Guideline for Android Development

[PROJECT] Planetarium\_Nine Chronicles\_Co-dev

Exported on Jan 06, 2023

Table of Contents

1 Overview 3

2 Unity Editor 4

3 Android Studio 5

4 ADB (Android Debug Bridge) 6

5 Android Debug in Visual Studio 7

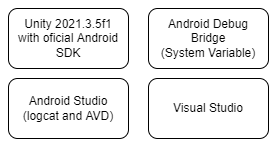
6 Build Settings (IMPORTANT) 8

6.1 Problem 1: Keystore 8

6.2 Problem 2: Change the backend between IL2CPP and mono 8

6.3 Problem 3: API level and storage problem 8

# Overview



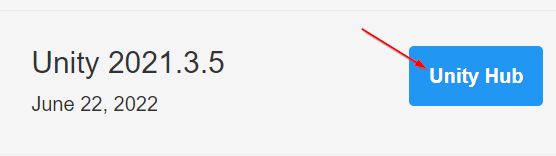
We need these four tools to develop the android game.

# Unity Editor

Download unity 2021.3.5f1 from Unity hub.

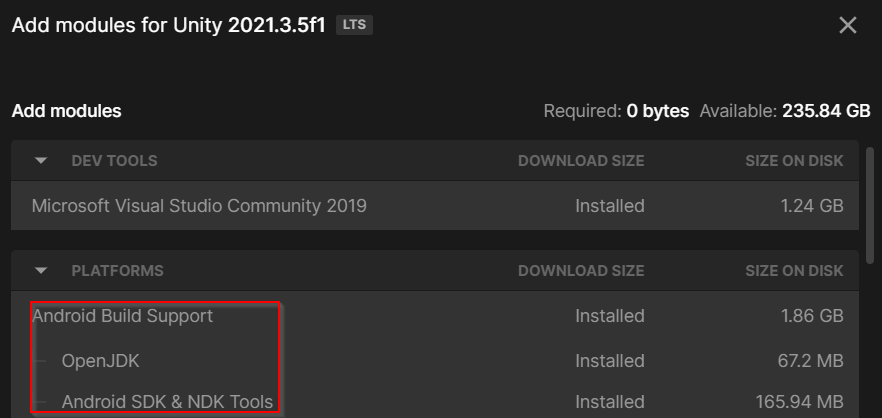
Firstly, install Unity hub. Then click the "unity hub" button on this website:

[Download Archive (unity.com)](https://unity.com/releases/editor/archive)



Only downloading by this method will enable you **to download the official Android SDK package**.

Then choose all android build support in Unity hub to download.



# Android Studio

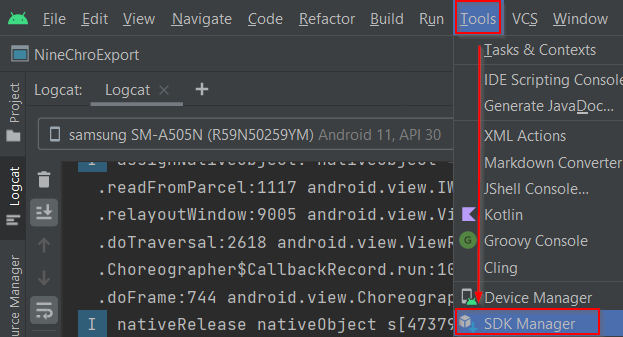
Just download the latest version of Android Studio. We don't talk about details because tutorials are everywhere on the internet.

We only use four tools in Android Studio:

1. Logcat (to display logs from Android and take screenshots)
2. Device Manager (to create Android Virtual Device)
3. Device File Explorer (to view file tree on Android device)
4. **SDK Manager** (to manage Android SDK)

Primarily, we need to care about SDK Manager.

**An Android SDK is a folder holds all Android development dependencies you chose.** Only **SDK Manager (see the below pic)** can manage an Android SDK.



First, to download Unity's official Android SDK. Unity official Android SDK is a special Android SDK that contains basic dependencies for building a unity Android game. However, it's not enough for our development (see this: [storage problem](#scroll-bookmark-5)).

**Run Android Studio as administrator first** andUse SDK Manager to localize unity Android SDK, then add the following contents to it:

* Android SDK Build-Tools-28.0.x (Please download all **28.0.x** tools).

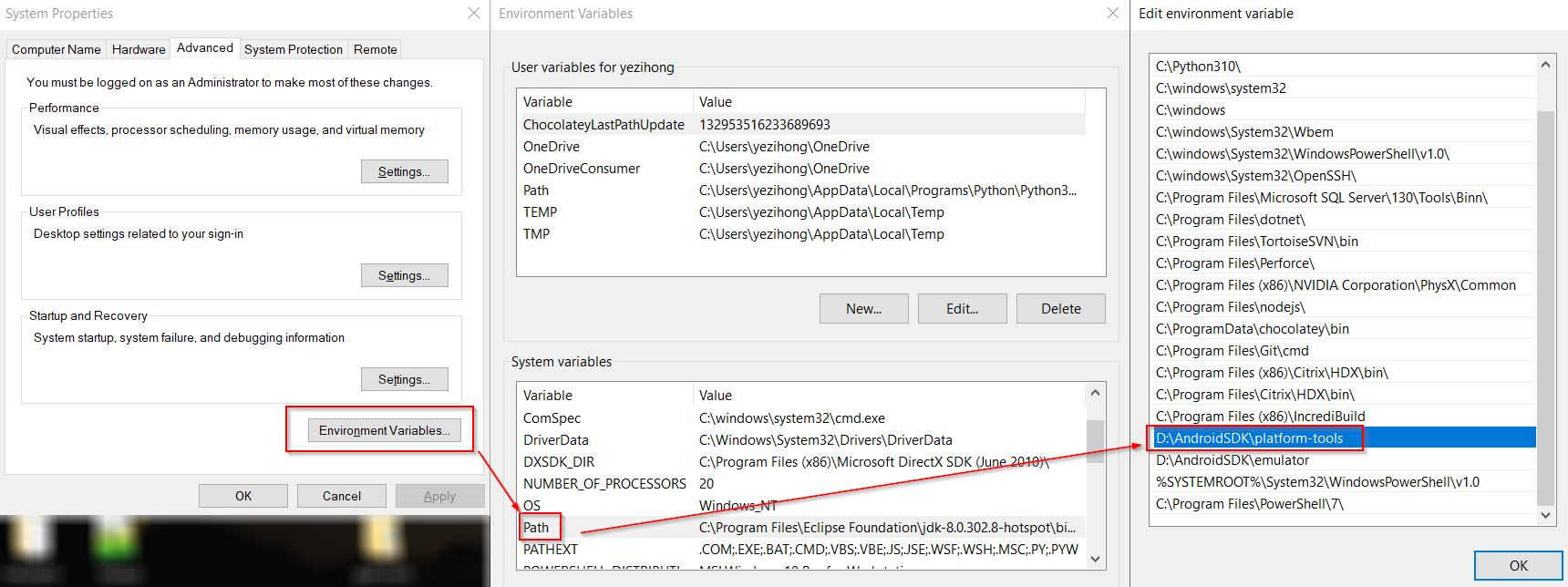
If you can't do this due to system permission, please run Android Studio as administrator.

# ADB (Android Debug Bridge)

We use ADB to install the APK package and support the use of logcat.

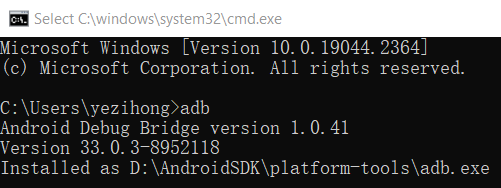
Firstly, find ldb.exe in your Android SDK (e.g, "D:\AndroidSDK\platform-tools\adb.exe"). The path format is **"???\YourAndroidSDKRootFolder\platform-tools"**.

Press **Win+S**, input "advanced settings", and press **Enter**. Follow the steps in the below picture:



Then click apply and ok.

Finally, open CMD and input "**adb**", you could see something like this:



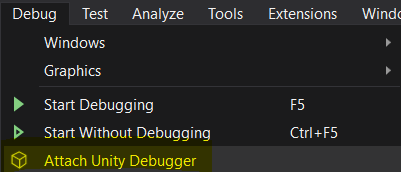
# Android Debug in Visual Studio

Check these two options when building the game.



Open CMD and input "**adb devices**", you should see the device ID when the device is ready to debug.

Then run the game on AVD (android virtual device) or on the android device. Click "**Attach Unity Debugger**" in VS (see the below picture). Choose the right project, then you can debug your game instance.

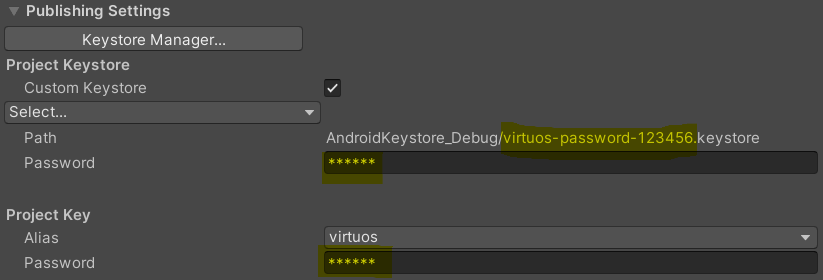


# Build Settings (IMPORTANT)

## Problem 1: Keystore

You should configure Keystore in "**Project Settings → Player → Publishing Settings**". The default password is "123456" (see the below picture). You can use your own Keystore by the same method.

Unity can't remember the password of Keystore so you should enter the password when you run the editor every time.



## Problem 2: Change the backend between IL2CPP and mono

Mono is a cross-platform virtual machine. IL2CPP compiles C# code to MSIL (Microsoft Intermediate Language) and to C++ finally. The theory underneath is complicated, but it is easy to switch between them.

Go to "**Project Settings → Player → Other Settings**", you can choose mono or IL2CPP in "**Scripting Backend**". This will only affect the build result (because the editor always runs on mono).

## Problem 3: API level and storage problem

Go to "**Project Settings → Player → Other Settings → Target API Level**", and change this to "**Android 9.0 'Pie' API Level 28**". We use this lower target API level because storage permission management is complicated at the newer API level. By using API 28, we can use legacy storage management in a game where it's much easier to store data in virtual external storage ("/storage/emulated/0/...").