액티비티에서 터치 이벤트와 키 이벤트를 직접 처리하려면, 이벤트 발생할 때 자동으로 호출되는 함수만 액티비티 내에 재정의하면 된다.

● 터치 이벤트(onTouchEvent()): 액티비티에 보이는 내용을 사용자가 손가락으로 조작하는 일과, 액티비티 화면을 터치해서 손가락으로 상하좌우 어던 방향으로 밀었는지를 알아낼 때는 터치 이벤트로 처리해서 구현한다. 터치 이벤트가 발생할 때 콜백 함수를 액티비티 내에 정의하는 것만으로도 이벤트 처리가 가능하다.

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    return super.onTouchEvent(event);
}
```

onTouchEvent 메서드가 호출되는 터치 이벤트는 3가지가 있으며, 이 메서드의 매개변수로 식별해서 사용할 수 있다 (event.getAction()).

MotionEvent.ACTION_DOWN	화면에 터치된 순간의 이벤트
MotionEvent.ACTION_UP	터치를 떼는 순간의 이벤트
MotionEvent.ACTION_MOVE	터치한 후 이동하는 순간의 이벤트
MotionEvent.ACTION_CANCEL	터치가 취소되는 순간의 이벤트

MotionEvent 객체의 메서드

event.getAction()	발생한 이벤트 종류를 감지하여 리턴해준다
event.getX()	이벤트가 발생한 뷰 내에서의 좌표값을 반환한다
event.getY()	
event.getRawX()	화면 내에서 이벤트가 발생한 지점의 좌표값을 반환한다
event.getRawY()	

```
float initX;
@Override
public boolean onTouchEvent(MotionEvent event) {
    // 만약 화면에 터치된 순간의 이벤트가 발생하면,
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        // 화면 내에서 이벤트가 발생한 지점 X좌표값을 initX에 할당한다.
        initX = event.getRawX();
    }
}
```

● 키 이벤트(onKeyDown(), onBackPressed()): 사용자가 안드로이드 스마트폰의 키를 눌렀을 때 이벤트 처리를 해 주는 것. 안드로이드 화면에서 키보드가 올라오고 키를 눌러 글을 입력한다면 그 키보드는 소프트 키보드(soft keyboard)이고, 이것은 키 이벤트로 처리할 수 없다. 만일 소프트 키보드가 아닌 하드웨어 키보드가 제공되는 스마트폰이라면 해당 하드웨어 키보드의 키가 눌렸을 때 키 이벤트로 처리할 수는 있다. 하지만 안드로이드 스마트폰 대부분은 소프트 키보드를 사용한다. 때문에 키 이벤트 처리는 키보드 이외의 키 이벤트를 처리해주기 위해 자주 이용된다. 안드로이드 시스템 수준에서 제공하는 버튼들이 있는데, 이 버튼을 누르는 순간의 이벤트 처리를 위해 키 이벤트를 이용한다. 하지만 홈과 전원, 오버뷰 버튼은 일반 애플리케이션에서 이벤트 처리로 제어할 수 없기 대문에, 뒤로가기 버튼 처리가 대부분이다.

뒤로가기 버튼은 기본적으로 액티비티 화면을 전환하는 버튼인데, 개발자가 이벤트 처리를 통해 다르게 작동시키고 싶을 때 키 이벤트를 이용한다. 예를 들어, 뒤로가기 버튼을 눌렀을 때, "정말 종료하시겠습니까?" 또는 "종료하려면 한 번 더 누르세요."라는 메시지를 띄우는 작업을 키 이벤트로 처리해서 작성한다. 액티비티 내에서 키 이벤트를 처리하려면, 키 이벤트가 발생할 때 호출되는 이벤트 함수를 액티비티 내에 정의하기만 하면 된다. 매개 변수로 keyCode 값이 전달되기 때문에, 어느 버튼을 누든 건지 식별할 수 있다.

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    // 만약 누른 키가 뒤로가기 버튼이라면,
    if (keyCode == keyEvent.KEYCODE_BACK) {
        ....
    }
    return super.onKeyDown(keyCode, event);
}
```

onBackPressed() 함수는 뒤로가기 버튼 제어만을 목적으로 만들어졌으므로, 다른 키 이벤트는 처리할 수 없다.

```
@Override
public void onBackPressed() {
    super.onBackPressed();
}
```

```
package com.example.practiceeventhandling;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.widget.CompoundButton;
import android.widget.Toast;

import com.example.practiceeventhandling.databinding.ActivityMainBinding;

// 엑티비티 클래스 자체를 이벤트 핸들러로 만들기 위해 다음 두 개의 인터페이스를 클래스 선언 부분
// 상속 받고, 인터페이스의 추상 함수를 재정의
public class MainActivity extends AppCompatActivity implements View.OnClickListener, Com
    ActivityMainBinding binding;
```

```java
// onCreate() 함수에 이벤트 등록
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityMainBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    binding.bellName.setOnClickListener(this);
    binding.label.setOnClickListener(this);
    binding.repeatCheck.setOnCheckedChangeListener(this);
    binding.vibrate.setOnCheckedChangeListener(this);
    binding.onOff.setOnCheckedChangeListener(this);

}

private void showToast(String message){
    Toast toast=Toast.makeText(this, message, Toast.LENGTH_SHORT);
    toast.show();
}


@Override
public void onClick(View v) {
    if (v == binding.bellName) {
        showToast("bell text click event...");
    } else if (v == binding.label) {
        showToast("label text click event...");
    }
}

@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    if (buttonView == binding.repeatCheck) {
        showToast("repeat checkbox is " + isChecked);
    } else if (buttonView == binding.vibrate) {
        showToast("vibrate checkbox is " + isChecked);
    } else if (buttonView == binding.onOff) {
        showToast("switch is " + isChecked);
    }
}

// 터치 이벤트
double initX;
@Override
public boolean onTouchEvent(MotionEvent event) {
    // 만약 화면에 터치된 순간의 이벤트가 발생하면
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        // 화면 내에서 터치 이벤트가 발생한 지점의 좌표 X를 initX에 지정
        initX = event.getRawX();
    // 만약 터치를 떼는 순간의 이벤트가 발생하면
    } else if (event.getAction() == MotionEvent.ACTION_UP) {
        // initX와 현재 터치를 떼는 이벤트가 발생한 지점의 좌표 X의 값을 구해 diffX에 지정
        double diffX = initX - event.getRawX();

        // 만약 그 차이가 30 이상이면 왼쪽, -30 이하면 오른쪽으로 화면을 밀었다는 문구를 띄
        // 이 차이는 크면 커 질수록 확실하게 화면 끝까지 밀어야 문구가 출력됨.
        if (diffX > 30) {
            showToast("왼쪽으로 화면을 밀었습니다.");
        } else if(diffX < -30) {
```

```
                showToast("오른쪽으로 화면을 밀었습니다.");
            }
        }
        return true;
    }

    // 키 이벤트
    long initTime = 0L;
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        // 만약 눌러진 키 버튼이 뒤로가기 버튼이라면
        if (keyCode == KeyEvent.KEYCODE_BACK) {
            // 만약 initTime과 버튼을 누른 현재 시간이 3초 이상이라면 - back button을 누른 지
            if (System.currentTimeMillis() - initTime > 3000) {
                // 메세지 띄우기
                showToast("종료하려면 한 번 더 누르세요.");
                // 현재 시간을 initTime에 지정
                initTime = System.currentTimeMillis();
            } else {
                // 3초 이내에 BackButton이 두 번 눌린 경우 Activity 종료
                finish();
            }
            // true를 리턴하는 경우, 기존 시스템이 가지는 해당 키 이벤트에 대한 처리를
            // 무시하겠다는 의미이다.
            // 즉, Toast메시지만 띄우고 기존의 뒤로가기 버튼의 동작인 이전 액티비티 이동
            // 또는 앱 종료와 같은 기능을 무시하고자 할 경우 true로 리턴해 주면 된다.
            return true;
            // 만약 Toast메시지를 띄우고 기존의 동작을 수행하길 원한다면,
            // 이 리턴문을 삭제하면 된다.
        }
        // 뒤로가기 키를 제외한 나머지 키 이벤트에 대한 처리는 시스템에 맡긴다
        return super.onKeyDown(keyCode, event);
    }

    // 키 이벤트
//    long initTime = 0L;
//    @Override
//    public void onBackPressed() {
//        if (System.currentTimeMillis() - initTime > 3000) {
//            // 메세지 띄우기
//            showToast("종료하려면 한 번 더 누르세요.");
//            // 현재 시간을 initTime에 지정
//            initTime = System.currentTimeMillis();
//        } else {
//            // 3초 이내에 BackButton이 두 번 눌린 경우 Activity 종료
//            finish();
//        }
//    }


}
```

Resources: 깡쌤의 안드로이드 프로그래밍