Homework 1

Game Graphics
Prof. HyeongYeop Kang
siamiz@khu.ac.kr

Goal

- Setting up Android Studio development environment

- Filling in code lines in the vertex shader.
    - Vertex position
    - Vertex normal
    - Texture coordinates

- Filling in code lines in the functions of the scene class in GL program.

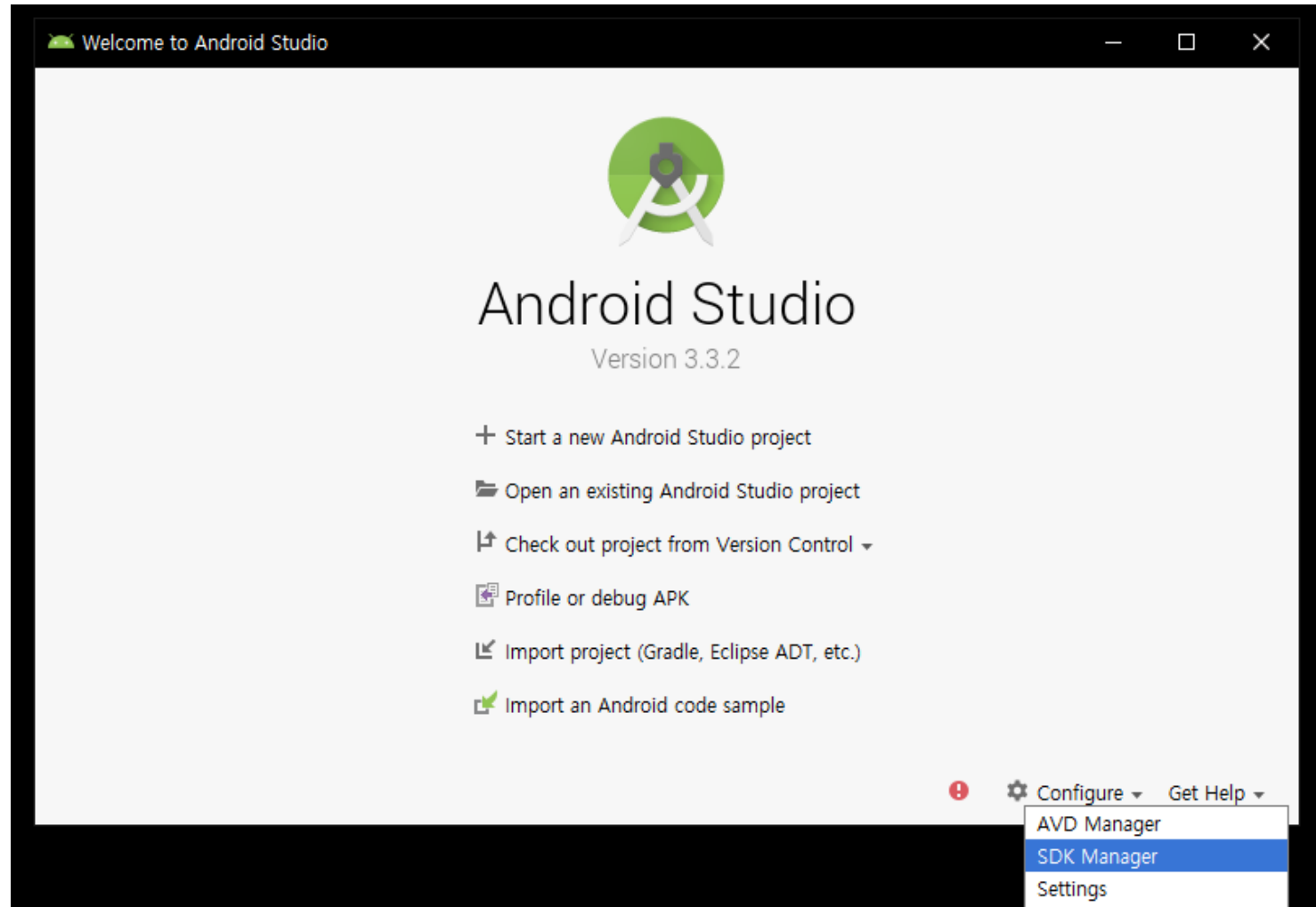- +10 points homework.

Android Studio

- **Android Studio** is the official integrated development environment (IDE) for the Android platform.

- Android Studio can be downloaded from the official website.
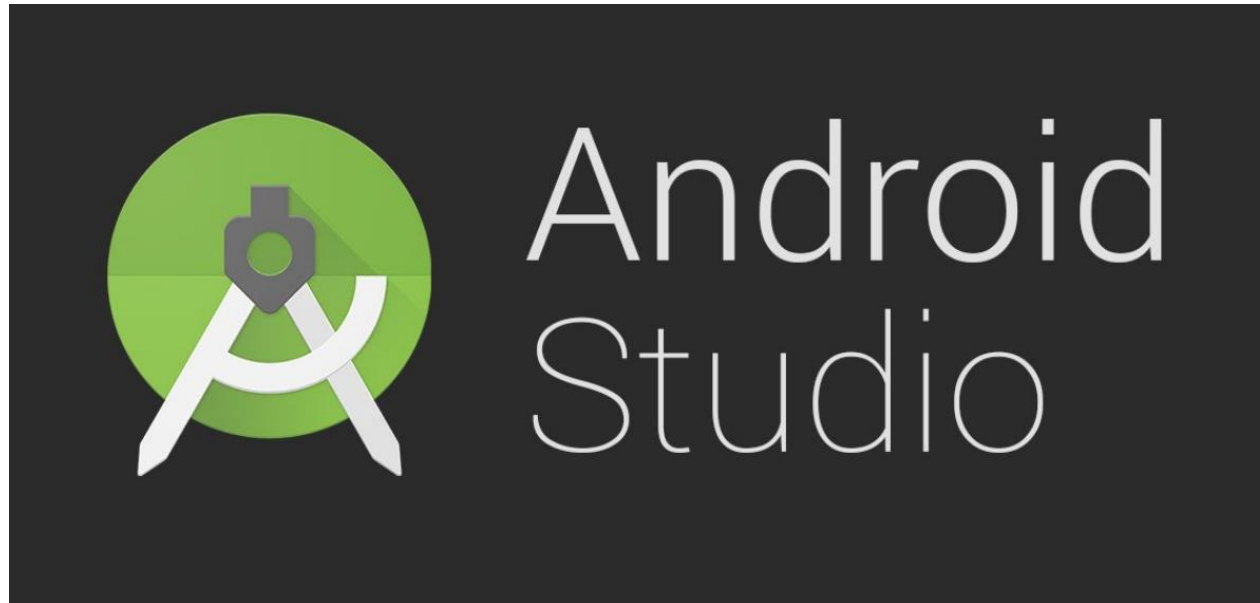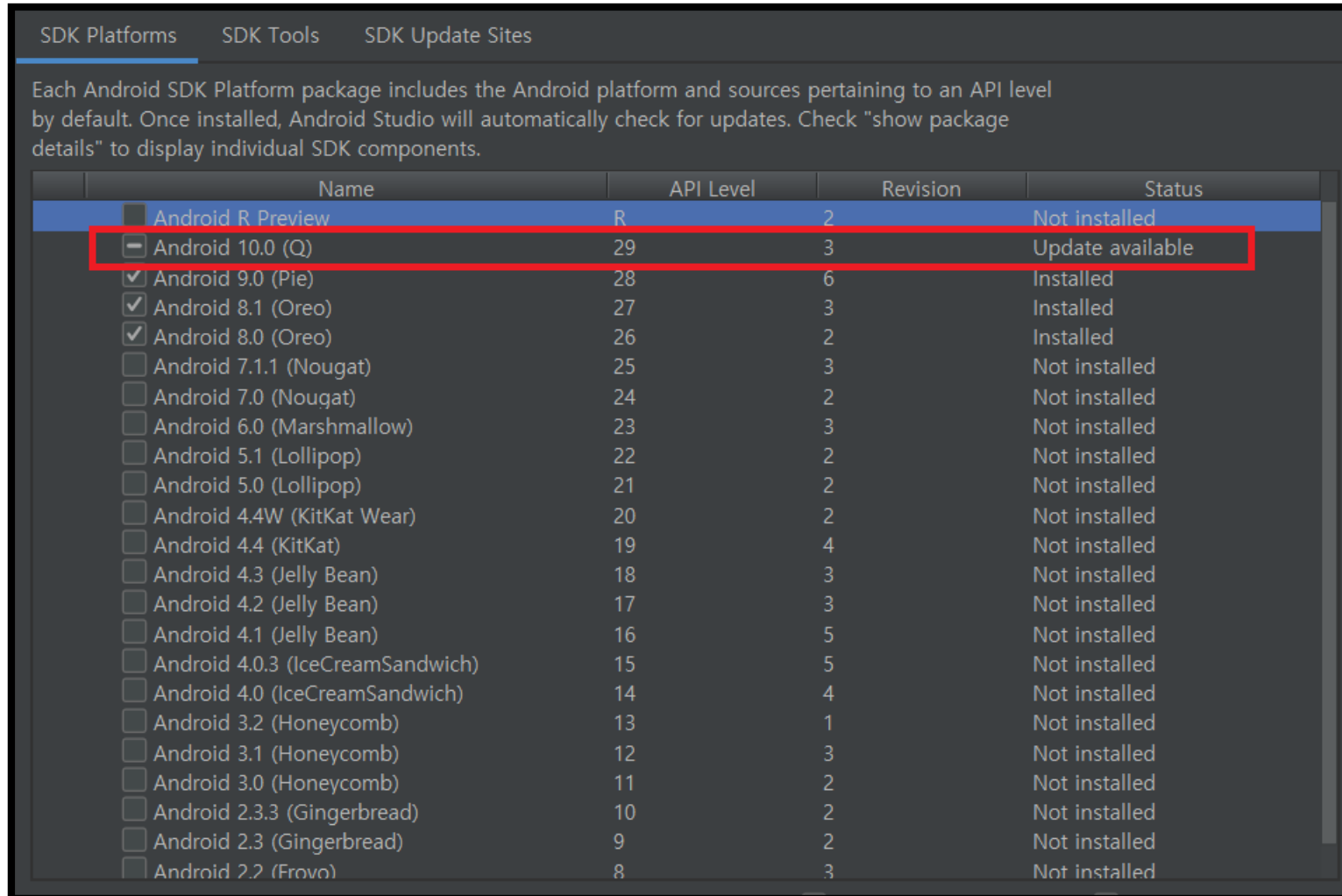
Android SDK

■ **Android SDK** can be installed through the SDK Manager in Android Studio.

Android SDK

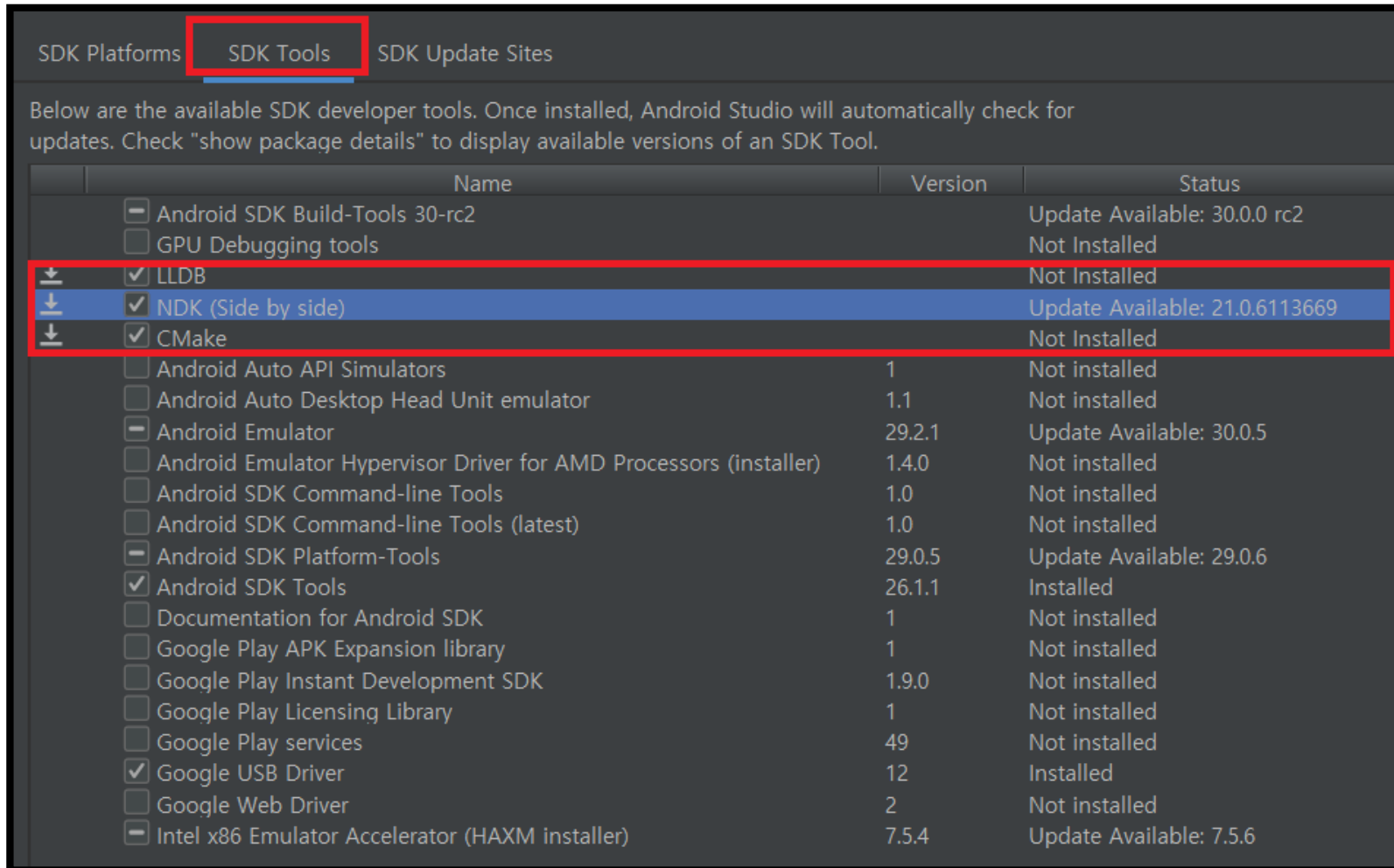- The **latest platform** (2020.03.26. Android 10.0 Q) will be installed automatically.

The image and the actual screen can be different.

Android SDK

To use C++ native language on Android, you need to install the following three tools.
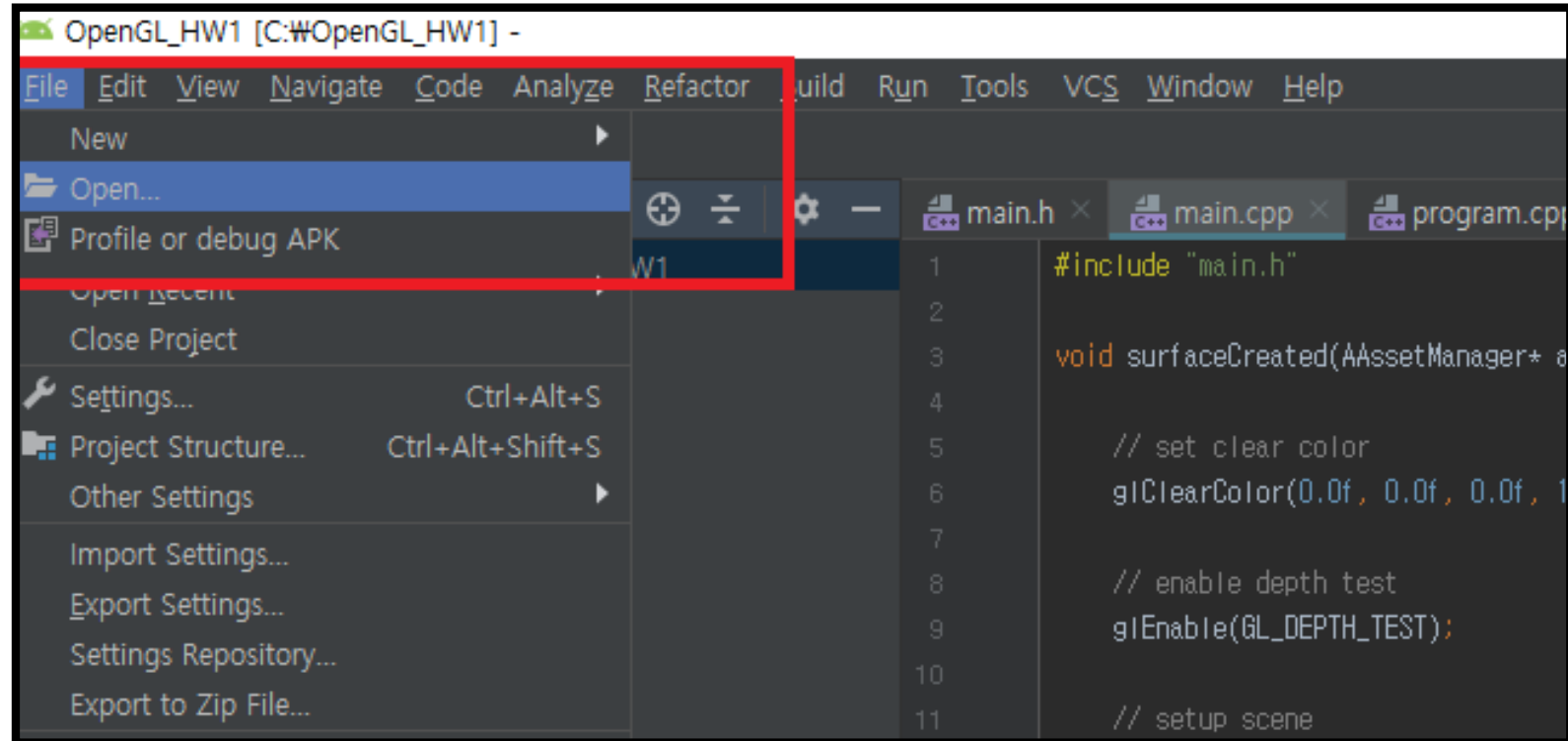
- CMake
- LLDB
- NDK

	Name	Version	Status
⊟	Android SDK Build-Tools 30-rc2		Update Available: 30.0.0 rc2
☐	GPU Debugging tools		Not Installed
☑	LLDB		Not Installed
☑	NDK (Side by side)		Update Available: 21.0.6113669
☑	CMake		Not Installed
☐	Android Auto API Simulators	1	Not installed
☐	Android Auto Desktop Head Unit emulator	1.1	Not installed
⊟	Android Emulator	29.2.1	Update Available: 30.0.5
☐	Android Emulator Hypervisor Driver for AMD Processors (installer)	1.4.0	Not installed
☐	Android SDK Command-line Tools	1.0	Not installed
☐	Android SDK Command-line Tools (latest)	1.0	Not installed
⊟	Android SDK Platform-Tools	29.0.5	Update Available: 29.0.6
☑	Android SDK Tools	26.1.1	Installed
☐	Documentation for Android SDK	1	Not installed
☐	Google Play APK Expansion library	1	Not installed
☐	Google Play Instant Development SDK	1.9.0	Not installed
☐	Google Play Licensing Library	1	Not installed
☐	Google Play services	49	Not installed
☑	Google USB Driver	12	Installed
☐	Google Web Driver	2	Not installed
⊟	Intel x86 Emulator Accelerator (HAXM installer)	7.5.4	Update Available: 7.5.6

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

SDK Platforms   SDK Tools   SDK Update Sites

The image and the actual screen can be different.
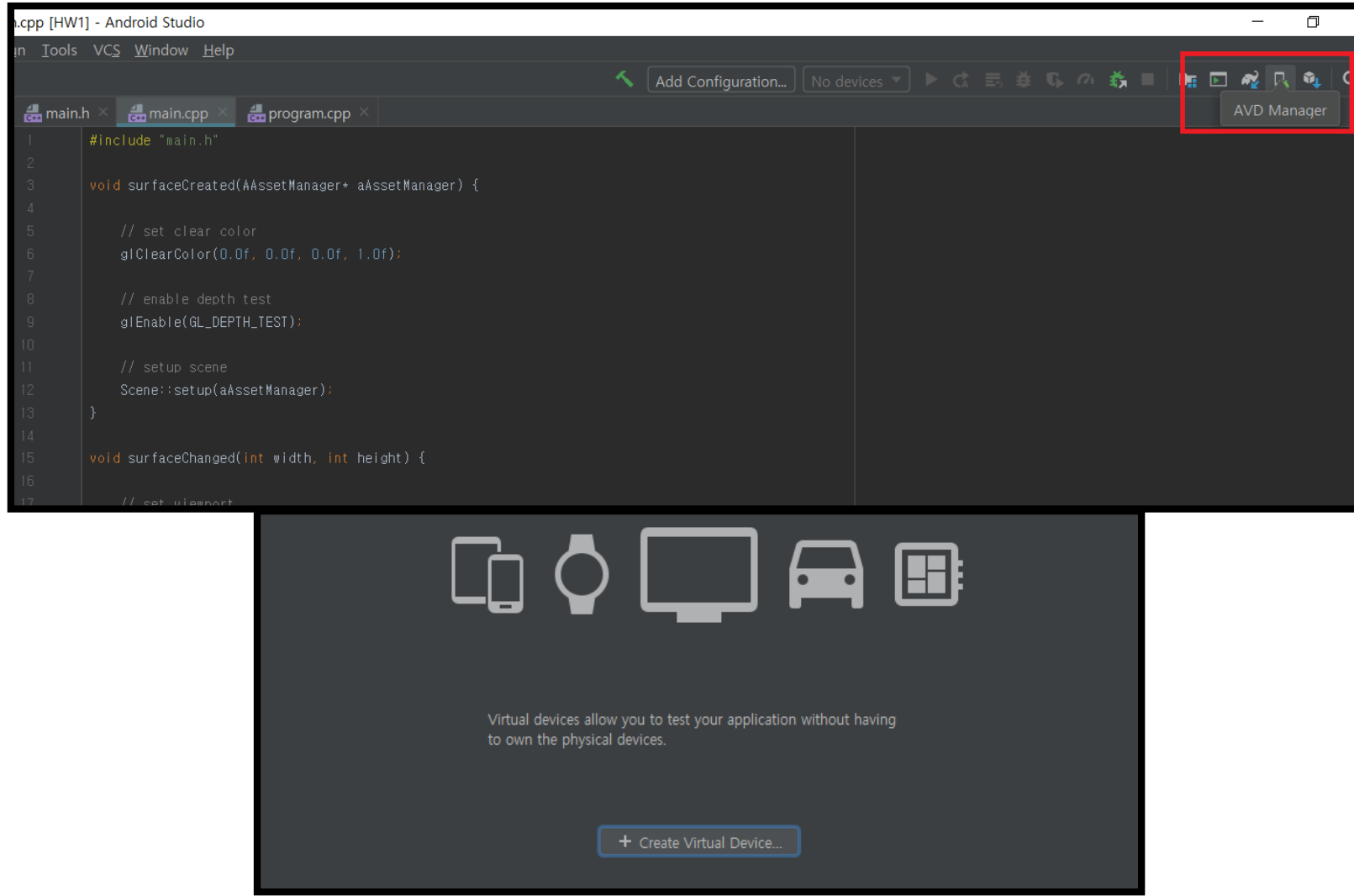
Open Project Folder

Click the [File] – [Open].

Create Virtual Device

After opening the project, let's create virtual device.

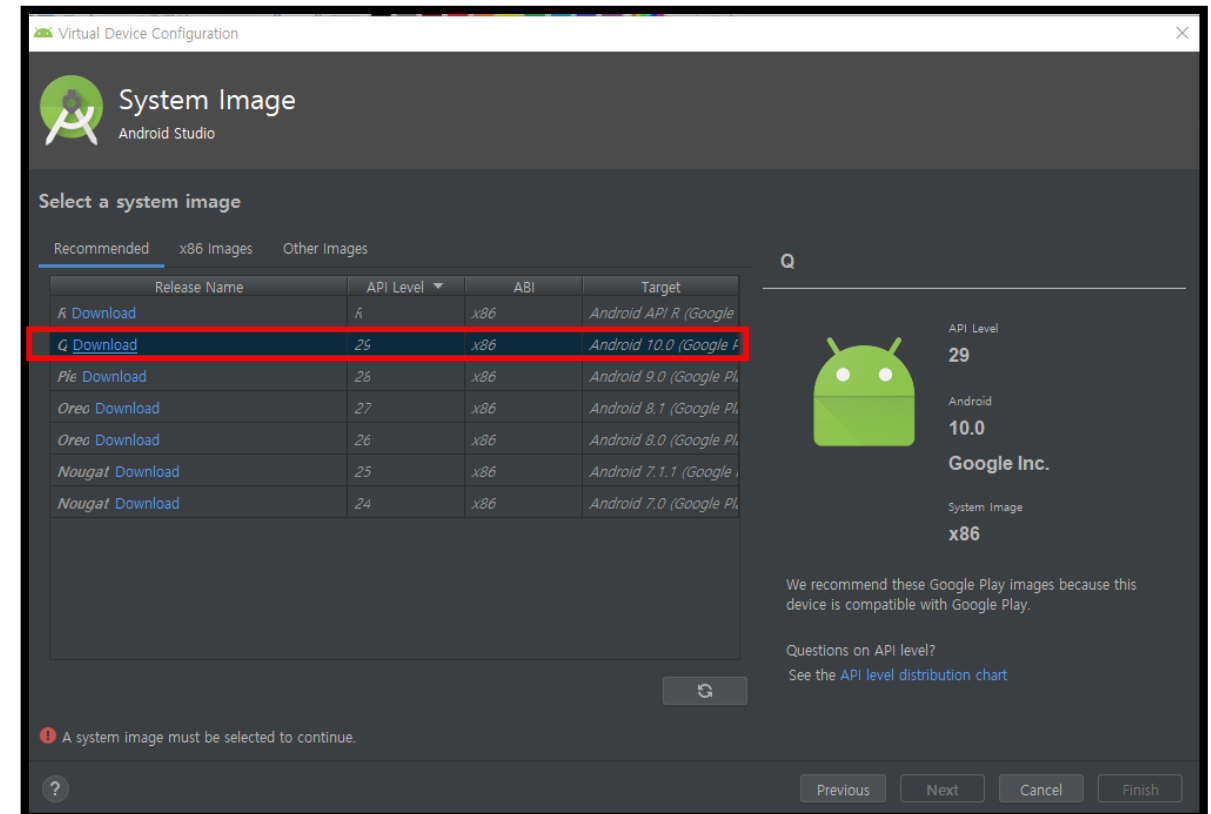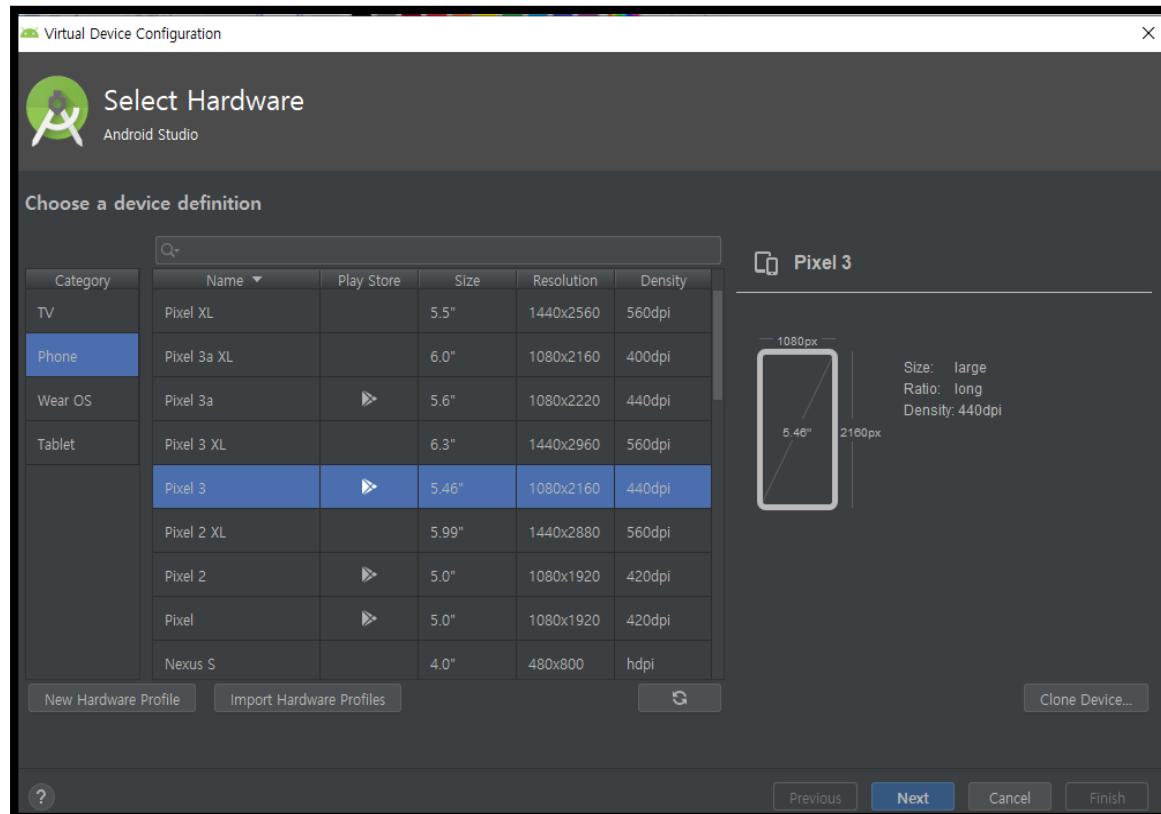- Click the AVD Manager and create virtual device.

Create Virtual Device

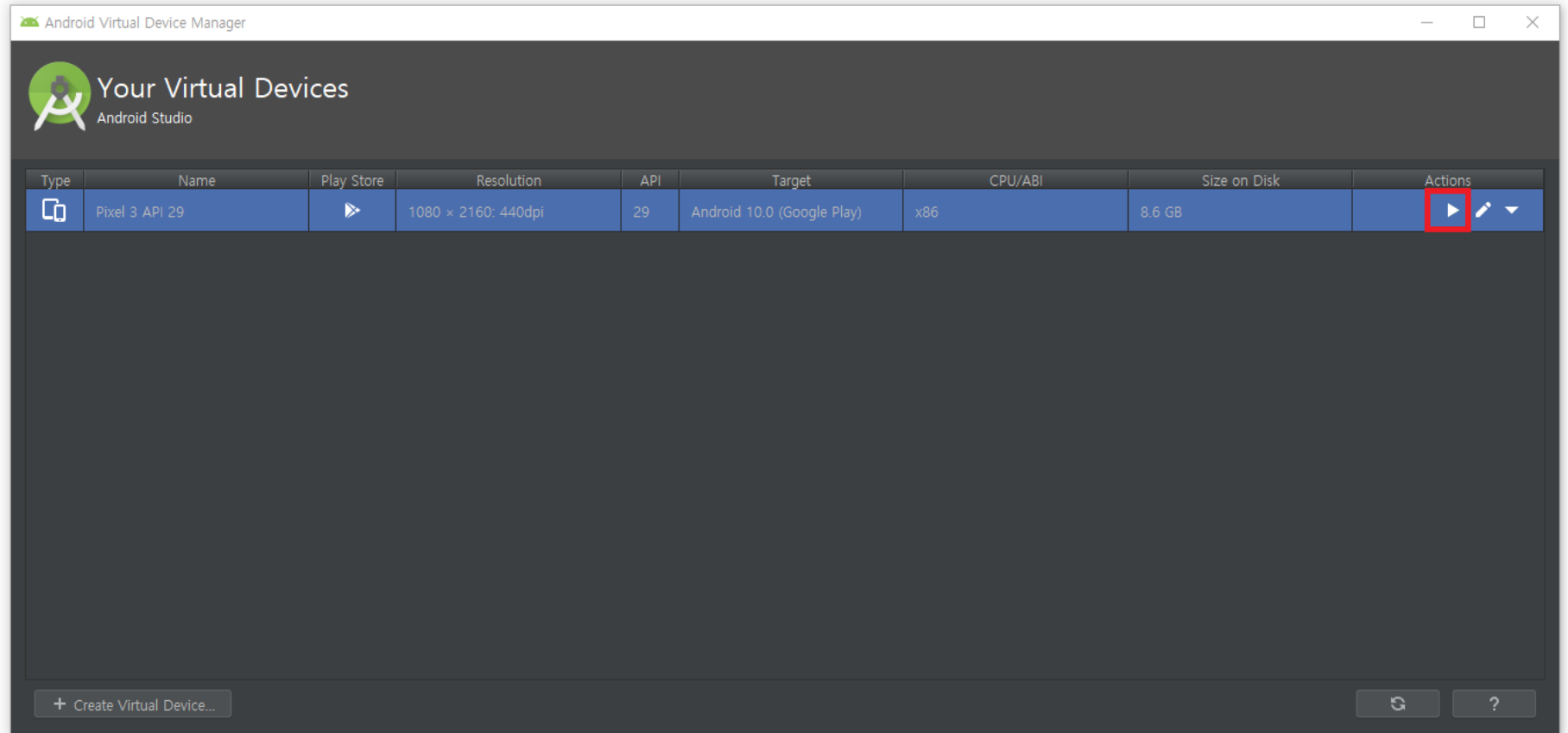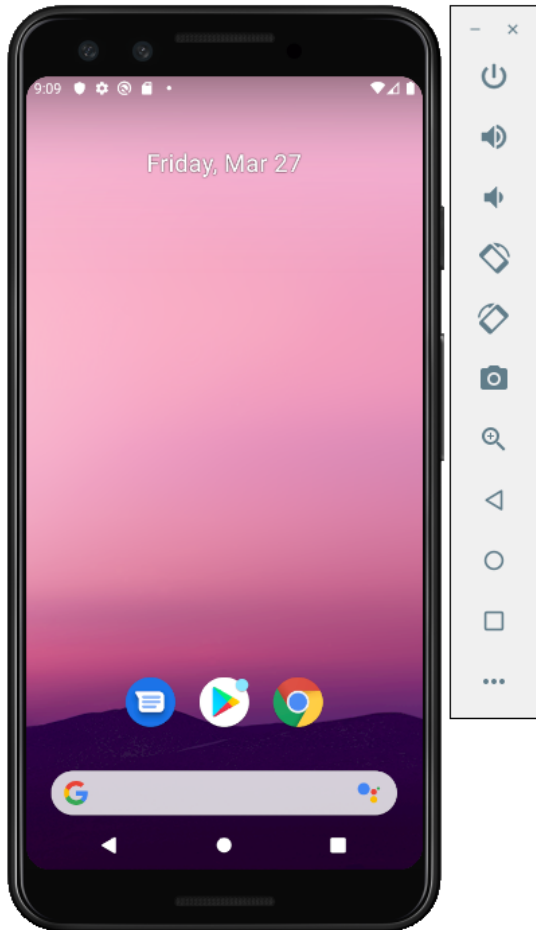Choose the device you prefer. (In my case, Pixel 3 with system Q.)

- If required, download the API.
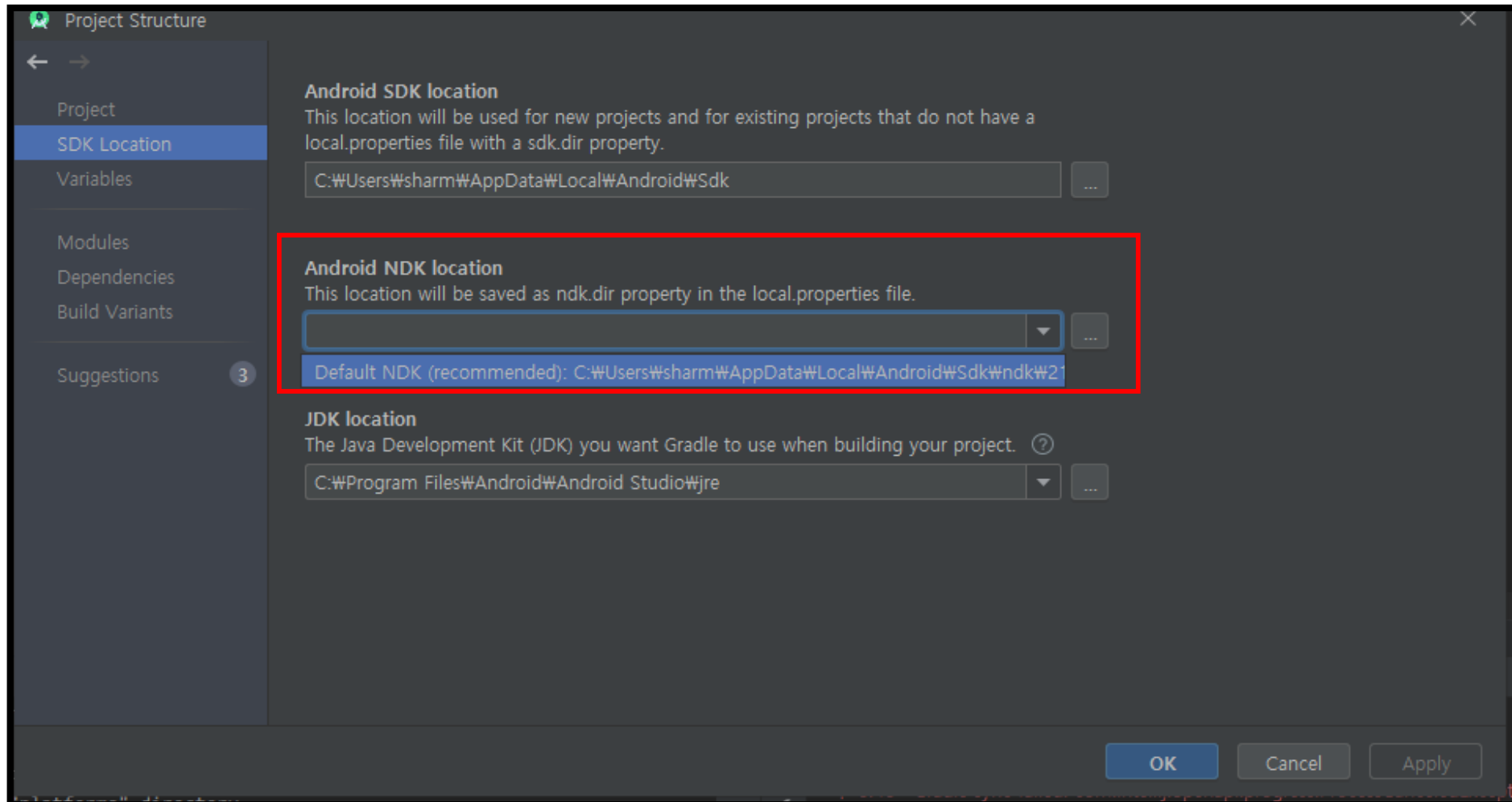
Create Virtual Device

After finishing the creation, click the launch button on the Actions tab.
Then, the virtual device shown on the left will appear.

NDK Location

Set a proper Android NDK location. (It varies depending on your computer setting!)

Ready?

Then click the [Run] – [Run] (or green triangle shown below.)

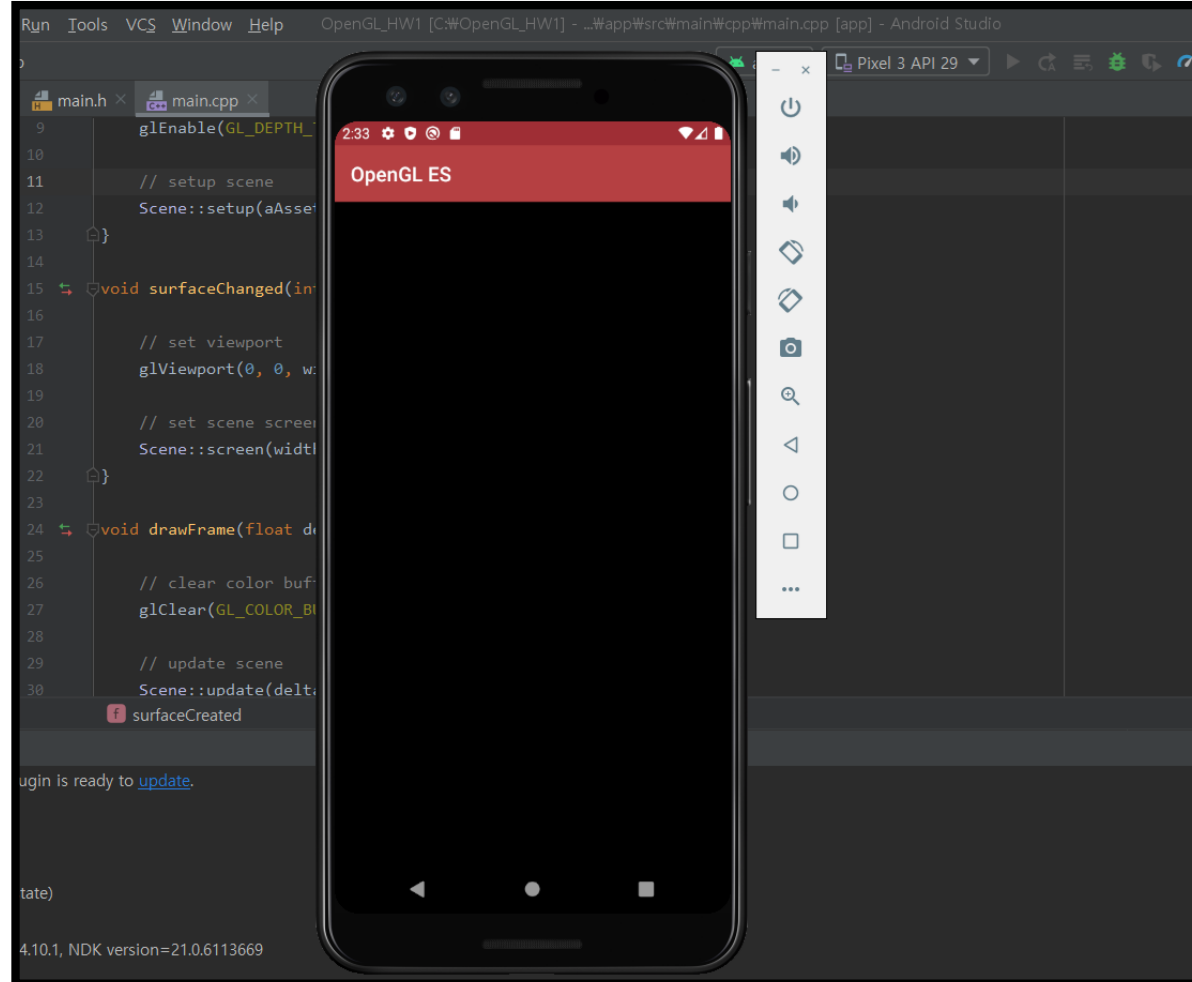- If required, several installation process will be performed (It takes several minutes.)

Ready?

The emulator shown bellow will appear. Now, you are ready.

Problem

Fill in lines.
- App/src/main/cpp/scene.cpp (setup and update)
- App/src/main/assets/vertex.glsl (main)

```cpp
    camera->eye = vec3(60.0f, 00.0f, 0.0f);

    // create light
    light = new Light(program);
    light->position = vec3(100.0f, 0.0f, 0.0f);

    // create floral texture
    flower = new Material(program, texFlowerData, texFlowerSize);

    // create teapot object
    teapot = new Object(program, flower, objTeapotVertices, objTeapotIndices,
                        objTeapotVerticesSize, objTeapotIndicesSize);

    //Set world matrix
    //teapot->worldMatrix;
}

void Scene::screen(int width, int height) {

    // set camera aspect ratio
    camera->aspect = (float) width / height;
}
```

```cpp
void Scene::update(float deltaTime) {
    static float angle = 0.0f;

    // use program
    program->use();

    // rotate the camera relative to the object
    // camera->eye = ;

    // setup camera and light
    camera->setup();
    light->setup();

    // draw teapot
    teapot->draw();
}
```

```glsl
uniform vec3 lightPos;

layout(location = 0) in vec3 position;
layout(location = 1) in vec3 normal;
layout(location = 2) in vec2 texCoord;

out vec3 v_normal;
out vec2 v_texCoord;
out vec3 v_lightDir;

void main() {

    // fill in the lines below
    //gl_Position = ;
    //v_normal = ;
    //v_texCoord = ;

    // do not touch below
    vec3 posWS = (worldMat * vec4(position, 1.0)).xyz;
    v_lightDir = normalize(lightPos - posWS);
}
```

Problem

Goal 1 (3points). Write your code at the setup() to transform the teapot as follows:

- Scale the teapot by 1.0, 2.0 and 0.5 along the x, y and z axis respectively.
- Rotate the teapot by 50° around the axis, (1, 1, 0).

Goal 2 (3points). Write your code at the update() to rotate the camera around the y-axis
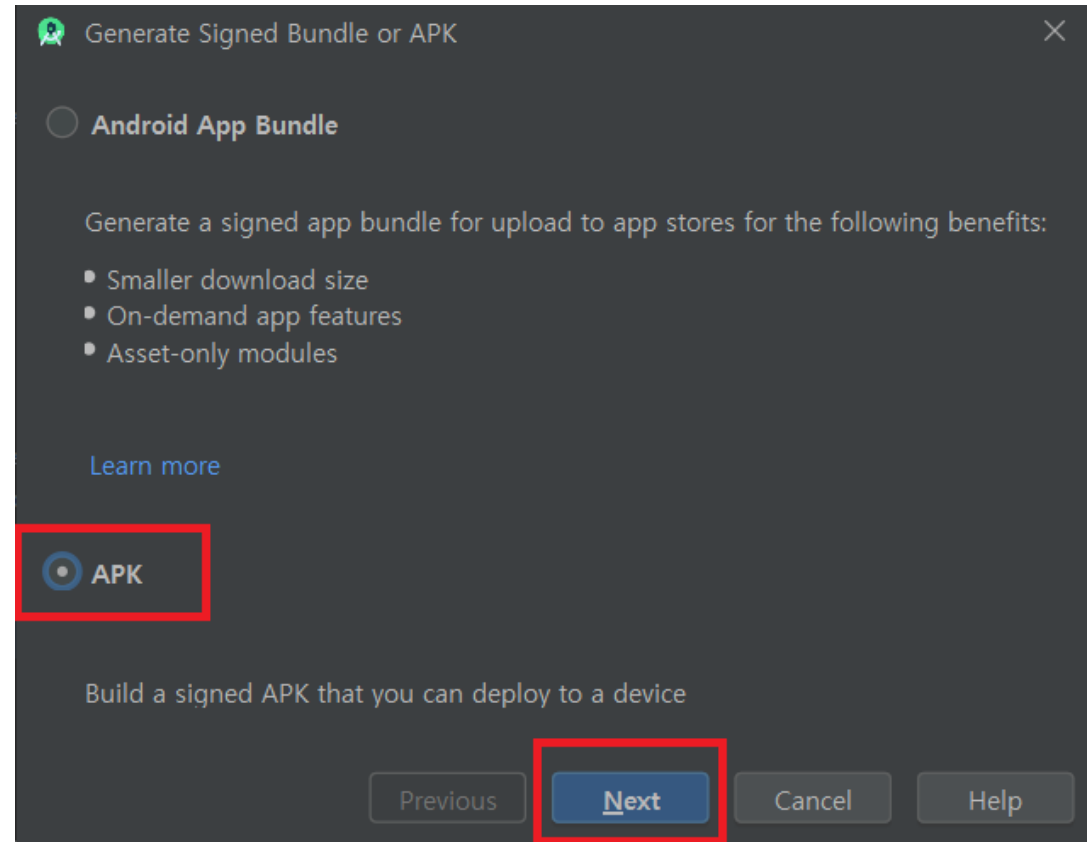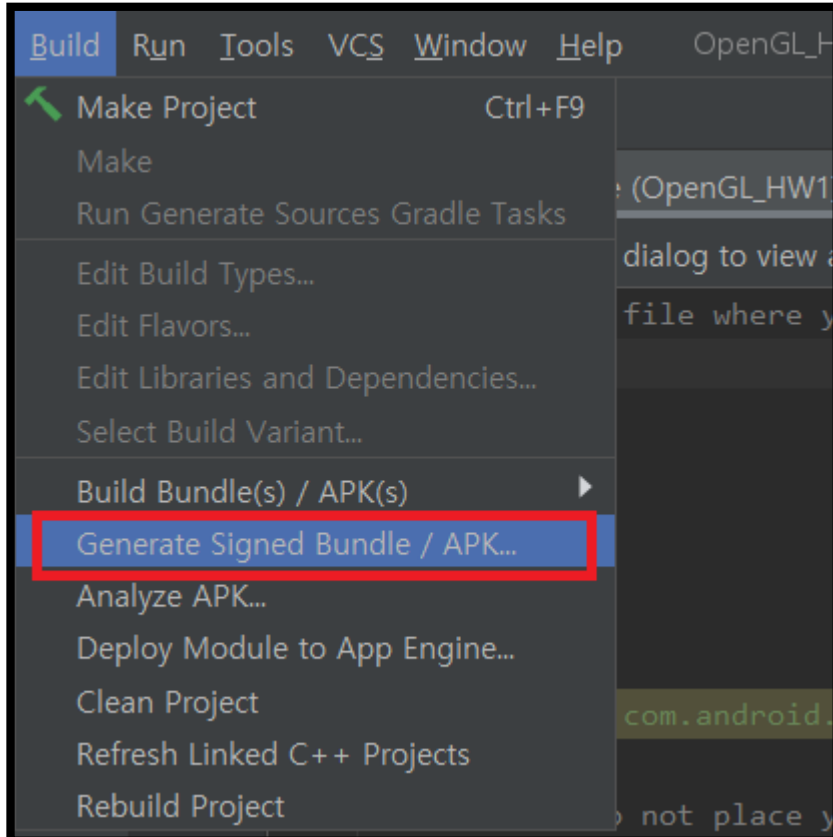
- Rotate the teapot by 2° for every update.

Goal 3 (4points). Write your code at the .glsl main().

- The clip-space vertex position must be calculated.
- The world-space vertex normal must be calculated accurately considering a non-uniform scaling (note that the lighting will look weird if vertex normals are incorrect.)
- The texture coordinates will be sent to the rasterizer stage with no change.

Note: do not use rotate() function!

Generate APK

Generate APK
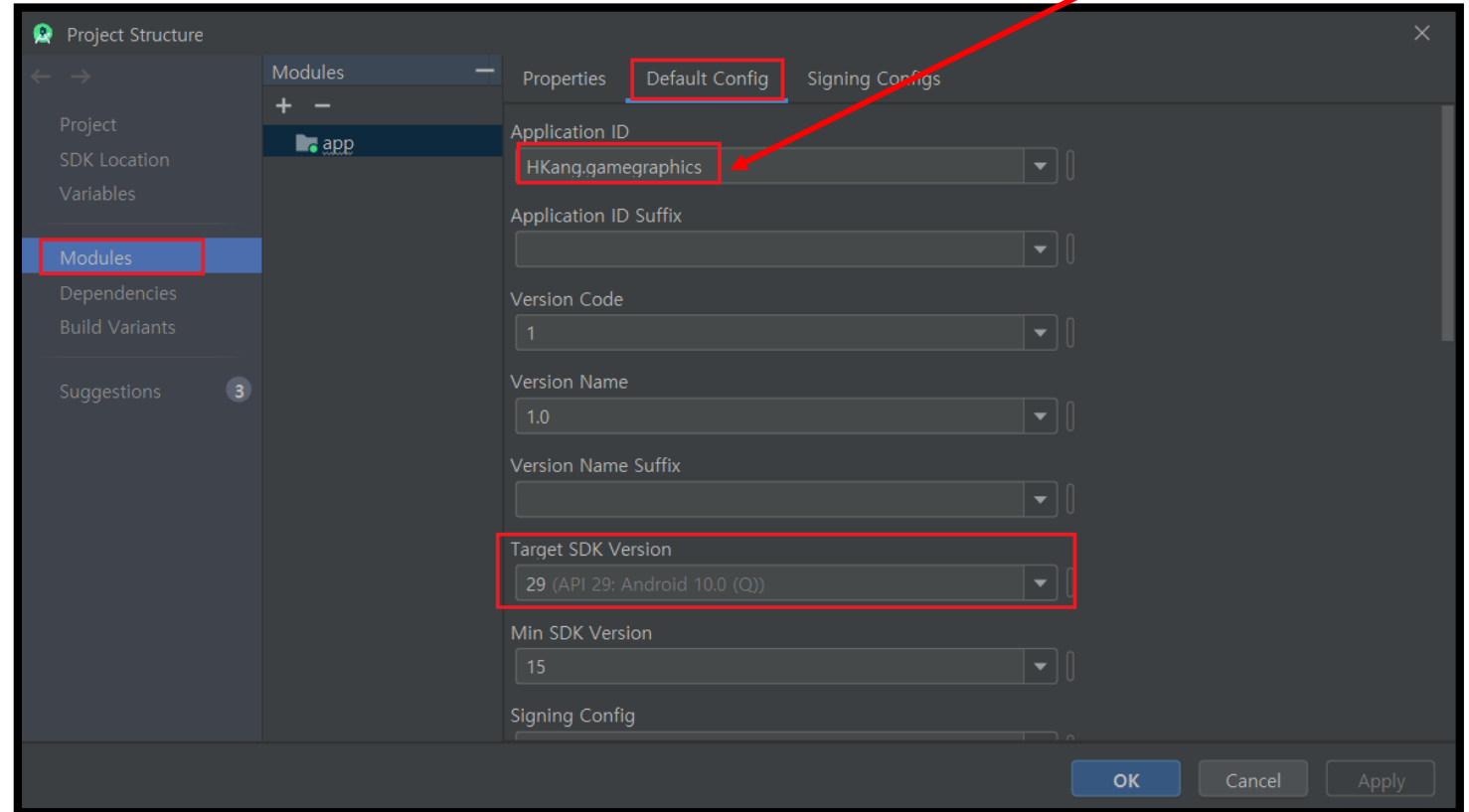
Generate APK

You have an error? Google may help you to generate APK ☺
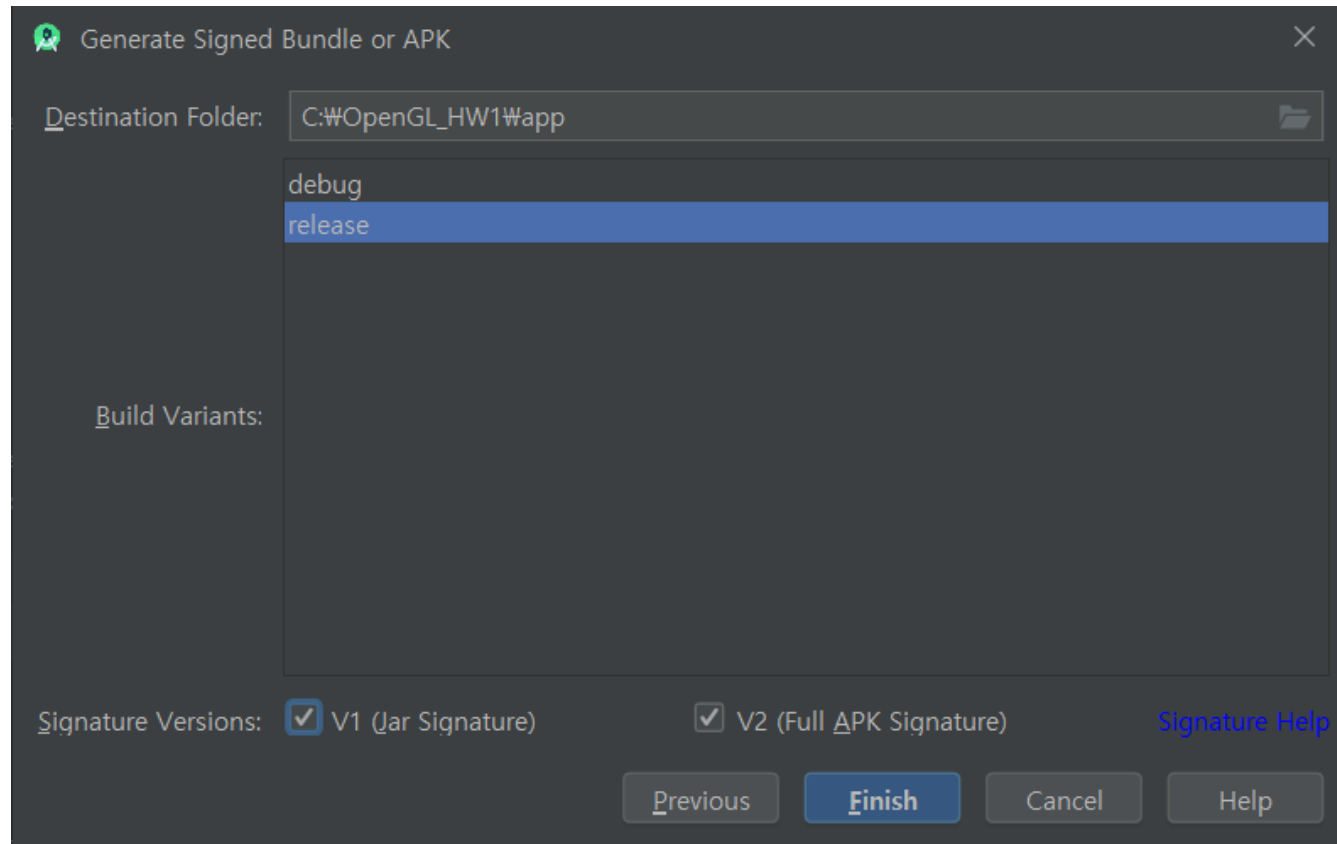
Homework 1
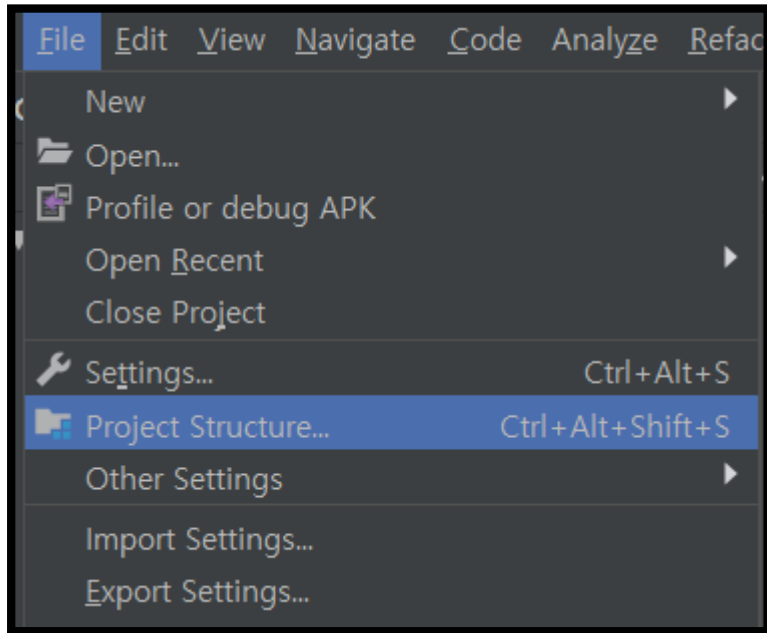
Compress these files with 'password' and upload to your git.
(Password should be uploaded through klas)

Submission

Deadline
- 4. 24. 23:50 (important!)

Submit followings to klas.
- Make an apk file and upload it to your git repository. Then, share your git URL and password.
  - ✓ APK does not work = 0 point!
- {student_number}_{name}.zip including vertex.glsl and scene.cpp.
  - ✓ Does not submit code = 0 point!

TA
- 이정은 (jeunlee0306@khu.ac.kr),

Office hour
- Monday, Wednesday 2:00 PM ~ 5:00 PM
- Contact TA by email before you visit.