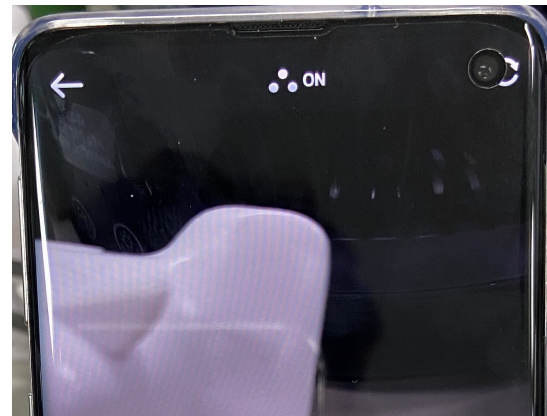


Camera Cutout(Notch) 처리 for Android (Android DisplayCutout 응용)

Context

1. Camera Notch(Cutout)
2. DisplayCutout
 - 1) Test App
 - 2) 적용
3. 기타
4. 참고 자료

1. 아이폰 X에서는 Notch라 불리는 부분이 있는데, 안드로이드에서는 Android 9 (API 레벨 28) 지원하는 단말기 부터 추가되었습니다.
2. 이름은 Notch가 아닌 **Cutout**로 불립니다.
3. Cutout(Notch)에는 UI적 이슈가 있습니다. **FullScreenDisplay**때 **Cutout**부분을 처리를 못한다는 점입니다.
4. Android도 이러한 부분을 피하지는 못하고, 하드웨어적인 이슈를 해결하기 전까지는 이러한 UI가 유지 될듯합니다.



Android에서는 Cutout대하여 대안으로 'DisplayCutout'을 내놓았습니다.

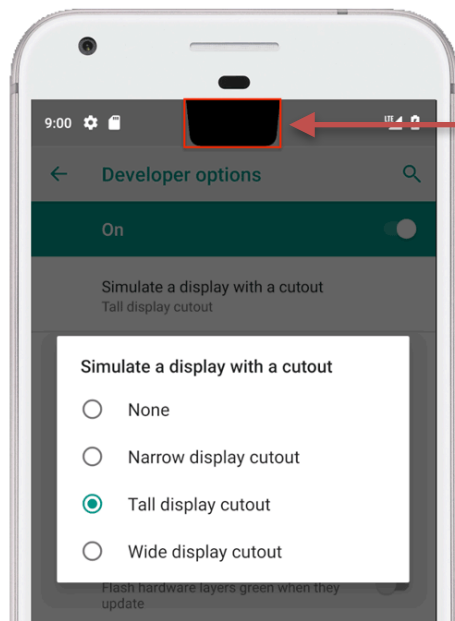
'DisplayCutout'은 Android 9 (API 레벨 28) 이상을 실행하는 기기에서 지원합니다.

컷 아웃 영역, 즉 **컷 아웃이 포함 된 디스플레이 표면의 가장자리에서 직사각형으로 작업하는 방법**을 포함하여 컷 아웃이있는 장치에 대한 지원을 구현하는 방법을 제시합니다.

이 방법을 응용하면 **Camera Cutout(Notch)**의 UI처리가 가능합니다.



1. 'DisplayCutout'는 안드로이드에서 기본 제공하는 library이므로, 따로 추가할 필요가 없습니다.
2. 'DisplayCutout'로 Cutout(Notch) Camera가 있는 경우, 반환 값을 Rect Class로 반환 합니다.
3. 반환하는 값의 형태는 **Rect(int left, int top, int right, int bottom)**인 4개의 정수 좌표이며, Canvas를 이용하여 선을 이으면 하기의 사진과 같이 Cutout(Notch) 위치에 사각형이 출력이 됩니다.
4. 값은 기기마다 다르게 출력이 됩니다. 그러므로, 모든 단말기에 Cutout(Notch) 값을 가져올 수 있습니다.



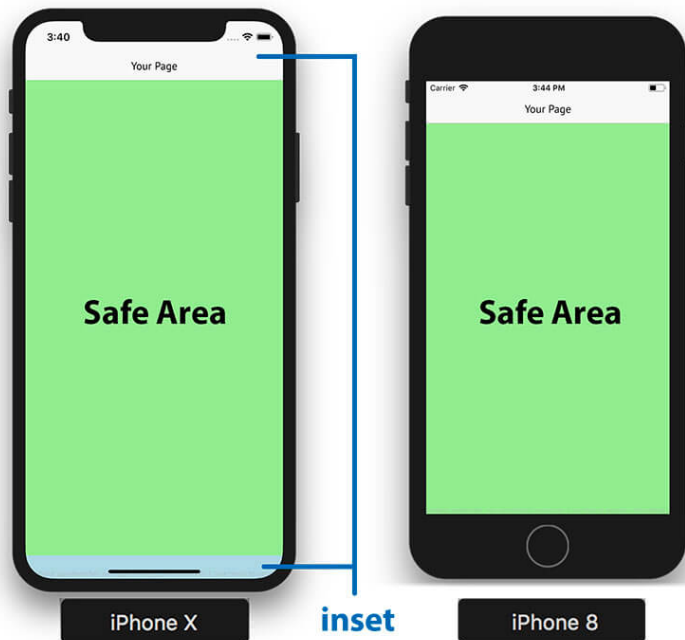
Rect(int left, int top, int right, int bottom)

Test App

Test App을 통하여 Cutout의 도출 과정을 설명합니다.

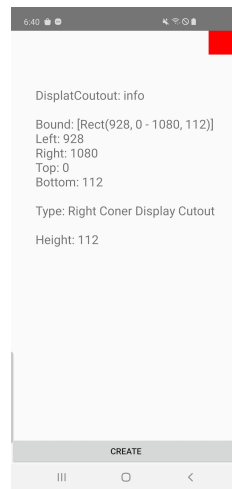
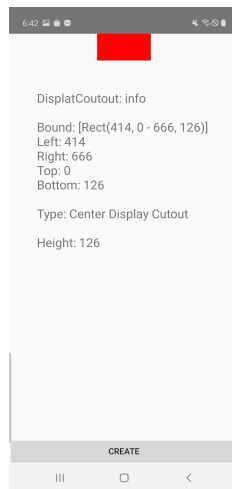
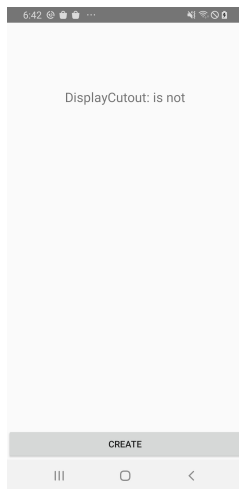
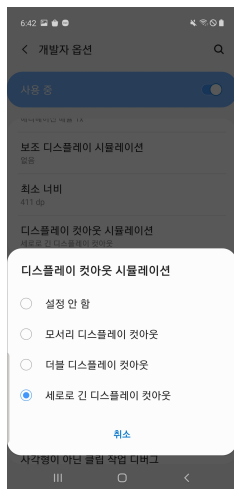
‘DisplayCutout’을 이용하여 테스트 앱을 구현하기

- Cutout의 Rect 좌표 값 외에도 Inset 좌표 값도 가져올 수 있습니다.
- 두(Rect, Inset) 값을 가지고 분석하여 추정 값을 도출 하였습니다.
- 추정 값 : 사각형 좌표 값, 좌표 값을 파싱한 값, **Cutout 타입, Cutout 높이**



‘DisplayCutout’을 이용하여 테스트 앱을 구현하기

- 기본 적으로 ‘**Android 단말기 설정 > 개발자 옵션**’ 에서 ‘**디스플레이 컷아웃 시뮬레이션**’을 지원하여 4가지 옵션을 선택을 할 수 있습니다.
- 테스트 앱에서 구현 후 테스트 결과,
하기 사진과 같이 **none, corner(left / right), double, center** 형태따라 다르게 표현이 됩니다.
- 4가지 옵션과 구현한 테스트 앱을 통하여 추정한 값의 정확성 확보하였습니다.



적용

Test App을 바탕으로 AnibearaR 앱에 적용한 결과를 설명합니다.

Android 앱에 'DisplayCutout' 적용하기

- 구현한 테스트 앱을 AnibearaR 앱에 적용합니다.

1. package > utils > 'DisplayCutoutHelper' 추가

- Activity에서 Cutout값을 쉽게 처리하도록 Helper클래스 생성하였습니다.

```
package com.anipen.anibearar.utils

import ...

/**
 * Created by YOO MyeongSeung on 2019-12-03.
 */

class DisplayCutoutHelper(private var activity: Activity) {
```

2. package > 'ScreenRecordUnityPlayerActivity'에 콜백함수 추가

- Unity와 Activity사이에서 통신이 가능하도록 함수를 추가하였습니다.

```
//화면에 뷰가 추가될때 호출
//getRootWindowInsets를 사용하기 위해 onAttachedToWindow()를 오버라이드
@Override
public void onAttachedToWindow() {
    super.onAttachedToWindow();
    new DisplayCutoutHelper(activity, this).initDisplaySettings();
}

//단말기의 컷아웃 정보를 배열로 가져온다.
/*
+ 배열 구조
+ String[0] : rect: MutableList<Rect>
+ String[1] : rectLeft: Int
+ String[2] : rectTop: Int
+ String[3] : rectRight: Int
+ String[4] : rectBottom: Int
+ String[5] : dType: String
+ String[6] : cHeight: Int
*/
@SuppressWarnings("unused")
public String[] getDisplayArray() {
    Log.d(TAG, "Start, getDisplayArray");
    return DisplayCutoutHelper.getDisplayArray();
}

//단말기의 컷아웃 상태를 체크한다.
/*
+ true : Cutout(North) 존재 0
+ false : Cutout(North) 존재 X
*/
@SuppressWarnings("unused")
public boolean checkDisplayBoolean() {
    Log.d(TAG, "Start, checkDisplay");
    return DisplayCutoutHelper.checkDisplayBoolean();
}
```

Android 앱에 'DisplayCutout' 적용하기

3. Logcat, 로그 확인 (Android)

Android -> *로 값을 보낼때 정상적으로 출력이 되는지 확인 하였습니다.

*로 값을 보낼때의 값은 String배열로 보내도록 구현하였습니다.

<배열 구조>

String[0] : rect: MutableList<Rect> - 사각형 좌표 값

String[1] : rectLeft: Int - 왼쪽 좌표

String[2] : rectTop: Int - 위쪽 좌표

String[3] : rectRight: Int - 오른쪽 좌표

String[4] : rectBottom: Int - 아래쪽 좌표

String[5] : dcType: String - Cutout 종류

String[6] : cHeight: Int - Cutout 높이

필요에 따라 String -> Integer로 형변환하여 사용하면 됩니다.

```
2019-12-05 18:40:00.507 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: Start, initDisCutoutSetting()
2019-12-05 18:40:00.508 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: Created, DisplayCutout
2019-12-05 18:40:00.508 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: DisplayCutout : [Rect(928, 0 - 1080, 112)]
2019-12-05 18:40:00.508 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: DisplayCutout : 928
2019-12-05 18:40:00.508 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: DisplayCutout : 0
2019-12-05 18:40:00.508 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: DisplayCutout : 1080
2019-12-05 18:40:00.508 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: DisplayCutout : 112
2019-12-05 18:40:00.508 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: DisplayCutout : Right Corner Display Cutout
2019-12-05 18:40:00.508 27876-27876/com.anipen.anibearar.dev D/DisplayCutoutHelper: DisplayCutout : 112
```

Android 문서

<https://developer.android.com/guide/topics/display-cutout>

<https://source.android.com/devices/tech/display/display-cutouts>

<https://developer.android.com/reference/android/view/DisplayCutout>

블로그_1

<https://thdev.tech/android/2018/07/07/Android-P-Preview-ChangeLog/>

블로그_2

<https://medium.com/exploring-android/exploring-android-p-display-cutouts-42885e8a2a96>

**Thanks
for
your
attention !**