Android Studio使用基础

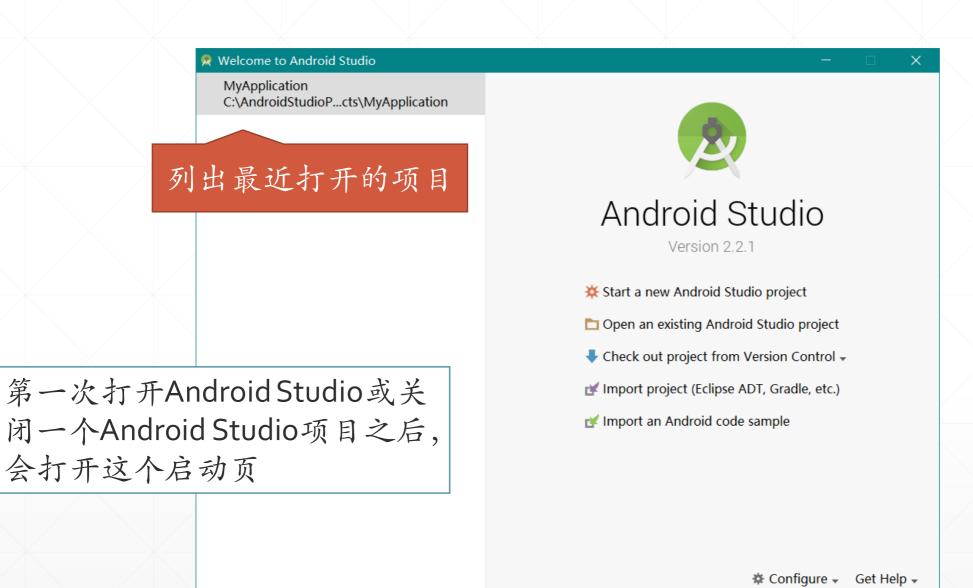
北京理工大学计算机学院金旭亮



|Android Studio是Google官方指定的Android开发环境, 是开发Android本地应用(Native APP)的主要工具



至2016年10月为止, Android Studio的最新版本是2.2.1, 本PPT介绍它的基本使用方法。



点击此可从下拉菜单中选择命令 配置Android Studio

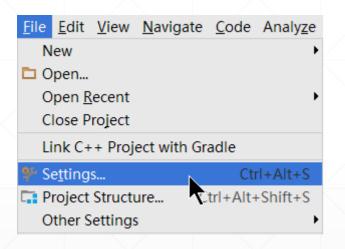
在启动页右下角可以进行的配置功能

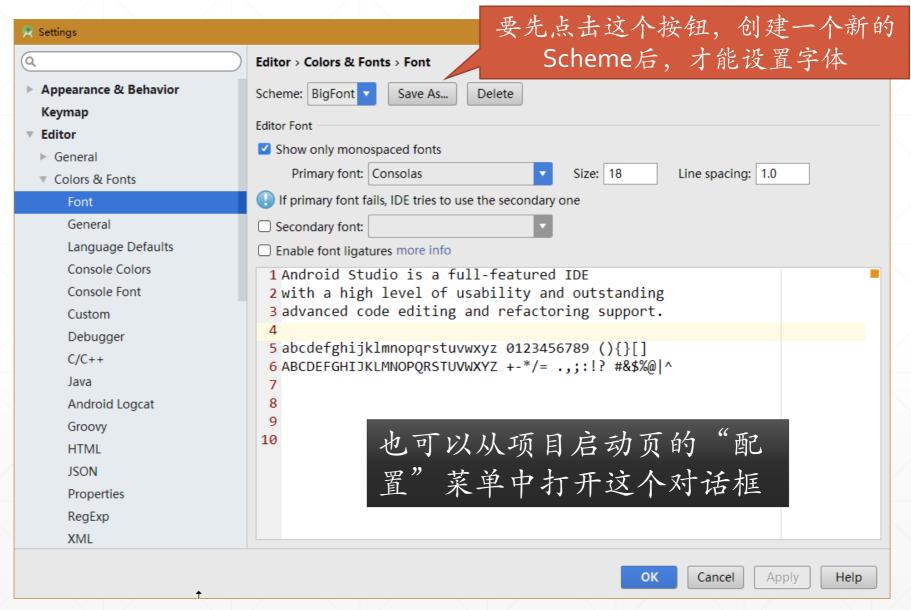


基本配置

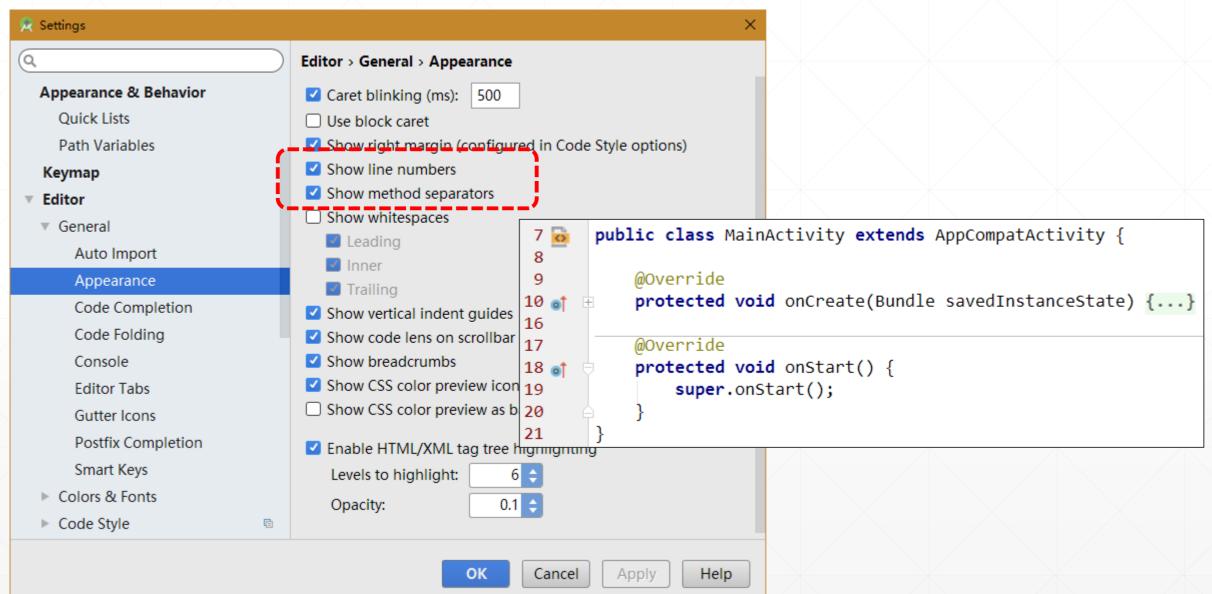
设置字体

从File菜单打开"配置"对话框





显示行号和方法分隔线



Android Studio应用开发流程

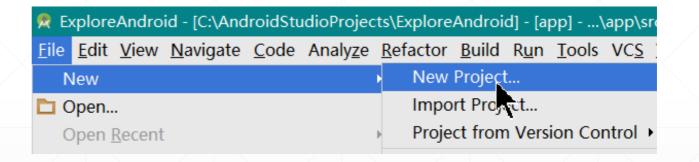
新建项目



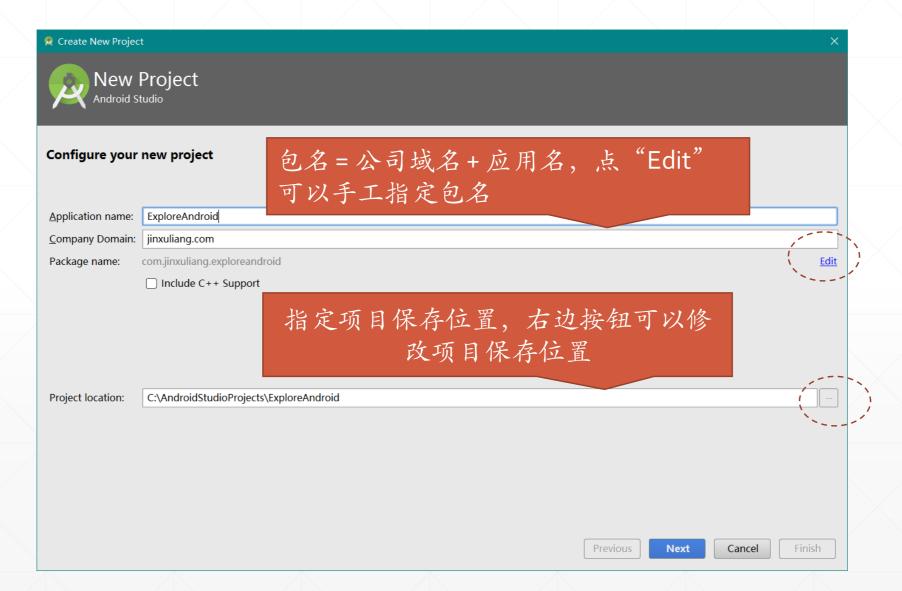
Android Studio

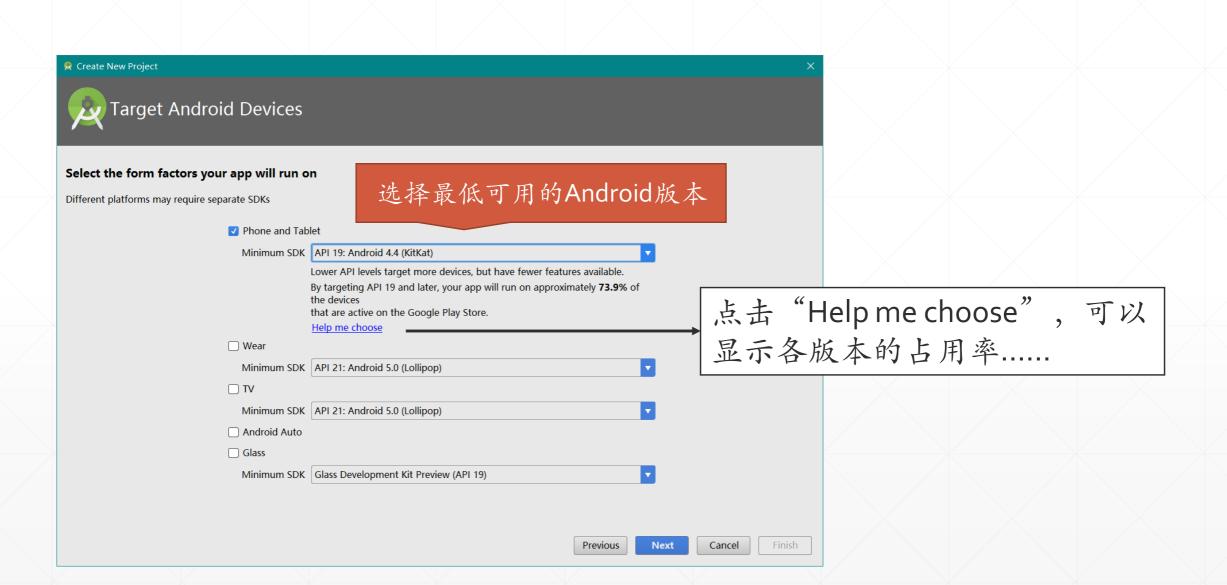
Version 2.2.1

- ★ Start a new Android Studio project
- Dopen an existing Android Studio project
- ♣ Check out project from Version Control ▼



设置项目名称与位置





ANDROID PLATFORM	API LEVELCUMULATIVE			
2.3 Gingerbread	10	STRIBUTION		
4.0 Ice Cream Sandwich	15	97.4%		
4.1 Jelly Bean	16	95.2%		
4.2 Jelly Bean	17	87.4%		
4.3 Jelly Bean	18	76.9%		
4.4 KitKat	19	73.9%		
5.0 Lollipop	21	40.5%		
5.1 Lollipop	22	24.1%		
6.0 Marshmallow	23	4.7%		

KitKat

Printing Framework

Print generic content Print images OEM print services

SMS Provider

Read and write SMS and MMS messages Select default SMS app

Wireless and Connectivity

Host NFC card emulation NFC reader mode Infrared support

Multimedia

Adaptive video playback
On-demand audio timestamps
Surface image reader
Peak and RMS audio
measurements
Loudness enhancer
Remote controllers
Closed captions

Animation and Graphics

Scenes and transitions Animator pausing Reusable bitmaps

User Content

Storage access framework External storage access Sync adapters

User Input

New sensor types, including step detector Batched sensor events Controller identities

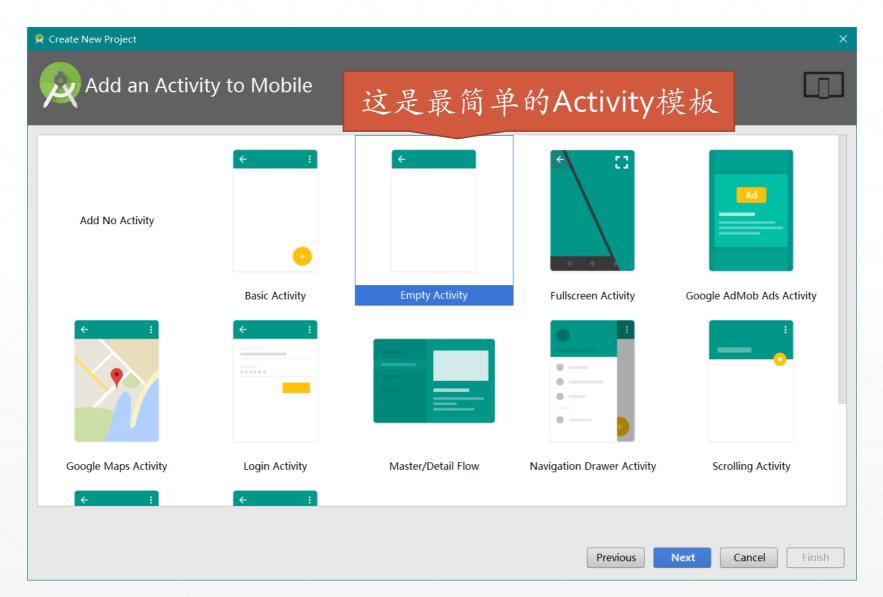
User Interface

Immersive full-screen mode Translucent system bars Enhanced notification listener Live regions for accessibility

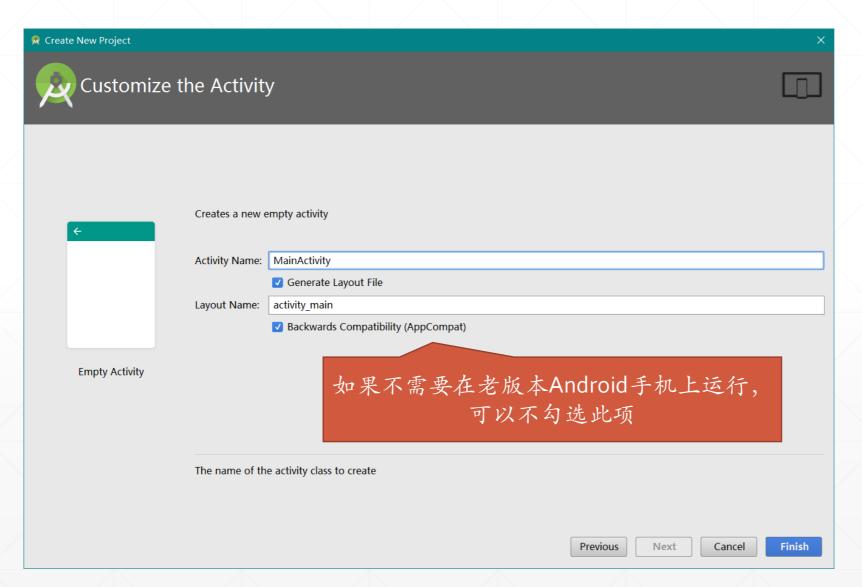
截止2016年10月Android 各版本的市场占有率

https://developer.android.com/about/versions/android-4.4.html

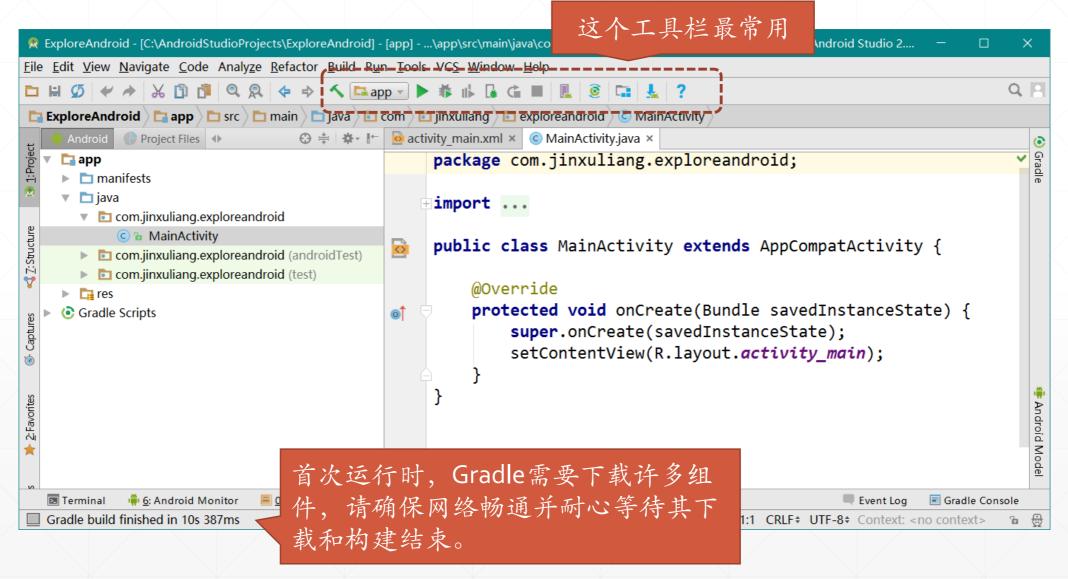
向项目中添加Activity



指定Activity和布局文件名



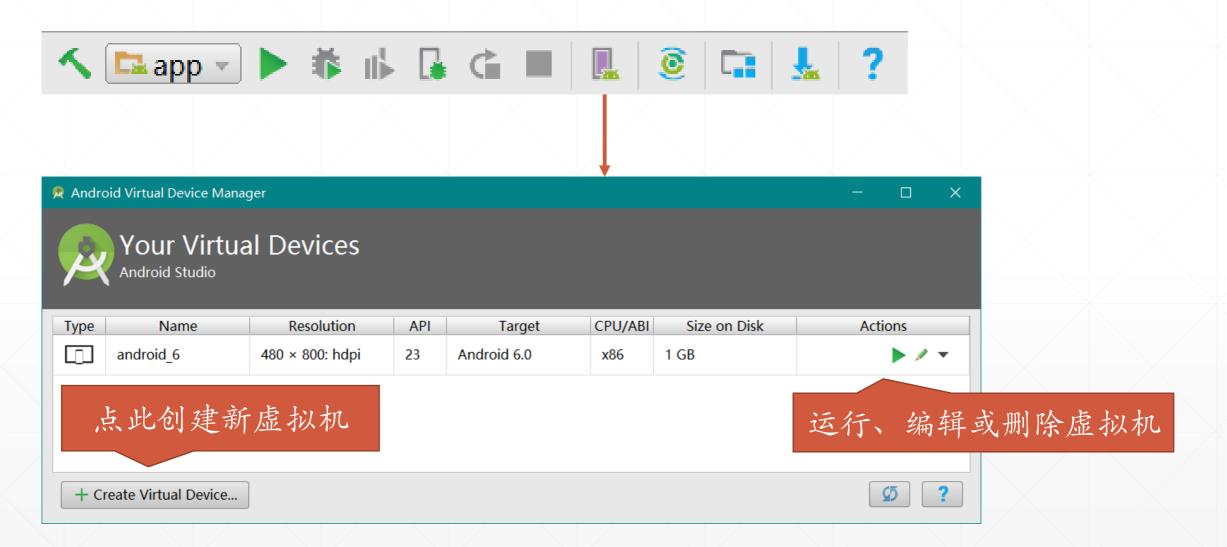
生成的项目模板



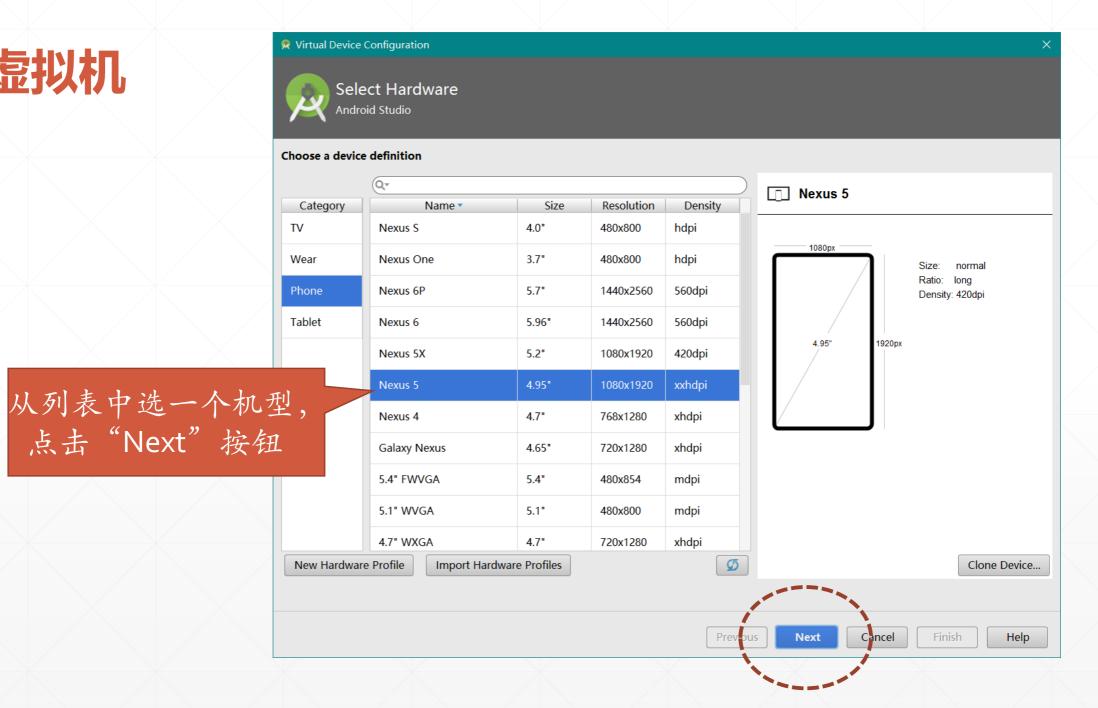
开发必用之工具栏命令



创建虚拟机



创建虚拟机





Virtual Device Configuration

指定虚拟机使用的Android版 本,已下载好的以粗体显示

Select a system image

Recommended x86 Images Other API Level ▼ Release Name ABI Target Nougat 24 x86 Android 7.0 **Nougat** Download 24 x86 64 Android 7.0 Marshmallow 23 x86 Android 6.0 Marshmallow Download 23 x86 64 Android 6.0 Lollipop Download 22 x86_64 Android 5.1 Lollipop Download 22 x86 Android 5.1 Lollipop 21 x86 Android 5.0 (with Google A Lollipop Download 21 x86 64 Android 5.0 (with Google A Lollipop 21 x86_64 Android 5.0 Lollipop x86 Android 5.0 21 KitKat Download 19 x86 Android 4.4 (with Google A KitKat Download Android 4.4 19 x86 Jelly Bean Download 18 x86 Android 4.3 (with Google A

API Level
24

Android
7.0

Android Open
Source Project

System Image
x86

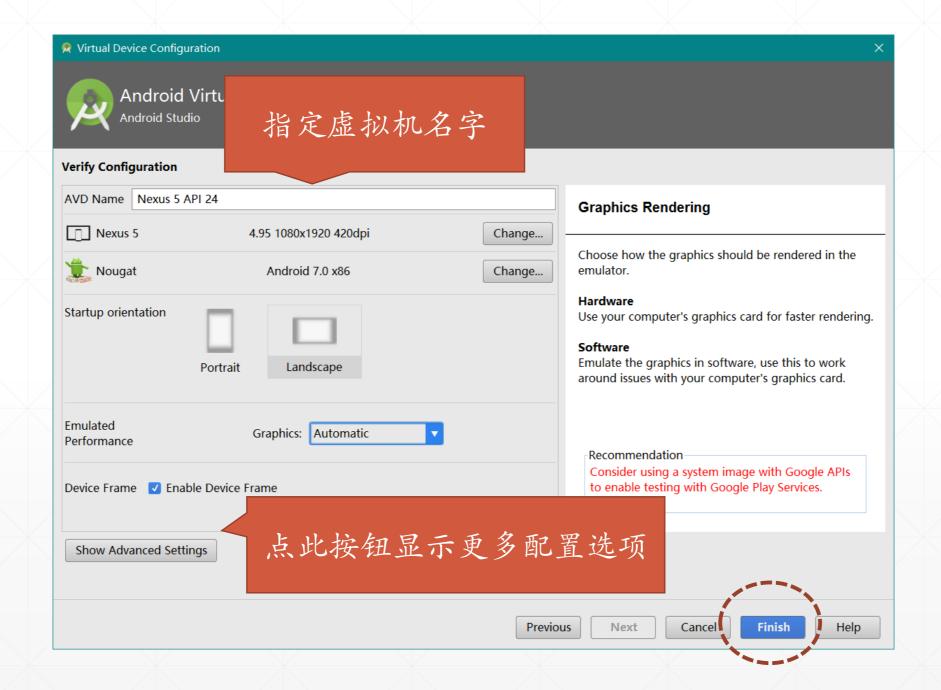
Recommendation
Consider using a system image with Google APIs to enable testing with Google Play Services.

See the API level distribution chart

Questions on API level?

ινυυyaι







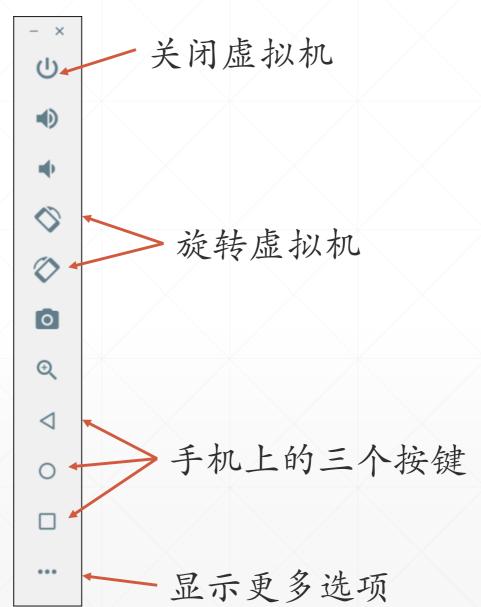


Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	android_6	480 × 800: hdpi	23	Android 6.0	x86	1 GB	▶ / ▼
	Nexus 5 API 24	1080 × 1920: xxh	24	Android 7.0	x86	650 MB	▶ / ▼

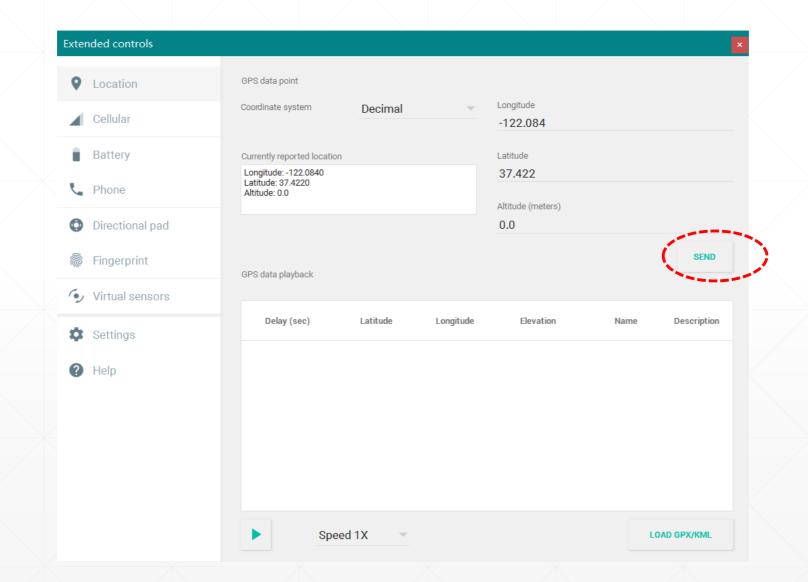
+ Create Virtual Device...

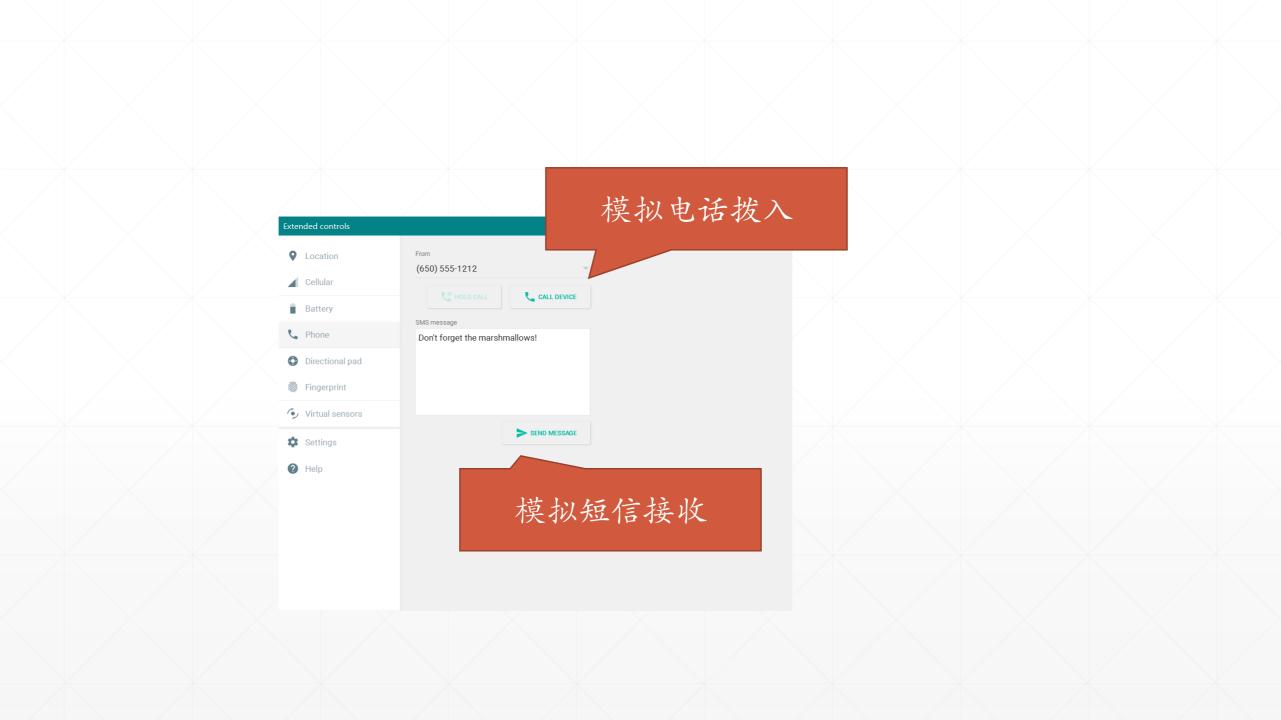
选中创建好虚拟机,点击"绿色三角"按钮启动它



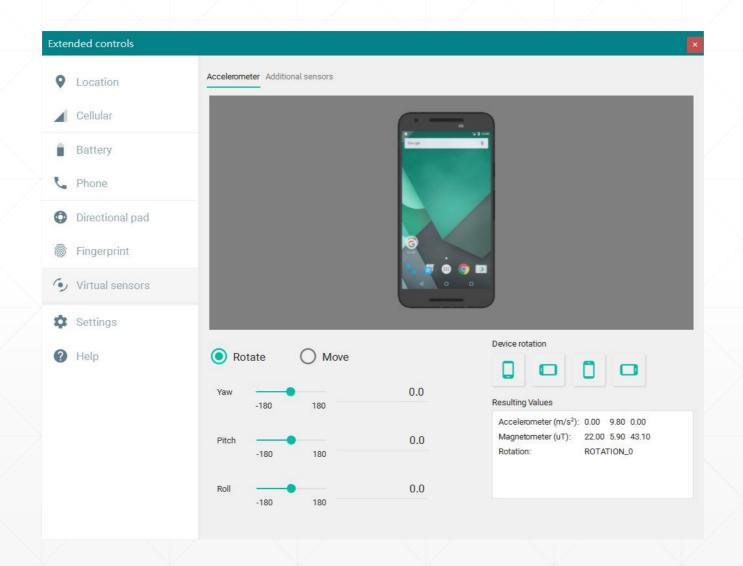


配置GPS位置座标,模拟地理位置功能

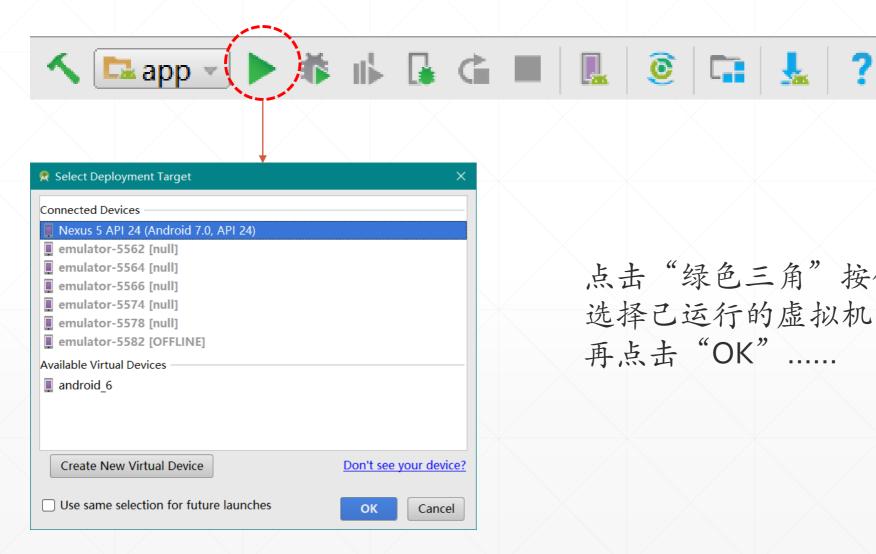




模拟各种传感器



运行程序



点击"绿色三角"按钮, 选择已运行的虚拟机, 再点击"OK"

等待Gradle构建项目, 生成调试 用的APK, 再部署到虚拟机上, 并自动运行之.....

点此结束示例程序的运行,

虚拟机重新显示桌面

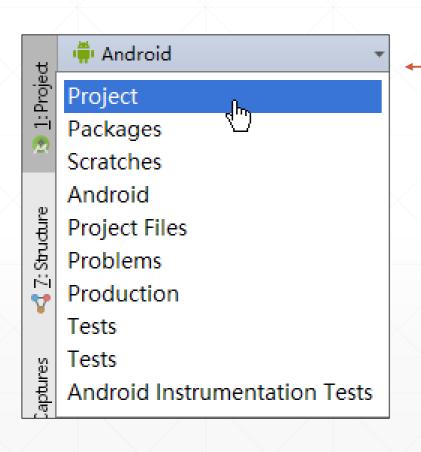


至此,使用Android Studio的一个完整的开发过程展示结束!

下面,我们展示一些Android Studio的使用技巧。

使用项目窗口 (Project Window)

Project Window用于展示项目成员

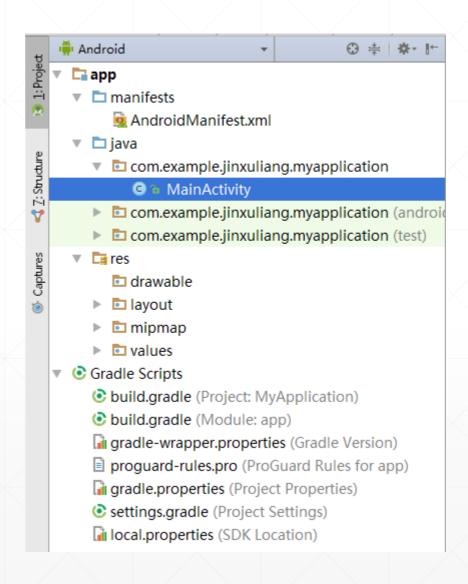


点击顶部下拉框,可以 切换视图

每一种视图,其实是以不同的方式呈现项目中的各种成员

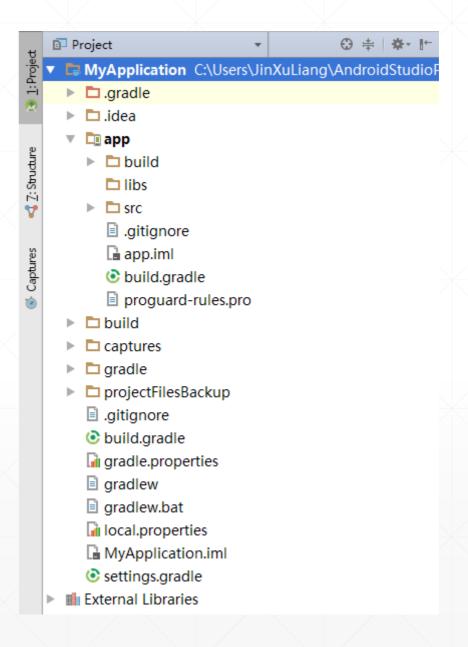
Android视图

这是最常用的视图,隐藏了 日常开发很少用到的文件, 将常用的文件分类以方便查 找和访问。



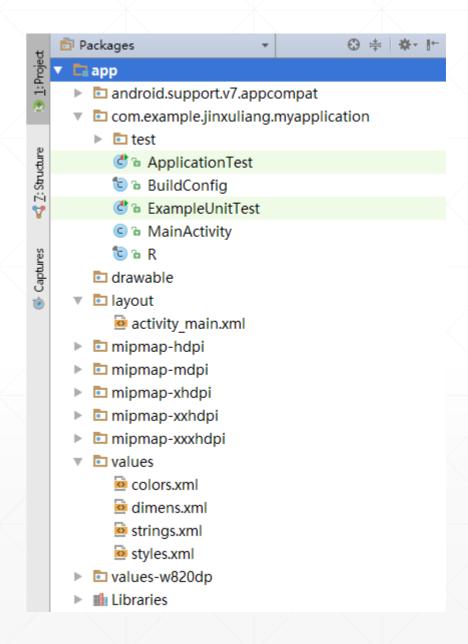
Project视图

展示真实的目录结构,与 Windows资源管理器中看 到的一样。

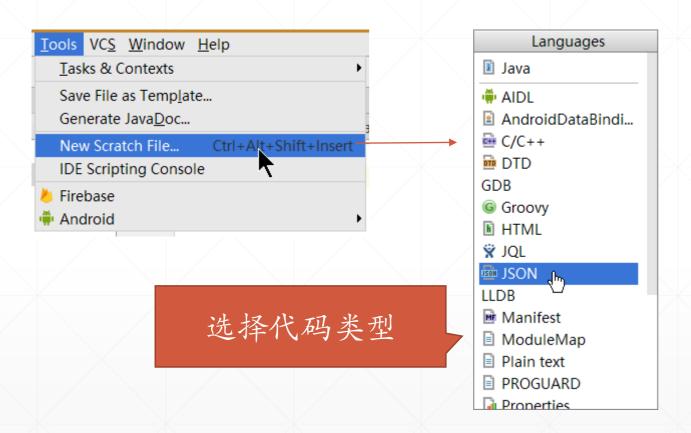


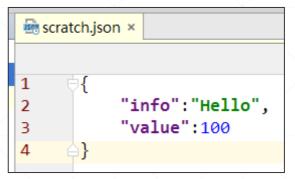
Packages视图

展示出项目中的所有包,第三方库和组件,以及相关的资源。



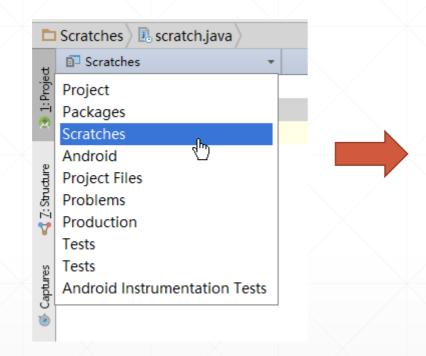
Scratches视图,可用于暂时保存一些代码



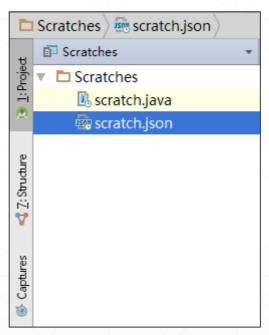


在打开的编辑窗口中输 入或粘贴代码

使用Scratches视图



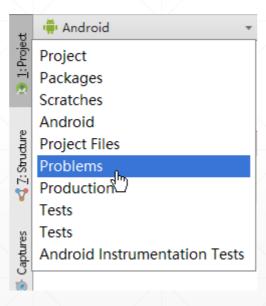
切换显示Scratches视图



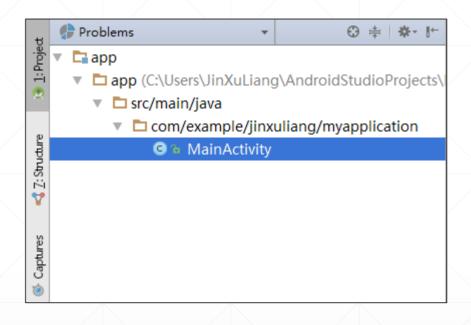
Scratches视图中展示了所有 代码片断,可以随时新建和 删除之

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toast.makeText(this,"中国人",Toast.LENGTH_LONG).show();
        nouse();
    }
}
```

代码中有错



切换到Problems视图



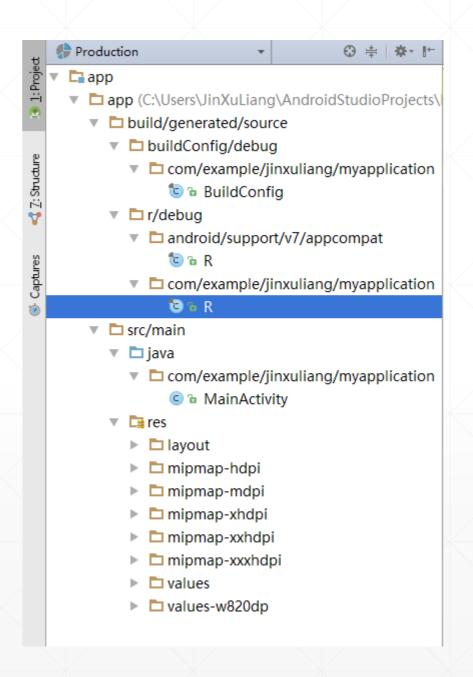
自动列出所有有错的文件,双击打开......

使用Problems视图

Production视图

可以用于查看IDE自动生成和 管理的一些文件(比如R.java)

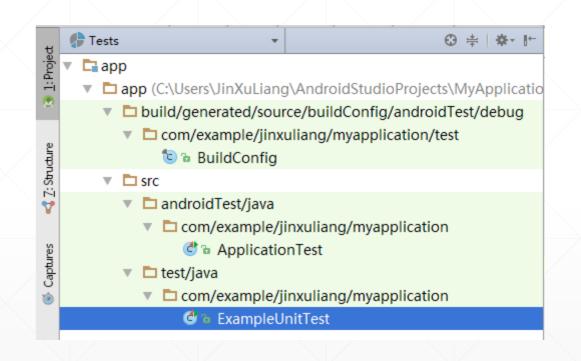
这些文件在发布时将会被打包到Apk中。



Tests视图

可以用于运行单元测试

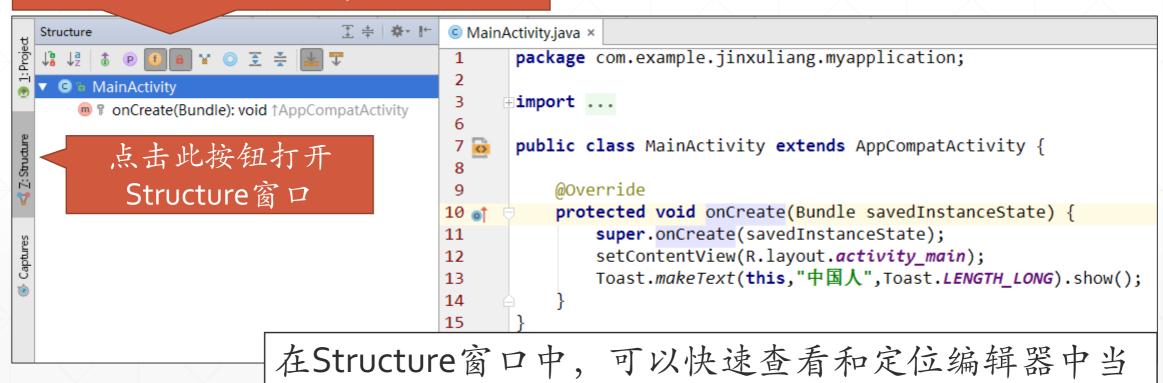
单元测试分为两类,一类是使用 JUnit针对纯Java SE代码编写的 测试,另一类是针对Android特 定代码的测试,需要使用 Android SDK所提供的特定测试 组件。



常用操作与开发技巧

查看代码全貌

工具栏上的这些按钮,可以过滤要显示的成员,比如只显示public成员



前打开文件中的成员

快速查找特定的文件

Search Everywhere Double Shift

Project View Alt+1

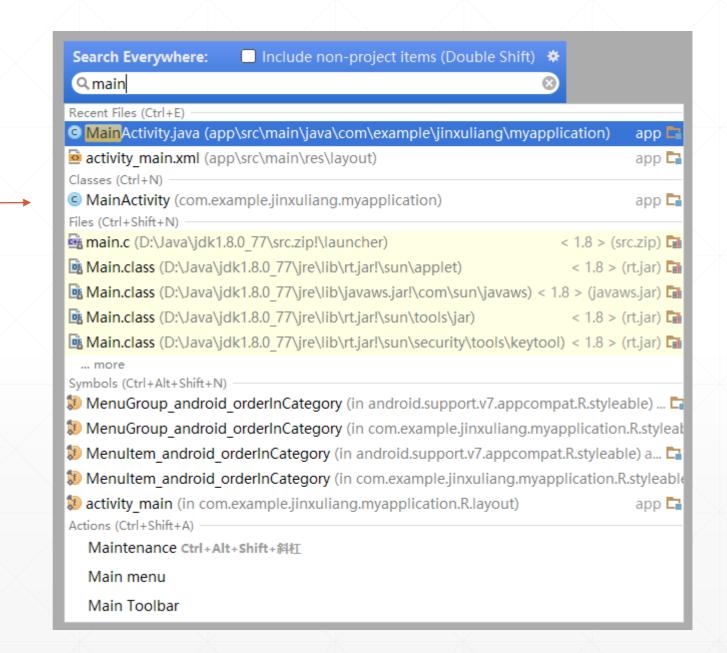
Go to File Ctrl+Shift+N

Recent Files Ctrl+E

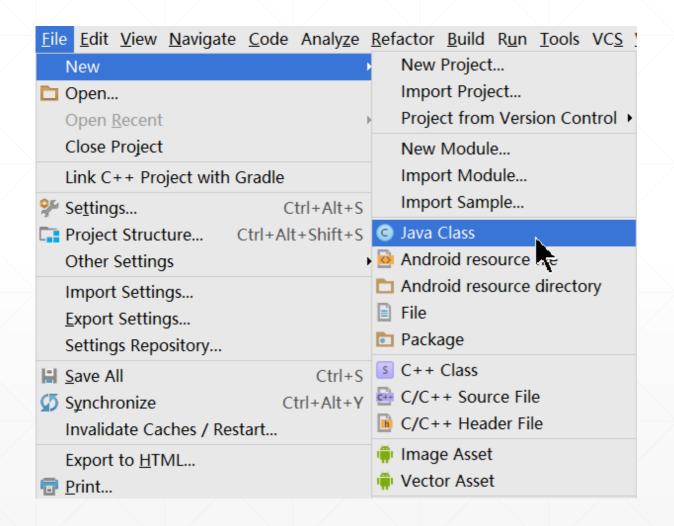
Navigation Bar Alt+Home

Drop files here to open

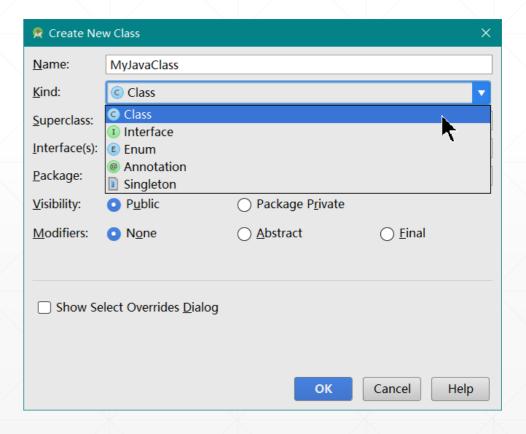
快速连按两下Shift键,可以查找文件、类等



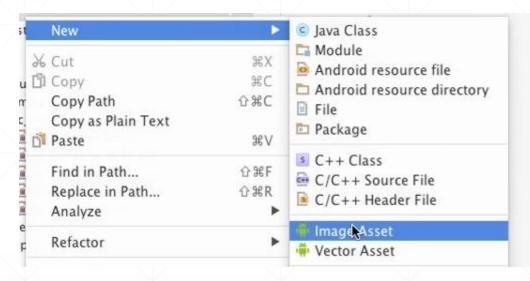
创建各种开发组件



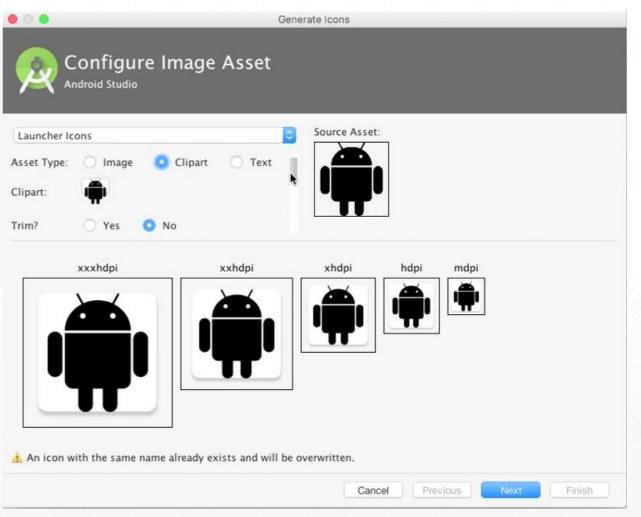
在File/New菜单中,包容创建各种开发组件的命令

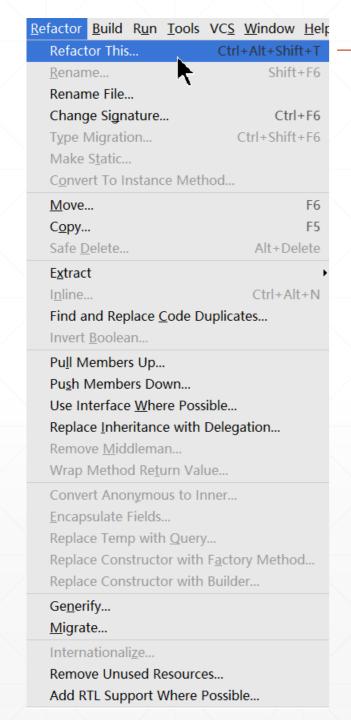


创建程序启动图标



使用Image Asset,可以创建显示在手机上的App图标





Refactor This	
<u>1</u> . Rename	ր _տ Shift+F6
2. Change Signature	☐ Ctrl+F6
<u>3</u> . Move	F6
<u>4</u> . Copy	F5
<u>5</u> . Safe Delete	Alt+Delete
Extract	
<u>6</u> . Delegate	
7. Interface	
8. Superclass	
9. Subclass	
0. Inline	Ctrl+Alt+N
Pull Members Up	
Push Members Down	
Use Interface Where Possible	
Replace Inheritance with Delegation	
Replace Constructor with Factory Method	
Generify	
Remove Unused Resources	

重构代码

Android Studio 集成了诸多 重构功能,使用方便。

将光标放到代码中的特定位置,Ctrl+Alt+Shift+T,可以调出重构快捷菜单,集中了最常用的重构功能

切换到相关的文件

点击这个图标,可以 快速地切换显示相关 联的文件

```
MainActivity.java ×
       package com.example.jinxuliang.myapplication;
1
       import ...
 6
       public class MainActivity extends AppCompatActivity {
         Go to Related Files
 9
           Layout Files
     activity_main.xml (layout)
                            onCreate(Bundle savedInstanceState) {
10
                           eate(savedInstanceState);
11
     AndroidManifest.xml
                          View(R.layout.activity_main);
12
                Toast.makeText(this,"中国人",Toast.LENGTH_LONG).show();
13
14
15
```

使用内置代码模板

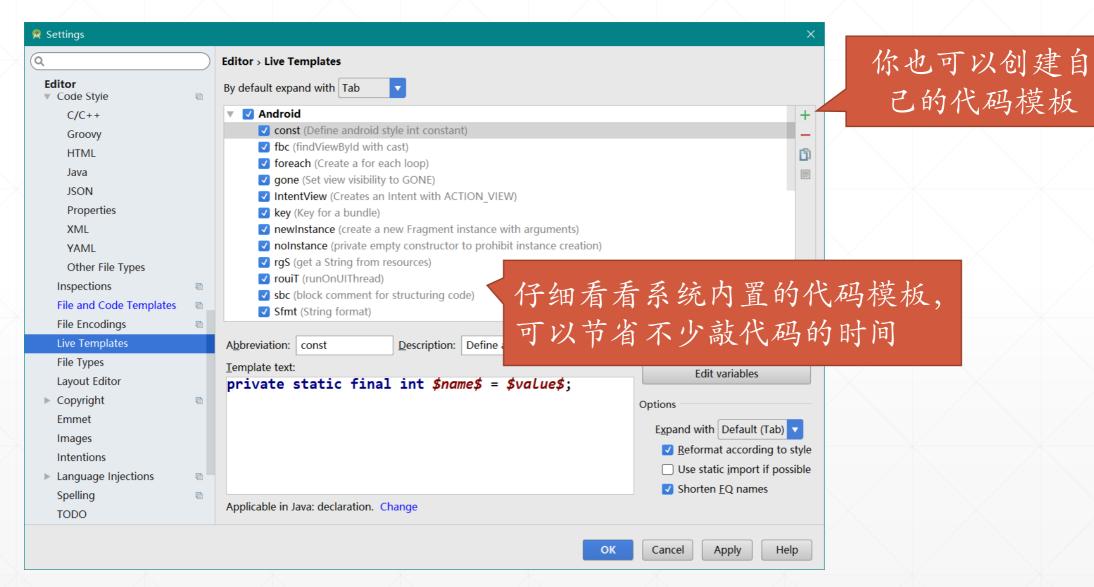
在代码中输入"const"再按Tab键, 会自动生成Java中的常量声明

```
public class MyJavaClass {
    con
    const Define android style int co...
    return x+y;
}
```

自动生成的代码框架

```
public class MyJavaClass {
    private static final int MYCONST=481;
    public int add(int x,int y){
        return x+y;
    }
}
```

查看Android Studio的内置代码模板



调出辅助代码编写快捷菜单

在代码空白处,按Alt+Insert键:

Generate

Constructor

toString()

Override Methods... Ctrl+O

Delegate Methods...

Copyright

App Indexing API Code

使用这个快捷菜单,可以快速 地创建类的构造方法、重写基 类的特定方法.....,很方便

一些有用的快捷键

₩ 选中多条语句,按ctrl+/,可以批量地给它们注释或取消注释

→ 立即显示智能提示: ctrl+p

```
String str="hello";
str.concat()

@NonNull String str
```

- 当代码格式混乱时,Ctrl+Alt+L,自动重新格式化代码。
- ✓ 上下移动整行或整个方法: ctrl + shift + 上下光标键

开发中常用窗口简介

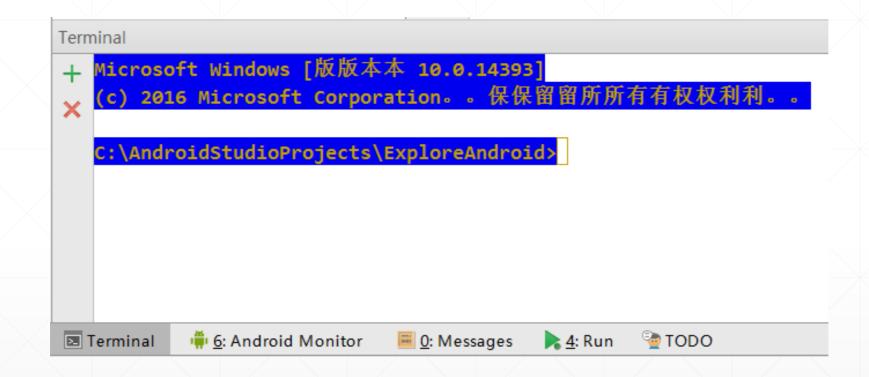
切换窗口

- Android Model
- Android Monitor
- Build Variants
- Captures
- Event Log
- * Favorites
- Gradle
- Gradle Console
- Messages
- Project
- ☆ Structure
- Terminal
- **™** TODO
- Gradle build finishe

每一个窗口,都有特定的用途,下面我们介绍一些常用的窗口。

鼠标移到左下角的灰色方块,会弹出一个菜单,可以用于切换窗口

终端窗口

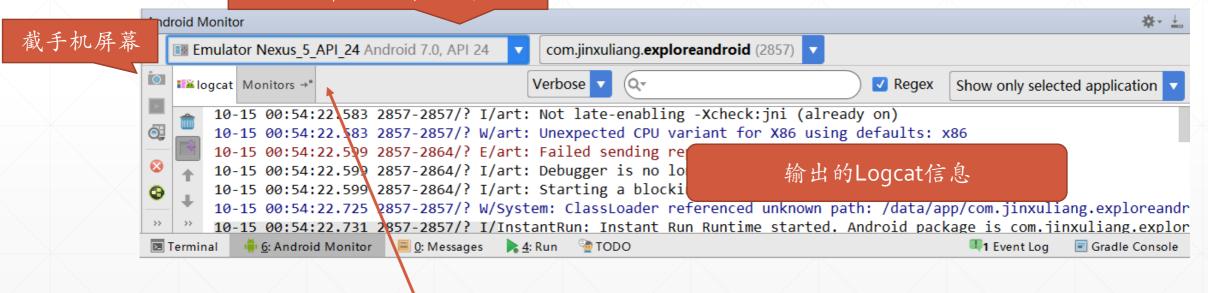


可以用于键入一些命令,与Window命令提示窗口用法一致,但当前版本的Android Studio在Windows 10下显示有BUG。

Monitor窗口

最重要的窗口,可以查看Log信息,占用内存,网速等,还可以截屏......

列出当所所连接的手机, 从中选一个查看



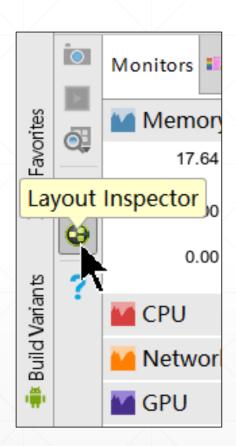
点击Monitor, 查看内存占用情况等相关信息.....

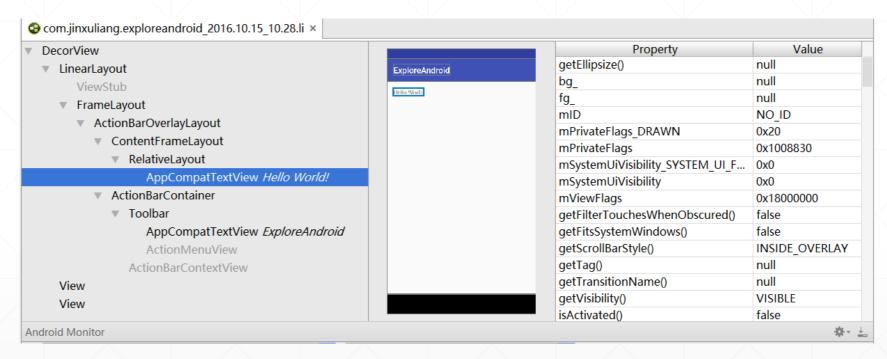
Monitors选项卡



可以用于查看内存占用, CPU、网络和GPU的使用情况, 对于程序性能优化很有用。

Layout Inspector





可以使用Layout Inspector查看真实的组件嵌套情况,从而优化界面装载时间。

TODO窗口

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

//TODO:这个代码还没完成
@Override
    protected void onStart() {
        super.onStart();
    }

}

TODO: Project Current File Scope Based

** Found 1 TODO item in 1 file
    ** Come.example.jinxuliang.myapplication (1 item in 1 file)
    ** © MainActivity.java
    ** (MainActivity.java
    ** (MainActivity.java
    ** (MainActivity.java)
```

❤ TODO

6: Android Monitor

0: Messages

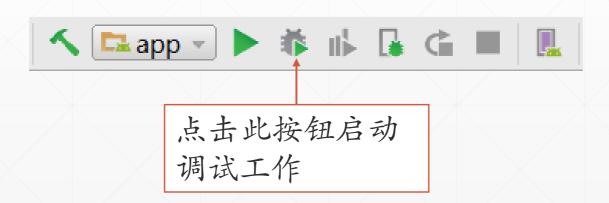
□ Terminal

使用TODO窗口显示插入在代码中的"便签条",以提醒自己完成特定的工作。

程序调试方法

在左部空白的地方点击,可以设置断点

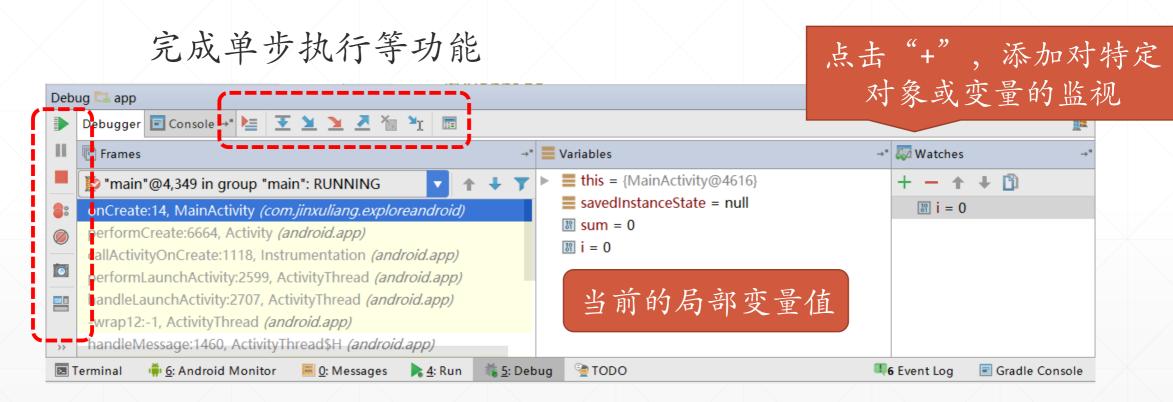
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    int sum = 0;
    for (int i = 0; i < 10; i++) {
        sum += i;
    }
}</pre>
```



暂停的程序

程序运行到断点处暂停, 鼠标移到变量上, 可以显示出变量的值

使用Debug窗口



继续与停止调试,查看断点等功能