

# Android Dual Screen development(Presentation) , casting the second page to the secondary screen

Modified on 2023-06-22 19:50:01

## 1. Background

For a recent development project with two screens, I needed to cast the second page to the secondary screen.

This requires the use of Android's dual-screen hetero-display (Presentation) technology, after some research, here is my notes.

That secondary screen we have is a car's rearview mirror (for streaming media), a long screen with a resolution of 400 \* 1920 px.

How to display the content that needs to be displayed on this screen in a correctly placed manner is a major focus in our development.

Moreover, we are using the tablet landscape for development, for the handling of dual-screen hetero display in landscape, is also a key point in our development.

Let's first understand the keys of dual-screen hetero-display (Presentation) technology. Android Presentation is a display device in Android.

It can connect to multiple display devices at the same time and display different content on different display devices.

It can be connected to an external monitor via HDMI or VGA to expand or switch the screen.

With Presentation, you can display different content on different screens, such as a presentation on a projector and notes on a laptop.

This allows for a better presentation of the information and improves the effectiveness of the presentation.

## 2. Creating a home screen layout

Just use MainActivity for the main screen.

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        //...  
    }  
}
```

## 3. Creating sub-screen layouts

Here, inheriting the MyPresentation interface,  
among others:

```
class MyPresentation(outerContext: Context, display: Display) :  
    Presentation(outerContext, display) {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.dialog_normal) //Bind the layout of the secondary screen  
                                                // display  
    }  
  
    //called when displaying  
    override fun onStart() {  
        super.onStart()  
    }  
  
    //called on disappearance  
    override fun onStop() {  
        super.onStop()  
    }  
}
```

#### 4. Casting the page to the secondary screen

There are multiple ways to cast the screen to the secondary screen, we just need to go and call these codes in MainActivity.

##### 4.1 Mode 1: MediaRouter

```
MediaRouter mediaRouter = (MediaRouter)  
getSystemService(Context.MEDIA_ROUTER_SERVICE);  
MediaRouter.RouteInfo route =  
mediaRouter.getSelectedRoute(MediaRouter.ROUTE_TYPE_LIVE_AUDIO);  
if (route != null) {  
    Display presentationDisplay = route.getPresentationDisplay();  
    if (presentationDisplay != null) {  
        MyPresentation myPresentation = new MyPresentation(MainActivity.this,  
presentationDisplay);  
        myPresentation.show();  
    }  
}
```

```
}
```

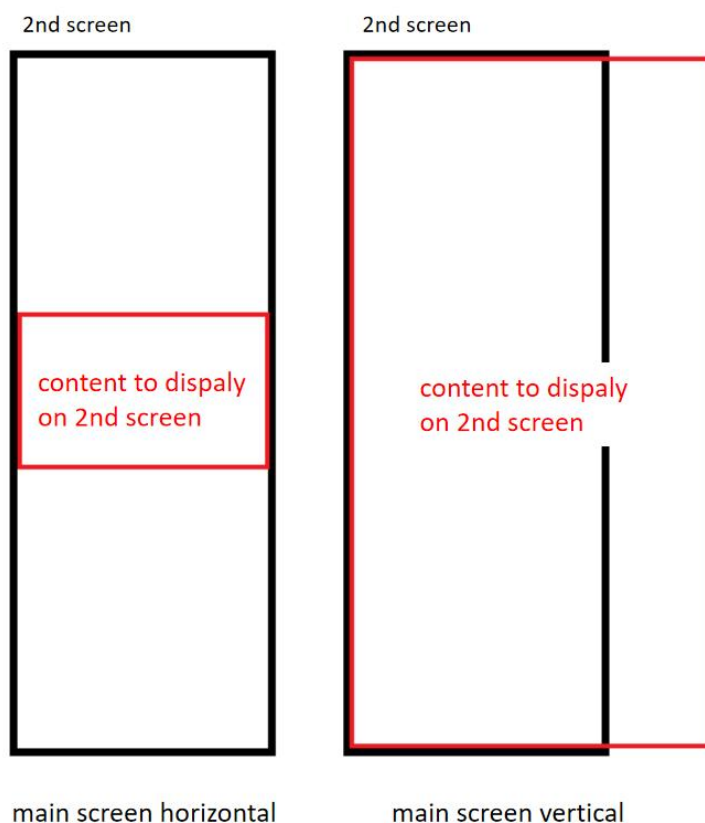
#### 4.2 Method 2: DisplayManager

```
DisplayManager mDisplayManager = (DisplayManager)
getSystemService(Context.DISPLAY_SERVICE);
Display[] displays = mDisplayManager.getDisplays();
if (displays.length > 1) {
    //displays[0] the main display, displays[1] the secondary display
    MyPresentation myPresentation = new MyPresentation(MainActivity.this, displays[1]);
    myPresentation.show();
}
```

#### 5. Handling of dual-screen hetero display in landscape mode

On our secondary screen, the display logic of the secondary screen is that if the main screen is vertical, the secondary screen is displayed as a secondary screen; if the main screen is horizontal, the secondary screen is displayed as a horizontal screen.

As shown in the figure below:



And the effect we want to achieve is that the main screen is horizontal and the main screen is vertical.

Then we came up with the idea of getting two Activity's that:

Activity1 is responsible for displaying the page on the main screen, while Activity2 is responsible for displaying the secondary screen.

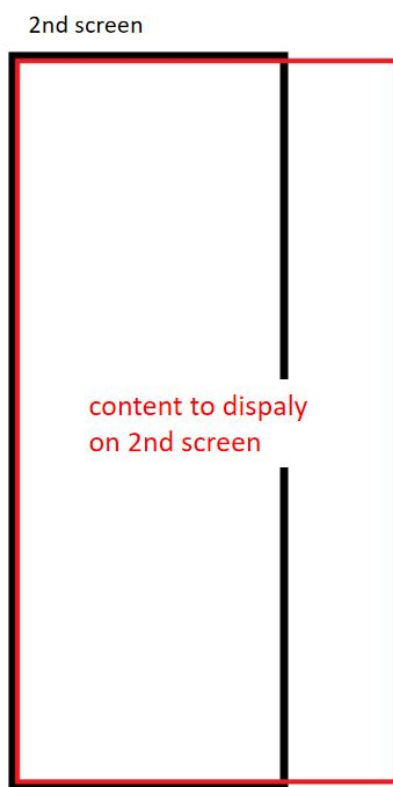
Start Activity2 first, then quickly start Activity1, and when Activity1 is closed, go ahead and close Activity2 as well.

This allows for the effect where the main and secondary screens are displayed in different orientations.

#### 6. Correct placement of the content to be displayed on the secondary screen

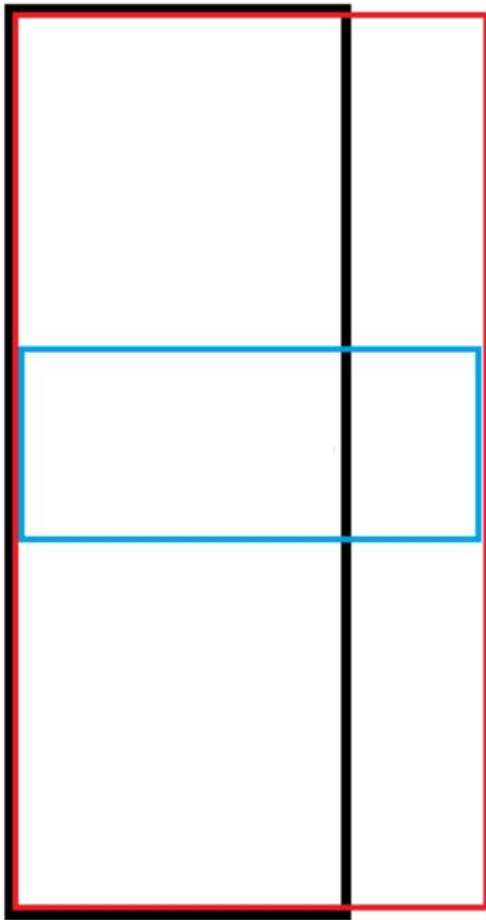
Since our secondary screen is a long screen with a resolution of 400 \* 1920 px, and we are now displaying it in landscape on the main screen,

and the secondary screen shows a vertical view by the above codes.



And what we're really expecting to show is the blue part.

2nd screen

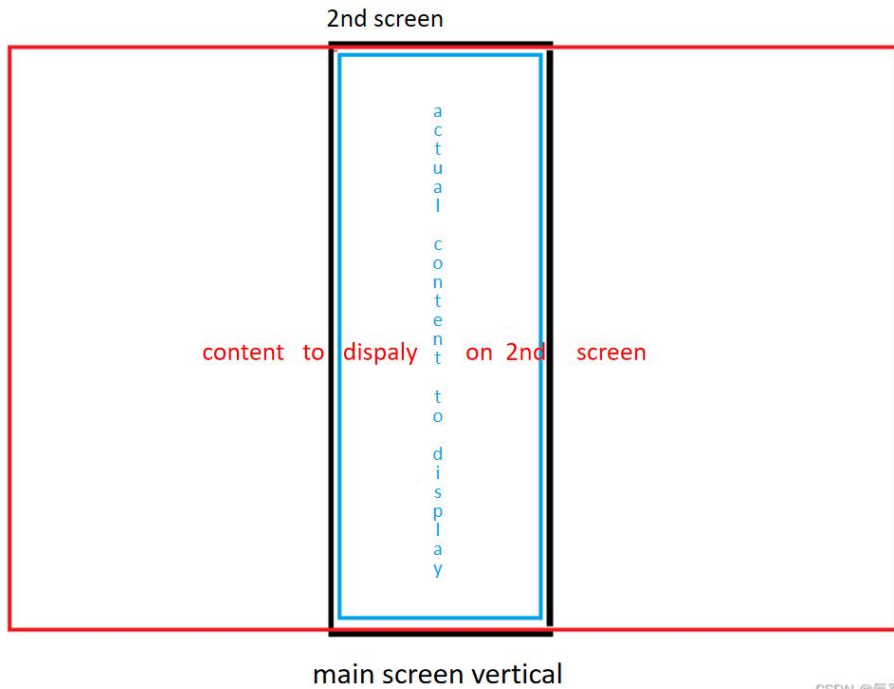


main screen vertical

So, we need to do a little rotation and displacement of the View:

```
val rotation: ObjectAnimator =  
    ObjectAnimator.ofFloat(binding.textureView, "rotation", 0f, 90f)  
val translationX: ObjectAnimator =  
    ObjectAnimator.ofFloat(binding.textureView, "translationX", 0f, -720f)  
val translationY: ObjectAnimator =  
    ObjectAnimator.ofFloat(binding.textureView, "translationY", 0f, 420f)  
val animatorSet = AnimatorSet()  
animatorSet.playSequentially(rotation, translationX, translationY)  
animatorSet.duration = 0  
animatorSet.start()
```

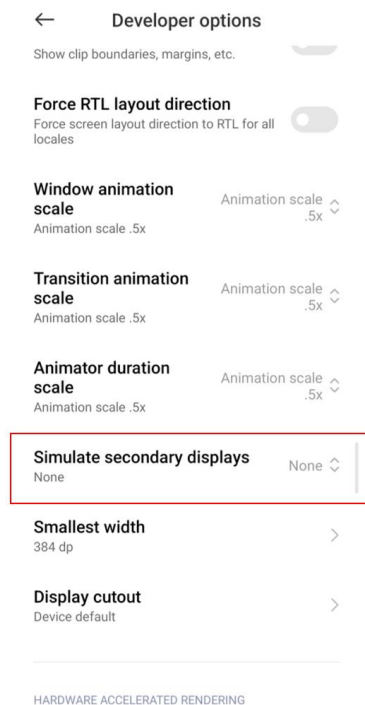
It ends up looking like this:



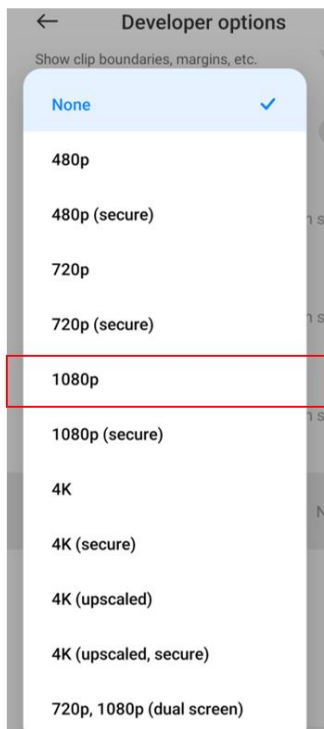
## 7. Others

### 7.1 How to debug without a 2nd screen ?

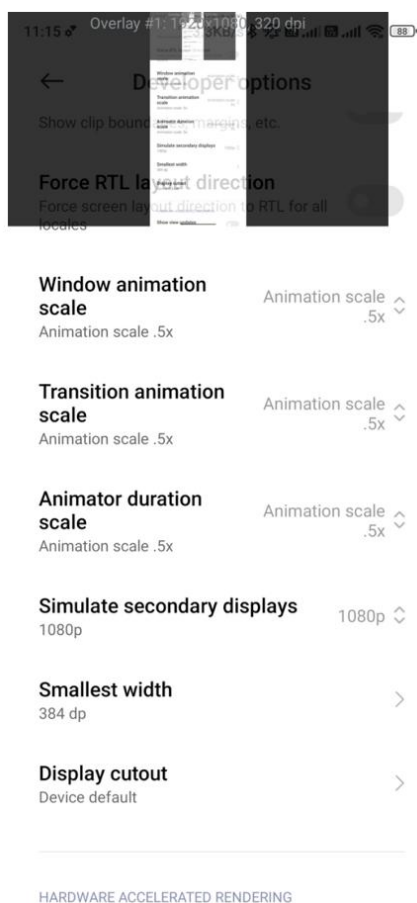
If you don't have a secondary screen on hand, you can turn on Emulate secondary display device in the developer options.



Here we can choose from 1080P, 720P, 4K, etc and they are all OK.



Once selected, the virtual secondary screen is displayed as follows:



In this way, we can debug via a virtual secondary screen in case we don't have a real one.

## 7.2 Application Scenarios of Dual-Screen Hetero-Display

Android Presentation can be used in the following scenarios:

Hotel information registration : Hotel staff can operate from one screen, while customers can view relevant information from the other screen.

Medical Devices: Some medical devices may require multiple screens to be connected at the same time so that doctors can view information about multiple patients at the same time.

Presentation: The presenter can play a presentation on one screen while displaying notes or relevant data on another screen.

Multi-screen collaboration: Multiple screens can be connected together and different content can be displayed on each screen to improve multitasking efficiency.

So Android's dual-screen hetero-display is quite useful in specific areas.

-----

Copyright: This article is an original blog post and is subject to CC 4.0 BY-SA copyright, please include a link to the original source and this statement.

Original link: <https://blog.csdn.net/ethanco/article/details/128662556>