Porting of Android on Raspberry Pi:

Much thanks goes out to Google, ProTekk, Canonical, and everyone else that I read a random paragraph here and snippet there.

Ready to begin?

1) Unless it's a completely fresh Ubuntu installation, many people might have the wrong version of Java installed. Let's fix that first.

The command below makes sure you're starting with a clean slate. Copy and paste it into a terminal (command prompt) window:

Code:

sudo apt-get remove openjdk-* icedtea-* icedtea6-*

If necessary, follow the on-screen instructions to remove any stray Java versions. Otherwise, move on to the next step.

2) Install the main build tools with this command:

Code:

sudo apt-get install openjdk-8-jdk git ccache automake lzop bison gperf build-essential zip curl zlib1g-dev zlib1g-dev:i386 g++-multilib python-networkx libxml2-utils bzip2 libbz2-dev libbz2-1.0 libghc-bzlib-dev squashfs-tools pngcrush schedtool dpkg-dev liblz4-tool make optipng maven

That's it on the package side of things.

3) "repo" communicates with git servers for all that precious source code. The next command will grab it:

Code:

mkdir ~/bin && curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo && chmod a+x ~/bin/repo

4) Use your favorite text editor to open ~/.bashrc - I like vim:

Code:

vim ~/.bashrc

5) At the very bottom (use the Page Down key) paste this code to a new line:

Code:

export PATH=~/bin:$PATH
export USE_CCACHE=1

6) Save it. In vim that would be Shift-zz and then Enter.

7) In the terminal, navigate to where you would like to download the Android source code. The commands below will make it in your home folder, but if you have limited space you may want to create it somewhere else. Faster is better, i.e. SSD would be best, USB external (even 3.0) will be comparatively slow. Here we go:

Code:

mkdir ~/android

cd ~/android

Sync Android Source (AOSP)
 * $ repo init -u https://github.com/tab-pi/platform_manifest -b nougat
 * $ repo sync

if any error related fetching use command
 * $ repo sync --force-sync

Build Kernel
 * Install gcc-arm-linux-gnueabihf (For Ubuntu: $ sudo apt install gcc-arm-linux-gnueabihf)
 * $ cd kernel/rpi
 * $ ARCH=arm scripts/kconfig/merge_config.sh arch/arm/configs/bcm2709_defconfig android/configs/android-base.cfg android/configs/android-recommended.cfg
 * $ ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- make zImage
 * $ ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- make dtbs

Install python mako module
  sudo apt-get install python-mako

Build Android source
 * $ source build/envsetup.sh
 * $ lunch rpi3-eng
 * $ make ramdisk systemimage

Help for build failure :
   https://github.com/tab-pi/device_brcm_rpi3/wiki/Build-Errors

Prepare SD card
 ### Partitions of the card should be set-up like following:
  1. p1 512MB for BOOT : Do fdisk : W95 FAT32(LBA) & Bootable, mkfs.vfat
  2. p2 1024MB for /system : Do fdisk, new primary partition, mkfs.ext4
  3. p3 512MB for /cache : Do fdisk, mkfs.ext4
  4. p4 remainings for /data : Do fdisk, mkfs.ex4
  5. Set volume label for each partition - system, cache, userdata
  : use -L option of mkfs.ext4, e2label command, or -n option of mkfs.vfat

Write system partition
  * $ cd out/target/product/rpi3
  * $ sudo dd if=system.img of=/dev/<p2> bs=1M

Copy kernel & ramdisk to BOOT partition
  * device/brcm/rpi3/boot/* to p1:/
  * kernel/rpi/arch/arm/boot/zImage to p1:/
  * kernel/rpi/arch/arm/boot/dts/bcm2710-rpi-3-b.dtb to p1:/
  * kernel/rpi/arch/arm/boot/dts/overlays/vc4-kms-v3d.dtbo to p1:/overlays/vc4-kms-v3d.dtbo
  * out/target/product/rpi3/ramdisk.img to p1:/

HDMI_MODE : If DVI monitor does not work, try followings for p1:/config.txt
hdmi_group=2

hdmi_mode=85

How to set up Android-TV launcher :
https://github.com/tab-pi/device_brcm_rpi3/wiki#how-to-apply-android-tv-leanback-launcher

Graphics HAL of this build : https://github.com/anholt/mesa/wiki/VC4

To Debug the android using serial port:

https://developer.android.com/things/hardware/raspberrypi.html