

18. Using the camera

Requesting camera permission

- If your app uses the device camera, you must explicitly state that it will use that hardware in your app's **AndroidManifest.xml** file.

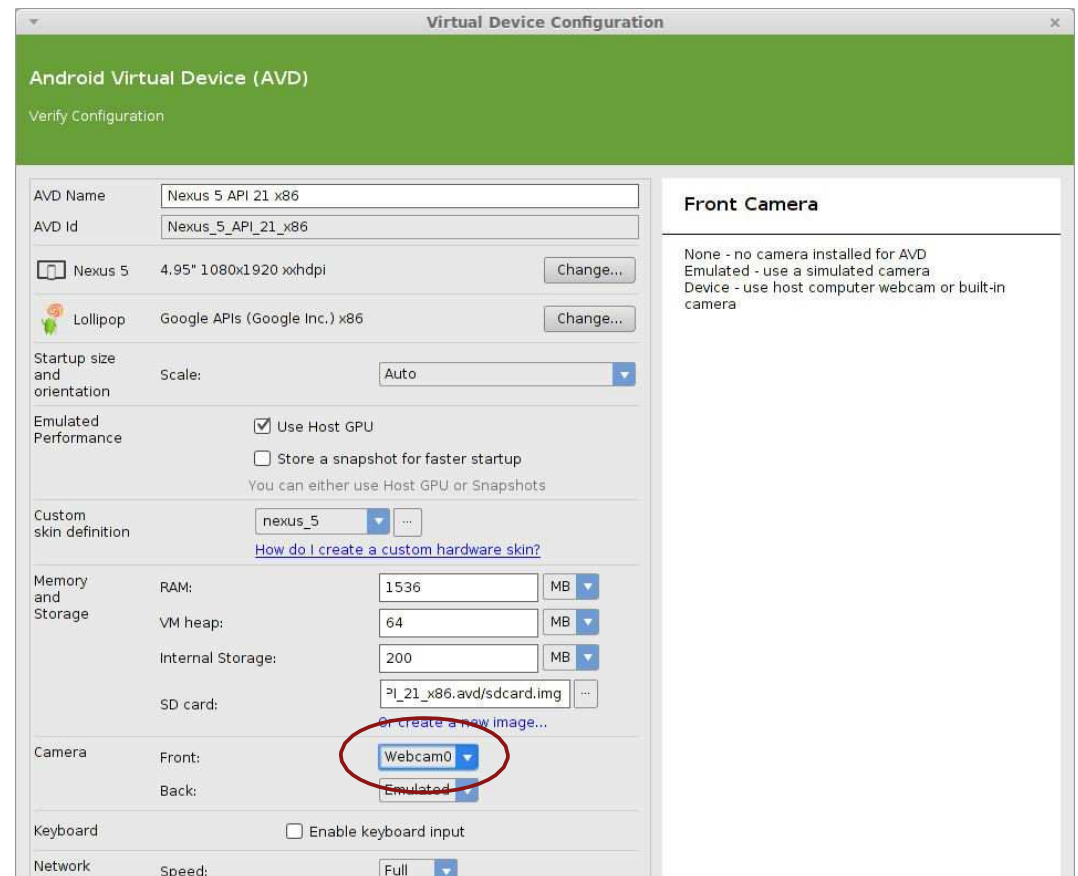


```
<manifest ...>  
    <uses-permission  
        android:name="android.permission.CAMERA" />  
    <uses-feature  
        android:name="android.hardware.camera"  
        android:required="true" />  
    ...  
</manifest>
```



AVD and camera

- It is difficult to emulate a camera in the Android Virtual Device.
- In AVD Manager, you can try setting the device to use your laptop's webcam as the virtual camera, but it may crash.
 - AVD Manager → Edit → Advanced Settings → Camera
- Alternatively, just deploy your app to a physical device to test camera functionality.



Camera code example

- Example that takes a picture and places it into an ImageView:

```
private static final int REQ_CODE_TAKE_PICTURE = 90210;
...
Intent picIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
startActivityForResult(picIntent, REQ_CODE_TAKE_PICTURE);
...

@Override
protected void onActivityResult(int requestCode, int resultCode,
                                Intent intent) {
    if (requestCode == REQ_CODE_TAKE_PICTURE
        && resultCode == RESULT_OK) {
        Bitmap bmp= (Bitmap) intent.getExtras().get("data");
        ImageView img= (ImageView) findViewById(R.id.camera_image);
        img.setImageBitmap(bmp);
    }
}
```

The Camera class ([link](#))

- In our previous example, we used the built-in photo-taking activity using an Intent.
 - This is simple but allows very little customization.



- For more precise camera control, look up the Camera class:

```
Camera cam = Camera.open();
Camera.Parameters params = cam.getParameters();
params.setPreviewSize(w, h);
cam.setParameters(params);
cam.setDisplayOrientation(degrees);    // degrees
cam.startPreview();
cam.takePicture(...);    // pass "callbacks" to run later
cam.stopPreview();
cam.release();
```