

Porting στο Android

• Buildozer

(<https://github.com/kivy/buildozer>)

Κατά τη διάρκεια του project ήταν το κύριο εργαλείο το οποίο χρησιμοποιήσαμε για να εξάγουμε το αρχείο .apk του διαχειριστή επαφών. Ωστόσο, συναντήσαμε αρκετά προβλήματα - τα οποία αναλύονται παρακάτω - λόγω του είδους της πλατφόρμας στην οποία προοριζόταν η εφαρμογή. Πρώτα, όμως, θα περιγραφούν συνοπτικά τα βήματα που ακολουθήσαμε για να εξάγουμε ένα αρχείο .apk.

Αρχικά, τοποθετήσαμε τα αρχεία main.py και GUI.kv σε ένα κοινό φάκελο και δώσαμε την εντολή *\$ buildozer init*

η οποία δημιουργεί ένα αρχείο buildozer.spec το οποίο περιέχει τις απαραίτητες ρυθμίσεις για να γίνει η εξαγωγή των αρχείων σε .apk.

Ένα δείγμα του αρχείου φαίνεται παρακάτω (στην προεπιλεγμένη μορφή).

```
[app]

# (str) Title of your application
title = My Application

# (str) Package name
package.name = myapp

# (str) Package domain (needed for android/ios packaging)
package.domain = org.test

# (str) Source code where the main.py live
source.dir = .

# (list) Source files to include (let empty to include all the files)
source.include_exts = py,png,jpg,kv,atlas
```

```

# (list) List of inclusions using pattern matching
#source.include_patterns = assets/*,images/*.png

# (list) Source files to exclude (let empty to not exclude anything)
#source.exclude_exts = spec

# (list) List of directory to exclude (let empty to not exclude anything)
#source.exclude_dirs = tests, bin

# (list) List of exclusions using pattern matching
#source.exclude_patterns = license,images/*/*.jpg

# (str) Application versioning (method 1)
version = 0.1

# (str) Application versioning (method 2)
# version.regex = __version__ = ['"](.*)['"]
# version.filename = %(source.dir)s/main.py

# (list) Application requirements
# comma separated e.g. requirements = sqlite3,kivy
requirements = python3,kivy

# (str) Custom source folders for requirements
# Sets custom source for any requirements with recipes
# requirements.source.kivy = ../../kivy

# (list) Garden requirements
#garden_requirements =

# (str) Presplash of the application
#presplash.filename = %(source.dir)s/data/presplash.png

# (str) Icon of the application
#icon.filename = %(source.dir)s/data/icon.png

# (str) Supported orientation (one of landscape, portrait or all)
orientation = portrait

# (list) List of service to declare
#services = NAME:ENTRYPOINT_TO_PY,NAME2:ENTRYPOINT2_TO_PY

#
# OSX Specific
#

#
# author = © Copyright Info

# change the major version of python used by the app
osx.python_version = 3

# Kivy version to use
osx.kivy_version = 1.9.1

#
# Android specific
#

# (bool) Indicate if the application should be fullscreen or not
fullscreen = 0

```

```
# (string) Presplash background color (for new android toolchain)
# Supported formats are: #RRGGBB #AARRGGBB or one of the following names:
# red, blue, green, black, white, gray, cyan, magenta, yellow, lightgray,
# darkgray, grey, lightgrey, darkgrey, aqua, fuchsia, lime, maroon, navy,
# olive, purple, silver, teal.
#android.presplash_color = #FFFFFF

# (list) Permissions
#android.permissions = INTERNET

# (int) Android API to use
#android.api = 19

# (int) Minimum API required
#android.minapi = 9

# (int) Android SDK version to use
#android.sdk = 20

# (str) Android NDK version to use
#android.ndk = 9c

# (bool) Use --private data storage (True) or --dir public storage (False)
#android.private_storage = True

# (str) Android NDK directory (if empty, it will be automatically downloaded.)
#android.ndk_path =

# (str) Android SDK directory (if empty, it will be automatically downloaded.)
#android.sdk_path =

# (str) ANT directory (if empty, it will be automatically downloaded.)
#android.ant_path =

# (bool) If True, then skip trying to update the Android sdk
# This can be useful to avoid excess Internet downloads or save time
# when an update is due and you just want to test/build your package
# android.skip_update = False

# (str) Android entry point, default is ok for Kivy-based app
#android.entrypoint = org.renpy.android.PythonActivity

# (list) Pattern to whitelist for the whole project
#android.whitelist =

# (str) Path to a custom whitelist file
#android.whitelist_src =

# (str) Path to a custom blacklist file
#android.blacklist_src =

# (list) List of Java .jar files to add to the libs so that pyjnius can access
# their classes. Don't add jars that you do not need, since extra jars can slow
# down the build process. Allows wildcards matching, for example:
# OUYA-ODK/libs/*.jar
#android.add_jars = foo.jar,bar.jar,path/to/more/*.jar

# (list) List of Java files to add to the android project (can be java or a
# directory containing the files)
#android.add_src =

# (list) Android AAR archives to add (currently works only with sdl2_gradle
# bootstrap)
```

```

#android.add_aars =

# (list) Gradle dependencies to add (currently works only with sdl2_gradle
# bootstrap)
#android.gradle_dependencies =

# (list) Java classes to add as activities to the manifest.
#android.add_activites = com.example.ExampleActivity

# (str) python-for-android branch to use, defaults to stable
#p4a.branch = stable

# (str) OUYA Console category. Should be one of GAME or APP
# If you leave this blank, OUYA support will not be enabled
#android.ouya.category = GAME

# (str) Filename of OUYA Console icon. It must be a 732x412 png image.
#android.ouya.icon.filename = %(source.dir)s/data/ouya_icon.png

# (str) XML file to include as an intent filters in <activity> tag
#android.manifest.intent_filters =

# (str) launchMode to set for the main activity
#android.manifest.launch_mode = standard

# (list) Android additional libraries to copy into libs/armeabi
#android.add_libs_armeabi = libs/android/*.so
#android.add_libs_armeabi_v7a = libs/android-v7/*.so
#android.add_libs_x86 = libs/android-x86/*.so
#android.add_libs_mips = libs/android-mips/*.so

# (bool) Indicate whether the screen should stay on
# Don't forget to add the WAKE_LOCK permission if you set this to True
#android.wakelock = False

# (list) Android application meta-data to set (key=value format)
#android.meta_data =

# (list) Android library project to add (will be added in the
# project.properties automatically.)
#android.library_references =

# (str) Android logcat filters to use
#android.logcat_filters = *:S python:D

# (bool) Copy library instead of making a libpymodules.so
#android.copy_libs = 1

# (str) The Android arch to build for, choices: armeabi-v7a, arm64-v8a, x86
android.arch = armeabi-v7a

#
# Python for android (p4a) specific
#

# (str) python-for-android git clone directory (if empty, it will be
# automatically cloned from github)
#p4a.source_dir =

# (str) The directory in which python-for-android should look for your own
# build recipes (if any)
#p4a.local_recipes =

```

```

# (str) Filename to the hook for p4a
#p4a.hook =

# (str) Bootstrap to use for android builds
# p4a.bootstrap = sdl2

# (int) port number to specify an explicit --port= p4a argument (eg for
bootstrap flask)
#p4a.port =

#
# iOS specific
#

# (str) Path to a custom kivy-ios folder
#ios.kivy_ios_dir = ../kivy-ios

# (str) Name of the certificate to use for signing the debug version
# Get a list of available identities: buildozer ios list_identities
#ios.codesign.debug = "iPhone Developer: <lastname> <firstname> (<hexstring>)"

# (str) Name of the certificate to use for signing the release version
#ios.codesign.release = %(ios.codesign.debug)s

[buildozer]

# (int) Log level (0 = error only, 1 = info, 2 = debug (with command output))
log_level = 1

# (int) Display warning if buildozer is run as root (0 = False, 1 = True)
warn_on_root = 1

# (str) Path to build artifact storage, absolute or relative to spec file
# build_dir = ../.buildozer

# (str) Path to build output (i.e. .apk, .ipa) storage
# bin_dir = ../bin

#
-----
#   List as sections
#
#   You can define all the "list" as [section:key].
#   Each line will be considered as a option to the list.
#   Let's take [app] / source.exclude_patterns.
#   Instead of doing:
#
#[app]
#source.exclude_patterns = license,data/audio/*.wav,data/images/original/*
#
#   This can be translated into:
#
#[app:source.exclude_patterns]
#license
#data/audio/*.wav
#data/images/original/*
#

#
-----

```

```
# Profiles
#
# You can extend section / key with a profile
# For example, you want to deploy a demo version of your application without
# HD content. You could first change the title to add "(demo)" in the name
# and extend the excluded directories to remove the HD content.
#
#[app@demo]
#title = My Application (demo)
#
#[app:source.exclude_patterns@demo]
#images/hd/*
#
# Then, invoke the command line with the "demo" profile:
#
#buildozer --profile demo android debug
```

Κάποιες αξιοσημείωτες ρυθμίσεις είναι:

- title: Το όνομα της εφαρμογής
- package name: Το όνομα του πακέτου που εγκαθίσταται
- package domain: Ο φορέας/προγραμματιστής που αναπτύσσει την εφαρμογή
- source dir: Ο φάκελος στον οποίο βρίσκεται ο πηγαίος κώδικας (το αρχείο main.py και το αρχείο .kv), ως προεπιλογή ορίζεται ο φάκελος που δημιουργήθηκε το buildozer.spec
- source.include_exts: επεκτάσεις επιπρόσθετων αρχείων που θα συμπεριληφθούν στο .apk
- version: η έκδοση της εφαρμογής
- requirements: απαραίτητα προγράμματα που απαιτούνται να συμπεριληφθούν στο .apk έτσι ώστε η εφαρμογή να είναι εκτελέσιμη π.χ. python3,kivy
- orientation: ο προσανατολισμός της οθόνης
- fullscreen: αν η εφαρμογή θα καταλαμβάνει όλη την οθόνη ή όχι
- android.permissions: δικαιώματα που απαιτούνται να χορηγηθούν από το χρήστη κατά την εγκατάσταση
- android.api: για ποια έκδοση android προορίζεται η εφαρμογή

- android.sdk: έκδοση του android sdk που θα χρησιμοποιηθεί
- android.ndk: έκδοση του android ndk που θα χρησιμοποιηθεί
- r4a.brach: ποιο branch θα χρησιμοποιηθεί από το github για να γίνει clone η python-for-android (προεπιλογή: stable)
- android.arch: η αρχιτεκτονική του επεξεργαστή για την οποία προορίζεται η εφαρμογή
- log.level: τι θα εμφανίζεται στο terminal κατά τη δημιουργία του .apk (π.χ. log.level = 0 θα εμφανίζονται μόνο τα errors)

Στη συνέχεια η εντολή

```
$ buildozer android release
```

δηλώνει ότι η εφαρμογή προορίζεται για android (δηλαδή θα δημιουργηθεί .apk) και είναι έτοιμη για κυκλοφορία.

Ύστερα, το buildozer προετοιμάζει την εξαγωγή του .apk ελέγχοντας όλα τα dependencies, εγκαθιστά από το github την python-for-android η οποία θα παίξει βασικό ρόλο στη μεταφορά των αρχείων από python/kivy σε android. Στη συνέχεια ελέγχει αν είναι εγκατεστημένα τα ακόλουθα android packages:

- android sdk
- android ndk
- ant

Μετά, αναλόγως το android api που έχουμε ορίσει στο buildozer.spec κατεβαίνουν τα απαραίτητα build-tools αφού γίνει ανανέωση των repositories του εκάστοτε πακέτου/πακέτων. Ύστερα, αναλαμβάνει η python-for-android. Πρώτα, ελέγχει για τα dependencies της (τα οποία έχουν προετοιμαστεί από το buildozer), μετά κατεβάζει το bootstrap (sdl2 στο συγκεκριμένο project) και τα απαραίτητα recipes (ειδικά scripts που θα εγκαταστήσουν τα απαραίτητα προγράμματα για να γίνει compile σε android ο πηγαίος κώδικας στη σωστή αρχιτεκτονική). Ποια recipes θα κατέβουν ορίζονται κυρίως από το πεδίο requirements (στο buildozer.spec) και τα υπόλοιπα είναι άλλα recipes που εξαρτώνται από τα requirements. Αφού γίνει η εγκατάσταση αυτών (εδώ χρησιμοποιείται κυρίως ο gcc και η cython) τότε ελέγχεται αν

υπάρχουν κάποια επιπρόσθετα python modules που πρέπει να εγκατασταθούν. Στη συνέχεια, εισάγεται ο πηγαίος κώδικας του project, συνδέεται με τις απαραίτητες libraries (που έχουν εγκατασταθεί προηγουμένως με τα recipes της python-for-android) και ξεκινάει το packaging για android. Τέλος, χρησιμοποιείται ένα java build tool (στο συγκεκριμένο project χρησιμοποιείται το gradle) το οποίο ολοκληρώνει το build και η python-for-android δημιουργεί το .apk και προσθέτει τη version της εφαρμογής.

Κύρια προβλήματα που αντιμετωπίσαμε

- Η έκδοση του android-ndk (που κατέβαζε το buildozer): το buildozer ως προεπιλογή κατεβάζει την έκδοση 9c ωστόσο η Google δεν διατηρεί πλέον αυτή την έκδοση στα ndk-archives με αποτέλεσμα το link που έχει το buildozer να είναι απαρχαιωμένο. Λύση του προβλήματος είναι η αλλαγή της έκδοσης του πεδίου android.ndk (στο buildozer.spec) σε έκδοση 10 και πάνω.
- Ασυμβατότητα android sdk και android ndk: το buildozer (master branch) υποστηρίζει συγκεκριμένο συνδυασμό εκδόσεων των 2 συγκεκριμένων προγραμμάτων (<https://github.com/kivy/kivy/wiki/Android-SDK-NDK-Information>)
- Έκδοση του openjdk: αν χρησιμοποιηθεί η έκδοση openjdk-9 (στο project χρησιμοποιήθηκε το openjdk-8) τότε προκύπτει ένα error της μορφής (<https://stackoverflow.com/questions/47150410/failed-to-run-sdkmanager-list-android-sdk-with-java-9>). Επίσης, άλλη λύση του προβλήματος αυτού είναι η εγκατάσταση του gradler.
- Έκδοση της Cython: η kivy υποστηρίζει συγκεκριμένες εκδόσεις της cython και δε συνίσταται η εγκατάσταση της πιο πρόσφατης έκδοσης της τελευταίας (<https://kivy.org/doc/stable/installation/deps-cython.html>)
- Branch της python-for-android: αν χρησιμοποιηθεί το stable branch από το github (<https://github.com/kivy/python-for-android/tree/stable>)

τότε το buildozer αναφέρει ότι υπάρχουν conflicting dependencies όταν

requirements = python3,kivy

(<https://github.com/kivy/python-for-android/issues/1557>)

Αυτό συμβαίνει διότι στο stable branch δεν έχει προστεθεί ακόμα recipe για python3. Λύση του προβλήματος είναι να χρησιμοποιηθεί το master branch.

(<https://github.com/kivy/python-for-android/tree/master>).

Ωστόσο, με τη χρήση του συγκεκριμένου branch καταλήγουμε στο επόμενο error.

- recipe for target 'Programs/python.o' failed:

[INFO]: Building sdl2_image for arm64-v8a[INFO]: Building sdl2_mixer for arm64-v8a

[INFO]: Building sdl2_ttf for arm64-v8a

[INFO]: Building python3 for arm64-v8a

[INFO]: -> directory context /home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/python3/arm64-v8a__ndk_target_21/python3/android-build

[WARNING]: Doing some hacky stuff to link properly

[INFO]: -> running cp /home/giorgos/.buildozer/android/platform/android-ndk-r13b/platforms/android-21/arch-arm64/usr/lib/crtbegin_so.o ./

[INFO]: -> running cp /home/giorgos/.buildozer/android/platform/android-ndk-r13b/platforms/android-21/arch-arm64/usr/lib/crtend_so.o ./

[INFO]: -> running configure --host=aarch64-linux-android --build=x86_64-pc-linux-gnu --enable-shared --disable-ipv6 ac_cv_file__dev_ptmx=yes ac_cv_file__dev_ptc=no --withou...(and 87 more)

working: please run ./configure [INFO]: -> running make all INSTSONAME=libpython3.7m.so

Exception in thread background thread for pid 23571:python.o' failed

Traceback (most recent call last):

File "/usr/lib/python2.7/threading.py", line 801, in __bootstrap_inner
self.run()

File "/usr/lib/python2.7/threading.py", line 754, in run
self.__target(*self.__args, **self.__kwargs)

File "/home/giorgos/.local/lib/python2.7/site-packages/sh.py", line 2170, in background_thread
handle_exit_code(exit_code)

File "/home/giorgos/.local/lib/python2.7/site-packages/sh.py", line 1929, in fn
return self.command.handle_command_exit_code(exit_code)

File "/home/giorgos/.local/lib/python2.7/site-packages/sh.py", line 672, in handle_command_exit_code
raise exc

ErrorReturnCode_2:

RAN: /usr/bin/make all INSTSONAME=libpython3.7m.so

STDOUT:

/home/giorgos/.buildozer/android/platform/android-ndk-r13b/toolchains/llvm/prebuilt/linux-x86_64/bin/clang -target aarch64-none-linux-android -gcc-toolchain /home/giorgos/.buildozer/android/platform/android-ndk-r13b/toolchains/aarch64-linux-android-4.9/prebuilt/linux-x86_64 -c -Wno-unused-result -Wsign-compare -Wunreachable-code -DNDEBUG -g -fwrapv -O3 -Wall -fPIC --sysroot=/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot -D__ANDROID_API__=21 -isystem /home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include/aarch64-linux-android -I/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include -fPIC --sysroot=/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot -D__ANDROID_API__=21 -isystem /home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include/aarch64-linux-android -I/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Wstrict-prototypes -Werror=implicit-function-declaration -IObjects -IInclude -IPython -I. -I/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/python3/arm64-v8a__ndk_target_21/python3/Include -fPIC --sysroot=/home/giorgos/.buildozer/android/platform/android-

```
ndk-r13b/sysroot -D__ANDROID_API__=21 -isystem /home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include/aarch64-linux-android -I/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include -fPIC --sysroot=/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot -D__ANDROID_API__=21 -isystem /home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include/aarch64-linux-android -I/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include -DPy_BUILD_CORE -o Programs/python.o /home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/python3/arm64-v8a__ndk_target_21/python3/Programs/python.c
In file included from /home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/python3/arm64-v8a__ndk_target_21/python3/Programs/python.c:3:
/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/python3/arm64-v8a__ndk_target_21/python3/Include/Python.h:25:10: fatal error:
'stdio.h' file not found
```

```
#include <stdio.h>
^
```

1 error generated.

Makefile:779: recipe for target 'Programs/python.o' failed

make: *** [Programs/python.o] Error 1

STDERR:

Traceback (most recent call last):

```
File "/usr/lib/python2.7/runpy.py", line 174, in _run_module_as_main
  "__main__", fname, loader, pkg_name)
File "/usr/lib/python2.7/runpy.py", line 72, in _run_code
  exec code in run_globals
File "/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/python-for-android-new-toolchain/pythonforandroid/toolchain.py", line 1039, in <module>
  main()
File "/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/python-for-android-new-toolchain/pythonforandroid/toolchain.py", line 1033, in main
  ToolchainCL()
File "/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/python-for-android-new-toolchain/pythonforandroid/toolchain.py", line 553, in __init__
  getattr(self, args.subparser_name.replace('-', '_'))(args)
File "/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/python-for-android-new-toolchain/pythonforandroid/toolchain.py", line 150, in wrapper_func
  build_dist_from_args(ctx, dist, args)
File "/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/python-for-android-new-toolchain/pythonforandroid/toolchain.py", line 191, in build_dist_from_args
  build_recipes(build_order, python_modules, ctx)
File "pythonforandroid/build.py", line 633, in build_recipes
File "pythonforandroid/python.py", line 240, in build_arch
File "pythonforandroid/logger.py", line 178, in shprint
File "/home/giorgos/.local/lib/python2.7/site-packages/sh.py", line 720, in next
  self.wait()
File "/home/giorgos/.local/lib/python2.7/site-packages/sh.py", line 651, in wait
  self.handle_command_exit_code(exit_code)
File "/home/giorgos/.local/lib/python2.7/site-packages/sh.py", line 672, in handle_command_exit_code
  raise exc
sh.ErrorReturnCode_2:
```

RAN: /usr/bin/make all INSTSONAME=libpython3.7m.so

STDOUT:

```
/home/giorgos/.buildozer/android/platform/android-ndk-r13b/toolchains/llvm/prebuilt/linux-x86_64/bin/clang -target aarch64-none-linux-android -gcc-toolchain /home/giorgos/.buildozer/android/platform/android-ndk-r13b/toolchains/aarch64-linux-android-4.9/prebuilt/linux-x86_64 -c -Wno-unused-result -Wsign-compare -Wunreachable-code -DNDEBUG -g -fwrapv -O3 -Wall -fPIC --sysroot=/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot -D__ANDROID_API__=21 -isystem /home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include/aarch64-linux-
```

```

android      -I/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include      -fPIC
--sysroot=/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot -D__ANDROID_API__=21 -
isystem      /home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include/aarch64-linux-
android -I/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include -std=c99 -Wextra
-Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Wstrict-prototypes -
Werror=implicit-function-declaration -lObjects -lInclude -lPython -l.
-I/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/python3/arm64-
v8a__ndk_target_21/python3/Include -fPIC --sysroot=/home/giorgos/.buildozer/android/platform/android-
ndk-r13b/sysroot -D__ANDROID_API__=21 -isystem /home/giorgos/.buildozer/android/platform/android-
ndk-r13b/sysroot/usr/include/aarch64-linux-android -I/home/giorgos/.buildozer/android/platform/android-ndk-
r13b/sysroot/usr/include -fPIC --sysroot=/home/giorgos/.buildozer/android/platform/android-ndk-r13b/
sysroot -D__ANDROID_API__=21 -isystem
/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include/aarch64-linux-android
-I/home/giorgos/.buildozer/android/platform/android-ndk-r13b/sysroot/usr/include -DPy_BUILD_CORE -o
Programs/python.o /home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/
python3/arm64-v8a__ndk_target_21/python3/Programs/python.c
In file included from /home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/
python3/arm64-v8a__ndk_target_21/python3/Programs/python.c:3:
/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build/build/other_builds/python3/arm64-
v8a__ndk_target_21/python3/Include/Python.h:25:10: fatal error:
'stdio.h' file not found
#include <stdio.h>
^
1 error generated.
Makefile:779: recipe for target 'Programs/python.o' failed
make: *** [Programs/python.o] Error 1

```

STDERR:

```

# Command failed: /usr/bin/python -m pythonforandroid.toolchain create --dist_name=Test --bootstrap=sdl2
--requirements=python3,kivy --arch arm64-v8a --copy-libs --color=always
--storage-dir="/home/giorgos/Desktop/Pong_apk/.buildozer/android/platform/build"
#
# Buildozer failed to execute the last command
# The error might be hidden in the log above this error
# Please read the full log, and search for it before
# raising an issue with buildozer itself.
# In case of a bug report, please add a full log with log_level = 2

```

(Δε βρέθηκε λύση του προβλήματος αυτού)

ΣΗΜΕΙΩΣΗ: χρησιμοποιήθηκε και ο κώδικας του pong (<https://kivy.org/doc/stable/tutorials/pong.html>) για να αποκλειστεί η πιθανότητα ο κώδικας της εφαρμογής μας να μην είναι λειτουργικός και επιπρόσθετα να μην μπορεί να μετατραπεί σε μορφή .apk. Επίσης, αν χρησιμοποιηθεί το stable branch της python-for-android και ως requirements ορίσουμε python2,kivy τότε το .apk εξάγεται κανονικά.

• Python-for-android

Ένας εναλλακτικός τρόπος για να γίνει το packaging στο android (χωρίς την αυτοματοποιημένη διαδικασία που παρέχει το buildozer)

είναι να γίνει εγκατάσταση του python-for-android, να εγκατασταθούν τα android sdk και ndk (και να προστεθούν οι εκδόσεις τους και η τοποθεσία των αρχείων στο *PATH*). Με το συγκεκριμένο τρόπο εμφανίστηκαν τα ίδια προβλήματα που συναντήσαμε και στο buildozer. Αναλυτικά η διαδικασία εγκατάστασης της python-for-android μπορεί να βρεθεί στο <https://python-for-android.readthedocs.io/en/latest/quickstart/#usage>

• Virtual Environment

Δεν αποτελεί αμιγώς ξεχωριστό τρόπο αλλά μία καλή προσέγγιση του packaging για το android είναι να χρησιμοποιηθεί ένα virtual environment για να μπορούν να αναιρεθούν γρήγορα οι αλλαγές στα packages και ουσιαστικά να μην επηρεαστεί το υπόλοιπο λειτουργικό σύστημα. Αναλυτικότερα αναφέρονται στο <https://kivy.org/doc/master/installation/installation-linux-venvs.html> Επίσης, πολλοί χρήστες στο github πρότειναν το porting να γίνει μέσω του conda (<https://anaconda.org/conda-forge/kivy>)

• Virtual Machine

Στο επίσημο site της kivy παρέχεται ένα τροποποιημένο virtual machine το οποίο έχει προεγκατεστημένη την kivy, το buildozer και τα λοιπά πακέτα (<https://kivy.org/doc/stable/guide/packaging-android-vm.html>). Ωστόσο, για το τελευταίο error που αναφέρεται παραπάνω (στην ενότητα buildozer) δεν βρέθηκε κάποια λύση.

• Docker

(<https://github.com/kivy/buildozer>)

Λόγω της πολυπλοκότητας της εγκατάστασης του docker ο συγκεκριμένος τρόπος απορρίφθηκε.

• Kivy Launcher

(<https://kivy.org/doc/stable/guide/packaging-android.html#packaging-your-application-for-the-kivy-launcher>)

Λόγω έλλειψης χρόνου και για να γίνει το porting στο android χρησιμοποιήθηκε ο kivy launcher. Πιο συγκεκριμένα, εγκαταστήσαμε στο android μέσω του playstore τη συγκεκριμένη εφαρμογή, δημιουργήσαμε το φάκελο PyContacts στο android (/sdcard/kivy/), τοποθετήσαμε εκεί τα αρχεία main.py, GUI.kv και android.txt μέσα στο οποίο περιέχονται κάποιες βασικές πληροφορίες για την εφαρμογή (απαιτείται από τον kivy launcher).

Android.txt

```
title=PyContacts  
author=Omada45  
orientation=portrait
```

Τέλος, για να εκτελεστεί η εφαρμογή απλώς ανοίγουμε τον kivy launcher και επιλέγουμε το PyContacts.