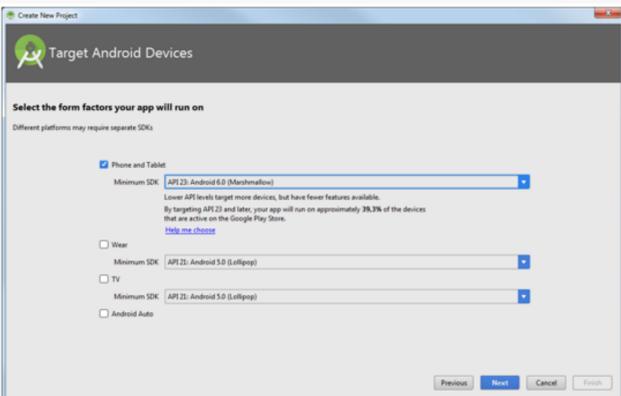


创建机器人应用程序

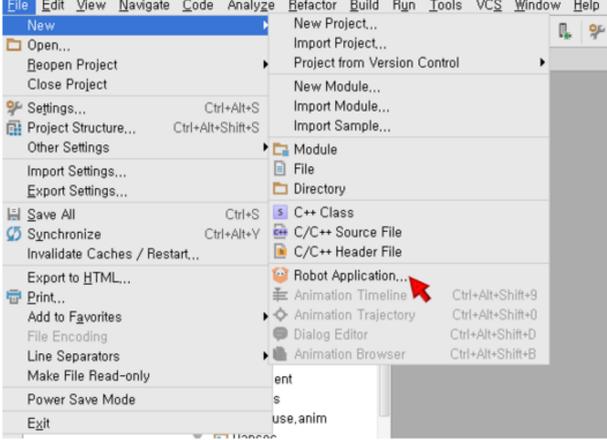
1. 创建项目

要创建机器人应用程序，我们首先创建一个标准的Android项目，然后我们将其转换为机器人应用程序。

Step	Action
	在 Android Studio 中选择: File > New > New Project 为您的应用程序命名
	勾选 Phone and Tablet 选项, 并选择 Android 6.0 (Marshmallow) 版本 
	按默认设置完成余下初始化选项

2. 创建机器人应用程序

现在您已经创建了Android项目，让我们将其转换为机器人应用程序：

Step	Action
	创建项目后，选择 File > New > Robot Application . 
	选择最低 Robot SDK 版本和Module进行robotify. 选择 OK .
	将项目与Gradle文件同步

结果

您的项目现在是机器人应用程序！

因此，您应该看到以下更改：

- `robotsdk.xml` 文件加入到了 `assets/robot` 目录下
- 工具栏出现了一些新的可用工具
- **QiSDK** 依赖已被添加到您的模块下的 `build.gradle` 文件
- 您的 `AndroidManifest.xml` 中添加了一个 `uses-feature` 标记

`robotsdk.xml` 文件包含项目（之前选择一个）的最低机器人SDK版本

新的可用工具允许您启动新的机器人虚拟机



和连接到真实的机器人



QiSDK 依赖为您提供与Pepper交互的所有功能

`uses-feature` 标签指出您的应用程序使用控制Pepper，并将只可用于机器人平板电脑上

3. 实现QiSDK和机器人生命周期

我们已经配置了一个机器人应用程序，现在是时候让我们的代码控制Pepper的身体了

Step	Action
	在 <code>onCreate</code> 方法中注册 <code>MainActivity</code> 到 QiSDK ： <pre>@Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); // Register the RobotLifecycleCallbacks to this Activity. QiSDK.register(this, this); }</pre>
	在 <code>onDestroy</code> 方法中取消注册: <pre>@Override protected void onDestroy() { // Unregister the RobotLifecycleCallbacks for this Activity. QiSDK.unregister(this, this); super.onDestroy(); }</pre> <p>提示：您还可以使用 <code>QiSDK.unregister(this)</code> 注销此活动的所有 <code>RobotLifecycleCallbacks</code>，注意不要无意中删除应该保持注册的回调</p>
	使 <code>MainActivity</code> class 继承 <code>RobotActivity</code> ： <pre>public class MainActivity extends RobotActivity</pre>
	使 <code>MainActivity</code> class 实现 <code>RobotLifecycleCallbacks</code> 接口: <pre>public class MainActivity extends RobotActivity implements RobotLifecycleCallbacks</pre>
	重写 <code>onRobotFocusGained</code> ， <code>onRobotFocusLost</code> 和 <code>onRobotFocusRefused</code> 方法: <pre>@Override public void onRobotFocusGained(QiContext qiContext) { // The robot focus is gained. } @Override public void onRobotFocusLost() { // The robot focus is lost. } @Override public void onRobotFocusRefused(String reason) { // The robot focus is refused. }</pre> <p>为什么？当相关Activity获得机器人Focus时，<code>onRobotFocusGained</code>方法被调用。获Focus后，我们可以在机器人上执行各种操作。当<code>onRobotFocusLost</code>方法被调用时，相关Activity失去了Focus，操作将无法在Pepper上运行。</p> <p>重要</p> <pre>onRobotFocusGained</pre> 和 <pre>onRobotFocusLost</pre> 方法在后台中运行, 当同步使用QiSDK时，UI thread 不会被阻塞