

Collision Avoidance

ΜΑΘΗΜΑ: Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες

ΠΡΩΤΟ ΠΑΡΑΔΟΤΕΟ

ΟΜΑΔΑ ΕΚΠΟΝΗΣΗΣ ΕΡΓΑΣΙΑΣ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ	A.M.
ΓΙΩΤΗ ΓΕΩΡΓΙΑ	1115201200026
ΓΟΥΛΕΤΑΣ ΜΙΧΑΛΗΣ	1115201200028
ΔΗΛΑΚΗ ΑΓΓΕΛΙΚΗ	1115201200031
ΧΑΤΖΗΕΛΕΥΘΕΡΙΟΥ ΚΩΝ/ΝΑ	1115201200198

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΚΟΝΕΣ	1
ΕΙΣΑΓΩΓΗ	2
Τεχνολογίες συστήματος ανάπτυξης:	4
ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ	4
ΠΗΓΕΣ.....	9

ΕΙΚΟΝΕΣ

Εικόνα 1: Warning Message	2
Εικόνα 2: Λογότυπο εφαρμογής	2
Εικόνα 3: Options Menu	3
Εικόνα 4: Settings	3
Εικόνα 5: Invalid Value	6
Εικόνα 6: Exit	6
Εικόνα 7: Settings	6
Εικόνα 8: Quit	7
Εικόνα 9: Main Activity	7
Εικόνα 10: Settings Loaded	7
Εικόνα 11: Settings Saved.....	8

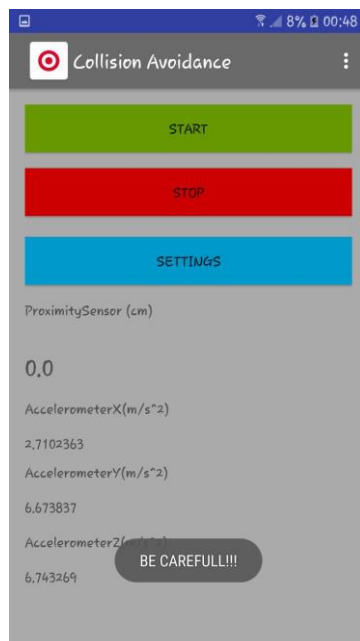
ΕΙΣΑΓΩΓΗ

Στα πλαίσια του μαθήματος, υλοποιήσαμε ένα σύστημα ανίχνευσης συγκρούσεων πεζών μέσω μιας εφαρμογής σε γλώσσα προγραμματισμού Java και της πλατφόρμας AndroidStudio. Η εφαρμογή λαμβάνει τιμές από τους αισθητήρες του κινητού τερματικού. Στην υλοποίησή μας χρησιμοποιήσαμε τους αισθητήρες **Proximity Sensor** και **Accelerometer Sensor**. Η εφαρμογή, με βάση κάποιες τροποποιήσιμες παραμέτρους ειδοποιεί τον χρήστη σε περίπτωση επικείμενης σύγκρουσης. Οι παράμετροι αυτοί αντιστοιχούν στα κατώφλια που θέτει ο χρήστης στα **Settings** της εφαρμογής. Συγκεκριμένα, θέσαμε ως κριτήριο ενεργοποίησης του οπτικού και ηχητικού σήματος τα εξής γεγονότα:

- η τιμή του Proximity αισθητήρα να είναι μικρότερη ή ίση από το κατώφλι που θέτει ο χρήστης και
- η τιμή του Z άξονα του Accelerometer αισθητήρα να είναι μεγαλύτερη από το κατώφλι που θέτει ο χρήστης.

Επιλέξαμε αυτή τη συνθήκη θεωρώντας ότι η συσκευή ανιχνεύει συγκρούσεις με την πρόσοψη.

Η ειδοποίηση γίνεται με τη μορφή ηχητικού και οπτικού μηνύματος. Το οπτικό μήνυμα έχει τη μορφή **Android Toast** και το ηχητικό μήνυμα γίνεται με την αναπαραγωγή ενός ήχου ειδοποίησης (warning.mp3). Τα μηνύματα ειδοποίησης (ηχητικό και οπτικό) παραμένουν ενεργά για διάρκεια ίση με την διάρκεια εμφάνισης της πιθανότητας σύγκρουσης. Η εφαρμογή παραμένει ανοικτή και λειτουργική και μετά από χρήση του **android home button**.



Εικόνα 1: Warning Message

Η πληροφορία των αισθητήρων παρουσιάζεται στην κεντρική οθόνη της εφαρμογής με την εξής δομή:

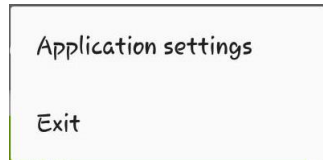
- τύπος αισθητήρα,
- μονάδα μέτρησης και
- τιμή

Η εφαρμογή διαθέτει **menu bar** που περιλαμβάνει το λογότυπο (ic_launcher) και το όνομα της εφαρμογής (**Collision Avoidance**) καθώς και **options menu**.



Εικόνα 2: Λογότυπο εφαρμογής

Το **options menu** περιλαμβάνει την επιλογή των ρυθμίσεων (**Application settings**) και την επιλογή εξόδου από την εφαρμογή (**Exit**).



Εικόνα 3: Options Menu

Οι ρυθμίσεις της εφαρμογής επιτρέπουν την παραμετροποίηση των αισθητήρων που χρησιμοποιήθηκαν και πιο συγκεκριμένα:

- i. τη συχνότητα ελέγχου των τιμών (**Sampling Rate σε Hz**). Ο χρήστης στην ουσία επιλέγει κάποια περίοδο δειγματοληψίας την οποία στη συνέχεια μετατρέπουμε σε συχνότητα. Η μετατροπή αυτή γίνεται μέσω της κλάσης **OnClickListenerStartMonitoring**.
- ii. το κατώφλι στο οποίο θα ενεργοποιείται η ειδοποίηση για τον Proximity Sensor (**Proximity Sensor Alarm Value σε cm**) και

ΠΑΡΑΤΗΡΗΣΗ: Και στις δύο συσκευές που δοκιμάσαμε την εφαρμογή παρατηρήσαμε πως ο αισθητήρας λαμβάνει δύο τιμές, 0.0 ή 1.0 (API 4.2.2)/8.0 (API 6.0.1). Παρ' όλα αυτά, αποφασίσαμε να αφήσουμε το seek bar για πιθανή μελλοντική χρήση με περισσότερες διακριτές τιμές.

- iii. τα κατώφλια για τον Accelerometer Sensor καθώς για αυτόν επιστρέφονται 3 τιμές, μία για κάθε άξονα (**Accelerometer Sensor Alarm Value X,Y,Z σε m/s²**), εκ των οποίων λαμβάνουμε υπόψιν τον τελευταίο. Ο τρόπος ρύθμισης των παραμέτρων γίνεται μέσω **sliders**.

ΠΑΡΑΤΗΡΗΣΗ: Παρ' όλα αυτά, αποφασίσαμε να αφήσουμε τα δύο επιπλέον seek bar για πιθανή μελλοντική χρήση.



Εικόνα 4: Settings




Collision Avoidance

Τέλος, ο χρήστης έχει τη δυνατότητα να κλείσει την εφαρμογή(**exit**) με 2 τρόπους:

- i. είτε μέσα από το **options menu** με τη χρήση αντίστοιχης επιλογής
- ii. είτε με τη χρήση του **android back button**

Και στις δύο περιπτώσεις εμφανίζεται **Dialog box** για την επιβεβαίωση της εξόδου.

Τεχνολογίες συστήματος ανάπτυξης:

-  Java Oracle SE 8
-  Android API 4.2.2 και 6.0.1
-  Android Studio (Εργαλείο ανάπτυξης της εφαρμογής Android)

ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ

- **data (package):** Στο συγκεκριμένο πακέτο διαχειριζόμαστε:
 - a. **DataSettings (class)** : την κλάση στην οποία αρχικοποιούνται οι αρχικές τιμές της συχνότητας (**frequency**) του κατωφλιού του πρώτου αισθητήρα (**firstsensorvalue**) και των κατωφλιών για τον κάθε άξονα του δεύτερου αισθητήρα. Επίσης, κάθε φορά που ο χρήστης πατάει το κουμπί **SAVE** οι τιμές τροποποιούνται και αποθηκεύονται στις αντίστοιχες μεταβλητές.
 - b. **SensorsData (class)**: την κλάση στην οποία αρχικοποιούνται οι αρχικές τιμές των αισθητήρων για την πρώτη φορά που θα ανοίξει ο χρήστης την εφαρμογή, δηλαδή, πριν ξεκινήσει η παρακολούθηση τους χωρίς να είναι ορατές στην αρχική οθόνη. Επίσης, κατά τη διάρκεια της παρακολούθησης ανανεώνονται συνεχώς και είναι ορατές στον χρήστη.
- **datahandler (package)**

DataHandler (class) : Η συγκεκριμένη κλάση διαχειρίζεται την ανάγνωση (**LoadFromDisk**) και την εγγραφή (**SaveToDisk**) από και προς τον δίσκο, αντίστοιχα, των τιμών των seek bars για τις μεταβλητές του **DataSettings**.
- **savelisteners (package)**

ListenersStorage (class) : Σε αυτή την κλάση αποθηκεύονται οι listeners που παρακολουθούν τις εναλλαγές των αισθητήρων για να μπορούν να είναι προσπελάσιμες μέσω του ονόματος της κλάσης (**public static**) απο τις κλάσεις **OnClickListenerStartMonitoring** και **OnClickListenerStopMonitoring** που θα τους χρειαστούν αμετάβλητους ώστε κάθε φορά να μην δημιουργούνται νέοι listeners αλλά οι υπάρχοντες να δείχνουν στο ίδιο αντικείμενο με το πάτημα των κουμπιών **START/STOP**.
- **onclicklisteners (package)**: Το πακέτο **onclicklisteners** περιέχει τις κλάσεις που προσδιορίζουν την ενέργεια που θα εκτελεστεί όταν πατηθεί το αντίστοιχο κουμπί (γεγονός). Είναι η διεπαφή ανάμεσα στα γεγονότα, δηλαδή τα αντίστοιχα πατήματα κουμπιών, και στις ενέργειες που θα εκτελεστούν στη συνέχεια ανάλογα με το κάθε κουμπί.

Collision Avoidance

- a. **OnClickListenerSaveData (class):** Η διεπαφή ανάμεσα στο πάτημα του κουμπιού **SAVE** και της ενέργειας που θα εκτελεστεί στη συνέχεια, δηλαδή την εγγραφή των δεδομένων στον δίσκο.
- b. **OnClickListenerSettings (class):** Η διεπαφή ανάμεσα στο πάτημα του κουμπιού **SETTINGS** και της ενέργειας που θα εκτελεστεί στη συνέχεια, δηλαδή τη δημιουργία του νέου activity Settings.
- c. **OnClickListenerStartMonitoring (class):** Η διεπαφή ανάμεσα στο πάτημα του κουμπιού **START** και της ενέργειας που θα εκτελεστεί στη συνέχεια. Αρχικά, διαβάζονται οι τιμές κατωφλίου και συχνότητας δειγματοληψίας για να ξεκινήσει η παρακολούθηση των επιλεγμένων αισθητήρων. Ταυτόχρονα, γίνεται η σύνδεση των listeners των αισθητήρων μέσω της κλάσης **ListenersStorage**.

*ΠΑΡΑΤΗΡΗΣΗ: Στα API όπου δοκιμάσαμε την εφαρμογή υπάρχει η παράμετρος της συχνότητας. Μέσα στην κλάση **OnClickListenerStartMonitoring** γίνεται μετατροπή της συχνότητας που θέτει ο χρήστης στα settings σε περίοδο με μονάδα μέτρησης τα *microsecond(μs)*. Η συνάρτηση *registerlistener* παίρνει σαν τρίτη παράμετρο την περίοδο, ώστε να την μεταβιβάσει στην *OnSensorChanged*, η οποία καλείται από την πρώτη.*

- d. **OnClickListenerStopMonitoring (class):** Η διεπαφή ανάμεσα στο πάτημα του κουμπιού **STOP** και της ενέργειας που θα εκτελεστεί στη συνέχεια. Σταματάει η παρακολούθηση των αισθητήρων και ταυτόχρονα γίνεται η αποσύνδεση των listeners των αισθητήρων μέσω της κλάσης **ListenersStorage**.
- **onprogresslisteners (package):** Το πακέτο **onprogresslisteners** περιέχει τις κλάσεις που προσδιορίζουν την ενέργεια που θα εκτελεστεί όταν υπάρξει αλλαγή στον δρομέα του κάθε seek bar (γεγονός).
- a. **OnSamplingRateProgressListener (class):** Για κάθε εναλλαγή του seek bar της **συχνότητας**, η τιμή εμφανίζεται στο αντίστοιχο text view μόνο αν είναι μεγαλύτερη του 0. Διαφορετικά, με τη χρήση ενός toast ενημερώνεται ο χρήστης για μη αποδεκτή τιμή ώστε να την αλλάξει.
 - b. **OnFirstSensorProgressListener (class):** Για κάθε εναλλαγή του seek bar του **proximity**, η τιμή εμφανίζεται στο αντίστοιχο text view μόνο αν είναι μεγαλύτερη του 0. Διαφορετικά, με τη χρήση ενός toast ενημερώνεται ο χρήστης για μη αποδεκτή τιμή ώστε να την αλλάξει.
 - c. **OnSecondSensorProgressListenerX (class):** Για κάθε εναλλαγή του seek bar του άξονα **x** του **accelerometer**, η τιμή εμφανίζεται στο αντίστοιχο text view μόνο αν είναι μεγαλύτερη του 0. Διαφορετικά, με τη χρήση ενός toast ενημερώνεται ο χρήστης για μη αποδεκτή τιμή ώστε να την αλλάξει.
 - d. **OnSecondSensorProgressListenerY (class):** Για κάθε εναλλαγή του seek bar του άξονα **y** του **accelerometer**, η τιμή εμφανίζεται στο αντίστοιχο text view μόνο αν είναι μεγαλύτερη του 0. Διαφορετικά, με τη χρήση ενός toast ενημερώνεται ο χρήστης για μη αποδεκτή τιμή ώστε να την αλλάξει.
 - e. **OnSecondSensorProgressListenerZ (class):** Για κάθε εναλλαγή του seek bar του άξονα **z** του **accelerometer**, η τιμή εμφανίζεται στο αντίστοιχο text view μόνο αν είναι μεγαλύτερη του 0. Διαφορετικά, με τη χρήση ενός toast ενημερώνεται ο χρήστης για μη αποδεκτή τιμή ώστε να την αλλάξει.



Εικόνα 5: Invalid Value

- **onsensoreventlisteners (package):** Το πακέτο **onsensoreventlisteners** περιέχει τις κλάσεις που προσδιορίζουν την ενέργεια που θα εκτελεστεί όταν υπάρξει αλλαγή στην τιμή των αισθητήρων κατά την παρακολούθηση (γεγονός).
 - a. **OnProximitySensorEventListener (class):** Για κάθε εναλλαγή των τιμών του αισθητήρα **proximity**, οι τιμές αποθηκεύονται στην μεταβλητή **proximity** της κλάσης **SensorData**.
 - b. **OnAccelerometerSensorEventListenerX (class):** Αρχικά, δημιουργούμε 3 συναρτήσεις για την εγκατάσταση, την έναρξη και τον τερματισμό της μουσικής. Για κάθε εναλλαγή των τιμών του άξονα **x** του αισθητήρα **accelerometer**, οι τιμές αποθηκεύονται στην μεταβλητή **x** της κλάσης **SensorData**. Για κάθε εναλλαγή των τιμών του άξονα **y** του αισθητήρα **accelerometer**, οι τιμές αποθηκεύονται στην μεταβλητή **y** της κλάσης **SensorData**. Για κάθε εναλλαγή των τιμών του άξονα **z** του αισθητήρα **accelerometer**, οι τιμές αποθηκεύονται στην μεταβλητή **z** της κλάσης **SensorData**. Ορίζουμε τη συνθήκη ενεργοποίησης της ειδοποίησης, όπως περιγράφεται στην εισαγωγή, η οποία ελέγχεται σε αυτή την κλάση, καθώς η **OnSensorChange** καλείται συνεχώς λόγω της μεγάλης συχνότητας αλλαγής των τιμών των τριών αξόνων του **Accelerometer sensor**.
- **protoproject (package):** Το πακέτο που περιέχει τα δύο activities της υλοποίησής μας.
 - a. **MainActivity:** Πρόκειται για το βασικό activity της εφαρμογής. Ο χρήστης, ανοίγοντας την εφαρμογή βλέπει τα 3 κουμπιά **START**, **STOP**, **SETTINGS**, το **options menu**, το **logo** μαζί με το όνομα της εφαρμογής και τα ονόματα των αισθητήρων που θα παρακολουθηθούν. Με το πάτημα του κουμπιού **START**, εμφανίζονται οι εναλλαγές στις τιμές των αισθητήρων. Μέσω του **options menu** μπορεί να μεταβεί στις ρυθμίσεις της εφαρμογής με την επιλογή **Application Settings** ή να βγει από την εφαρμογή με την επιλογή του **exit**. Ανάλογα με την επιλογή που κάνει, εμφανίζεται αντίστοιχο **Android Toast**, όπως φαίνεται στις εικόνες.

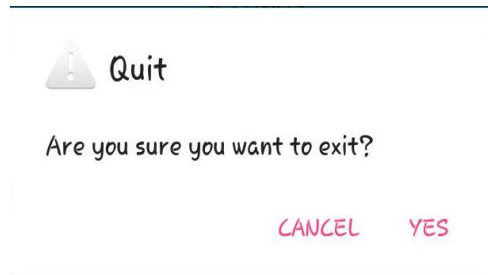


Εικόνα 6: Exit



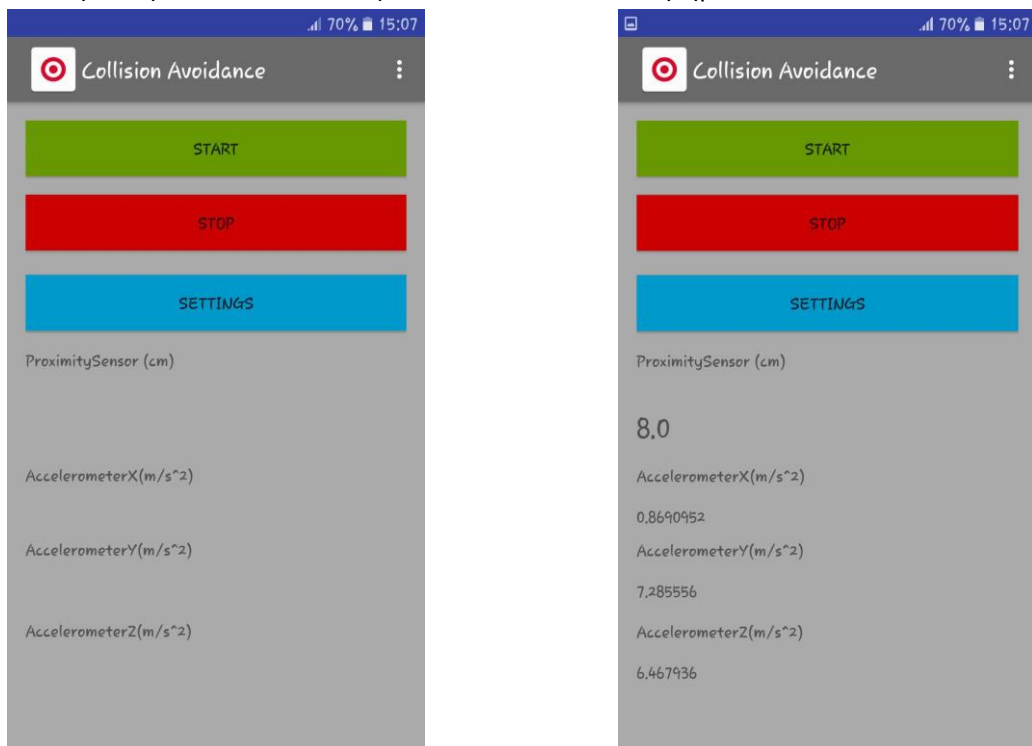
Εικόνα 7: Settings

Για την έξοδο από την εφαρμογή έχουμε δημιουργήσει την συνάρτηση **exitapplication()**, η οποία καλείται είτε όταν επιλέγεται έξοδος από το μενού, είτε όταν πατηθεί το **back button**. Η συγκεκριμένη συνάρτηση εμφανίζει ένα παράθυρο για επιβεβαίωση εξόδου.



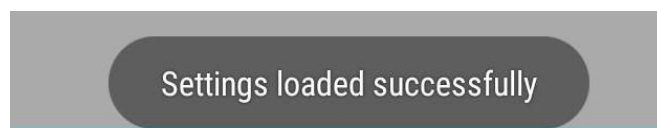
Εικόνα 8: Quit

Αν επιβεβαιωθεί η έξοδος αποδεσμεύεται από την μνήμη ο χώρος στον οποίο αποθηκεύτηκαν οι διευθύνσεις των listeners των αισθητήρων.

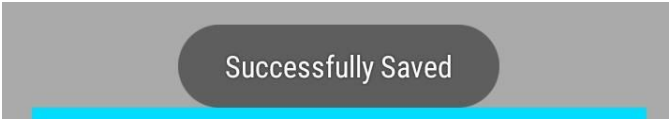


Εικόνα 9: Main Activity

- b. **SettingsActivity**: Πρόκειται για το δευτερεύον activity της εφαρμογής. Ο χρήστης μπαίνοντας στις ρυθμίσεις, ενημερώνεται με αντίστοιχο **toast** για το επιτυχές φόρτωμα των ρυθμίσεων από την μνήμη. Επιπλέον, βλέπει τα **5 seek bars** που έχουν ήδη αναφερθεί, το κουμπί **SAVE** και το **logo** μαζί με το όνομα της εφαρμογής. Με το πάτημα του κουμπιού **SAVE** αποθηκεύονται οι τρέχουσες τιμές των seek bar και εμφανίζεται αντίστοιχο μήνυμα **toast** για την επιτυχία ή μη της αποθήκευσης. Με το πάτημα του **back button** επιστρέφει και πάλι στο main activity της εφαρμογής.



Εικόνα 10: Settings Loaded



Εικόνα 11: Settings Saved

Collision Avoidance

ΠΗΓΕΣ

- [1] <http://www.oracle.com/technetwork/articles/java/index-137868.html>
- [2] <https://developer.android.com/reference/android/widget/SeekBar.OnSeekBarChangeListener.html>
- [3] <http://stackoverflow.com/questions/37244357/how-to-play-music-in-android-studio>
- [4] <http://stackoverflow.com/questions/3033135/android-seekbar-minimum-value>
- [5] <http://stackoverflow.com/questions/14376807/how-to-read-write-string-from-a-file-in-android>
- [6] <http://stackoverflow.com/questions/17800615/android-studio-default-project-directory>
- [7] <http://stackoverflow.com/questions/26615889/how-to-change-the-launcher-logo-of-an-app-in-android-studio>
- [8] <http://stackoverflow.com/questions/33068542/how-to-get-context-from-on-click-method-in-dialogbox>
- [9] <http://stackoverflow.com/questions/2257963/how-to-show-a-dialog-to-confirm-that-the-user-wishes-to-exit-an-android-activity>
- [10] <http://stackoverflow.com/questions/24993825/handle-back-button-in-alertdialog-builder>
- [11] <http://stackoverflow.com/questions/10346011/how-to-handle-back-button-with-in-the-dialog>
- [12] <http://stackoverflow.com/questions/28564343/need-for-popbackstack-in-onbackpressed>
- [13] <http://stackoverflow.com/questions/26472417/logo-are-not-displayed-in-actionbar-using-appcompat>
- [14] <http://stackoverflow.com/questions/14570512/how-to-get-the-parent-view-of-a-button-within-a-list-adapter>
- [15] <http://stackoverflow.com/questions/5447092/get-context-inside-onclickdialoginterface-v-int-buttonid>
- [16] https://developer.android.com/guide/topics/sensors/sensors_overview.html
- [17] <http://stackoverflow.com/questions/15326290/get-android-seekbar-value-and-display-it-on-screen>
- [18] <http://stackoverflow.com/questions/9792888/android-seekbar-set-progress-value>
- [19] <http://stackoverflow.com/questions/14376807/how-to-read-write-string-from-a-file-in-android>
- [20] <http://stackoverflow.com/questions/35481924/write-a-string-to-a-file>
- [21] <http://stackoverflow.com/questions/18100485/how-to-create-and-or-append-to-a-file-on-android>
- [22] <http://stackoverflow.com/questions/13426142/bufferedwriter-not-writing-everything-to-its-output-file>
- [23] <http://stackoverflow.com/questions/4072706/the-connection-to-adb-is-down-and-a-severe-error-has-occured>