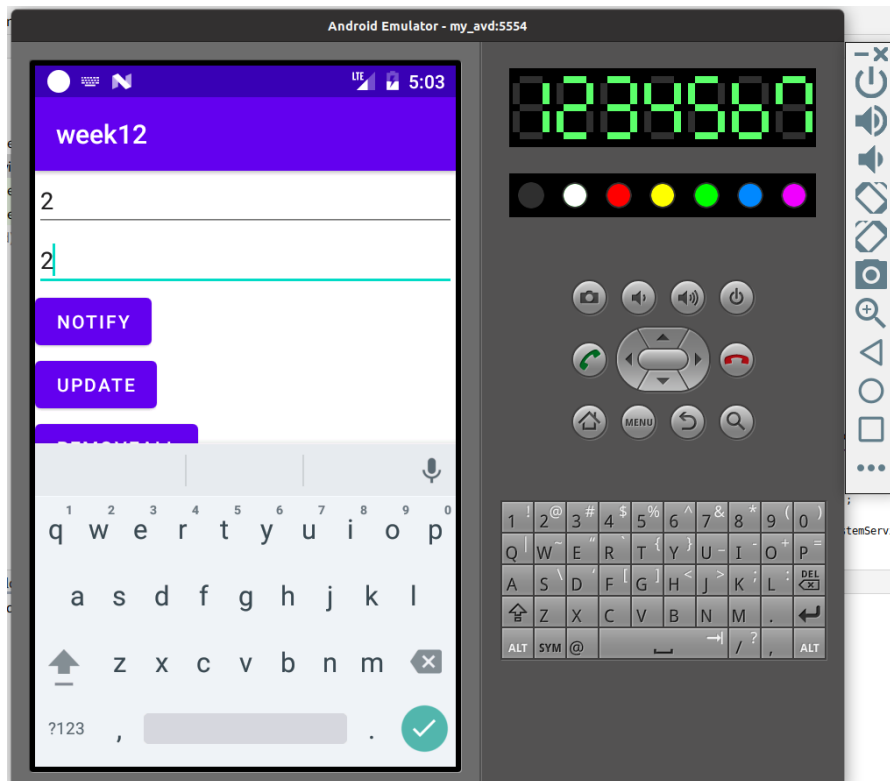
실행 화면입니다.



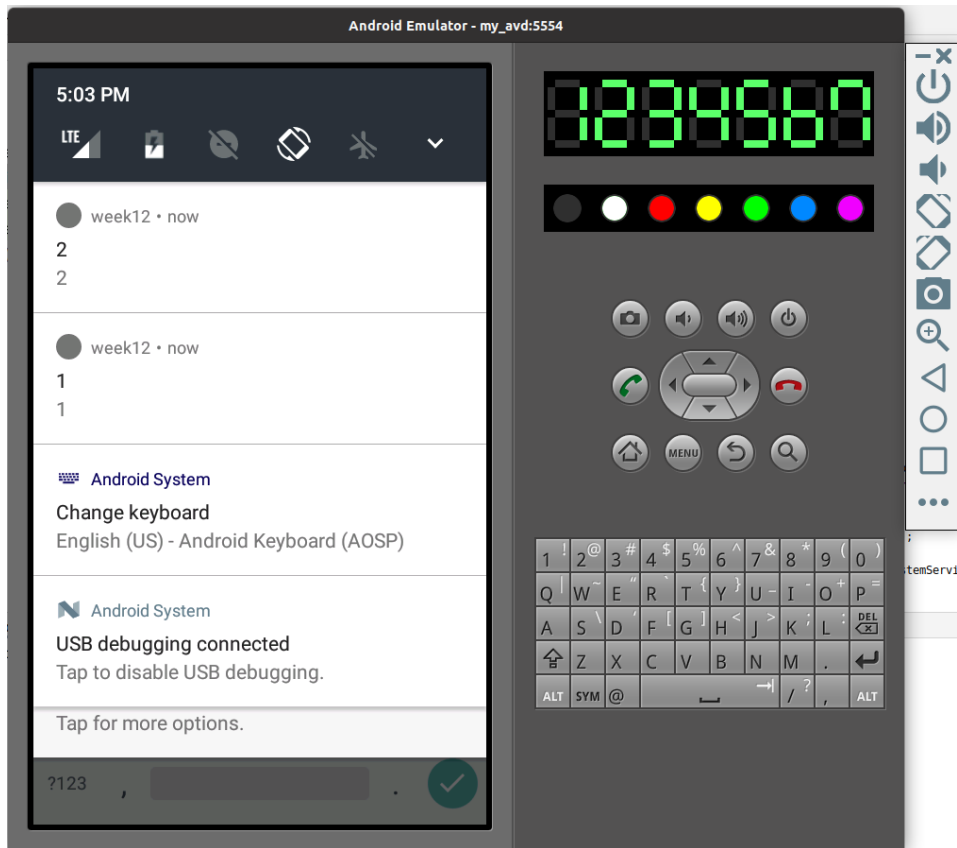
title 과 content 에 1 을 입력한 후 notify 를 누르면



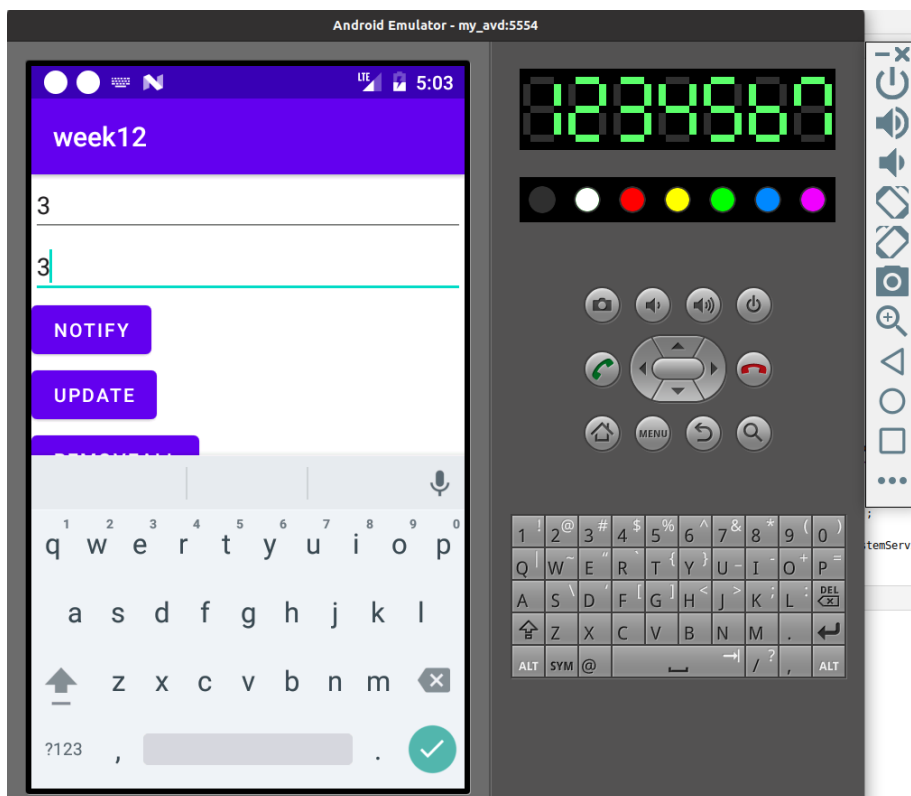
입력해준 title 과 content 에 맞추어 알림이 오는 것을 확인할 수 있습니다.



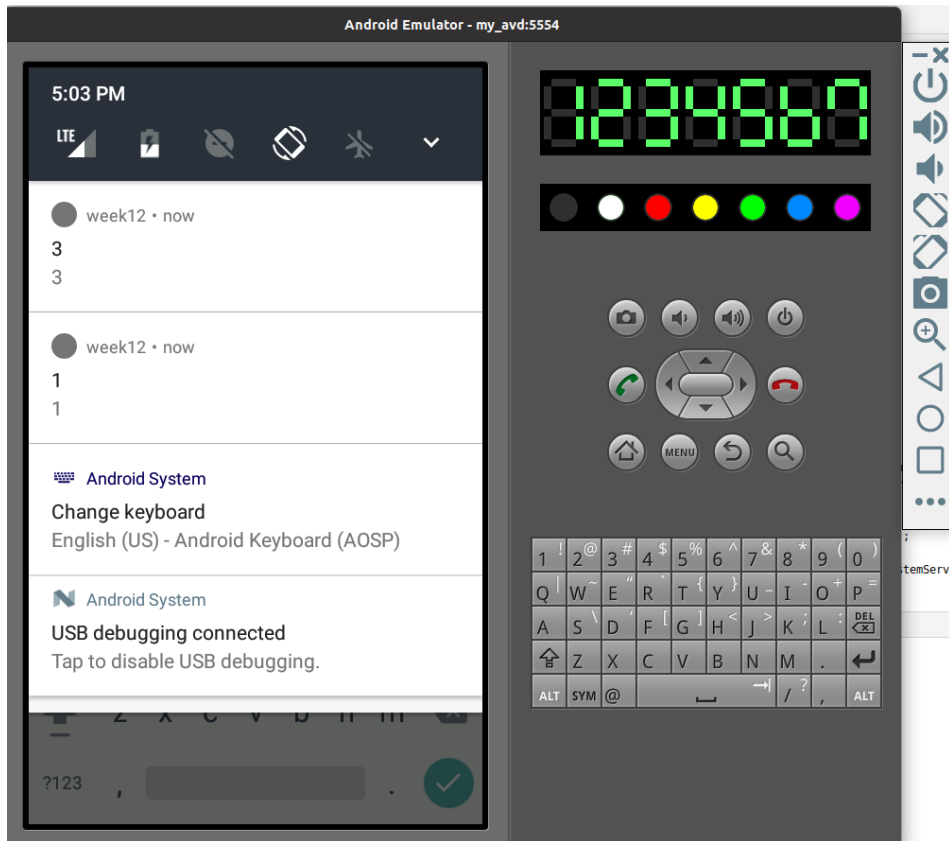
title 과 content 를 2 로 수정한 후 동일하게 notify 를 누르면



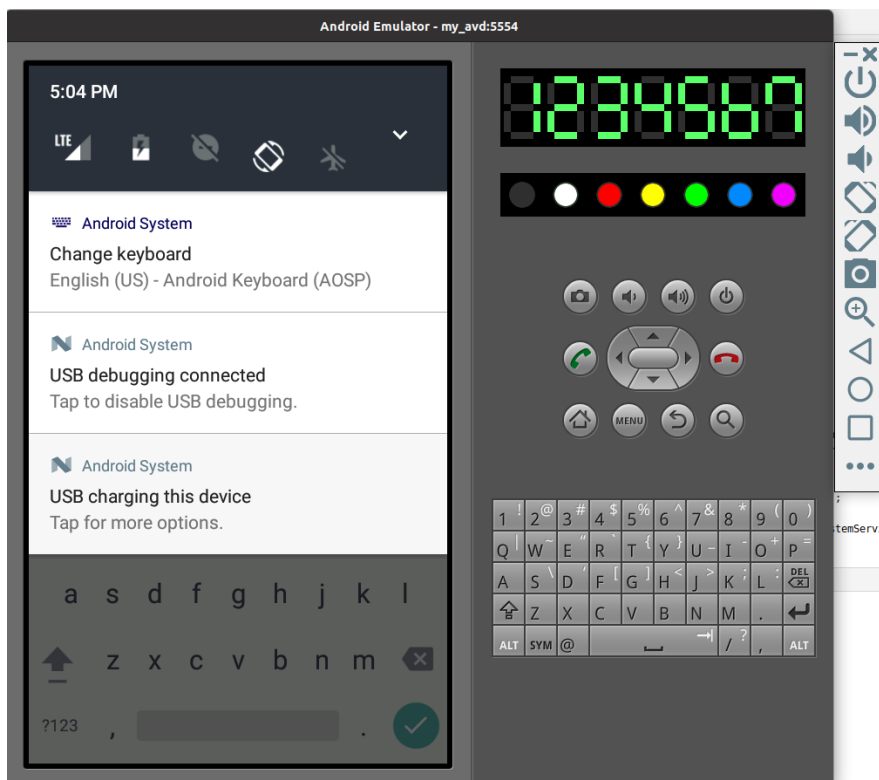
새로운 알림이 추가된 것을 볼 수 있습니다.



3 으로 수정한 후 update 를 누르면



가장 최근에 온 알림 2가 3으로 수정된 것을 확인할 수 있습니다. 이후 remove all 버튼을 누르면



모든 알림이 사라진 것을 확인할 수 있습니다.



