**Android Application using Python & Kivy**

**Aditya Singh(aditya107)**

**Prerequisites**:

**Pycharm IDE**

(Best IDE for Python) for better programing experience.

Installation of Pycharm

sudo apt install snapd snapd-xdg-open

sudo snap install pycharm-community –classic

**Kivy**

Kivy is an open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps. To run a simple kivy Program and packaging this to apk(Ubuntu 18.04). Just follow the following steps.

**Step -1 Installation of Kivy:**

sudo add-apt-repository ppa:kivy-team/kivy

sudo apt-get update

sudo apt-get install python-kivy (For Python 2)

sudo apt-get install python3-kivy (For Python 3)

**Dependencies for Kivy**

Commands

sudo apt-get install -y python-pip build-essential git python python-dev ffmpeg libsdl2-dev libsdl2-image-dev libsdl2-mixer-dev libsdl2-ttf-dev libportmidi-dev libswscale-dev libavformat-dev libavcodec-dev zlib1g-dev

\*In the following command use “python” and “python-dev” for Python 2, or “python3” and “python3-dev” for Python 3

Some other important packages related with kivy

**Kivy-Garden**

sudo pip install kivy-garden

**Kivy-examples**

sudo pip install kivy-examples

Some extra packages related with Kivy. One can install these packages directly from Pycharm.

**KivyCalendar**

**KivyOnTop**

**easykivy**

**kivy-automate**

**kivy-django**

**kivy-okapi**

Some important dependencies

**Cython**- You must have a cython version of 0.25 otherwise you will get a compilation error. To install cython 0.25 just use this command.

sudo pip install 'cython == 0.25'

**Installation of Pygame**

sudo apt-get install python-pygame

**Simple Kivy Program**

from kivy.app import App

from kivy.uix.label import Label

class SimpleKivy(App):

def build(self):

return Label(text ="Hello World!")

if \_\_name\_\_ == "\_\_main\_\_":

SimpleKivy().run()

To package this Kivy program there are different method but the easiest one is using Buildozer.

**Buildozer** is a tool that aim to package mobiles application easily. It automates the entire build process and download the prerequisites like python-for-android, Android SDK, NDK, etc.

Buildozer manage a file named **buildozer.spec** in your application directory, describing your application requirements and settings such as title, icon, included modules etc. It will use the specification file to create a package for Android, iOS, and more.

**Installation of buildozer (Python3) ( Not recommended )**

git clone<https://github.com/kivy/buildozer.git>

cd buildozer

sudo python3 setup.py install

**Installation of buildozer (Python2) ( Recommended )**

sudo pip2 install buildozer

**Some important dependencies**

sudo pip2 install -U colorama

sudo pip2 install -U appdirs

sudo pip2 install -U appdirs

sudo pip2 install -U sh

**Installation of ADB (Android Debug Bridge)**

**To compile the kivy program** first go to your project directory and run the following commands.

buildozer init

This will create a buildozer.spec file. Just go to your project directory and check whether this file is there or not.

buildozer android debug deploy run

This will download Apache ANT, Android SDK, Android NDK and install in your system if you are running first time.

**Possible errors during compilation**

javac compilation error

This is due to the version of jdk. Buildozer use jdk version 7 and 8. It's better to use jdk 8. If you have many jdk version installed in your system you can select jdk7 or jdk8 by using this command.

sudo update-alternatives --config java

Apk file copy error after successful build

This is due to a bug in the latest version of buildozer. To remove this follow the following instructions.

Open android.py file in /usr/local/lib/python2.7/dist-packages/buildozer/targets and do the following changes:

1. at file top insert : from distutils.version import LooseVersioin
2. From line 790 replace with below code:

# XXX found how the apk name is really built from the title  
 \_\_sdk\_dir = self.android\_sdk\_dir  
 build\_tools\_versions = os.listdir(join(\_\_sdk\_dir, 'build-tools'))  
 build\_tools\_versions = sorted(build\_tools\_versions, key=LooseVersion)  
 build\_tools\_version = build\_tools\_versions[-1]

gradle\_files = ["build.gradle", "gradle", "gradlew"]  
 is\_gradle\_build = any((exists(join(dist\_dir, x)) for x in gradle\_files)) and build\_tools\_version >= '25.0'

To check the debug output you need to change the log level = 2 in **buildozer.spec** file which was created in project directory.

**After successful build one apk file will be created in bin directory of your project.**