

안드로이드 개발자를 위한 접근성 개발 가이드

Screen reader (Talkback)

삼성전자 김민욱



목차

- Talkback 소개 3
- 접근성을 높이기 위한 앱 개발 시 고려해야 할 점 9
- 구현 팁
- 나아 가야 할 방향
- Appendix

Talkback 소개(1/4)

TalkBack은 Android 기기에 포함된 Google 스크린 리더.
TalkBack을 사용하면 기기를 보지 않고도 제어할 수 있음.



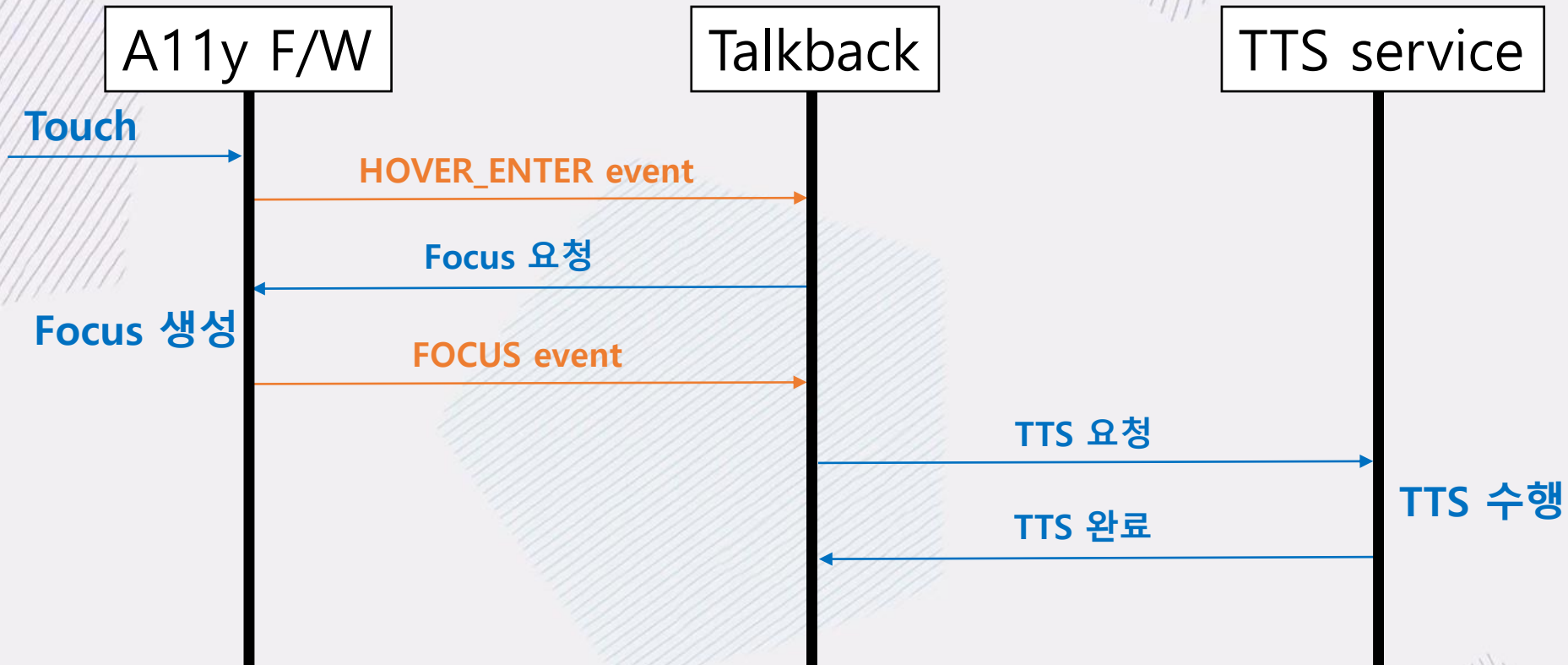
- 터치를 통해 화면을 탐색 시 **FOCUS** 효과 후 **TTS**(Text-To-Speech) 발생
- 진입이나 동작이 가능한 경우, **더블 탭**을 통해 수행 가능
- Scroll 시 **두 손가락 스와이프**(Swipe)을 통해 동작
- 여러 손가락에 대한 제스처 지원 및 커스텀 동작

기본 정보 - <https://support.google.com/accessibility/android/answer/6283677?hl=ko>

Talkback 소개(2/4)

App의 화면은 View 단위로 구성되며, Touch시 대상 View에서 발생하는 Event에 의해 Talkback에 전달 되어 Focus 및 TTS를 처리.

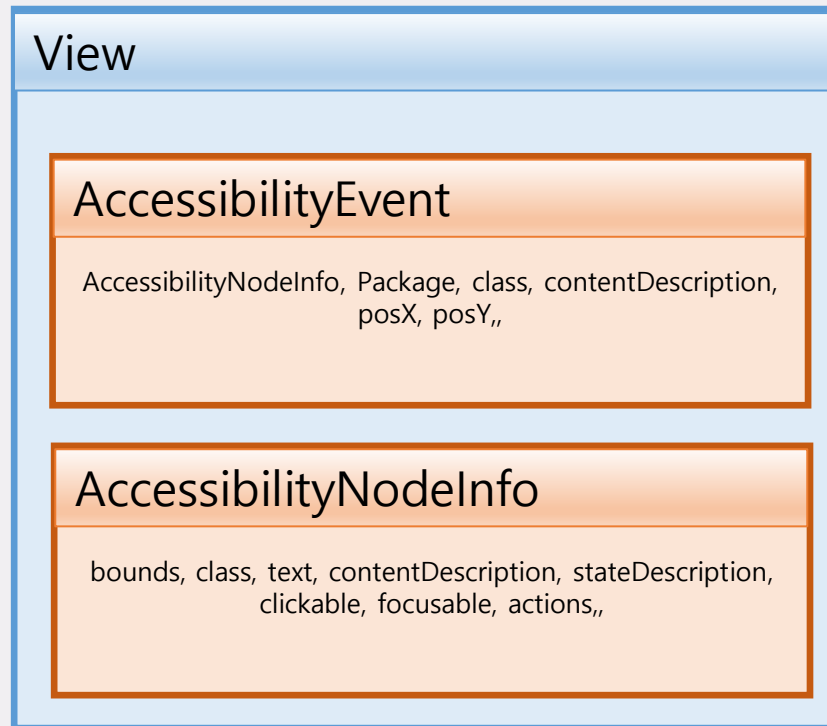
Talkback은 A11y F/W으로부터 전달되는 Event를 기반으로 동작.



Talkback 소개(3/4)

Click, Long click, Scroll 등 Action이 View에 발생 하면 AccessibilityEvent에 AccessibilityNodeInfo를 넣어 Talkback으로 전달 됨.

Action



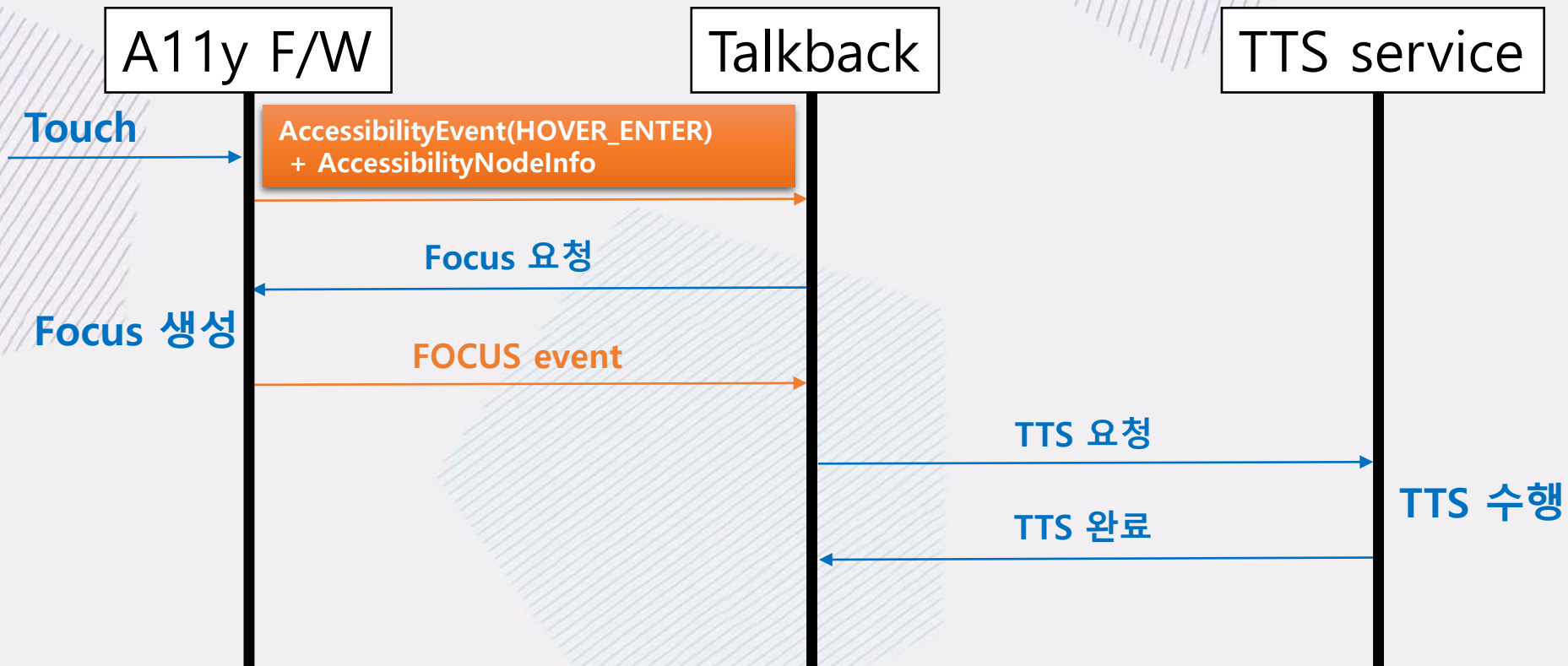
Talkback
(AccessibilityService)

```
@Override  
public void onAccessibilityEvent(AccessibilityEvent event) {  
    ...  
}
```

Talkback code - <https://github.com/google/talkback>

Talkback 소개(4/4)

Touch시 대상 View에서 AccessibilityEvent (HOVER_ENTER) 를 Talkback에 전달 함.



접근성을 높이기 위한 앱 개발시 고려해야 할 점(1/3)

1. Touch listener 사용 보단 Click listener를 사용

- O OS부터 Touch -> Click 기반으로 변경 됨.
- Touch listener의 문제점
: Click, Hint

2. Focusable view에 대해 Text 설정을 통해 Unlabelled 되지 않게 구현

- Focusable view의 조건
: Visibility, Text, Focusable 속성, Actions(Clickable/Long clickable/Checkable,,)
- Icon이나 Image에 대해 contentDescription을 설정
- 의미 없는 View에 대해서는 Focusable view의 조건 제거

접근성을 높이기 위한 앱 개발시 고려해야 할 점(2/3)

3. 접근성 용도가 아닌 AccessibilityService 구현은 지양

- 디바이스 성능에 부하를 줄 수 있음.
- AccessibilityService로 구현된 App의 경우 구글 인증 요구 됨.
(android:isAccessibilityTool="true" 지정이 되어 있지 않으면 24시간 마다 Notification이 발생)

<https://developer.android.com/reference/android/accessibilityservice/AccessibilityServiceInfo>

4. Webview 구현 지양

- 성능 열세
 - a. Webview의 경우 한 페이지에 Load 하는 데이터의 양이 많음.
 - b. Google lib(Chromium lib) 기반으로 되어 있어 내부 분석이 어려움.

접근성을 높이기 위한 앱 개발시 고려해야 할 점(3/3)

5. FrameLayout 보다는 LinearLayout 사용 권장

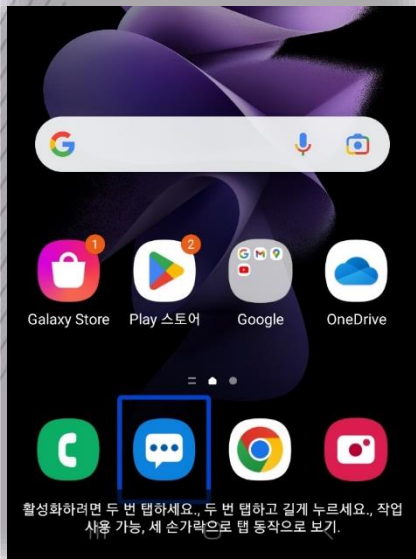
- 문제점

- a. Focus order의 문제
- b. 보이지 않는 영역 임에도 불구하고 Focus가 발생할 수 있음. Visibility 처리 문제.

구현 팁

힌트[Hint] (1/2)

Clickable, Long clickable, Checkable, 특정 View 등에 대해 사용성을 알려주는 기능



Actionable 속성	힌트 텍스트
Checkable	전환 하려면 두 번 탭하세요.
Clickable	활성화 하려면 두 번 탭하세요.
LongClickable	두 번 탭하고 길게 누르세요.

(View가 활성화 되어 있고 Actionable 속성이 true인 경우에만 발생.)

구현 팁

힌트[Hint] (2/2)

Hint의 일부분을 변경 하는 방법

.java

```
import android.view.accessibility.AccessibilityNodeInfo.AccessibilityAction;
import android.view.accessibility.AccessibilityNodeInfo;
import android.view.View.AccessibilityDelegate;

private final AccessibilityDelegate playBtnAccDelegate = new AccessibilityDelegate() {
    @Override
    public void onInitializeAccessibilityNodeInfo(View host, AccessibilityNodeInfo info) {
        super.onInitializeAccessibilityNodeInfo(host, info);
        info.addAction(new AccessibilityAction(AccessibilityNodeInfo.ACTION_CLICK, "재생"));
        info.addAction(new AccessibilityAction(AccessibilityNodeInfo.ACTION_LONG_CLICK, "길게 재생"));
    }
};

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        View playBtn = findViewById(R.id.play_btn);
        playBtn.setAccessibilityDelegate(playBtnAccDelegate);
    }
}
```

.xml

```
..

<Button android:id="@+id/play_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play" />

</LinearLayout>
```

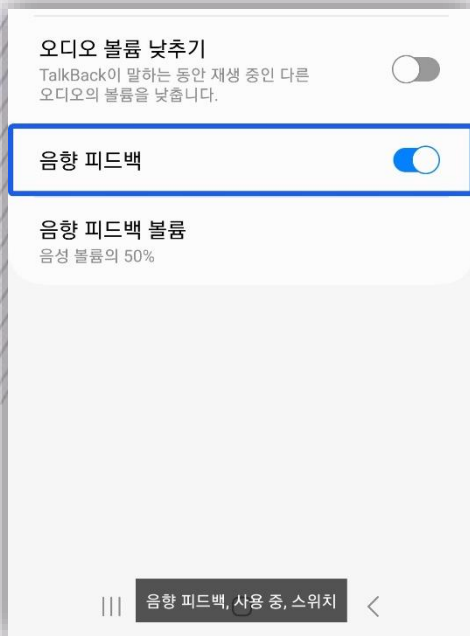
Actionable 속성	힌트 텍스트
Checkable	OO 작업을 하려면 두 번 탭하세요.
Clickable	OO 작업을 하려면 두 번 탭하세요.
LongClickable	OO 작업을 하려면 두 번 탭하고 길게 누르세요.

TTS : "Play, 버튼, 재생 작업을 하려면 두 번 탭하세요. 길게 재생 하려면 두 번 탭하고 길게 누르세요."

구현 팁

RoleDescription

버튼, 스위치, 이미지와 같이 요소 유형과 관련된 값 변경 가능



.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    View view = findViewById(R.id.view);
    ViewCompat.setAccessibilityDelegate(view, new RoleDescriptionAccessibilityDelegate("Gift Box"));
}

private static class RoleDescriptionAccessibilityDelegate extends AccessibilityDelegateCompat {
    private final String mRoleDescription;
    public RoleDescriptionAccessibilityDelegate(String roleDescription) {
        mRoleDescription = roleDescription;
    }

    @Override
    public void onInitializeAccessibilityNodeInfo(View host,
        AccessibilityNodeInfoCompat info) {
        super.onInitializeAccessibilityNodeInfo(host, info);
        info.setRoleDescription(mRoleDescription);
    }
}
```

TTS : "음향 피드백, 사용 중, Gift Box "

구현 팁

Announcement

사용자 지정 음성 피드백 생성 가능

1. View의 announceForAccessibility API 활용

```
View deleteView = findViewById(R.id.delete_view);  
  
// 특정 조건시 ex) 삭제  
deleteView.announceForAccessibility("삭제 되었습니다.");
```

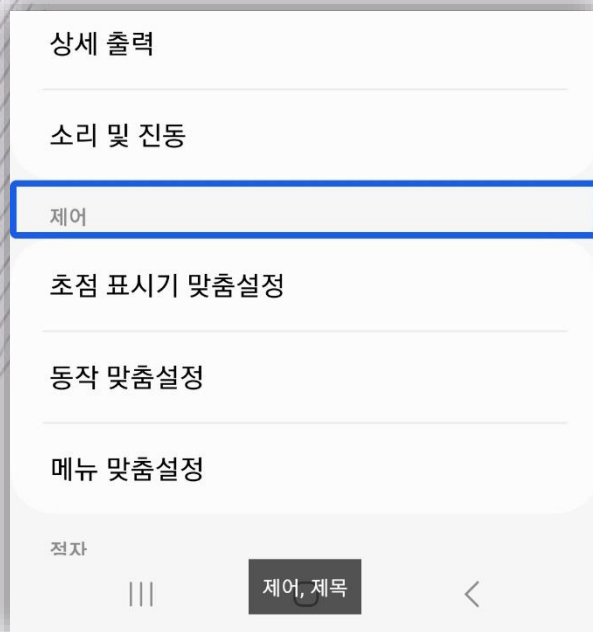
2. AccessibilityEvent 직접 생성

```
import android.view.accessibility.AccessibilityManager;  
import android.view.accessibility.AccessibilityEvent;  
  
...  
AccessibilityManager manager = (AccessibilityManager)  
mContext.getSystemService(Context.ACCESSIBILITY_SERVICE);  
if (manager != null && manager.isEnabled()) {  
    AccessibilityEvent event =  
        AccessibilityEvent.obtain(AccessibilityEvent.TYPE_ANNOUNCEMENT);  
    event.getText().clear();  
    event.getText().add("삭제 되었습니다.");  
    event.setPackageName(getPackageName());  
    manager.sendAccessibilityEvent(event);  
}
```

구현 팁

Heading

화면상에 Category로 쓰이는 View에 속성 지정함으로써 "제목"을 붙일 수 있음



```
.xml
..
<TextView android:id="@+id/play_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="@style/categoryStyle"
    android:accessibilityHeading="true"
    android:text="시스템" />
</LinearLayout>
```

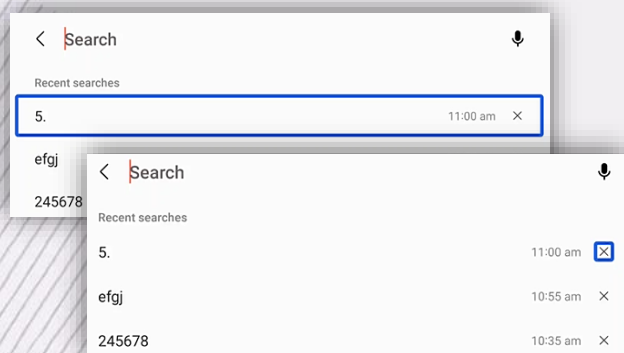
TTS : "시스템, 제목"

<https://developer.android.com/guide/topics/ui/accessibility/principles?hl=ko#java>

구현 팁

Focus 중첩 수정(1/4)

Action(Click)이 처리될 수 있는 상위 뷰만 Focusable이 가능하도록 수정 필요



"5, 11:00 am"

"Delete, Button"

수정 방안

TTS

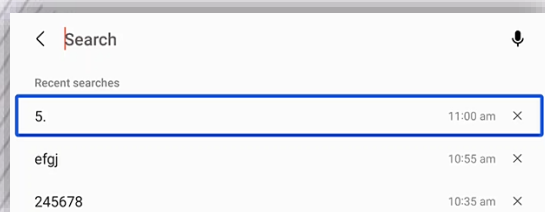
- | | |
|-----------------------------------|--------------------------------|
| 1. Talkback이 X 버튼을 인지하지 못하게 함 | "5., 11:00 am" |
| 2. X 버튼의 Focusable 하지 못하도록 변경 | "5., 11:00 am, Delete, Button" |
| 3. 상위 View에 contentDescription 설정 | "5., 11:00 am, Delete, Button" |

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:clickable="true"
    android:orientation="horizontal" >
    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="5." />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="11:00 am" />
    <ImageView
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:clickable="true"
        android:contentDescription="Delete"
        android:src="@drawable/ic_holder" />
</LinearLayout>
```

구현 팁

Focus 중첩 수정(2/4)

Action(Click)이 처리될 수 있는 상위 뷰만 Focusable이 가능하도록 수정 필요



"5, 11:00 am"

수정 방안

TTS

1. Talkback이 X 버튼을 인지하지 못하게 함 "5., 11:00 am"

AsIs

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:clickable="true"
  android:orientation="horizontal" >
  <TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="5." />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="11:00 am" />
  <ImageView
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:clickable="true"
    android:contentDescription="Delete"
    android:src="@drawable/ic_holder" />
</LinearLayout>
```

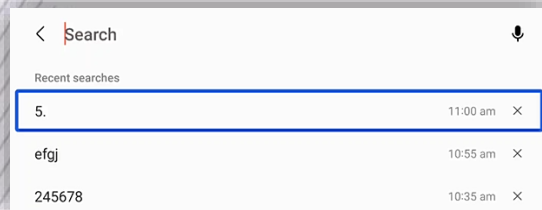
ToBe

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:clickable="true"
  android:orientation="horizontal" >
  <TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="5." />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="11:00 am" />
  <ImageView
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:clickable="true"
    android:importantForAccessibility="no"
    android:contentDescription="Delete"
    android:src="@drawable/ic_holder" />
</LinearLayout>
```

구현 팁

Focus 중첩 수정(3/4)

Action(Click)이 처리될 수 있는 상위 뷰만 Focusable이 가능하도록 수정 필요



"5, 11:00 am, Delete, Button"

수정 방안

TTS

2. X 버튼의 Focusable 하지 못하도록 변경 "5., 11:00 am, Delete, Button"

AsIs

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:clickable="true"
  android:orientation="horizontal" >
  <TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="5." />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="11:00 am" />
  <ImageView
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:clickable="true"
    android:contentDescription="Delete"
    android:src="@drawable/ic_holder" />
</LinearLayout>
```

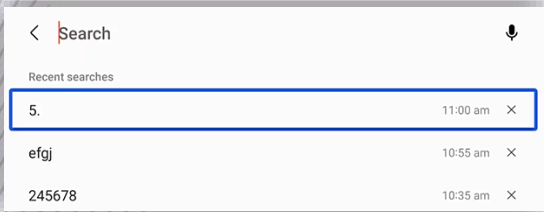
ToBe

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:clickable="true"
  android:orientation="horizontal" >
  <TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="5." />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="11:00 am" />
  <ImageView
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:contentDescription="Delete, Button"
    android:src="@drawable/ic_holder" />
</LinearLayout>
```

구현 팁

Focus 중첩 수정(4/4)

Action(Click)이 처리될 수 있는 상위 뷰만 Focusable이 가능하도록 수정 필요



"5, 11:00 am, Delete, Button"

수정 방안

TTS

3. 상위 View에 contentDescription 설정

"5., 11:00 am, Delete, Button"

AsIs

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:clickable="true"
  android:orientation="horizontal" >
  <TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="5." />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="11:00 am" />
  <ImageView
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:clickable="true"
    android:contentDescription="Delete"
    android:src="@drawable/ic_holder" />
</LinearLayout>
```

ToBe

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:clickable="true"
  android:contentDescription="5., 11:00 am, Delete, Button"
  android:orientation="horizontal" >
  <TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="5." />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="11:00 am" />
  <ImageView
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:src="@drawable/ic_holder" />
</LinearLayout>
```

나아 가야 할 방향

- ML 기반의 서비스 적용 및 확대
ex) Icon detection
- Multi window UI에 대한 접근성 강화
- Watch (Wear OS) 사용성 향상
- 앱 개발자 인식 개선

Fin.

Thank you



Appendix



Log 전달 방법

1. 설정 앱 > 접근성 > Talkback > 설정 > 고급 설정 > 개발자 설정 > **로그 출력 수준 : VERBOSE**
2. 설정 앱 > 휴대전화 정보 > 소프트웨어 정보 > 빌드 번호 → 개발자 옵션이 나올 때 까지 클릭
3. 설정 앱 > 개발자 옵션 > **삼성 상세 디버깅 로그 : On** > **Device reboot**
4. 문제점 재현
5. Log 전달 (삼성 멤버스, 자세한 문제점 설명)
ex) “Talkback이 갤러리 앱에서 제대로 동작을 안해요.”
=> “갤러리 > 사진 선택 > 편집 > 회전 버튼을 눌렀을 때 동작을 안해요. ”