



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Πτυχιακή Εργασία**



Τροποποιήθηκε μερικώς για ανάρτηση στο GitHub

## Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

## Ευχαριστίες

Η εκπόνηση αυτής της εργασίας και η ανάπτυξη της σχετικής εφαρμογής δε θα μπορούσε να έχει πραγματοποιηθεί με επιτυχία χωρίς την καθοριστική συνεισφορά πολλών ανθρώπων οι οποίοι στάθηκαν στο πλευρό μου καθ' όλη τη διάρκεια ενασχόλησής μου με την εν λόγω υλοποίηση. Στο πλαίσιο αυτό θα ήθελα να ευχαριστήσω:

Τον επιβλέποντα καθηγητή μου όπου με την καθοδήγησή του η εφαρμογή αναπτύχθηκε χωρίς να αντιμετωπιστούν τεχνικές δυσκολίες καθώς και με τις συμβουλές του συνέβαλε ώστε η εφαρμογή να αποκτήσει σημαντική λειτουργικότητα και κατ' επέκταση περισσότερο ενδιαφέρον για τον τελικό χρήστη.

Την οικογένεια μου οι οποίοι αφιέρωσαν πολλές ώρες ώστε να χρησιμοποιήσουν και να δοκιμάσουν την εφαρμογή εντοπίζοντας προβλήματα και παραλείψεις συμβάλλοντας με αυτό τον τρόπο στο καλύτερο δυνατό τελικό αποτέλεσμα.

Τους φίλους μου που με το ενεργό τους ενδιαφέρον έκανα προτάσεις ώστε η εφαρμογή να βελτιωθεί περισσότερο και να φτάσει σε ισάξιο επίπεδο με άλλες υλοποιήσεις που υπάρχουν στην αγορά.

## Περίληψη

Η συγκεκριμένη εργασία παρουσιάζει την ανάπτυξη μιας εφαρμογής Android πραγματικού χρόνου, πολλαπλών παικτών η οποία επιτρέπει στους χρήστες να παίξουν το γνωστό παιχνίδι στρατηγικής «Ναυμαχία» έχοντας ως αντιπάλους άλλους χρήστες που μπορούν να βρίσκονται σε οποιαδήποτε τοποθεσία ανά τον κόσμο. Η υλοποίηση της εφαρμογής γίνεται με χρήση της γλώσσας προγραμματισμού Java καθώς επίσης χρησιμοποιείται η απομακρυσμένη βάση δεδομένων Firebase ώστε να επιτευχθεί η δυνατότητα παιχνιδιού με οποιονδήποτε χρήστη όσο μακριά και αν βρίσκεται σε πραγματικό χρόνο. Μέσω του μοντέρνου γραφικού σχεδιασμού και των πολλαπλών τρόπων ανάδρασης κατά τη διάρκεια του παιχνιδιού επιτρέπει στον τελικό χρήστη να συμμετέχει σε «ναυμαχίες» μέσα από μια ολοκληρωμένη και διαδραστική εμπειρία παιχνιδιού.

Λέξεις Κλειδιά: Java, Firebase, Εφαρμογή Android, Πραγματικού χρόνου, Πολλαπλών παικτών

## Abstract

This paper presents the development of a real-time, multiplayer Android application that allows users to play the well-known strategy game 'Battleship' against other users who may be located anywhere in the world. The implementation of the application is carried out using the Java programming language, and also utilizes the Firebase remote database to enable gameplay with any user, no matter how far they are, in real time. Through modern graphic design and multiple interaction methods during the game, the application provides the end user with the opportunity to participate in 'battleships' through a complete and interactive gaming experience.

Key Words: Java, Firebase, Android app, Real-time, Multiplayer

## Αφιερώσεις

Η συγγραφή αυτής της εργασίας και η υλοποίηση της αντίστοιχης εφαρμογής αφιερώνεται σε όλους εκείνους που με στήριξαν και με ενθάρρυναν να συνεχίσω να προσπαθώ και να εξελίσσομαι.

Πρώτα απ' όλα στους καθηγητές και τους συμφοιτητές μου οι οποίοι καθ' όλη τη διάρκεια των σπουδών στάθηκαν στο πλευρό μου, με ενίσχυσαν με νέες γνώσεις και με βοήθησαν να φτάσω στο επίπεδο που βρίσκομαι σήμερα.

Τέλος, στην οικογένειά μου και στους φίλους μου όπου με τη στήριξη και τις συμβουλές τους με όπλισαν με υπομονή και επιμονή ώστε να επιτυγχάνω κάθε φορά το καλύτερο δυνατό τελικό αποτέλεσμα.

# Πίνακας Περιεχομένων

Copyright © .....	ii
Ευχαριστίες .....	iii
Περίληψη .....	iv
Abstract.....	iv
Αφιερώσεις .....	v
Πίνακας Περιεχομένων .....	vi
Εισαγωγή.....	1
1. Ανασκόπηση Πεδίου .....	2
1.1 Παρόμοιες Εφαρμογές.....	2
1.2 Ορισμός Πλαισίου Ανάπτυξης της Εφαρμογής .....	7
2. Εγχειρίδιο Χρήστη .....	8
2.1 Οθόνη Καλωσορίσματος - Σύνδεση .....	8
2.2 Οθόνη Εγγραφής .....	9
2.3 Οθόνη «Αρχική» .....	10
2.4 Οθόνη «Παιχνίδι».....	11
2.5 Οθόνη «Στατιστικά» .....	13
2.6 Οθόνη «Ρυθμίσεις» .....	15
2.7 Οθόνη «Ναυμαχία» - Κεντρικό Παιχνίδι .....	18
2.8 Προσαρμογή για Ταμπλέτα .....	21
3. Υλοποίηση Εφαρμογής.....	22
3.1 Γραφική Διεπαφή .....	22
3.2 Επιχειρησιακή Λογική.....	24
3.2.1 StartActivity.java .....	25
3.2.2 SignUpActivity.java .....	25
3.2.3 HomeActivity.java.....	26
3.2.4 HomeFragment.java .....	26
3.2.5 GameFragment.java .....	27
3.2.6 StatisticsFragment.java.....	29
3.2.7 AccountFragment.java.....	29
3.2.8 BattleshipActivity.java .....	32
3.2.9 SystemOperations.java .....	34
3.3 Μοντέλα .....	34

<b>3.3.1 User.java</b> .....	35
<b>3.3.2 Opponent_User.java</b> .....	36
<b>4. Firebase</b> .....	37
<b>4.1 Περιγραφή &amp; Τρόπος Ενσωμάτωσης</b> .....	37
<b>4.2 Χρήση Κανόνων στη Real-Time Database</b> .....	40
<b>5. Android UI</b> .....	42
<b>5.1 Navigation Bar</b> .....	42
<b>5.2 RecyclerView</b> .....	43
<b>Συμπεράσματα &amp; Μελλοντικές Επεκτάσεις</b> .....	46
<b>Βιβλιογραφία</b> .....	47

## Εισαγωγή

Η εφαρμογή «Ναυμαχία» που υλοποιήθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας επιδιώκει να δώσει στο χρήστη όλα τα απαραίτητα μέσα ώστε να έχει μια ολοκληρωμένη, απρόσκοπτη και ποιοτική εμπειρία παιχνιδιού με οποιονδήποτε άλλον χρήστη ανά τον κόσμο σε πραγματικό χρόνο. Η επίτευξη των παραπάνω στόχων δε θα μπορούσε να επιτευχθεί χωρίς τη λεπτομερή ανάλυση και μελέτη των τεχνολογιών που υπάρχουν αυτή τη στιγμή διαθέσιμες. Για το σκοπό αυτό η εργασία διαρθρώνεται σε πέντε κύρια κεφάλαια στα οποία παρουσιάζονται διαφορετικές πτυχές ανάπτυξης της εφαρμογής:

Το πρώτο κεφάλαιο αφορά την ανασκόπηση του πεδίου, δηλαδή παρόμοιων εφαρμογών που έχουν υλοποιηθεί και είναι ήδη διαθέσιμες στην αγορά. Συγκεκριμένα, πραγματοποιείται μια αναλυτική παρουσίαση των υφιστάμενων εφαρμογών και των δυνατοτήτων τους καθώς επίσης και των μειονεκτημάτων που αυτές έχουν. Επίσης, παρουσιάζεται το πλαίσιο ανάπτυξης της εφαρμογής το οποίο διαμορφώθηκε από τα αποτελέσματα της ανασκόπησης πεδίου.

Το δεύτερο κεφάλαιο επικεντρώνεται στην λεπτομερή παρουσίαση του τρόπου χρήσης της εφαρμογής από την πλευρά του τελικού χρήστη. Πιο αναλυτικά, μέσω εικόνων παρουσιάζονται όλα τα βασικά σημεία της εφαρμογής όπως η διαδικασία εγγραφής και σύνδεσης, ο τρόπος εκκίνησης του παιχνιδιού και επιλογής αντιπάλου καθώς και η προβολή στατιστικών και η πραγματοποίηση ρυθμίσεων/παραμετροποιήσεων από τον χρήστη.

Το τρίτο κεφάλαιο αναλύει τις τεχνικές λεπτομέρειες υλοποίησης της εφαρμογής. Δηλαδή, στο κεφάλαιο αυτό περιγράφεται η διάρθρωση και η λειτουργία του κώδικα της εφαρμογής. Επίσης, γίνεται αναφορά στον τρόπο που γίνεται η διαχείριση και ο συντονισμός πολλαπλών ασύγχρονων λειτουργιών στον πηγαίο κώδικα.

Το τέταρτο κεφάλαιο αναδεικνύει βασικά χαρακτηριστικά της Firebase και συγκεκριμένα προβαίνει σε μια λεπτομερή παρουσίαση της RealTime Database καθώς και πως γίνεται η διαχείριση των κανόνων και των δικαιωμάτων χρήστη σε αυτή ώστε να διασφαλιστεί η ομαλή και ασφαλής λειτουργία της εφαρμογής.

Τέλος, το πέμπτο κεφάλαιο εστιάζει σε συγκεκριμένα γραφικά στοιχεία που χρησιμοποιήθηκαν στην ανάπτυξη της εφαρμογής. Αυτά τα γραφικά στοιχεία είναι ιδιαίτερα σημαντικά τόσο για την ποιοτική εμπειρία χρήσης όσο και για την ομαλή λειτουργία της εφαρμογής σε περιπτώσεις μεγάλης ποσότητας πληροφορίας προς προβολή.



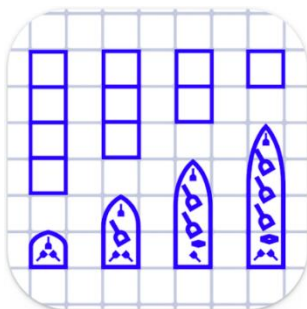
# 1. Ανασκόπηση Πεδίου

Το κεφάλαιο αυτό παρουσιάζει παρόμοιες υλοποιήσεις καθώς και πως αυτές συνέβαλαν στον προσδιορισμό του πλαισίου ανάπτυξης της εφαρμογής.

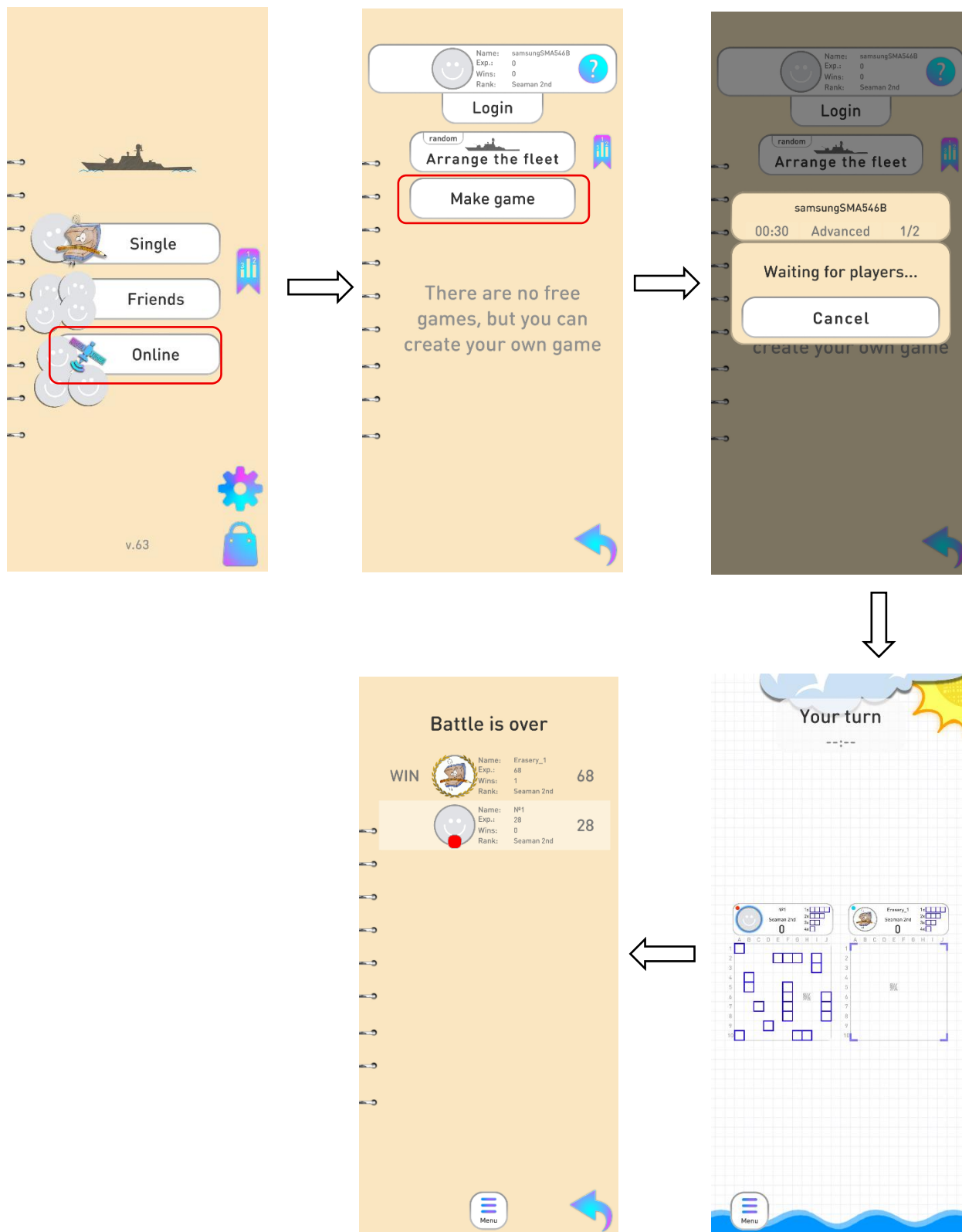
## 1.1 Παρόμοιες Εφαρμογές

Αρχικά κατά την έναρξη υλοποίησης της εφαρμογής μελετήθηκαν εφαρμογές/παιχνίδια που παρείχαν παρόμοια λειτουργικότητα και ήταν ήδη διαθέσιμες. Πιο συγκεκριμένα, εξετάστηκε ο τρόπος λειτουργίας τους, οι δυνατότητες που παρέχουν καθώς και τα μειονεκτήματα που παρουσιάζουν οι ακόλουθες εφαρμογές:

### Sea Battle Batalies



Η συγκεκριμένη εφαρμογή έχει αναπτυχθεί με χρήση της πλατφόρμας Unity με σκοπό να προσφέρει μια όμορφη εμπειρία στους παίκτες του παιχνιδιού. Επιλέγοντας ο χρήστης την επιλογή «Online» από το αρχικό menu κατευθύνεται στην οθόνη για την παραμετροποίηση και έναρξη του online παιχνιδιού με άλλους παίκτες. Από αυτή την οθόνη αφού ο χρήστης ορίσει τη διάταξη των πλοίων του πατάει το κουμπί «Make Game» περιμένοντας άλλους χρήστες να συνδεθούν ώστε να ξεκινήσει η Ναυμαχία. Όταν συγκεντρωθούν οι αντίπαλοι παίκτες πλέον το παιχνίδι ξεκινάει και ο χρήστης κατευθύνεται στην οθόνη του κυρίως παιχνιδιού όπου θα διαγωνιστεί με τον αντίπαλό του. Το παιχνίδι εξελίσσεται με τη συμμετοχή και την αλληλεπίδραση των δύο παικτών προσπαθώντας ο ένας να «χτυπήσει» πλοίο του αντιπάλου. Τέλος, ο παίκτης ο οποίος θα καταφέρει να βυθίσει πρώτος όλα τα πλοία του αντιπάλου είναι ο νικητής του συγκεκριμένου γύρου.

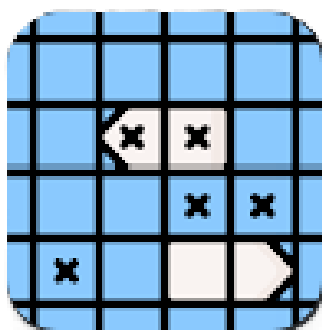


Παρουσίαση Εφαρμογής «Sea Battle Batalies»

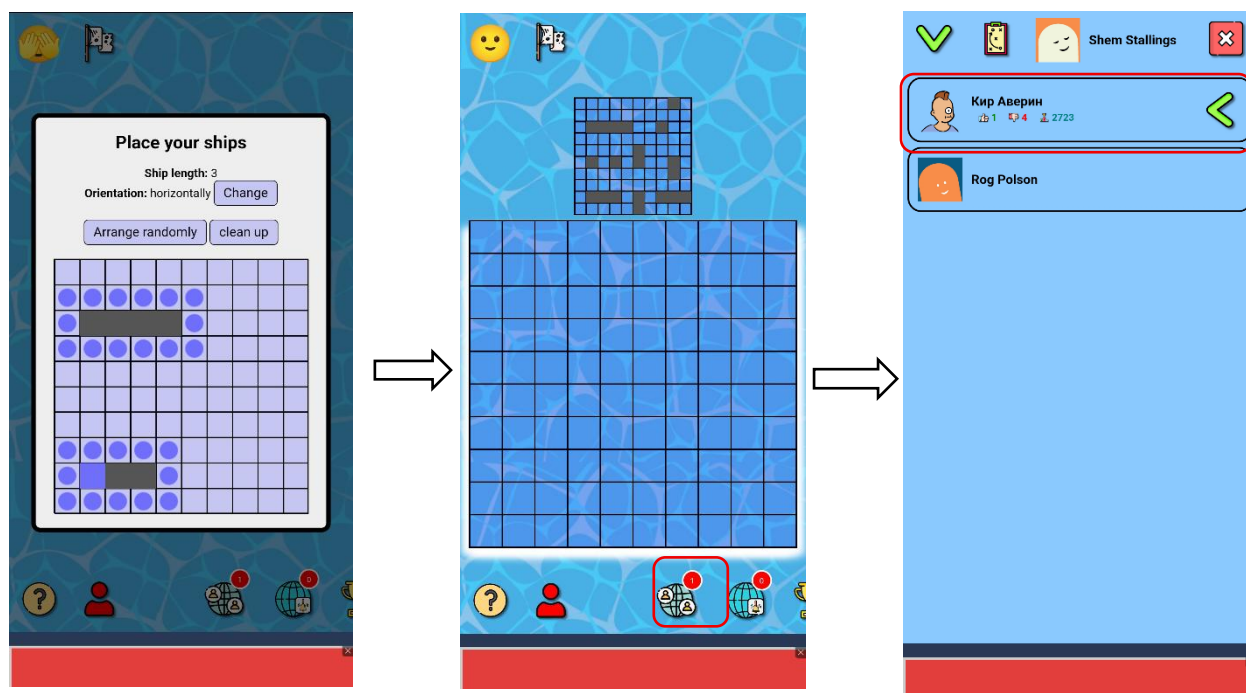
Η εφαρμογή παρόλο που χρησιμοποιεί ένα όμορφο και ποιοτικό περιβάλλον διεπαφής έχει και ορισμένα μειονεκτήματα τα οποία πρέπει να επισημανθούν και να ληφθούν υπόψη. Πιο αναλυτικά παρατηρήθηκαν τα ακόλουθα:

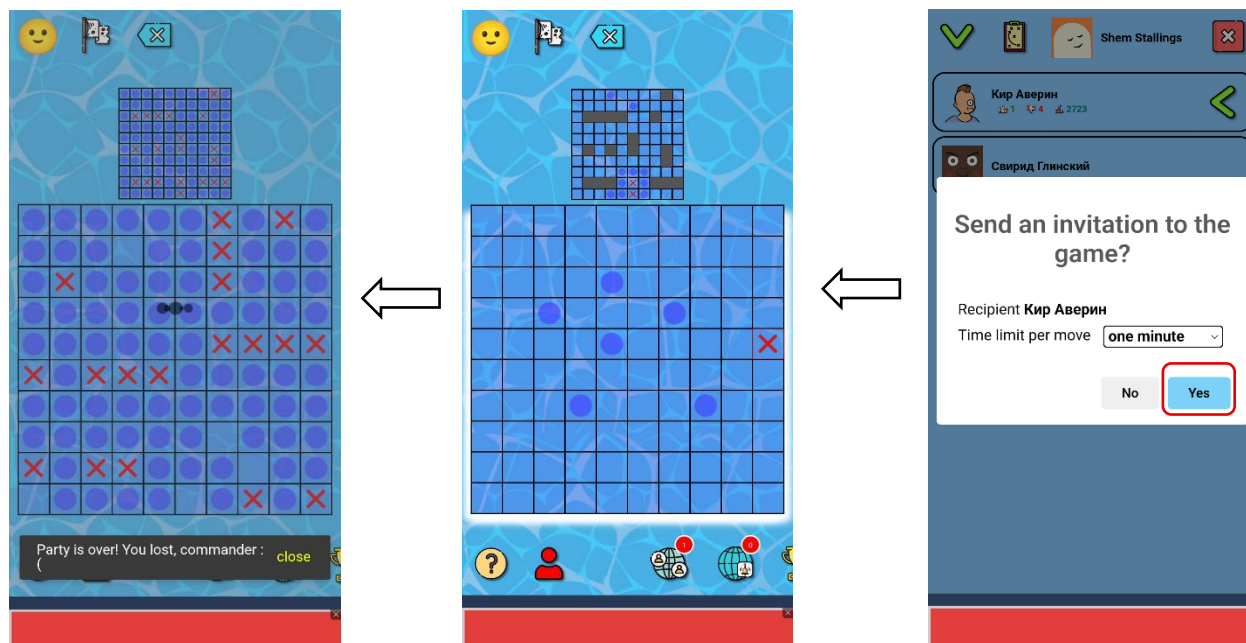
1. Κατά τη διαδικασία εκκίνησης του Online παιχνιδιού δε δίνεται η δυνατότητα στο χρήστη να επιλέξει αντίπαλο από τους διαθέσιμους εκείνη τη στιγμή. Αντιθέτως αναμένει έναν οποιονδήποτε παίκτη να γίνει διαθέσιμος ώστε να ξεκινήσει η ναυμαχία. Η μόνη επιλογή που δίνεται είναι να δημιουργήσει ομάδα από φίλους ώστε να παίξει αποκλειστικά με αυτούς όταν θα είναι συνδεδεμένοι ωστόσο αυτό σε κάθε περίπτωση δεν επιτρέπει την άμεση επιλογή συγκεκριμένου αντιπάλου.
2. Όπως φαίνεται και από τα στιγμιότυπα οθόνης που παρατίθενται κατά τη διάρκεια της ναυμαχίας δε γίνεται πλήρης εκμετάλλευση του μεγέθους της οθόνης με αποτέλεσμα ο χρήστης κατά την αλληλεπίδρασή του με την εφαρμογή κατά την εκτέλεση του παιχνιδιού ενδεχομένως να συναντήσει δυσκολίες στην επιλογή κελιού για την πραγματοποίηση επίθεσης σε πλοίο του αντιπάλου.
3. Τέλος, εντοπίστηκε ότι δεν υπάρχει η δυνατότητα επικοινωνίας ή ανάδρασης με το αντίπαλο παίκτη κατά τη διάρκεια εξέλιξης του παιχνιδιού. Αυτό θα μπορούσε να επιτευχθεί με την υλοποίηση κάποιας απλής υπηρεσίας chat ή με κάποιον προκαθορισμένο τρόπο «έκφρασης συναισθημάτων» στον αντίπαλο κάτι που θα προσέδιδε μεγαλύτερη ζωντάνια στην συνολική εμπειρία του παιχνιδιού.

### Battleship on Cells



Η εφαρμογή αυτή παρέχει στο χρήστη τη δυνατότητα να παίξει το παιχνίδι Ναυμαχία με οποιονδήποτε διαθέσιμο παίκτη υπάρχει εκείνη τη στιγμή διαθέσιμος μέσα από μια όμορφη και ποιοτική γραφική διεπαφή. Ο χρήστης πατώντας το αντίστοιχο κουμπί με τη «σφαίρα» βλέπει τη λίστα των διαθέσιμων παικτών εκείνη τη στιγμή. Στη συνέχεια επιλέγει έναν και στέλνει αίτημα για μάχη στον αντίπαλο. Εφόσον το αίτημα γίνει αποδεκτό (οι θέσεις των πλοίων έχουν οριστεί στην αρχή) τότε η ναυμαχία ξεκινάει οδηγώντας τον χρήστη στην οθόνη εξέλιξης του παιχνιδιού. Εκεί ο χρήστης βλέπει έναν κεντρικό πίνακα πατώντας στα κελιά του οποίου προσπαθεί να βρει και να βυθίσει τα πλοία του αντιπάλου. Ο παίκτης ο οποίος βύθισε πρώτος όλα τα πλοία του αντιπάλου είναι ο νικητής.





Παρουσίαση Εφαρμογής «Battleship on Cells»

Το παιχνίδι αυτό αν και έχει σχεδιαστεί ικανοποιητικά καλά ώστε ο τελικός χρήστης να μην αντιμετωπίσει προβλήματα κατά τη χρήση του έχει ορισμένες αστοχίες οι οποίες πρέπει να αναφερθούν. Πιο συγκεκριμένα εντοπίστηκαν τα ακόλουθα:

1. Ο χρήστης πρέπει να ορίσει τις θέσεις των πλοίων του σειριακά με βάση τη σειρά που του επιβάλλει η εφαρμογή. Από τη στιγμή που τοποθετήσει ένα πλοίο δε μπορεί να του αλλάξει θέση πάνω στον πίνακα παρά μόνο του δίνεται η δυνατότητα να καθαρίσει όλο τον πίνακα και να τοποθετήσει όλα τα πλοία από την αρχή. Αυτός ο περιορισμός στον τρόπο τοποθέτησης των πλοίων δε διευκολύνει την ομαλή εκκίνηση του παιχνιδιού καθώς δεν υπάρχει η δυνατότητα ο παίκτης να αλλάξει θέσεις στα πλοία που έχει ήδη τοποθετήσει στο ταμπλό.
2. Τέλος αν και τα γραφικά είναι όμορφα δεν είναι επαρκώς επιμελημένα και βέλτιστα τοποθετημένα με αποτέλεσμα να εμφανίζονται πολλά εικονίδια και μεγάλη ποσότητα πληροφορίας στην οθόνη του χρήστη χωρίς να είναι ξεκάθαρο που πρέπει να πατήσει ώστε να πετύχει το επιδιωκόμενο αποτέλεσμα π.χ. εμφάνιση λίστας διαθέσιμων αντιπάλων για πρόκληση σε μάχη.

## 1.2 Ορισμός Πλαισίου Ανάπτυξης της Εφαρμογής

Έχοντας λάβει υπόψη το θέμα της πτυχιακής εργασίας, το οποίο είναι η ανάπτυξη εφαρμογής που επιτρέπει στους χρήστες να παίξουν το παιχνίδι «Ναυμαχία» σε πραγματικό χρόνο με οποιονδήποτε διαθέσιμο παίκτη την εκάστοτε χρονική στιγμή, καθώς επίσης και τα μειονεκτήματα που υπάρχουν σε ήδη διαθέσιμες σχετικές εφαρμογές, όπως αναφέρθηκε παραπάνω, αποφασίστηκε η εφαρμογή να περιλαμβάνει τα ακόλουθα στοιχεία.

Να υλοποιηθεί μια εφαρμογή Android σε γλώσσα προγραμματισμού Java και χρήση της γλώσσας σήμανσης XML για τον σχεδιασμό της γραφικής διεπαφής. Με σκοπό να αντιμετωπιστούν προβλήματα που υφίστανται σε άλλες σχετικές υλοποιήσεις, η εφαρμογή θα χρησιμοποιεί το μητρικό UI framework της Google, Material UI, το οποίο θα προσφέρει έναν καθαρό σχεδιασμό που ταυτόχρονα δεν θα επιβαρύνει τους πόρους του συστήματος. Επίσης, για να επιτευχθεί ο συγχρονισμός των παικτών σε πραγματικό χρόνο θα γίνει χρήση της Firebase RealTime Database. Ο χρήστης θα έχει τη δυνατότητα να αλλάξει τις θέσεις των τοποθετημένων πλοίων του από τις σχετικές ρυθμίσεις με εύκολο τρόπο και κατά τη διάρκεια του παιχνιδιού θα μπορεί να επικοινωνήσει με τον αντίπαλο οποιαδήποτε στιγμή μέσω chat. Ακόμα, θα μπορεί να προσκαλέσει σε μάχη οποιονδήποτε διαθέσιμο αντίπαλο επιθυμεί χωρίς περιορισμούς. Επιπρόσθετα, η εφαρμογή θα δίνει στο χρήστη τη δυνατότητα προβολής στατιστικών στοιχείων με όμορφο γραφικό τρόπο τόσο για τον ίδιο όσο και για τους καλύτερους/χειρότερους παίκτες του παιχνιδιού. Τέλος, θα προσφέρεται πλήθος επιλογών παραμετροποίησης της εφαρμογής, όπως αλλαγή της γλώσσας εμφάνισης και του θέματος (φωτεινού/σκοτεινού).

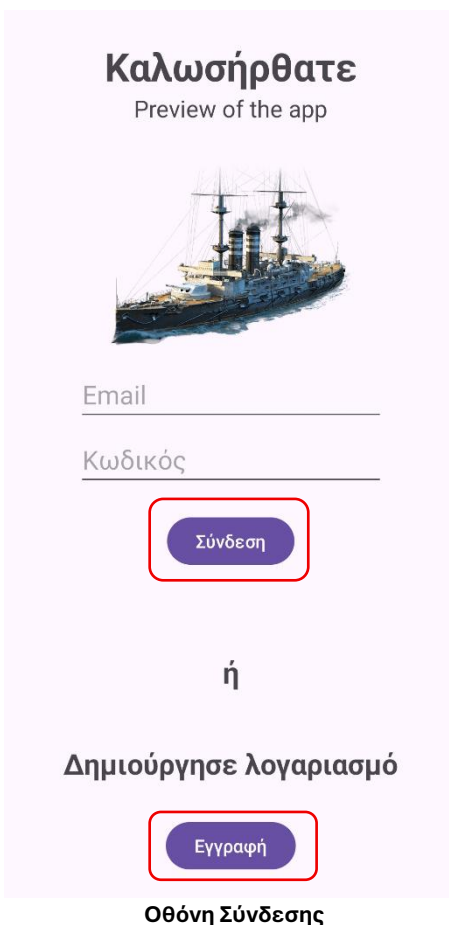
## 2. Εγχειρίδιο Χρήστη

Στο κεφάλαιο αυτό γίνεται αναλυτική παρουσίαση και περιγραφή του τρόπου χρήσης της εφαρμογής από τους χρήστες/παίκτες.

### 2.1 Οθόνη Καλωσορίσματος - Σύνδεση

Αυτή είναι η πρώτη οθόνη που χρήστης συναντά όταν ανοίγει για πρώτη φορά την εφαρμογή. Σε αυτή την οθόνη υπάρχουν οι εξής επιλογές είτε ο χρήστης να συμπληρώσει το Email και το Password και να πατήσει το κουμπί «Σύνδεση» για να συνδεθεί και να κατευθυνθεί στην κύρια οθόνη είτε να πατήσει το κουμπί «Εγγραφή», αν δεν έχει λογαριασμό, ώστε να ανακατευθυνθεί στη σελίδα εγγραφής νέου χρήστη και να ολοκληρώσει την εγγραφή του. Σε περίπτωση συμπλήρωσης λανθασμένων στοιχείων σύνδεσης τότε ο χρήστης θα ενημερωθεί με σχετικά μηνύματα σφάλματος.

Επισημαίνεται ότι αν ο χρήστης έχει συνδεθεί στην εφαρμογή και δεν έχει αποσυνδεθεί από αυτή πριν την κλείσει τότε ο συνδεδεμένος λογαριασμός διατηρείται στη μνήμη και θα πραγματοποιηθεί αυτόματη σύνδεση σε αυτόν όταν η εφαρμογή ανοίξει ξανά παρακάμπτοντας πλήρως την οθόνη σύνδεσης.



## 2.2 Οθόνη Εγγραφής

Αυτή η οθόνη επιτρέπει στον χρήστη να πραγματοποιήσει εγγραφή στην εφαρμογή ώστε να είναι σε θέση να τη χρησιμοποιήσει. Πιο αναλυτικά, απαιτείται να συμπληρώσει τα εξής στοιχεία: *Email*, *Κωδικό*, *Όνομα* και *Επώνυμο* και στη συνέχεια να πατήσει το κουμπί «Υποβολή». Πρέπει η συμπλήρωση των στοιχείων να γίνει σύμφωνα με τις οδηγίες που παρέχονται σε κάθε πεδίο καθώς και η διεύθυνση email να μην έχει χρησιμοποιηθεί ξανά για εγγραφή στην εφαρμογή. Αν κάποια/κάποιες από τις παραπάνω προϋποθέσεις δεν τηρείται τότε ένα μήνυμα σφάλματος θα εμφανιστεί δηλώνοντας ότι η διαδικασία εγγραφής δεν είναι δυνατό να ολοκληρωθεί. Εφόσον η εγγραφή ολοκληρωθεί επιτυχώς τότε ο χρήστης θα συνδεθεί αυτόματα και θα ανακατευθυνθεί στην κύρια οθόνη της εφαρμογής. Σε αυτό το σημείο επισημαίνεται ότι κατά τη διαδικασία της εγγραφής αποδίδονται αυτόματα θέσεις στα πλοία του χρήστη επομένως μπορεί να ξεκινήσει άμεσα να παίζει χωρίς να απαιτείται να προβεί σε τοποθέτησή τους – έχει τη δυνατότητα να δει και να αλλάξει τις θέσεις των πλοίων από την Οθόνη Ρυθμίσεων.

Οθόνη Εγγραφής



## 2.3 Οθόνη «Αρχική»

Η Αρχική οθόνη είναι η πρώτη οθόνη που συναντά ο χρήστης όταν συνδέεται ή προβαίνει σε επιτυχή εγγραφή στην εφαρμογή. Η οθόνη αυτή παρουσιάζει με όμορφα γραφικά τον τρόπο πλοήγησης στην εφαρμογή με χρήση του Navigation Bar το οποίο βρίσκεται στο κάτω μέρος της οθόνης. Συγκεκριμένα μέσω του Navigation Bar ο χρήστης μπορεί να πλοηγηθεί στις ακόλουθες ενότητες:

- Παιχνίδι. Εκεί μπορεί να δει τους διαθέσιμους αντιπάλους και να τους προκαλέσει σε «Ναυμαχία» στέλνοντας σχετικό αίτημα.
- Στατιστικά. Από την ενότητα αυτή μπορεί να δει χρήσιμα στατιστικά στοιχεία τόσο για τον ίδιο όσο και για τους άλλους παίκτες του παιχνιδιού.
- Ρυθμίσεις. Στην ενότητα αυτή έχει τη δυνατότητα να πραγματοποιήσει πλήθος ρυθμίσεων και παραμετροποιήσεων που αφορούν τόσο τη θέση των πλοίων του χρήστη στο ταμπλό όσο και την ίδια την εφαρμογή (π.χ. γλώσσα εμφάνισης).



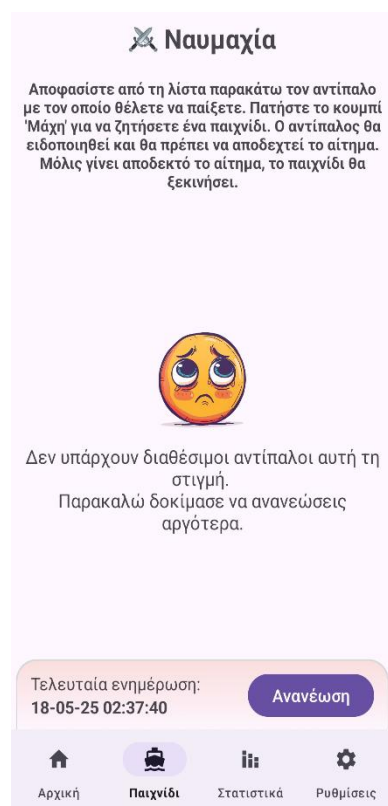
Οθόνη «Αρχική» (Α')



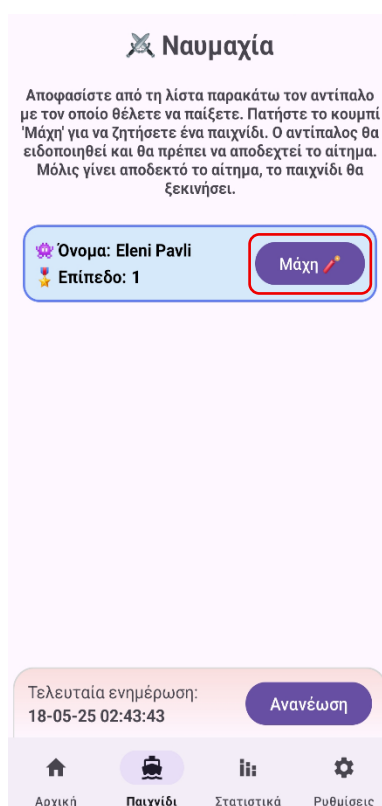
Οθόνη «Αρχική» (Β')

## 2.4 Οθόνη «Παιχνίδι»

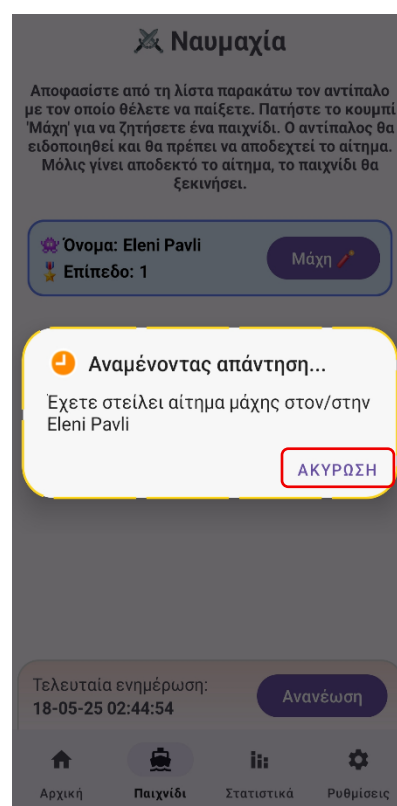
Η οθόνη αυτή επιτρέπει στον χρήστη να δει τους εκάστοτε διαθέσιμους αντιπάλους και να τους προκαλέσει σε ναυμαχία πατώντας το κουμπί «Μάχη». Τονίζεται ότι το παιχνίδι διαθέτει σύστημα μέτρησης πόντων και επιπέδων. Κάθε ένα επίπεδο είναι 100 πόντοι, δηλαδή 200 - 299 πόντοι είναι το Επίπεδο 2, 300 – 399 πόντοι είναι το Επίπεδο 3 και τα λοιπά. Κατά την αρχική εγγραφή ο χρήστης λαμβάνει 100 πόντους δηλαδή ανήκει στο Επίπεδο 1. Οι πόντοι αυξάνονται μόνο με νίκη επί του εκάστοτε αντιπάλου - ο παίκτης έχει τη δυνατότητα να προκαλέσει σε ναυμαχία τους αντιπάλους που είναι μέχρι ένα επίπεδο πάνω από το δικό του. Μετά την πρόκληση σε μάχη και μέχρι να απαντηθεί το σχετικό αίτημα από τον αντίπαλο ο παίκτης έχει τη δυνατότητα να το ακυρώσει πατώντας το κουμπί «Ακύρωση».



Οθόνη «Παιχνίδι» (Α')

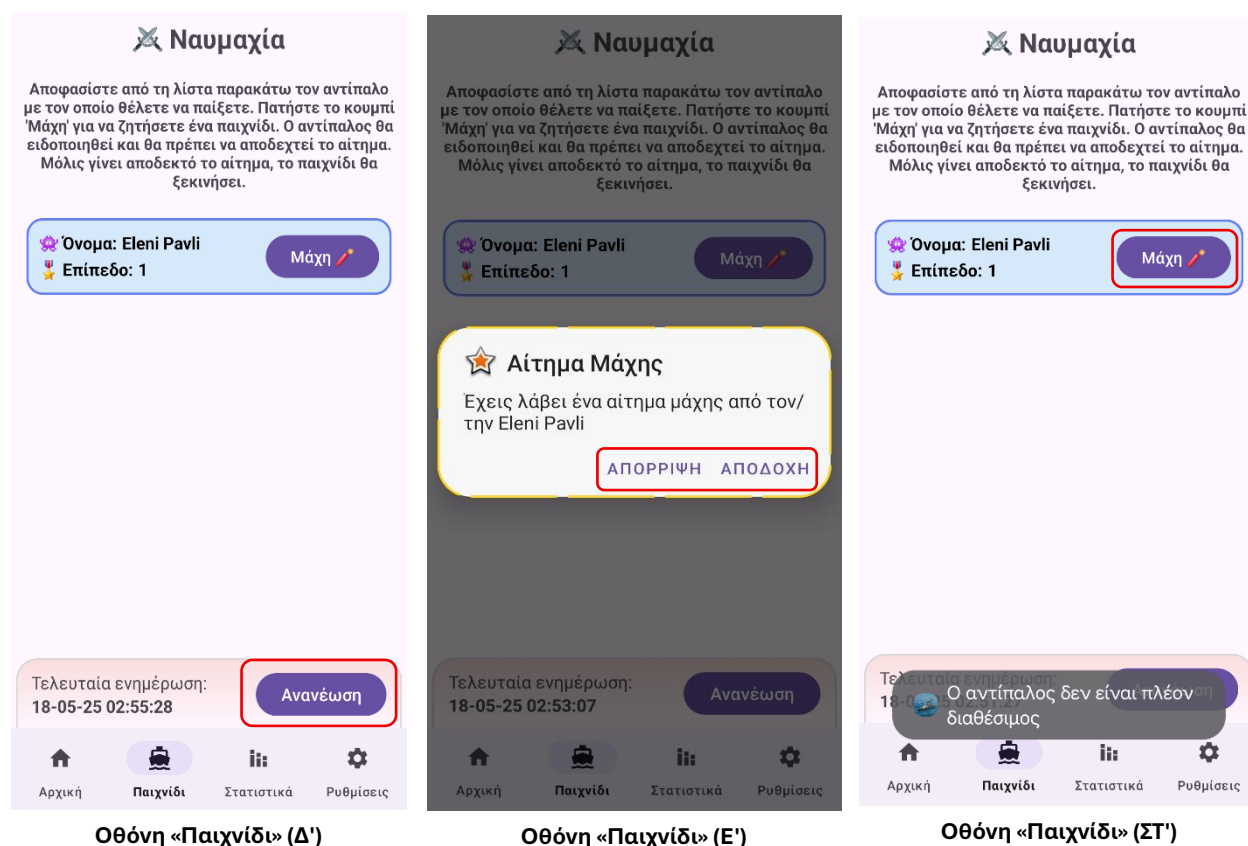


Οθόνη «Παιχνίδι» (Β')



Οθόνη «Παιχνίδι» (Γ')

Εδώ αξίζει να επισημανθεί ότι από τη στιγμή που εισέρχεται ο παίκτης στη συγκεκριμένη οθόνη αυτομάτως γίνεται ορατός ως διαθέσιμος για μάχη σε όλους τους υπόλοιπους ενεργούς παίκτες (εφόσον πληρούν το κριτήριο του επιπέδου που αναφέρθηκε νωρίτερα) και είναι πιθανό να λάβει αίτημα πρόκλησης σε μάχη από κάποιον από αυτούς. Τότε θα πρέπει ή να πατήσει το κουμπί «Αποδοχή» για να αποδεχτεί το αίτημα είτε το κουμπί «Απόρριψη» για να το απορρίψει. Επίσης, αν ο παίκτης θέλει να ανανεώσει τη λίστα των διαθέσιμων αντιπάλων πρέπει να πατήσει το κουμπί «Ανανέωση» που βρίσκεται στο κάτω μέρος της οθόνης. Τέλος αν για κάποιο λόγο ο αντίπαλος που επέλεξε να στείλει ο παίκτης αίτημα για μάχη δεν είναι πλέον διαθέσιμος θα ειδοποιηθεί με σχετικό μήνυμα.



## 2.5 Οθόνη «Στατιστικά»

Από την οθόνη αυτή ο χρήστης μπορεί να δει πλήθος διαφορετικών στατιστικών στοιχείων. Μέρος αυτών αποτελούν τα στατιστικά που αφορούν τους καλύτερους παίκτες ως προς το ποσοστό νικών που έχουν καταγράψει καθώς και τους καλύτερους και τους χειρότερους παίκτες ως προς τον υψηλότερο αριθμό νικών και ηττών αντίστοιχα. Κατά την εκκίνηση της οθόνης τα στατιστικά που αφορούν τους καλύτερους/χειρότερους παίκτες φορτώνονται δυναμικά, όταν ολοκληρωθεί η φόρτωση εμφανίζονται στον χρήστη οι 3 καλύτεροι παίκτες, δηλαδή εκείνοι με το υψηλότερο ποσοστό νικών. Στη συνέχεια ο χρήστης έχει τη δυνατότητα να πατήσει είτε το κουμπί «Νίκες» ώστε να εμφανιστούν οι 3 καλύτεροι παίκτες με τον μεγαλύτερο αριθμό νικών είτε το κουμπί «Ήττες» ώστε να εμφανιστούν οι 3 χειρότεροι παίκτες με τον μεγαλύτερο αριθμό ηττών. Αν θέλει να επιστρέψει πίσω στους 3 καλύτερους παίκτες με το μεγαλύτερο ποσοστό νικών πρέπει να πατήσει το κουμπί «Νίκες (%)».



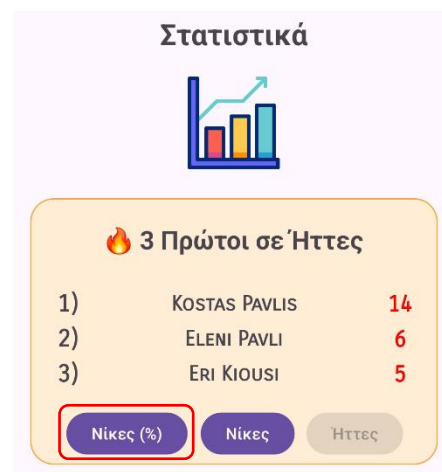
Οθόνη «Στατιστικά» (Α')



Οθόνη «Στατιστικά» (Β')



Οθόνη «Στατιστικά» (Γ')



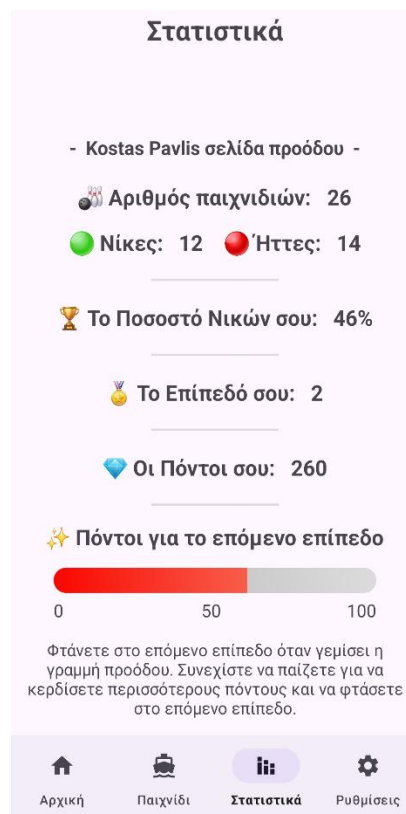
Οθόνη «Στατιστικά» (Δ')

Στην ίδια σελίδα ο χρήστης μπορεί να δει να παρουσιάζονται με όμορφο γραφικό τρόπο στατιστικά που αφορούν τη δική του επίδοση στο παιχνίδι μέχρι εκείνη τη στιγμή. Συγκεκριμένα μπορεί να δει:

- Τον συνολικό αριθμό παιχνιδιών στα οποία έχει συμμετάσχει (*Αριθμός παιχνιδιών*).
  - Τον αριθμό των παιχνιδιών στα οποία έχει νικήσει (*Νίκες*).
  - Τον αριθμό των παιχνιδιών στα οποία έχει χάσει (*Ήττες*).
  - Το ποσοστό των παιχνιδιών επί του συνόλου των οποίων έχει νικήσει (*Το Ποσοστό Νικών σου*).
  - Το επίπεδο στο οποίο βρίσκεται (*Το Επίπεδό σου*).
  - Το συνολικό αριθμό των πόντων που έχει συγκεντρώσει (*Οι Πόντοι σου*).
- Σημειώνεται ότι πόντοι κερδίζονται όταν ο παίκτης νικήσει σε ένα παιχνίδι τον αντίπαλό του και υπολογίζονται σύμφωνα με τον παρακάτω τύπο:

$$\text{Κερδισμένοι\_πόντοι} = \text{Επίπεδο\_Αντιπάλου} * 10$$

- Τους πόντους που υπολείπονται ώστε ο παίκτης να ανέβει επίπεδο μέσω χρήσης γραφικού (*Πόντοι για το επόμενο επίπεδο*). Υπενθυμίζεται ότι **1 Επίπεδο = 100 Πόντους**, δηλαδή 100 – 199 πόντοι = Επίπεδο 1, 200 – 299 πόντοι = Επίπεδο 2 και κ.λπ.



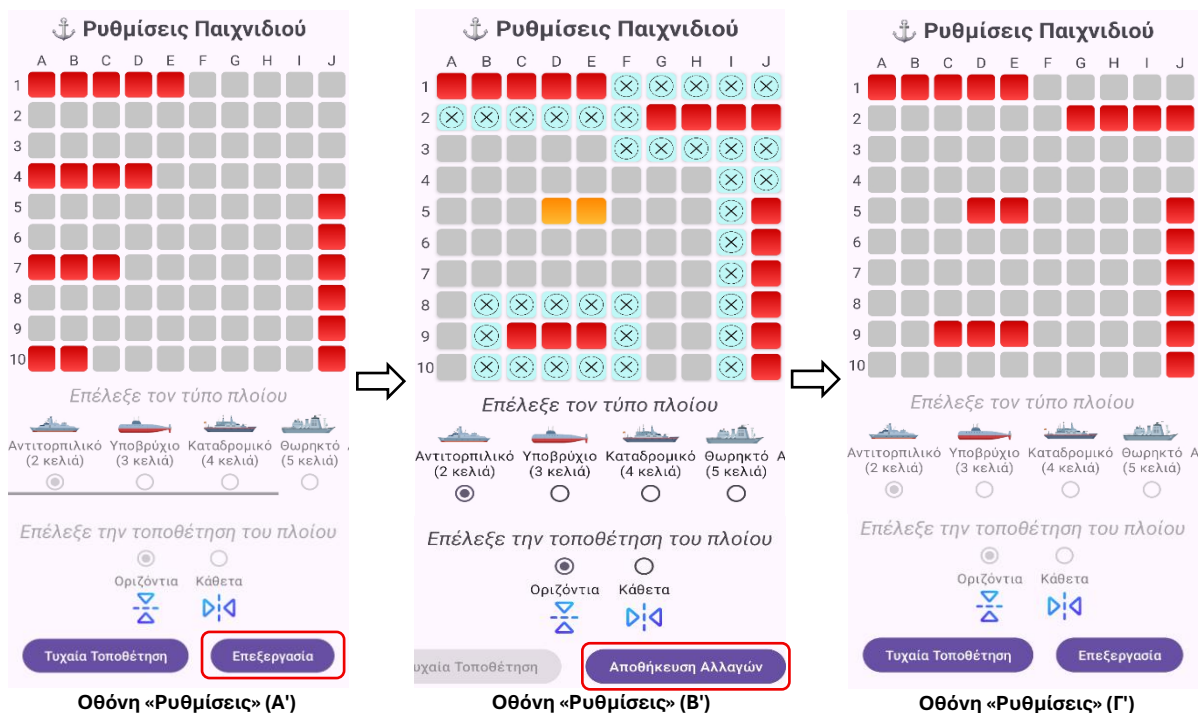
Οθόνη «Στατιστικά» (Δ')

## 2.6 Οθόνη «Ρυθμίσεις»

Από την οθόνη αυτή ο χρήστης μπορεί να πραγματοποιήσει όλες τις ρυθμίσεις που σχετίζονται με το παιχνίδι και την εφαρμογή. Πιο συγκεκριμένα μπορεί να προχωρήσει στις εξής ρυθμίσεις:

### A) Ρυθμίσεις Παιχνιδιού.

Από εδώ ο χρήστης μπορεί να αλλάξει τη θέση όπου βρίσκονται τα πλοία του στον σχετικό πίνακα. Για να επανατοποθετήσει τα πλοία έχει δύο επιλογές είτε να τοποθετηθούν με τυχαίο τρόπο πατώντας το κουμπί «Τυχαία Τοποθέτηση» είτε να επιλέξει το κουμπί «Επεξεργασία» ώστε να προχωρήσει σε τοποθέτηση των πλοίων στις θέσεις που εκείνος επιθυμεί. Όταν πατηθεί το κουμπί «Επεξεργασία» ενεργοποιείται η επιλογή για να *Επιλέξει τύπο πλοίου* προς επανατοποθέτηση και η επιλογή για να *Επιλέξει την τοποθέτηση του πλοίου* σε Οριζόντια ή Κάθετα. Επίσης το επιλεγμένο πλοίο εμφανίζεται με πορτοκαλί χρώμα στον πίνακα και ο χρήστης για να αλλάξει τη θέση του πρέπει να πατήσει ένα οποιοδήποτε κελί το οποίο θα αντιπροσωπεύει το πάνω-αριστερά μέρος του πλοίου ανάλογα με τον τρόπο τοποθέτησης που επιλέχθηκε. Αν η θέση που επέλεξε ο χρήστης δεν είναι έγκυρη, δηλαδή το πλοίο δε μπορεί να τοποθετηθεί τότε σχετικό προειδοποιητικό μήνυμα και ήχος εμφανίζονται. Επισημαίνεται ότι ένα πλοίο δε μπορεί να εφάπτεται με κάποιο άλλο όπως φαίνεται και από το γραφικό που δημιουργείται κατά τη διαδικασία επανατοποθέτησης. Τέλος, επιλέγει «Αποθήκευση Αλλαγών» για αποθήκευση.





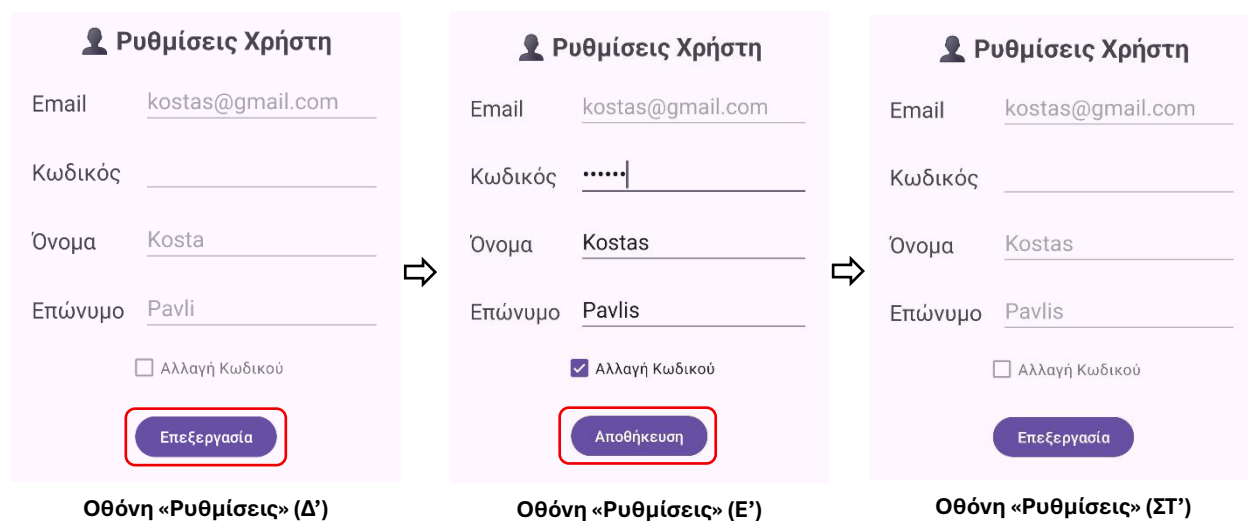
## Β) Ρυθμίσεις Χρήστη

Από αυτή την ενότητα της οθόνης «Ρυθμίσεις» ο χρήστης μπορεί να ενημερώσει τα στοιχεία του και να πραγματοποιήσει αλλαγές. Αναλυτικότερα μπορεί να προβεί σε επεξεργασία αποκλειστικά των παρακάτω στοιχείων :

- Κωδικός
- Όνομα
- Επώνυμο

Το email που δηλώθηκε κατά την εγγραφή δε μπορεί να τροποποιηθεί.

Η ενημέρωση των παραπάνω στοιχείων γίνεται αφού ο χρήστης πατήσει το κουμπί «Επεξεργασία», τότε τα πεδία θα ενεργοποιηθούν και θα μπορούν να πραγματοποιηθούν αλλαγές. Η αλλαγή του Κωδικού απαιτεί να τσεκαριστεί η επιλογή «Αλλαγή Κωδικού» για λόγους ασφαλείας. Όταν ο χρήστης ολοκληρώσει τη διαδικασία ενημέρωσης των στοιχείων του πρέπει να πατήσει υποχρεωτικά το κουμπί «Αποθήκευση» ώστε οι νέες τιμές που εισήγαγε να αποθηκευτούν.



### Γ) Ρυθμίσεις Συστήματος

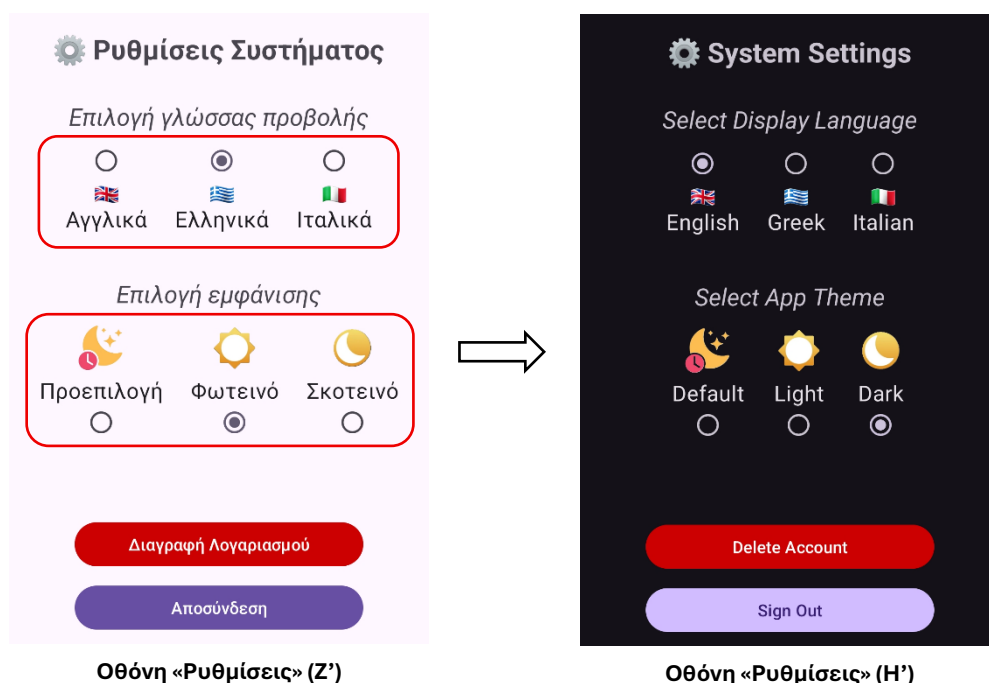
Η ενότητα αυτή επιτρέπει στο χρήστη να αλλάξει την εμφάνιση της εφαρμογής καθώς επίσης να αποσυνδεθεί ή να διαγράψει οριστικά τον λογαριασμό του. Με τη ρύθμιση *Επιλογή γλώσσας προβολής* μπορεί να αλλάξει τη γλώσσα εμφάνισης του συστήματος επιλέγοντας μια από τις ακόλουθες:

- Αγγλικά
- Ελληνικά
- Ιταλικά

Από τη ρύθμιση *Επιλογή εμφάνισης* δίνεται η δυνατότητα να επιλέξει κάτι ανάμεσα στα ακόλουθα:

- Προεπιλογή
- Φωτεινό (θέμα)
- Σκοτεινό (θέμα)

Επιπρόσθετα, μπορεί να προβεί σε αποσύνδεση από την εφαρμογή πατώντας το κουμπί «Αποσύνδεση». Υπενθυμίζεται ότι η αποσύνδεση είναι απαραίτητη αν κάποιος θέλει να συνδεθεί με τον ίδιο λογαριασμό από διαφορετική συσκευή. Τέλος, πατώντας το κουμπί «Διαγραφή Λογαριασμού» ο χρήστης μπορεί να προβεί στην οριστική διαγραφή του λογαριασμού αφού πρώτα απαντήσει θετικά στο σχετικό προειδοποιητικό μήνυμα που θα εμφανιστεί στην οθόνη.



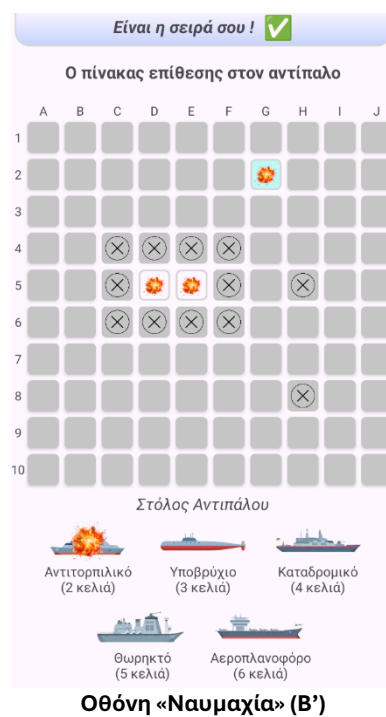
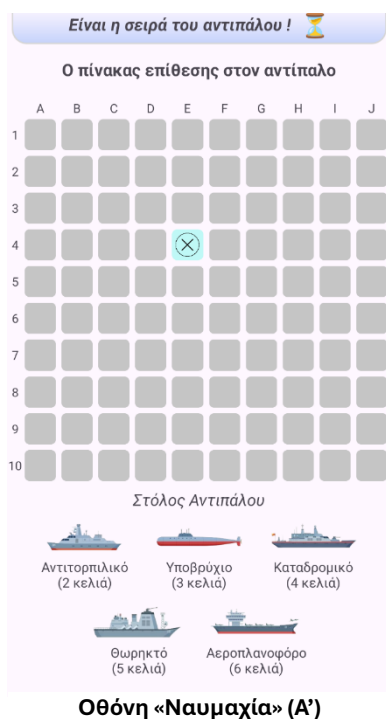


## 2.7 Οθόνη «Ναυμαχία» - Κεντρικό Παιχνίδι

Η οθόνη αυτή είναι το σημείο όπου εξελίσσεται η ναυμαχία με τον εκάστοτε αντίπαλο. Ο παίκτης οδηγείται σε αυτή τη σελίδα είτε όταν έχει απαντήσει θετικά σε αίτημα ναυμαχίας που έχει λάβει είτε όταν το αίτημα που έχει κάνει σε έναν αντίπαλο έχει λάβει θετική απάντηση. Στο πάνω μέρος της οθόνης υπάρχει η Μπάρα Κατάστασης σε αυτή προβάλλονται μηνύματα που είναι σημαντικά για τον χρήστη. Τα μηνύματα αυτά είναι:

- «Είναι η σειρά σου»: Ο παίκτης μπορεί να προσπαθήσει να χτυπήσει πλοίο του αντιπάλου.
- «Είναι η σειρά του αντιπάλου»: Ο παίκτης πρέπει να περιμένει τον αντίπαλο να εκτελέσει την κίνησή του.
- «Ο αντίπαλος απαντά...»: Ο αντίπαλος έχει εκτελέσει την επίθεσή του και ο παίκτης λαμβάνει το αποτέλεσμα – αμέσως μετά θα έχει το δικαίωμα να παίξει.

Στην οθόνη αυτή, επίσης, ο χρήστης παρατηρεί τον *Πίνακα επίθεσης στον αντίπαλο* όπου από εκεί πραγματοποιεί τις επιθέσεις του επιλέγοντας κάποιο από τα διαθέσιμα κουτάκια όταν έρθει η σειρά του. Με βάση το αποτέλεσμα της κάθε επίθεσης και αν αυτή ήταν επιτυχής ή όχι τότε το κάθε κουτάκι λαμβάνει την αντίστοιχη μορφή και ακούγεται σχετικός ήχος. Όταν ο παίκτης βυθίσει κάποιο πλοίο του αντιπάλου τότε εμφανίζεται ένα γραφικό καθώς επίσης αναπαράγεται και αντίστοιχος ήχος. Το κουτάκι με το γαλάζιο φόντο υποδηλώνει τη τελευταία κίνηση/επίθεση του παίκτη.

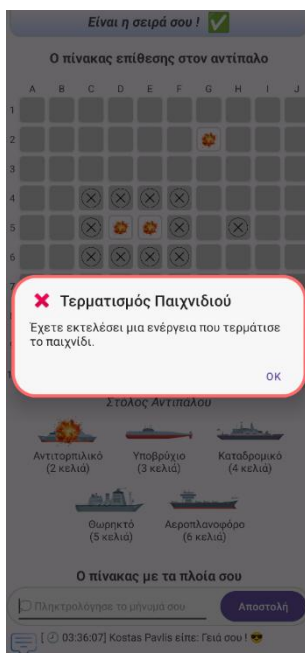


Στη συνέχεια ο παίκτης παρατηρεί την ενότητα ο *Πίνακας με τα πλοία σου* στον οποίο απεικονίζονται με κόκκινο πλαίσιο οι θέσεις των πλοίων του καθώς και οι επιθέσεις που λαμβάνει από τον αντίπαλο. Ανάλογα με το αν η επίθεση που έλαβε από τον αντίπαλο ήταν επιτυχής ή μη επιτυχής εμφανίζεται αντίστοιχο γραφικό και αναπαράγεται ήχος. Με γαλάζιο φόντο υποδηλώνεται η τελευταία επίθεση που έλαβε από τον αντίπαλο. Επιπρόσθετα μπορεί να δει τα πλοία του που δε έχουν βυθιστεί από το σχετικό γραφικό. Όταν ο αντίπαλος βυθίζει ένα πλοίο τότε και πάλι ακούγεται χαρακτηριστικός ήχος. Στο κάτω μέρος της οθόνης υπάρχει η περιοχή του Chat από την οποία ο παίκτης μπορεί να στείλει σύντομα μηνύματα στον αντίπαλό του πληκτρολογώντας το μήνυμα στο σχετικό πεδίο και πατώντας το κουμπί «Αποστολή». Από το ίδιο ακριβώς σημείο μπορεί να βλέπει τα μηνύματα που λαμβάνει από τον αντίπαλο καθώς και την ώρα λήψης τους.

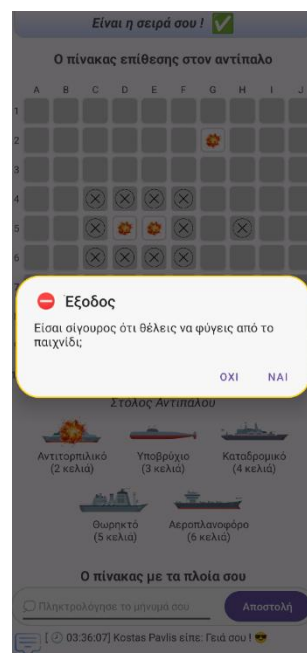


Οθόνη «Ναυμαχία» (Γ')

Αξίζει να σημειωθεί ότι κατά τη διάρκεια που το παιχνίδι είναι σε εξέλιξη η οθόνη δεν σβήνει από μόνη της. Αν ο χρήστης βάλει από μόνος του τη συσκευή σε αναστολή τότε το παιχνίδι τερματίζεται αυτόματα και ο αντίπαλος ενημερώνεται αντιστοίχως. Αν ο παίκτης κατά τη διάρκεια εξέλιξης του παιχνιδιού θέλει να αποχωρήσει πρέπει να πατήσει το κουμπί «πίσω» και να επιβεβαιώσει το μήνυμα αποχώρησης, τότε ο αντίπαλος θα ενημερωθεί σχετικά.

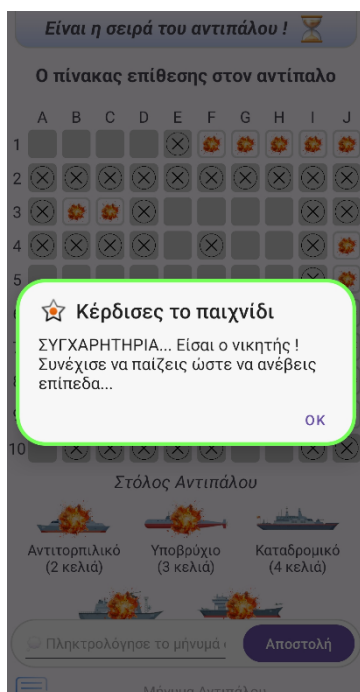


Οθόνη «Ναυμαχία» (Δ') – Μετά το κλείσιμο της συσκευής και επιστροφή του χρήστη στην εφαρμογή

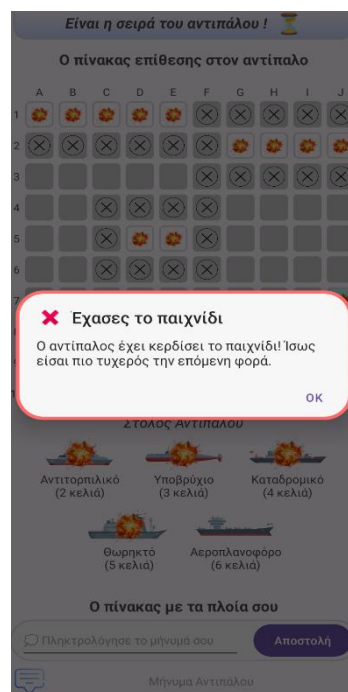


Οθόνη «Ναυμαχία» (Ε')

Ο παίκτης κερδίζει το παιχνίδι εφόσον καταφέρει πρώτος να βηθίσει όλα τα πλοία του αντιπάλου. Τότε σε κάθε περίπτωση, νίκη ή ήττα, εμφανίζεται σχετικό μήνυμα και ακούγεται αντίστοιχος ήχος.



Οθόνη «Ναυμαχία» (ΣΤ')



Οθόνη «Ναυμαχία» (Ζ')

## 2.8 Προσαρμογή για Ταμπλέτα

Η εφαρμογή έχει παραμετροποιηθεί ώστε να προσαρμόζεται κατάλληλα και να προσφέρει μία άνετη εμπειρία θέασης και παιχνιδιού στις μεγαλύτερες οθόνες.



Οθόνη Ταμπλέτας (Α')



Οθόνη Ταμπλέτας (Β')

### 3. Υλοποίηση Εφαρμογής

Το κεφάλαιο αυτό προσφέρει μια αναλυτική περιγραφή όλων των βασικών τμημάτων κώδικα που συντελούν στη λειτουργία αυτής της Android εφαρμογής. Πιο συγκεκριμένα αναλύει την δημιουργία της γραφικής διεπαφής με χρήση της γλώσσας σήμανσης XML, την υλοποίηση της επιχειρησιακής λογικής με τη γλώσσα προγραμματισμού Java καθώς και τον τρόπο χρήσης των μοντέλων που χρησιμοποιήθηκαν.

#### 3.1 Γραφική Διεπαφή

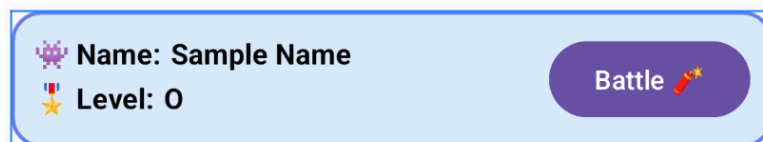
Για τη δημιουργία τους μεγάλου πλήθους γραφικών διεπαφών επιλέχθηκε η γλώσσα σήμανσης XML η οποία πέρα από την ευελιξία της στη δημιουργία προσαρμοστικών γραφικών αποτελεί και προεπιλογή της πλατφόρμας ανάπτυξης εφαρμογών για Android (Android Studio). Συγκεκριμένα υλοποιήθηκαν τόσα αρχεία XML όσες και οι οθόνες (Activities & Fragments) που υπάρχουν για προβολή περιεχομένου στον χρήστη κατά την πλοήγησή του στην εφαρμογή. Μια αντιστοίχιση των βασικών αρχείων XML με τις οθόνες που βλέπει ο χρήστης είναι η ακόλουθη:

- activity\_start.xml → Οθόνη Σύνδεσης
- activity\_sign\_up.xml → Οθόνη Εγγραφής
- activity\_home.xml: Φιλοξενεί τις ακόλουθες σελίδες για προβολή
  - fragment\_home.xml → Οθόνη «Αρχική»
  - fragment\_game.xml → Οθόνη «Παιχνίδι»
  - fragment\_statistics.xml → Οθόνη «Στατιστικά»
  - fragment\_account.xml → Οθόνη «Ρυθμίσεις»
- activity\_battleship.xml → Οθόνη «Ναυμαχία» - Κεντρικό Παιχνίδι

[Τα παραπάνω αρχεία βρίσκονται στον φάκελο *res/layout* του project]

Ωστόσο χρησιμοποιήθηκαν και πολλά αρχεία τύπου XML ώστε να ορίσουν τον τρόπο παρουσίασης μικρότερων γραφικών που εμφανίζονται στον χρήστη ως μέρος μιας οθόνης. Για παράδειγμα, ενδεικτικά, θα μπορούσαν να αναφερθούν τα ακόλουθα:

- opponent\_card.xml (Φάκελος layout): Το συγκεκριμένο αρχείο ορίζει τον τρόπο παρουσίασης των διαθέσιμων παικτών για μάχη στην Οθόνη «Παιχνίδι».

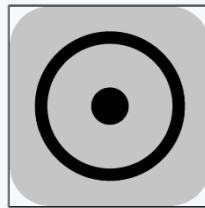


opponent\_card.xml

- `game_make_attack_successful.xml` (Φάκελος drawable): Το αρχείο αυτό προσδιορίζει τον τρόπο εμφάνισης του πλαισίου που βρίσκεται στο πλέγμα της Οθόνης «Ναυμαχία» όταν ο παίκτης πετύχει πλοίο του αντιπάλου.

`game_make_attack_successful.xml`

- `game_receive_attack.xml` (Φάκελος drawable): Το συγκεκριμένο αρχείο αποδίδει τον τρόπο εμφάνισης του πλαισίου που βρίσκεται στο πλέγμα της Οθόνης «Ναυμαχία» όταν ο αντίπαλος έχει χτυπήσει μια περιοχή στο πλέγμα του παίκτη όπου δεν υπάρχει πλοίο – άστοχη βολή.

`game_receive_attack.xml`

- `won_game_alert_dialog_background.xml` (Φάκελος drawable): Το αρχείο χρησιμοποιείται για να μορφοποιήσει το υπόβαθρο (background) του Alert Dialog που εμφανίζεται στην Οθόνη «Ναυμαχία» όταν ο παίκτης έχει κερδίσει τον αντίπαλό του. Συγκεκριμένα προσδίδει στο Alert Dialog πράσινο περίγραμμα και πιο έντονες κυρτές γωνίες για καλύτερο αισθητικό αποτέλεσμα.

`won_game_alert_dialog_background.xml`

Όπως γίνεται αντιληπτό υπάρχουν πολλά περισσότερα παρόμοια αρχεία xml μέσα στο project της εφαρμογής τα οποία αντιστοίχως μορφοποιούν συγκεκριμένα τμήματά της προσδίδοντας μια όμορφη εμφάνιση για τον τελικό χρήστη.

Επιπρόσθετα, για να γίνει εφικτή η εμφάνιση της εφαρμογής σε παραπάνω από μια γλώσσες έγινε εκτεταμένη χρήση των String Resources. Τα String Resources επιτρέπουν την αποθήκευση της μετάφρασης των λέξεων / προτάσεων που χρησιμοποιούνται σε ολόκληρη την εφαρμογή σε αρχεία XML. Συγκεκριμένα υπάρχουν τα εξής αρχεία xml:

- **strings.xml** - Αγγλικά, η προεπιλογή σε επίπεδο αρχείων
- **strings.xml (el)** – Ελληνικά
- **strings.xml (it)** – Ιταλικά

[Τα αρχεία βρίσκονται στον φάκελο *res/values/strings/filename.xml*]

Κάθε ένα από αυτά τα αρχεία χρησιμοποιείται ώστε να εμφανιστεί όλο το κείμενο που υπάρχει στην εφαρμογή σε μια από τις παραπάνω γλώσσες όταν ο χρήστης επιλέξει να αλλάξει τη γλώσσα εμφάνισης από τις ρυθμίσεις της εφαρμογής.

Τέλος, για να επιτευχθεί η καλύτερη προσαρμογή της εφαρμογής σε συσκευές με μεγαλύτερη οθόνη (π.χ. Ταμπλέτες) δημιουργήθηκαν τα ακόλουθα επιπλέον αρχεία:

- **dimens.xml** – Πλάτος οθόνης < 600dp (π.χ. κινητά)
- **dimens.xml (sw600dp)** – Πλάτος οθόνης > 600dp (π.χ. ταμπλέτες)

[Τα αρχεία βρίσκονται στον φάκελο *res/values/dimens/filename.xml*]

Τα αρχεία αυτά περιέχουν δηλωμένες τις διαστάσεις των περισσότερων γραφικών στοιχείων που υπάρχουν στην εφαρμογή. Στο κάθε αρχείο είναι αποθηκευμένες οι αντίστοιχες διαστάσεις για το συγκεκριμένο μέγεθος οθόνης / είδος συσκευής. Η εφαρμογή ελέγχει με αυτοματοποιημένο τρόπο τις διαστάσεις της οθόνης στην οποία λειτουργεί και συγκεκριμένα αν το πλάτος της οθόνης είναι μικρότερο από 600dp λαμβάνει τις διαστάσεις που υπάρχουν στο αρχείο *dimens.xml* για να διαμορφώσει το UI. Αν το πλάτος της οθόνης είναι μεγαλύτερο από 600dp τότε χρησιμοποιείται το αρχείο *dimens.xml (sw600dp)* παρέχοντας με αυτό τον τρόπο μια προσαρμοσμένη γραφική διεπαφή για κάθε είδος συσκευής.

### 3.2 Επιχειρησιακή Λογική

Η επιχειρησιακή λογική βρίσκεται στα αρχεία .java τα οποία βρίσκονται στον φάκελο Business\_Logic. Υπάρχει ένα αρχείο Java για κάθε οθόνη του χρήστη το οποίο εκτελεί όλες τις απαραίτητες λειτουργίες της οθόνης αυτής. Αναλυτικότερα τα αρχεία Java της εφαρμογής είναι τα εξής.

### 3.2.1 StartActivity.java

Το συγκεκριμένο αρχείο αντιστοιχεί στην οθόνη καλωσορίσματος – σύνδεσης του χρήστη (κλάση StartActivity) και αποτελείται από τις ακόλουθες μεθόδους:

- **void onCreate(...)** Η συγκεκριμένη μέθοδος καλείται πριν το Activity γίνει ορατό στον χρήστη με σκοπό να πραγματοποιηθούν διάφορες αρχικοποιήσεις. Το πιο σημαντικό στοιχείο της συγκεκριμένης μεθόδου είναι ότι ελέγχει αν ο χρήστης είναι ήδη συνδεδεμένος στην εφαρμογή πραγματοποιώντας έλεγχο στο αντικείμενο FirebaseUser. Αν είναι συνδεδεμένος προσπερνά την οθόνη σύνδεσης και φορτώνει άμεσα τα στοιχεία του χρήστη από τη βάση δεδομένων οδηγώντας τον στο Home Activity το οποίο αναλύεται παρακάτω. Σε διαφορετική περίπτωση αν ο χρήστης δεν είναι συνδεδεμένος εμφανίζεται κανονικά η οθόνη καλωσορίσματος – σύνδεσης.
- **void sign\_in(...)** Η μέθοδος αυτή καλείται όταν ο χρήστης πατήσει το κουμπί «Σύνδεση» με σκοπό να συνδεθεί. Ο κώδικας ελέγχει αν ο χρήστης έχει συμπληρώσει τα πεδία Email και Κωδικός και αν ναι τότε ελέγχει αν υπάρχει άτομο με τα εν λόγω στοιχεία στη βάση. Αν επιτευχθεί ταυτοποίηση ελέγχεται στη συνέχεια αν το Android\_Id της συσκευής που επιχειρεί σύνδεση ο χρήστης ταυτίζεται με αυτό που υπάρχει αποθηκευμένο στη βάση με στόχο να διασφαλιστεί ότι ο χρήστης συνδέεται από μια μόνο συσκευή. Ο χρήστης περνάει επιτυχώς από τον έλεγχο του Android\_Id μόνο αν στη βάση δεν υπάρχει Android\_Id καταχωρημένο για εκείνον ή εάν το Android\_Id της βάσης ταυτίζεται με αυτό της συσκευής που επιχειρεί σύνδεση. Εφόσον περάσει και αυτό τον έλεγχο η σύνδεση εγκρίνεται, τα στοιχεία του φορτώνονται από τη βάση και κατευθύνεται στο Home Activity ώστε να συνεχίσει την πλοήγησή του στην εφαρμογή. Σε διαφορετική περίπτωση που δεν περάσει κάποιον από τους παραπάνω ελέγχους η σύνδεση απορρίπτεται και προειδοποιητικά μηνύματα εμφανίζονται.
- **void sign\_up(...)** Η μέθοδος αυτή καλείται όταν ο χρήστης πατήσει το κουμπί «Εγγραφή» και αυτό που κάνει είναι να φορτώσει και να τον ανακατευθύνει στο Sign Up Activity στο οποίο εμφανίζεται η φόρμα από την οποία ο χρήστης μπορεί να προχωρήσει σε εγγραφή.

### 3.2.2 SignUpActivity.java

Το αρχείο αυτό αντιστοιχεί στην οθόνη εγγραφής του χρήστη στην εφαρμογή (κλάση SignUpActivity) και αποτελείται από τις ακόλουθες μεθόδους:



- **void onCreate(...)** Η μέθοδος αυτή είναι υπεύθυνη να τρέξει πριν την εμφάνιση του Activity στον χρήστη με σκοπό να υλοποιήσει ορισμένες απαραίτητες αρχικοποιήσεις. Όπως για παράδειγμα όταν πατηθεί το κουμπί «πίσω» να καταστρέφεται το συγκεκριμένο Activity μέσω κλήσης της συνάρτησης `finish()`.
- **void submit\_sign\_up(...)** Η μέθοδος αυτή καλείται όταν ο χρήστης πατήσει το κουμπί «Υποβολή» με σκοπό να υποβάλλει τα στοιχεία που συμπλήρωσε για εγγραφή στο παιχνίδι. Αφού ελεγχθεί ότι τα σχετικά πεδία δεν είναι κενά τότε προσπαθεί να εγγράψει το χρήστη αρχικά στο Authentication Service της Firebase. Αν η εγγραφή ήταν επιτυχής τότε καλείται η μέθοδος `register_user_to_dp()` που αναλύεται παρακάτω.
- **void register\_user\_to\_db(...)** Η μέθοδος αυτή αναλαμβάνει να δημιουργήσει για τον νέο χρήστη τις εγγραφές που απαιτούνται στους κόμβους Users και Games της βάσης. Αν η διαδικασία ολοκληρωθεί επιτυχώς τότε ο νέος χρήστης συνδέεται αυτόματα και ανακατευθύνεται στο Home Activity ώστε να συνεχίσει την πλοήγησή του στην εφαρμογή. Αν η διαδικασία δεν ολοκληρωθεί επιτυχώς πέρα από τα σχετικά μηνύματα που εμφανίζονται γίνεται προσπάθεια όσες εγγραφές έγιναν στη βάση μέχρι εκείνη τη στιγμή στα πλαίσια της εγγραφής του νέου χρήστη να αναστραφούν και να διαγραφούν.

### 3.2.3 HomeActivity.java

Το αρχείο αυτό (κλάση HomeActivity) δεν περιέχει κάποια ιδιαίτερη επιχειρησιακή λογική καθώς αποτελεί το «κιβώτιο» μέσα στο οποίο φορτώνονται τα Fragments για την πλοήγηση του χρήστη μέσα στην εφαρμογή μέσω του Navigation Bar. Συγκεκριμένα υπάρχει μόνο η ακόλουθη μέθοδος:

- **void onCreate(...)** Στη μέθοδο αυτή γίνονται τυπικές αρχικοποιήσεις και ορίζεται το BottomNavigationView αντικείμενο το οποίο θα διαχειρίζεται με αυτοματοποιημένο τρόπο τον χειρισμό της εναλλαγής των Fragments «Αρχική», «Παχνίδι», «Στατιστικά» και «Ρυθμίσεις» κατά την πλοήγηση του συνδεδεμένου χρήστη στο κεντρικό τμήμα της εφαρμογής.

### 3.2.4 HomeFragment.java

Το αρχείο αυτό (κλάση HomeFragment) όπως δηλώνει και το όνομα του περιέχει τον κώδικα Java ενός Fragment, δηλαδή ενός στοιχείου το οποίο φορτώνεται προς εμφάνιση στον χρήστη από το Navigation Bar στην επιλογή

«Αρχική». Το αρχείο δεν περιέχει καθόλου πηγαίο κώδικα καθώς το Fragment χρησιμοποιείται μόνο για απλή εμφάνιση πληροφοριών στον χρήστη, οι οποίες έχουν δηλωθεί στο αντίστοιχο αρχείο XML. Οι μόνες τυπικές μέθοδοι που περιλαμβάνονται είναι οι ακόλουθες:

- **void onCreate(...)** Η μέθοδος αυτή δεν περιέχει καμία επιχειρησιακή λογική και χρησιμοποιείται για να αρχικοποιήσει μεταβλητές / δεδομένα του Fragment.
- **void onCreateView(...)** Η μέθοδος αυτή δεν περιέχει επιχειρησιακή λογική και χρησιμοποιείται για να αρχικοποιήσει στοιχεία του UI προς εμφάνιση.

### 3.2.5 GameFragment.java

Το αρχείο αυτό περιέχει τον κώδικα Java για το Fragment που αντιστοιχεί στην επιλογή «Παιχνίδι» του Navigation Bar (κλάση GameFragment) και δίνει τη δυνατότητα προβολής των διαθέσιμων παικτών και την πρόκληση αυτών σε ναυμαχία. Συγκεκριμένα περιλαμβάνει τις ακόλουθες μεθόδους:

- **void onCreate(...)** Η μέθοδος αυτή δεν περιέχει καμία επιχειρησιακή λογική και χρησιμοποιείται για να αρχικοποιήσει μεταβλητές / δεδομένα του Fragment.
- **void onCreateView(...)** Η μέθοδος αυτή αρχικοποιεί διάφορα UI στοιχεία στον κώδικα Java του Fragment.
- **void onResume(...)** Η μέθοδος αυτή εκτελείται κάθε φορά που το Fragment εμφανίζεται στην οθόνη (είτε αρχική εκκίνηση είτε αφύπνιση της συσκευής) μεριμνώντας ώστε να κάνει τον χρήστη ορατό στους άλλους παίκτες κάνοντας ενημέρωση των αντίστοιχων πεδίων στη βάση. Επίσης ενεργοποιεί του κατάλληλους listeners ώστε σε περίπτωση που υπάρξει αίτημα μάχης από άλλον παίκτη να δημιουργηθεί σχετικό Alert Dialog που θα ενημερώσει σχετικά τον χρήστη. Τέλος, ανανεώνει τη λίστα των διαθέσιμων αντιπάλων καλώντας τη μέθοδο *find\_opponents()* που θα αναλυθεί στη συνέχεια.
- **void find\_opponents(...)** Η μέθοδος αυτή αναζητά στη βάση τους διαθέσιμους παίκτες εκείνη τη χρονική στιγμή ανακτώντας τα μοναδικά id τους. Αφού ανακτήσει τα id των διαθέσιμων παικτών προβαίνει σε εύρεση των επιμέρους στοιχείων των παικτών (Όνομα, Επώνυμο, Επίπεδο) ώστε να ελέγξει και να εμφανίσει μόνο τους παίκτες που βρίσκονται μέχρι ένα επίπεδο πάνω από το επίπεδο του παίκτη που έκανε την αναζήτηση. Όπως γίνεται αντιληπτό η συγκεκριμένη

διαδικασία ανάκτησης των επιμέρους στοιχείων κάθε παίκτη είναι σύνθετη και το πιο σημαντικό ασύγχρονη για κάθε αίτημα που γίνεται στη βάση. Αυτό δημιούργησε την ανάγκη εύρεσης τρόπου ώστε οι διαθέσιμοι παίκτες να εμφανιστούν στον χρήστη μόνο όταν έχουν ολοκληρωθεί όλες οι παραπάνω ασύγχρονες λειτουργίες, για αυτό έγινε χρήση του **CountDownLatch** της Java. Με αυτόν τον τρόπο κατέστη εφικτό ένα Thread που δημιουργήθηκε μετά την πραγματοποίηση όλων των αιτημάτων στη βάση να αναμένει την ολοκλήρωσή τους πριν επιτρέψει την προβολή των διαθέσιμων αντιπάλων στον παίκτη. Συγκεκριμένα κάθε ασύγχρονη λειτουργία ανάγνωσης από τη βάση όταν ολοκληρώνεται μειώνει έναν κοινό μετρητή και όταν το Thread διαπιστώσει ότι ο μετρητής έχει μηδενιστεί, άρα έχουν ολοκληρωθεί όλες οι ασύγχρονες συναλλαγές με τη βάση, εκτελεί τη μέθοδο *fill\_the\_UI()* η οποία αναλύεται παρακάτω με σκοπό να δημιουργήσει τη λίστα των διαθέσιμων αντιπάλων.

- **void fill\_the\_UI(...)** Η μέθοδος αυτή διαβάζει τα στοιχεία των διαθέσιμων παικτών που έχουν συγκεντρωθεί σε σχετικές λίστες και χρησιμοποιώντας τον Adapter του RecyclerView, το οποίο αναλύεται στο *Κεφάλαιο 5 – Android UI* οδηγεί τα δεδομένα προς εμφάνιση στην οθόνη.
- **void request\_battler(...)** Η μέθοδος αυτή εκτελείται όταν ο χρήστης πατήσει το κουμπί «Μάχη» σε κάποιον από τους διαθέσιμους παίκτες για ναυμαχία. Συγκεκριμένα, ενημερώνει κατάλληλα τη βάση ώστε ο παίκτης να φαίνεται ως μη διαθέσιμος και ο αντίπαλος να ενημερωθεί για το σχετικό αίτημα. Αν το αίτημα γίνει αποδεκτό από τον αντίπαλο τότε ο χρήστης οδηγείται στην οθόνη που θα εξελιχθεί το παιχνίδι (*BattleshipActivity.java*). Αν το αίτημα δε γίνει αποδεκτό από τον αντίπαλο ή ο ίδιος ο παίκτης το ακυρώσει τότε ο παίκτης γίνεται ξανά διαθέσιμος για παιχνίδι με την κατάλληλη ενημέρωση της βάσης και το σχετικό Alert Dialog απομακρύνεται.
- **String current\_date\_time(...)** Η μέθοδος αυτή ανακτά την τρέχουσα ημερομηνία και ώρα του συστήματος και την επιστρέφει στη μέθοδο που την κάλεσε.
- **void play\_request\_sound(...)** Η μέθοδος αυτή χρησιμοποιείται για να αναπαραχθεί ένας ήχος ειδοποίησης όταν ο παίκτης λάβει αίτημα μάχης από κάποιον αντίπαλο.
- **void onPause(...)** Η μέθοδος αυτή εκτελείται κάθε φορά που είτε η οθόνη κλείνει είτε ο χρήστης φεύγει από την οθόνη «Παιχνίδι» με τη

χρήση του Navigation Bar. Βασική λειτουργία της είναι να κάνει τον χρήστη να φαίνεται μη διαθέσιμος στη βάση δεδομένων καθώς και να απενεργοποιήσει οποιουσδήποτε ενεργούς listeners υπήρχαν για τον εντοπισμό αιτημάτων μάχης από αντιπάλους.

### 3.2.6 StatisticsFragment.java

Το αρχείο αυτό περιέχει τον κώδικα για το Fragment που αντιστοιχεί στην επιλογή «Στατιστικά» από το Navigation Bar (κλάση StatisticsFragment). Πιο αναλυτικά περιλαμβάνει τις ακόλουθες μεθόδους:

- **void onCreate(...)** Η μέθοδος αυτή δεν περιέχει καμία επιχειρησιακή λογική και χρησιμοποιείται για να αρχικοποιήσει μεταβλητές / δεδομένα του Fragment.
- **void onCreateView(...)** Η μέθοδος αυτή χρησιμοποιείται κυρίως για να αρχικοποιήσει διάφορα UI στοιχεία στον κώδικα Java. Επίσης χρησιμοποιείται για να εμφανίσει τα στατιστικά στοιχεία που αφορούν τον ίδιο τον χρήστη (Επίπεδο, Νίκες, Ήττες, Πόντους κ.λπ.). Τέλος, εκτελεί τις απαραίτητες ενέργειες ώστε να ανακτηθούν από τη βάση οι 3 πρώτοι παίκτες σε ποσοστό νικών και σε αριθμό νικών και ηττών – κατά τη διαδικασία αυτή γίνεται και πάλι χρήση του *CountDownLatch* που προαναφέρθηκε.
- **Void best\_quality\_selection(...)** Η μέθοδος αυτή καλείται όταν ο χρήστης πατήσει το κουμπί «Νίκες (%)» με σκοπό να εμφανιστούν στο σχετικό σημείο της οθόνης οι 3 πρώτοι παίκτες σε ποσοστό νικών πραγματοποιώντας σχετικές αλλαγές / ενημερώσεις στο UI.
- **void wins\_selection(...)** Η μέθοδος αυτή καλείται όταν ο χρήστης πατήσει το κουμπί «Νίκες» με σκοπό να εμφανιστούν στο σχετικό σημείο της οθόνης οι 3 πρώτοι παίκτες σε αριθμό νικών πραγματοποιώντας σχετικές αλλαγές / ενημερώσεις στο UI.
- **void losses\_selection(...)** Η μέθοδος αυτή καλείται όταν ο χρήστης πατήσει το κουμπί «Ήττες» με σκοπό να εμφανιστούν στο σχετικό σημείο της οθόνης οι 3 πρώτοι παίκτες σε αριθμό ηττών πραγματοποιώντας σχετικές αλλαγές / ενημερώσεις στο UI.

### 3.2.7 AccountFragment.java

Το αρχείο αυτό Java (κλάση AccountFragment) περιέχει την επιχειρησιακή λογική για την υλοποίηση όλων των ρυθμίσεων και παραμετροποιήσεων που

μπορεί να κάνει ο χρήστης σε επίπεδο εφαρμογής και παιχνιδιού. Συγκεκριμένα περιλαμβάνει τις ακόλουθες μεθόδους:

- **void onCreate(...)** Η μέθοδος αυτή δεν περιέχει καμία επιχειρησιακή λογική και χρησιμοποιείται για να αρχικοποιήσει μεταβλητές / δεδομένα του Fragment.
- **void onCreateView(...)** Η μέθοδος αυτή χρησιμοποιείται για να αρχικοποιήσει διάφορα UI στοιχεία στον κώδικα Java καθώς και να ορίσει τη συμπεριφορά αυτών σε περίπτωση ανάδρασης με τον χρήστη (π.χ. ποια μέθοδο θα καλέσει ένα κουμπί όταν πατηθεί από τον χρήστη), δηλαδή πραγματοποιείται ορισμός των event listeners. Τέλος, η μέθοδος είναι αρμόδια και για τη δημιουργία γραφικών στο UI τα οποία φορτώνονται δυναμικά με χαρακτηριστικό παράδειγμα το πλέγμα προβολής των θέσεων των πλοίων του χρήστη από το οποίο στη συνέχεια θα έχει τη δυνατότητα να αλλάξει τις θέσεις τους.
- **void onSaveInstanceState(...)** Η μέθοδος αυτή χρησιμοποιείται για να αποθηκεύσει προσωρινά στη μνήμη τις τιμές κάποιων μεταβλητών/ρυθμίσεων όταν το Fragment επανεκκινείται ώστε να μη χαθούν. Αυτό είναι απαραίτητο καθώς ρυθμίσεις όπως η αλλαγή του θέματος εμφάνισης και της γλώσσας εμφάνισης της εφαρμογής απαιτούν την επανεκκίνηση του Fragment ώστε να γίνουν ορατές στον χρήστη.
- **void onViewStateRestored(...)** Η μέθοδος αυτή αναλαμβάνει να κάνει ακριβώς την αντίθετη ενέργεια από την προηγούμενη μέθοδο. Δηλαδή ελέγχει αν υπάρχει αποθηκευμένη πληροφορία στη μνήμη για το Fragment που μόλις δημιουργείται και αν υπάρχει να επαναφέρει τις τιμές που είχαν αποθηκευτεί στις αντίστοιχες μεταβλητές ώστε να διασφαλιστεί η συνοχή των δεδομένων.
- **void edit\_fields(...)** Η μέθοδος αυτή εκτελείται όταν ο χρήστης πατήσει το κουμπί «Επεξεργασία» για να επεξεργαστεί τα προσωπικά του δεδομένα (π.χ. Όνομα, Επώνυμο, Κωδικός). Συγκεκριμένα ενεργοποιεί τα σχετικά παιδιά επιτρέποντας την επεξεργασία τους. Επίσης αλλάζει το κείμενο του κουμπιού σε «Αποθήκευση». Τώρα όταν ο χρήστης πατήσει «Αποθήκευση» θα απενεργοποιήσει τα πεδία και θα ενημερώσει τη βάση δεδομένων με τις νέες τιμές.
- **void relocate(...)** Η μέθοδος αυτή ενεργοποιείται όταν ο χρήστης πατήσει το κουμπί «Τυχαία Τοποθέτηση» ώστε το σύστημα να επανατοποθετήσει με τυχαίο τρόπο τα καράβια του για μάχη. Αφού

εκτελέσει τους απαραίτητους ελέγχους ώστε να είναι σίγουρο ότι τα πλοία βρίσκονται σε επιτρεπτές νέες θέσεις εμφανίζει στον χρήστη τις νέες θέσεις των πλοίων, κάνοντας χρήση του ειδικού πλέγματος που υπάρχει και τέλος αποθηκεύει τις νέες θέσεις στη βάση.

- **void edit\_ships(...)** Η μέθοδος αυτή εκτελείται όταν ο χρήστης πατήσει το κουμπί «Επεξεργασία» ώστε να τροποποιήσει χειροκίνητα τις θέσεις των πλοίων του. Συγκεκριμένα ενεργοποιεί τα γραφικά στοιχεία που είναι απαραίτητα για να δοθεί η δυνατότητα στο χρήστη να επιλέξει νέες θέσεις για τα πλοία του. Επίσης, αλλάζει το κείμενο του κουμπιού σε «Αποθήκευση Αλλαγών». Όταν ολοκληρώσει ο χρήστης την επεξεργασία της θέσης των πλοίων πατάει το κουμπί «Αποθήκευση Αλλαγών» και τότε τα γραφικά στοιχεία απενεργοποιούνται μη επιτρέποντας περαιτέρω αλλαγές στις θέσεις των πλοίων. Τέλος, αποθηκεύει τις νέες θέσεις των πλοίων στη βάση δεδομένων.
- **void ship\_button\_click(...)** Η συνάρτηση καλείται όταν ο χρήστης επιλέξει μια νέα θέση στο πλέγμα ώστε να αλλάξει τη θέση του πλοίου που έχει επιλέξει προς μετακίνηση. Στη συνέχεια καλεί την *change\_vessels\_positions()* με ορίσματα ανάλογα με το είδος του πλοίου προς μετακίνηση που έχει επιλέξει ο χρήστης ώστε να ολοκληρωθεί η σχετική διαδικασία.
- **void change\_vessels\_positions(...)** Η μέθοδος αυτή εκτελείται κάθε φορά που ο χρήστης προσπαθεί τη χειροκίνητη αλλαγή της θέσης ενός πλοίου του. Αν η επανατοποθέτηση που επιθυμεί είναι έγκυρη τότε το πλοίο εμφανίζεται στο πλέγμα στη νέα του θέση και η πληροφορία της νέας θέσης αποθηκεύεται τοπικά. Αν η επανατοποθέτηση που επιθυμεί δεν είναι έγκυρη τότε δημιουργείται σχετικό μήνυμα ενημέρωσης του χρήστη και καλείται η συνάρτηση *play\_invalid\_sound()* ώστε να αναπαραχθεί ήχος σφάλματος.
- **void switch\_buttons(...)** Η μέθοδος αυτή χρησιμοποιείται για την αλλαγή του πλαισίου γύρω από τις θέσεις που καταλαμβάνουν τα πλοία ώστε να γίνει σαφές στον χρήστη ότι κατά τη διάρκεια της χειροκίνητης αλλαγής θέσης των πλοίων ένα πλοίο δε μπορεί να εφάπτεται με ένα άλλο.
- **void ship\_radio\_selected(...)** Η μέθοδος αυτή εκτελείται κάθε φορά που ο χρήστης επιλέξει διαφορετικό πλοίο για χειροκίνητη επανατοποθέτηση από τα αντίστοιχα Radio Buttons. Όταν ένα νέο πλοίο επιλέγεται τότε το χρώμα που συμβολίζει το συγκεκριμένο πλοίο

στο πλέγμα προβολής αλλάζει ώστε να το διαφοροποιήσει από τα υπόλοιπα τοποθετημένα πλοία.

- **void play\_invalid\_sound(...)** Η μέθοδος αυτή χρησιμοποιείται αποκλειστικά για την αναπαραγωγή ήχου σφάλματος όταν ο χρήστης προσπαθήσει να τοποθετήσει χειροκίνητα ένα πλοίο σε μη έγκυρη θέση στο πλέγμα προβολής της θέσης των πλοίων.
- **void sign\_out(...)** Η μέθοδος αυτή καλείται όταν ο χρήστης πατήσει το κουμπί «Αποσύνδεση». Πιο αναλυτικά, ενημερώνει τη βάση δεδομένων τοποθετώντας κενό Android (Device) Id, ώστε ο χρήστης να μπορεί να συνδεθεί από διαφορετική συσκευή αν το επιθυμεί. Τέλος, οδηγεί τον χρήστη στην Οθόνη Καλωσορίσματος – Σύνδεσης.
- **void delete\_account(...)** Η μέθοδος αυτή εκτελείται όταν ο χρήστης πατήσει το κουμπί «Διαγραφή Λογαριασμού» και επιβεβαιώσει την ενέργεια της διαγραφής από το σχετικό Alert Dialog που θα εμφανιστεί. Τότε, διαγράφει όλες τις εγγραφές που αφορούν τον χρήστη στη βάση δεδομένων και τον κατευθύνει στην Οθόνη Καλωσορίσματος – Σύνδεσης.

### 3.2.8 BattleshipActivity.java

Το αρχείο αυτό περιέχει τον κώδικα Java που αντιστοιχεί στην Οθόνη «Παιχνίδι» (κλάση BattleshipActivity) στην οποία πραγματοποιείται η ναυμαχία με τον εκάστοτε αντίπαλο. Περιέχει τις ακόλουθες μεθόδους:

- **void onCreate(...)** Η μέθοδος αυτή καλείται κατά τη δημιουργία του Activity προκειμένου να πραγματοποιήσει τις απαραίτητες αρχικοποιήσεις μεταβλητών και UI στοιχείων τα οποία είναι απαραίτητα για την εξέλιξη του παιχνιδιού. Ακόμα ορίζει τους Event Listeners όλων των κουμπιών αλλά το πιο σημαντικό ενεργοποιεί τους Firebase Listeners οι οποίοι επιτρέπουν τη λήψη τόσο των επιθέσεων του αντιπάλου όσο και την λήψη μηνυμάτων που ενδεχομένως θα αποστέλλει μέσω του Chat.
- **void ship\_attack\_click(...)** Η μέθοδος αυτή εκτελείται κάθε φορά που ο παίκτης επιλέξει ένα κουμπί από το πλέγμα για να πραγματοποιήσει επίθεση στον αντίπαλο. Αρχικά, ελέγχεται τοπικά αν χτυπήθηκε κάποιο πλοίο του αντιπάλου στη συγκεκριμένη θέση, εμφανίζεται το αντίστοιχο γραφικό και αναπαράγεται ο σχετικός ήχος (επιτυχίας ή αποτυχίας). Στη συνέχεια ενημερώνεται η βάση δεδομένων με τη θέση που επέλεξε να χτυπήσει ο παίκτης ώστε να ενημερωθεί αντίστοιχα και

ο αντίπαλος. Τέλος, απενεργοποιεί τα κουμπιά του πλέγματος ώστε ο χρήστης να μη μπορεί να πραγματοποιήσει νέα επίθεση μέχρι να έρθει πάλι η σειρά του.

- **void buttons\_grid\_creation(...)** Η συγκεκριμένη μέθοδος καλείται κατά την αρχικοποίηση του Activity στη μέθοδο *onCreate()* ώστε να δημιουργηθούν μέσα από τον κώδικα Java τα δύο πλέγματα κουμπιών που υπάρχουν στην οθόνη. Το ένα πλέγμα χρησιμοποιείται ώστε ο παίκτης να πραγματοποιεί τις επιθέσεις του στον αντίπαλο και το άλλο, μικρότερο σε διάσταση, πλέγμα χρησιμοποιείται αποκλειστικά ώστε ο παίκτης να έχει οπτική αποτύπωση των επιθέσεων που κάνει ο αντίπαλος στα πλοία του.
- **void disable\_buttons(...)** Η μέθοδος αυτή χρησιμοποιείται για να απενεργοποιήσει, όποτε αυτό απαιτείται, τα κουμπιά που βρίσκονται περιμετρικά ενός συγκεκριμένου κουμπιού του πλέγματος που έχει προσδιοριστεί στα ορίσματα της συνάρτησης. Αυτό γίνεται ώστε να αποτραπεί η χρήση των περιμετρικών κουμπιών από τον παίκτη σε συγκεκριμένες περιπτώσεις.
- **void sound\_player(...)** Η μέθοδος αυτή είναι αρμόδια να αναπαράγει έναν συγκεκριμένο ήχο ανάλογα με το όρισμα που θα λάβει κατά την κλήση της. Οι ήχοι που αναπαράγει είναι οι εξής: άστοχη επίθεση, επιτυχής επίθεση σε πλοίο και καταστροφή ολόκληρου του πλοίου.
- **String current\_date\_time(...)** Η μέθοδος αυτή ανακτά την τρέχουσα ώρα του συστήματος και την επιστρέφει στη μέθοδο που την κάλεσε. Χρησιμοποιείται για την εμφάνιση της ώρας λήψης ενός μηνύματος στο chat από τον αντίπαλο.
- **void onResume(...)** Η μέθοδος αυτή εκτελείται όταν το Activity γίνεται ορατό στην οθόνη, κυρίως χρησιμοποιείται στην περίπτωση αφύπνισης του κινητού από κατάσταση αναμονής. Συγκεκριμένα, αν το Battleship Activity είναι ενεργό και μεταβεί η συσκευή σε κατάσταση αναμονής τότε η ναυμαχία αυτομάτως ακυρώνεται οπότε όταν ο χρήστης ανοίξει ξανά τη συσκευή η μέθοδος αυτή εμφανίζει σχετικό μήνυμα για την ακύρωση του παιχνιδιού στον παίκτη.
- **void onPause(...)** Η συγκεκριμένη μέθοδος εκτελείται όταν ο χρήστης βάλει από μόνος του τη συσκευή σε κατάσταση αναμονής, δηλαδή κλείσει την οθόνη του κινητού. Τότε η ναυμαχία ακυρώνεται, η βάση δεδομένων ενημερώνεται για τον τερματισμό του υπό εξέλιξη παιχνιδιού και κατ' επέκταση ο αντίπαλος ενημερώνεται αντίστοιχα. Τέλος, απενεργοποιούνται όλοι οι ενεργοί Firebase Listeners που είχαν



οριστεί κατά την εκκίνηση του παιχνιδιού από τη στιγμή που το παιχνίδι τερματίστηκε και δεν υπάρχει λόγος να συνεχιστεί η ακρόαση συμβάντων στη βάση δεδομένων.

### 3.2.9 SystemOperations.java

Το αρχείο αυτό περιέχει την κλάση «SystemOperations» με στατικές μεθόδους οι οποίες καλούνται κυρίως από το *AccountFragment.java* για την εκτέλεση ενεργειών που σχετίζονται με την εφαρμογή και τη λειτουργία της. Συγκεκριμένα αποτελείται από τις εξής μεθόδους:

- **boolean internetConnectionAvailability(...)** Η μέθοδος αυτή ελέγχει αν υπάρχει σύνδεση στο Internet τη στιγμή που εκτελείται. Αν υπάρχει επιστρέφει true αν δεν υπάρχει επιστρέφει false στη μέθοδο που την κάλεσε.
- **void set\_display\_language(...)** Η μέθοδος είναι υπεύθυνη με βάση το όρισμα που θα λάβει να ενημερώσει τα Shared Preferences με την αντίστοιχη γλώσσα εμφάνισης της εφαρμογής και να επανεκκινήσει το Activity που την κάλεσε.
- **Context setLocale(...)** Η συνάρτηση αυτή ορίζει μια νέα γλώσσα εμφάνισης στην εφαρμογή ανάλογα με το όρισμα που έλαβε και επιστρέφει ένα αντικείμενο τύπου Context.
- **String getLanguage(...)** Η συγκεκριμένη μέθοδος λαμβάνει την τρέχουσα γλώσσα εμφάνισης της εφαρμογής, η οποία είναι αποθηκευμένη στα Shared Preferences, και επιστρέφει σχετικό String στο οποίο περιέχεται η γλώσσα που ανακτήθηκε.
- **void set\_Appearance\_Theme(...)** Η συνάρτηση αυτή με βάση το όρισμα που θα λάβει ενημερώνει το θέμα εμφάνισης της εφαρμογής καθώς επίσης καταχωρεί τον τύπο του θέματος που επιλέχθηκε στα Shared Preferences.
- **int get\_Appearance\_Theme(...)** Η συνάρτηση αυτή ανακτά από τα Shared Preferences το επιλεγμένο θέμα εμφάνισης συστήματος που έχει αποθηκευτεί και επιστρέφει έναν ακέραιο με τιμές 0, 1 ή 2 ανάλογα με το θέμα το οποίο έχει επιλεγεί.

## 3.3 Μοντέλα

Η συγκεκριμένη εφαρμογή χρησιμοποιεί δύο μοντέλα για τη διατήρηση των απαραίτητων δεδομένων κατά τη λειτουργία της. Το ένα είναι η κλάση User και το άλλο η κλάση Opponent\_User, τα οποία είναι και τα δύο singleton καθώς μόνο ένας παίκτης

μπορεί να είναι συνδεδεμένος στην εφαρμογή και επιτρέπεται να παίζει το πολύ με έναν αντίπαλο κάθε χρονική στιγμή. Οι κλάσεις αυτές διατηρούνται στον φάκελο Models και περιλαμβάνονται στα ακόλουθα αρχεία.

### 3.3.1 User.java

Το αρχείο αυτό περιέχει τη κλάση User, η οποία αντιπροσωπεύει τον εκάστοτε συνδεδεμένο χρήστη και δηλώνει τα πεδία – attributes που περιλαμβάνονται στη κλάση και είναι τα:

- **String user\_id:** Το μοναδικό User ID που αποδίδεται από τη Firebase στο χρήστη
- **String first\_name:** Το όνομα του χρήστη
- **String last\_name:** Το επώνυμο του χρήστη
- **String email:** Το email του χρήστη
- **List<String> ships:** Οι θέσεις που είναι τοποθετημένα τα 5 πλοία του χρήστη
- **int level:** Το επίπεδο που βρίσκεται ο χρήστης
- **int points:** Οι πόντοι που έχει συγκεντρώσει ο χρήστης
- **int wins:** Ο αριθμός των παιχνιδιών που κέρδισε
- **int losses:** Ο αριθμός των παιχνιδιών που έχασε
- **String device\_id:** Το μοναδικό αναγνωριστικό συσκευής στην οποία είναι συνδεδεμένος ο χρήστης τη συγκεκριμένη χρονική στιγμή
- **double win\_rate:** Το ποσοστό των παιχνιδιών επί του συνόλου των οποίων κέρδισε

Για όλα τα πεδία που αναφέρθηκαν έχουν οριστεί Getters και Setters για να διευκολύνουν την πρόσβαση στα δεδομένα.

Επιπρόσθετα, η κλάση αυτή περιέχει και κάποιες μεθόδους οι οποίες περιλαμβάνουν κώδικα για την ενημέρωση συγκεκριμένων πεδίων της βάσης δεδομένων. Αυτές είναι οι ακόλουθες:

- **void updateUser(...)** η οποία ενημερώνει όνομα, επώνυμο, email και θέσεις πλοίων
- **void updateDeviceID(...)** η οποία ενημερώνει το αναγνωριστικό της συσκευής Android
- **void updateLevel(...)** η οποία ενημερώνει το επίπεδο
- **void updatePoints(...)** η οποία ενημερώνει του πόντους

- **void updateWins(...)** η οποία ενημερώνει τον συνολικό αριθμό των νικών
- **void updateLosses(...)** η οποία ενημερώνει τον συνολικό αριθμό των ηττών

### 3.3.2 Opponent\_User.java

Το αρχείο αυτό περιέχει τη κλάση Opponent\_User η οποία είναι μια πιο απλοποιημένη και εξατομικευμένη εκδοχή της κλάσης User και χρησιμοποιείται κατά τη διάρκεια που ο παίκτης βρίσκεται σε ναυμαχία με κάποιον αντίπαλο. Συγκεκριμένα περιλαμβάνει τα εξής πεδία:

- **String id:** Το μοναδικό User ID που αποδίδεται από τη βάση δεδομένων στον εκάστοτε αντίπαλο χρήστη
- **String first\_name:** Το όνομα του εκάστοτε αντίπαλου χρήστη
- **String last\_name:** Το επώνυμο του εκάστοτε αντίπαλου χρήστη
- **int level:** Το επίπεδο του εκάστοτε αντίπαλου χρήστη
- **int opponent\_selection:** Ένα αναγνωριστικό που δηλώνει αν ο αντίπαλος προκάλεσε πρώτος σε ναυμαχία ή όχι
- **List<String> ships:** Οι θέσεις που είναι τοποθετημένα τα 5 πλοία του εκάστοτε αντίπαλου χρήστη

Για όλα τα παραπάνω πεδία έχουν οριστεί Getters για τη διευκόλυνση της πρόσβασης στα δεδομένα. Επισημαίνεται ότι η αρχικοποίηση των πεδίων γίνεται αποκλειστικά στον constructor της κλάσης.

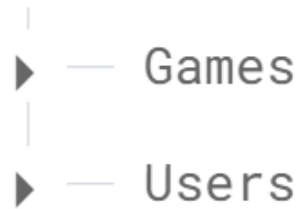
## 4. Firebase

Η Firebase είναι η βάση δεδομένων που επιλέχθηκε ώστε να γίνεται η αποθήκευση των δεδομένων της εφαρμογής καθώς κρίθηκε ότι ήταν η καλύτερη δυνατή λύση.

### 4.1 Περιγραφή & Τρόπος Ενσωμάτωσης

Η Firebase είναι μία εφαρμογή ιστού η οποία έχει ως κεντρικό χαρακτηριστικό της την αποθήκευση των δεδομένων σε μορφή JSON. Η αποθήκευση των δεδομένων σε μορφή JSON δεν απαιτεί την εκτέλεση ερωτημάτων (queries) για την ανάγνωση, εισαγωγή, επεξεργασία και διαγραφή των εγγραφών κάτι που την καθιστά μη σχεσιακή βάση δεδομένων (No SQL). Οι μη-σχεσιακές βάσεις δεδομένων έχουν πολλαπλά οφέλη έναντι των σχεσιακών βάσεων δεδομένων με σημαντικότερο πλεονέκτημα τη δυνατότητα ύπαρξης πολλαπλών εγγραφών με διαφορετική μορφή η μια από την άλλη χωρίς να χρειάζεται η ξεχωριστή δήλωση του κάθε τύπου εγγραφής (π.χ. σε πίνακα) – στις μη-σχεσιακές βάσεις κάθε τύπος εγγραφής είναι αποδεκτός οποιαδήποτε στιγμή. Η Firebase ακόμα προσφέρει πλήθος από υπηρεσίες τις οποίες μπορούν να ενσωματώσουν οι προγραμματιστές και να δημιουργήσουν υψηλής ποιότητας εφαρμογές. Στη συγκεκριμένη εφαρμογή έγινε χρήση δύο εξαιρετικά χρήσιμων υπηρεσιών του Firebase Authentication και του Real-Time Database.

- ✓ Η υπηρεσία **Firebase Authentication** επιτρέπει την ταυτοποίηση των χρηστών κατά τη σύνδεσή τους στην εφαρμογή μέσω της επαλήθευσης των στοιχείων τους (Email & Κωδικός) που έχουν δηλώσει κατά την εγγραφή. Πιο αναλυτικά, η συγκεκριμένη υπηρεσία αποθηκεύει το Email, τον κωδικό πρόσβασης καθώς και ένα μοναδικό ID που δημιουργείται αυτόματα για τον κάθε χρήστη κατά τη διαδικασία της εγγραφής. Επίσης, μια πολύ σημαντική λειτουργικότητα που παρέχει είναι η ανίχνευση της ταυτότητας του συνδεδεμένου χρήστη καθ' όλη τη διάρκεια που παραμένει συνδεδεμένος στην εφαρμογή και εκτελεί ενέργειες ανάγνωσης/εγγραφής στη βάση δεδομένων (Real-Time Database).
- ✓ Η υπηρεσία **Real-Time Database** είναι αυτή που παρέχει την ίδια τη βάση δεδομένων και τη δυνατότητα αποθήκευσης των εγγραφών που είναι απαραίτητες για τη λειτουργία της εφαρμογής / παιχνιδιού. Η δομή της βάσης δεδομένων είναι η ακόλουθη:

**Δομή Βάσης Δεδομένων (Α')**

Από τη ρίζα προκύπτουν δύο βασικοί κόμβοι. Ο κόμβος «Games» και ο κόμβος «Users». Κάθε χρήστης έχει μια εγγραφή σε κάθε έναν από αυτούς τους δύο κόμβους και συγκεκριμένα ο κάθε χρήστης είναι ένας υπό-κόμβος με αναγνωστικό υπό-κόμβου το μοναδικό ID που του έχει αποδοθεί από τη βάση κατά την εγγραφή. Ο κόμβος «Users» αποθηκεύει τα πεδία που παρουσιάστηκαν στο *Κεφάλαιο 3.3.1* και σχετίζονται με όλα τα διαθέσιμα στοιχεία που υπάρχουν για τον εκάστοτε χρήστη (Όνομα, Επώνυμο, θέσεις πλοίων κ.λπ.). Αυτό διαπιστώνεται στην παρακάτω εικόνα.

**Δομή Βάσης Δεδομένων (Β')**

Ο κόμβος «Games» είναι αυτός που περιέχει τα πεδία όπου γίνεται η ανταλλαγή της πληροφορίας κατά τη διάρκεια των ναυμαχιών. Η ανταλλαγή της πληροφορίας κατά την εξέλιξη της εκάστοτε ναυμαχίας γίνεται από την εγγραφή του χρήστη που έλαβε το αίτημα για μάχη. Διαθέτει τα πεδία που φαίνονται στην παρακάτω εικόνα.



Δομή Βάσης Δεδομένων (Γ')

Ανάλυση του εκάστοτε πεδίου:

- **is\_available:** Δηλώνει αν είναι διαθέσιμος για μάχη ο χρήστης στον οποίο αντιστοιχεί ο γονικός κόμβος.
- **player\_1:** Περιέχει το μοναδικό αναγνωριστικό του χρήστη στον οποίο αντιστοιχεί ο γονικός κόμβος (ο χρήστης που δέχτηκε το αίτημα για ναυμαχία).
- **player\_2:** Συμπληρώνεται με το μοναδικό αναγνωριστικό του αντιπάλου όταν ξεκινήσει η ναυμαχία (ο χρήστης που έστειλε το αίτημα για ναυμαχία).
- **player\_1\_response & player\_2\_response:** Συμπληρώνεται με την κωδικοποιημένη πληροφορία που περιλαμβάνει τη θέση που έκανε την εκάστοτε επίθεση ο κάθε χρήστης. Επίσης μπορεί να περιλαμβάνει και άλλη πληροφορία όπως αν ο αντίπαλος κέρδισε το παιχνίδι.
- **player\_1\_message & player\_2\_message:** Συμπληρώνεται με το μήνυμα που μπορεί να στείλει ο ένας παίκτης στον άλλον.
- **request\_response:** Συμπληρώνεται με την απάντηση (θετική ή αρνητική) που δίνει ο χρήστης στον οποίο αντιστοιχεί ο γονικός κόμβος στον χρήστη που τον προκάλεσε σε ναυμαχία και αναμένει την απάντησή του.

Η Real-Time Database επιτρέπει τη ζωντανή παρακολούθηση των αλλαγών που γίνονται σε εγγραφές της βάσης δεδομένων. Αυτό είναι πολύ χρήσιμο και

απαραίτητο για την αποτελεσματική λειτουργία της συγκεκριμένης εφαρμογής. Για αυτό πολλά πεδία από τα παραπάνω που αναφέρθηκαν όπως τα *player\_1\_response*, *player\_2\_response*, *player\_1\_message*, *player\_2\_message* παρακολουθούνται από κατάλληλους Event Listeners κατά τη διάρκεια της εκάστοτε ναυμαχίας ώστε να είναι εφικτή η επικοινωνία του ενός παίκτη με τον άλλο και κατ' επέκταση να καταστεί δυνατή η εξέλιξη του παιχνιδιού.

## 4.2 Χρήση Κανόνων στη Real-Time Database

Η Real-Time Database δίνει τη δυνατότητα να οριστούν κανόνες οι οποίοι μπορούν να περιορίσουν την πρόσβαση στη βάση σε κάποιες κατηγορίες χρηστών. Πιο αναλυτικά, μπορεί να εφαρμοστεί ειδική πολιτική για συγκεκριμένες λειτουργίες της βάσης όπως ανάγνωση, εισαγωγή, επεξεργασία και διαγραφή εγγραφών. Στη συγκεκριμένη εφαρμογή οι κανόνες που ορίστηκαν στη βάση είναι οι ακόλουθοι.

```
1  {
2    "rules": {
3      "Users": {
4        ".read": "auth != null",
5        "$uid": {
6          ".write": "auth != null && auth.uid === $uid"
7        }
8      },
9      "Games": {
10       ".read": "auth != null",
11       "$uid": {
12         ".write": "auth.uid === $uid",
13         "$sub": {
14           ".write": "auth.uid === $uid || (data.exists() && newData.exists())"
15         },
16         "player_1": {
17           ".write": "auth.uid === $uid"
18         },
19         "is_available": {
20           ".write": "auth.uid === $uid"
21         }
22       }
23     }
24   }
25 }
```

Κανόνες Βάσης Δεδομένων

Οι κανόνες αυτοί ορίζουν ότι ένας μη συνδεδεμένος χρήστης δε μπορεί να κάνει καμία ενέργεια στη βάση (*auth.uid != null* ή *auth.uid === \$uid*). Αυτό καθίσταται εφικτό από τις δυνατότητες που παρέχει το Firebase Authentication, όπως προαναφέρθηκε, και μπορεί να αναγνωρίσει αν κάποιο αίτημα που φτάνει στη βάση δεδομένων προέρχεται από συνδεδεμένο χρήστη. Αναλυτικότερα, όλοι οι συνδεδεμένοι χρήστες μπορούν να διαβάσουν όλες τις εγγραφές που υπάρχουν στους κόμβους «Users» και «Games» ώστε να εξασφαλιστεί η ομαλή λειτουργία του παιχνιδιού. Ωστόσο, στον

κόμβο «Users» κάθε χρήστης μπορεί μόνο ο ίδιος να τροποποιήσει τα στοιχεία που βρίσκονται στον δικό του υπό-κόμβο και κανένας άλλος. Αντίστοιχα στον κόμβο «Games» και στον υπό-κόμβο του κάθε χρήστη όλοι μπορούν να αλλάζουν τις τιμές των πεδίων αλλά δε μπορούν να τα διαγράψουν ή να δημιουργήσουν νέα. Επίσης, σημαντικά πεδία όπως το `is_available` και `player_1` δε μπορούν να τροποποιηθούν από κανένα άλλον παρά μόνο από τον χρήστη στον οποίο ανήκουν. Με αυτό τον τρόπο διασφαλίζεται η ακεραιότητα των δεδομένων και ότι δε θα συμβεί κακόβουλη ενέργεια και επεξεργασία από μη εξουσιοδοτημένους χρήστες.

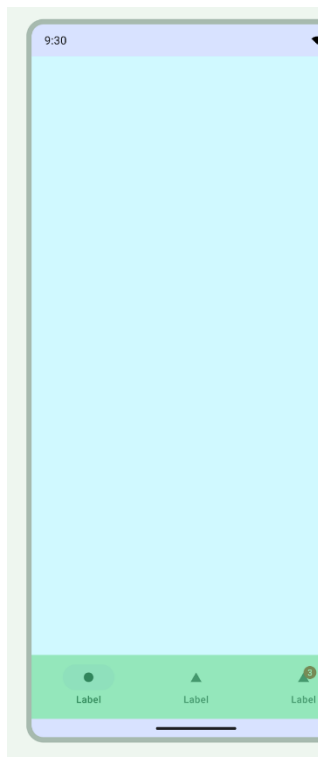


## 5. Android UI

Η Google έχει δημιουργήσει για τις εφαρμογές Android ένα ολοκληρωμένο σύνολο γραφικών στοιχείων (Material Design Components), τα οποία συνεισφέρουν στη δημιουργία όμορφων και λειτουργικών γραφικών διεπαφών για τις οθόνες της εκάστοτε εφαρμογής. Στη συγκεκριμένη υλοποίηση χρησιμοποιήθηκε πλήθος από αυτά τα γραφικά στοιχεία, όπως «Button», «Progress Indicators», «Checkbox» κ.λπ.. Ορισμένα από αυτά παρουσιάζουν ιδιαίτερο ενδιαφέρον και αξίζει να αναλυθούν περαιτέρω στις επόμενες ενότητες.

### 5.1 Navigation Bar

Για την πλοήγηση στις βασικές οθόνες της εφαρμογής ο συνδεδεμένος χρήστης χρησιμοποιεί ένα Bottom Navigation Bar. Η επιλογή χρήσης του Navigation Bar που παρέχεται από τη Google έναντι μίας μη τυποποιημένης υλοποίησης προκρίθηκε αφενός λόγω της δυνατότητας πλήρους συμμόρφωσης με τα σχεδιαστικά πρότυπα που ορίζει η Google στην εναλλαγή οθονών με χρήση πλοήγησης και αφετέρου λόγω της οικειότητας που έχει ο χρήστης με αυτό το ευρέως χρησιμοποιούμενο σύστημα πλοήγησης στο κάτω μέρος της οθόνης.



Navigation Bar  
Πηγή: [Google](#)

Η χρήση του Navigation Bar, επίσης, μειώνει τον κώδικα που χρειάζεται να γραφεί καθώς ο κώδικας παρέχεται έτοιμος και βελτιστοποιημένος οδηγώντας στην αποφυγή λαθών και λειτουργικών προβλημάτων που ενδεχομένως θα υπήρχαν σε αντίθετη περίπτωση. Για να καταστεί το Navigation Bar λειτουργικό έπρεπε να δημιουργηθούν δύο πρόσθετα αρχεία XML τα οποία θα παρείχαν τις απαραίτητες πληροφορίες για τη φόρτωση διαφορετικής οθόνης σε κάθε επιλογή από το μενού στο κάτω μέρος της οθόνης. Συγκεκριμένα υλοποιήθηκαν τα ακόλουθα αρχεία:

➤ **navigation\_menu.xml:** Ορίζει ποια κουμπιά, δηλαδή τι θα εμφανίζεται στο μενού προς επιλογή από τον χρήστη. Στην εφαρμογή ορίζονται τα ακόλουθα «Αρχική», «Παιχνίδι», «Στατιστικά» και «Ρυθμίσεις». Το αρχείο αυτό ορίζει με κάθε λεπτομέρεια τον τρόπο εμφάνισης του κουμπιού δηλαδή και εικονίδιο και τίτλο.

- **navigation\_graph.xml:** Το αρχείο αυτό προσδιορίζει ποια οθόνη «Fragment» πρέπει να φορτωθεί και να εμφανιστεί σε κάθε επιλογή που θα κάνει ο χρήστης από τη μπάρα επιλογών στο κάτω μέρος της οθόνης. Με άλλα λόγια, ορίζεται ποιο αρχείο XML θα πρέπει να εμφανιστεί κάθε φορά ζητείται η πλοήγηση σε μια διαφορετική οθόνη.

Η σύνδεση και ο συντονισμός των λειτουργιών που υπάρχουν στα αρχεία *navigation\_menu.xml* και *navigation\_graph.xml* γίνεται αποκλειστικά μέσα από τη κλάση Navigation UI που δίνεται από το Android για αυτό τον σκοπό. Με αυτόν τον τρόπο παρέχεται μία αποδοτική υπηρεσία πλοήγησης του χρήστη μεταξύ των κεντρικών σελίδων της εφαρμογής.

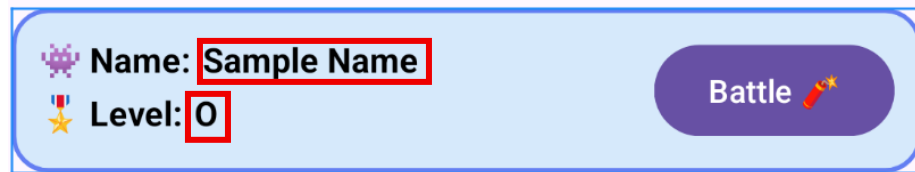
## 5.2 Recycler View

Η εμφάνιση της λίστας με τους διαθέσιμους αντιπάλους στην Οθόνη «Παιχνίδι» της εφαρμογής ήταν μια μεγάλη πρόκληση καθώς αν οι διαθέσιμοι αντίπαλοι αυξηθούν πολύ σε αριθμό αυτό συνεπάγεται και έναν πολύ μεγάλο όγκο από UI στοιχεία που θα πρέπει να φορτωθούν για προβολή. Για να αντιμετωπιστεί το συγκεκριμένο ζήτημα και να περιορισθούν οι πιθανότητες η εφαρμογή να δεσμεύσει πολλούς πόρους σε μνήμη RAM για την προβολή της λίστας των διαθέσιμων αντιπάλων στην οθόνη του χρήστη έγινε χρήση του RecyclerView. Το RecyclerView αποτελείται από ένα σύνολο αντικειμένων και μεθόδων τα οποία λειτουργούν συγχρονισμένα με σκοπό κάθε «κάρτα» (στοιχείο UI) που δείχνει τον εκάστοτε διαθέσιμο αντίπαλο να επαναχρησιμοποιείται χωρίς να χρειάζεται να καταστρέφεται ολοκληρωτικά όταν παύει να είναι ορατή στον χρήστη (αν ο χρήστης κάνει scroll και χαθεί από το οπτικό του πεδίο τότε αυτή η «κάρτα» δεν καταστρέφεται ως οντότητα αλλά αντιθέτως τα περιεχόμενα-πεδία της ενημερώνονται ώστε να δείχνει τον επόμενο διαθέσιμο αντίπαλο συμβάλλοντας με αυτόν τον τρόπο στην αποδοτική διαχείριση της μνήμης).

Για τη λειτουργία του RecyclerView απαιτήθηκε η υλοποίηση του Adapter, ο οποίος είναι ο κεντρικός συντονιστής που είναι υπεύθυνος για τη λειτουργία της δυναμικής λίστας προβολής. Ο κώδικας για τον Adapter είναι στο αρχείο *Opponents\_Recycle\_Adapter.java* (κλάση *Opponents\_Recycle\_Adapter*) το οποίο βρίσκεται στον φάκελο *Business\_Logic*. Η κλάση αυτή περιέχει τα ακόλουθα σημαντικά στοιχεία (ιδιότητες και μέθοδοι):

- **List<Opponent\_Card> opponentCards:** Αποθηκεύει τα αντικείμενα τύπου *Opponent\_Card* που προορίζονται για εμφάνιση στον χρήστη.

- **onCreateViewHolder(...):** Χρησιμοποιείται για τη δημιουργία του γραφικού τρόπου εμφάνισης κάθε «κάρτας» από το σχέδιο που υπάρχει στο αρχείο *R.layout.opponent\_card*.
- **onBindViewHolder(...):** Καλείται από το RecyclerView όταν θέλει να εμφανίσει έναν νέο διαθέσιμο αντίπαλο σε ένα υπάρχον *OpponentViewHolder* αντικείμενο. Συγκεκριμένα, αναλαμβάνει να γεμίσει τα πεδία του αντικειμένου αυτού με τις πραγματικές τιμές (Όνομα, Επώνυμο, κ.λπ.) του εκάστοτε αντιπάλου στη λίστα *opponentCards* ο οποίος βρίσκεται προς εμφάνιση τη δεδομένη στιγμή.



Αντιστοίχιση των πεδίων προς εμφάνιση

- **int getItemCount(...):** Η μέθοδος αυτή είναι υπεύθυνη για την επιστροφή του πλήθους των στοιχείων της λίστας *opponentsCards*.
- **static class OpponentViewHolder:** Η συγκεκριμένη κλάση περιέχει δηλωμένα ως ιδιότητες τα γραφικά αντικείμενα προς εμφάνιση στην «κάρτα» κάθε αντιπάλου και παρέχει αντιστοίχιση με τα γραφικά αντικείμενα που υπάρχουν στο αρχείο *R.layout.opponent\_card*. Αντικείμενα αυτής της κλάσης διατηρούν τα δεδομένα των διαθέσιμων αντιπάλων που εμφανίζονται κάθε χρονική στιγμή στο οπτικό πεδίο της οθόνης του χρήστη.
- **void updateOpponentsCard(...):** Η μέθοδος αυτή ενημερώνει τη λίστα *opponentCards* με νέους αντιπάλους και στη συνέχεια ενημερώνει τον Adapter ότι πρέπει να ανανεώσει την λίστα που βλέπει ο χρήστης.

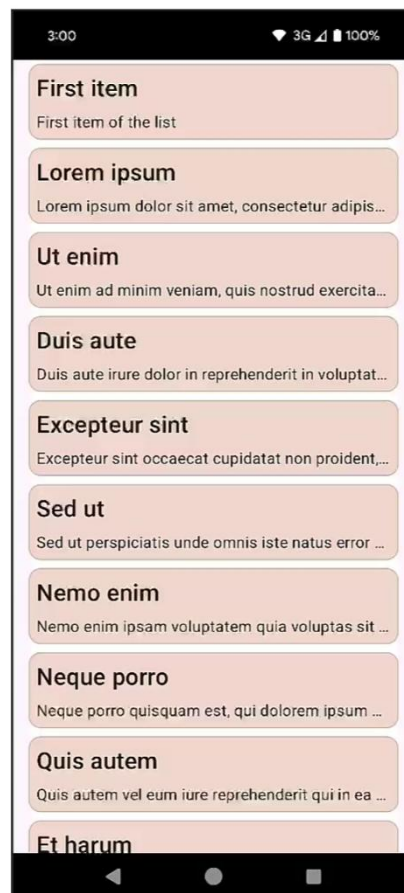
Επίσης, για να είναι λειτουργικό το RecyclerView έπρεπε να δημιουργηθεί και ένα αρχείο *Opponent\_Card.java* (κλάση *Opponent\_Card*) στον φάκελο *Models* το οποίο θα αποθηκεύει στις ιδιότητές του τα στοιχεία του κάθε αντιπάλου που προορίζεται για εμφάνιση στη λίστα. Επομένως, η κλάση αυτή χρησιμοποιείται από τις μεθόδους του Adapter που προαναφέρθηκαν και αποτελείται από τις ακόλουθες ιδιότητες:

- **String full\_name:** Το πλήρες ονοματεπώνυμο του εκάστοτε αντιπάλου
- **int level:** Το επίπεδο του εκάστοτε αντιπάλου
- **String opponent\_id:** Το μοναδικό αναγνωριστικό χρήστη του αντιπάλου (Δεν εμφανίζεται στο UI)

- **GameFragment gameFragment:** Μία αναφορά στο ενεργό αντικείμενο Game Fragment. Χρησιμοποιείται εσωτερικά σε περίπτωση πρόκλησης του εμφανιζόμενου αντιπάλου σε μάχη (Δεν εμφανίζεται στο UI)

Η αρχικοποίηση των πεδίων αυτών γίνεται από τον constructor και επιπλέον για όλες τις παραπάνω ιδιότητες έχουν δημιουργηθεί οι αντίστοιχοι Getters για την πρόσβαση στο περιεχόμενό τους.

Με την υλοποίηση των ανωτέρω επιτεύχθηκε η λειτουργία μιας δυναμικής λίστας στην οθόνη «Παιχνίδι» για την προβολή των διαθέσιμων αντιπάλων η οποία δεν επιβαρύνει τους πόρους της συσκευής του χρήστη όσο μεγάλος και αν είναι ο αριθμός των αντιπάλων προς εμφάνιση. Μια ενδεικτική μορφή του RecyclerView δίνεται στην παρακάτω εικόνα.



Λίστα RecyclerView

Πηγή: [Google](#)

## Συμπεράσματα & Μελλοντικές Επεκτάσεις

Η εφαρμογή που αναπτύχθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας και αναλύθηκε εκτενώς στα προηγούμενα κεφάλαια αποτελεί μια σύγχρονη και πλήρως λειτουργική εφαρμογή που δίνει τη δυνατότητα στους χρήστες της να παίξουν το διάσημο παιχνίδι «Ναυμαχία» με οποιονδήποτε αντίπαλο επιλέξουν οπουδήποτε στον κόσμο. Σημαντικό σε αυτό το σημείο είναι να επισημανθούν οι περιορισμοί που υπήρχαν κατά τη διάρκεια υλοποίησης της εφαρμογής όπως για παράδειγμα ήταν η δυσκολία στον έλεγχο της ορθής λειτουργίας της. Από τη στιγμή που επρόκειτο για μια εφαρμογή πολλαπλών παικτών ήταν μεγάλη πρόκληση να υπάρξουν ταυτόχρονα συνδεδεμένες στο περιβάλλον ανάπτυξης τουλάχιστον 3 κινητές συσκευές στις οποίες να έχει εγκατασταθεί και να εκτελείται η εφαρμογή ώστε να μπορέσει να ελεγχθεί η ομαλή λειτουργία της καθώς και η αναμενόμενη συμπεριφορά της σε διάφορα σενάρια χρήσης ώστε να μπορέσει να επιλυθεί οποιοδήποτε πρόβλημα υπήρχε στον κώδικα. Επιπρόσθετα, η προσαρμογή που έγινε για χρήση σε ταμπλέτες (συσκευές με μεγαλύτερη οθόνη) απαιτούσε τη δημιουργία ενός ακόμα εξομοιωτή στο περιβάλλον ανάπτυξης ώστε να μπορέσει να πιστοποιηθεί ότι η εμφάνιση της εφαρμογής ήταν η αναμενόμενη και σε αυτή την περίπτωση. Όλοι οι παραπάνω περιορισμοί αντιμετωπίστηκαν με χρήση τόσο εξωτερικών συσκευών για να ελαφρυνθεί ο φόρτος του κυρίως συστήματος που γινόταν η ανάπτυξη όσο και με εικονικές συσκευές / εξομοιωτές που έτρεχαν παράλληλα στο περιβάλλον ανάπτυξης. Παρά τις όποιες δυσκολίες αντιμετωπίστηκαν η χρήση των κατάλληλων εργαλείων και υπηρεσιών όπως η Firebase για την υλοποίηση της βάσης δεδομένων, τα εξειδικευμένα γραφικά στοιχεία του Android που ενισχύουν την απόδοση της εφαρμογής σε περιπτώσεις που είναι αναγκαία η διαχείριση μεγάλης ποσότητας πληροφορίας καθώς και η χρήση βέλτιστων προγραμματιστικών τεχνικών σε επίπεδο κώδικα δημιουργήθηκε μια ποιοτική και άξια αναφοράς εφαρμογή.

Η εφαρμογή προσφέρει μεγάλο εύρος δυνατοτήτων στον χρήστη τόσο για την παραμετροποίησή της (γλώσσα εμφάνισης, θέμα εμφάνισης κ.λπ.) όσο και για τις ρυθμίσεις του παιχνιδιού (αναδιάταξη πλοίων τυχαία και χειροκίνητα κ.λπ.). Θα μπορούσε ωστόσο να επεκταθεί περαιτέρω και να εμπλουτιστεί με επιπλέον δυνατότητες που θα έκαναν την εμπειρία του χρήστη ακόμα πιο συναρπαστική. Μία ενδιαφέρουσα προσθήκη θα ήταν η δυνατότητα για την πραγματοποίηση «Ναυμαχίας» με τον υπολογιστή χωρίς την ύπαρξη φυσικού αντιπάλου, με κλιμακούμενο ενδεχομένως επίπεδο δυσκολίας, κάτι που θα επέτρεπε στον χρήστη να χρησιμοποιεί την εφαρμογή και να παίζει το παιχνίδι είτε όταν δε βρίσκει διαθέσιμους αντιπάλους είτε όταν η σύνδεση στο διαδίκτυο δεν είναι εφικτή (π.χ. ταξίδι με αεροπλάνο). Αυτή η προσθήκη θα αποτελούσε ένα μεγάλο άλμα για την εφαρμογή καθώς θα μεγέθυνε τις δυνατότητές της και θα της επέτρεπε να παρέχει μια ακόμα πιο ολοκληρωμένη εμπειρία στους χρήστες της.

## Βιβλιογραφία

### [Επιστημονικές Δημοσιεύσεις]

Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." *International Journal of Computer Applications* 179.46 (2018): 49-53.

[Application of Firebase in Android App Development-A Study \(Link\)](#)

Mawlood-Yunis, Abdul-Rahman. "ListView, ScrollList, Date and Time Pickers, and RecyclerView." *Android for Java Programmers*. Cham: Springer International Publishing, 2022. 279-324.

[ListView, ScrollList, Date and Time Pickers, and RecyclerView - Android for Java Programmers \(Link\)](#)

Patsakis, Constantinos & Alepis, Efthymios. (2017). Trapped by the UI: The Android case. 10.1007/978-3-319-66332-6\_15.

[Trapped by the UI: The Android case | ResearchGate \(Link\)](#)

Tselepatiotis, Michail & Alepis, Efthimios. (2023). Design of Real-Time Multiplayer Word Game for the Android Platform Using Firebase and Fuzzy Logic. 1-8.

10.1109/IISA59645.2023.10345901.

[Design of Real-Time Multiplayer Word Game for the Android Platform Using Firebase and Fuzzy Logic | ResearchGate \(Link\)](#)

Alepis, Efthymios & Virvou, Maria. (2014). Object-Oriented User Interfaces for Personalized Mobile Learning. 10.1007/978-3-642-53851-3.

[Object-Oriented User Interfaces for Personalized Mobile Learning | ResearchGate \(Link\)](#)

Alepis, Efthymios & Virvou, Maria & Drakoulis, Savas. (2014). Human smartphone interaction: Exploring smartphone senses. IISA 2014 - 5th International Conference on Information, Intelligence, Systems and Applications. 44-48. 10.1109/IISA.2014.6878825.

[Human smartphone interaction: Exploring smartphone senses | ResearchGate \(Link\)](#)

Kontogianni, Aristeia & Kabassi, Katerina & Alepis, Efthymios. (2018). Designing a Smart Tourism Mobile Application: User Modelling Through Social Networks' User Implicit Data: 10th International Conference, SocInfo 2018, St. Petersburg, Russia, September 25-28, 2018, Proceedings, Part II. 10.1007/978-3-030-01159-8\_14.

[Designing a Smart Tourism Mobile Application: User Modelling Through Social Networks' \(Link\)](#)

[\*User Implicit Data: 10th International Conference, SocInfo 2018, St. Petersburg, Russia, September 25-28, 2018, Proceedings, Part II | ResearchGate \(Link\)\*](#)

Alepis, Efthymios & Patsakis, Constantinos. (2017). Hey Doc, Is This Normal?: Exploring Android Permissions in the Post Marshmallow Era. 53-73. 10.1007/978-3-319-71501-8\_4. [\*Hey Doc, Is This Normal?: Exploring Android Permissions in the Post Marshmallow Era | ResearchGate \(Link\)\*](#)

Hagos, Ted & Zechner, Mario & DiMarzio, J.F. & Green, Robert. (2020). Beginning Android Games Development, From Beginner to Pro. 10.1007/978-1-4842-6121-7. [\*Beginning Android Games Development: From Beginner to Pro | Springer \(Link\)\*](#)

### **[Διαδικτυακές Πηγές]**

Android App “Sea Battle Batalies”

[\*Sea Battle Batalies - Google Play \(Link\)\*](#)

Android App “Battleship on Cells”

[\*Battleship on Cells - Google Play \(Link\)\*](#)

CountDownLatch in Java

[\*CountDownLatch - GeeksforGeeks \(Link\)\*](#)

Realtime Database data format

[\*Structure Your Database - Firebase Realtime Database \(Link\)\*](#)

Realtime Database security rules

[\*Security Rules language - Firebase Security Rules \(Link\)\*](#)

Android UI layout guide

[\*Layout basics - Android Developers \(Link\)\*](#)

Android UI elements

[\*Material Components - Android Developers \(Link\)\*](#)

RecyclerView usage and implementation

[\*Create dynamic lists with RecyclerView - Android Developers \(Link\)\*](#)