ΑΘΑΝΑΣΙΟΣ ΔΕΛΗΣ 03117103

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ

1. Σχεδιασμός και υλοποίηση λογικής (55%)

Από αυτό το σημείο και έπειτα όταν θα αναφέρομαι στο μέρος 1. , την λειτουργικότητα δηλαδή θα λέω το Παιχνίδι, ενώ για τις λύσεις του μέρους 2. η Διεπαφή.
Το παιχνίδι υλοποιήθηκε σε Eclipse IDE JDK 15 JavaFx 11, δοκιμάστηκε και με Runtime JDK 11 και δούλευε κανονικά. Έχει δοκιμαστεί μόνο σε Eclipse αν και ως Project ξεκίνησε σε Visual Studio Code. Αποτελείται από 10 κλάσεις .java και ένα μορφότυπο τύπου .fxml . Έγινε επίσης χρήση SceneBuilder.
Τυπικά, το παιχνίδι παίζεται χρησιμοποιώντας 4 ταμπλό , δύο για κάθε παίκτη. Στο ένα ταμπλό κάθε παίκτης τοποθετεί τα πλοία του και καταγράφει τις βολές του αντιπάλου και στο δεύτερο ταμπλό καταγράφει τις δικές του βολές.

Επί της ουσίας ως αντικείμενα υπάρχουν μόλις 2 ταμπλό τα οποία εμπειριέχουν όλη την πληροφορία για το Παιχνίδι καθώς και την πληροφορία για την Διεπαφή.
Ως υλοποίηση 4 ταμπλό, θεωρήθηκε να οριστούν 4 μεταβλητές, 2 για κάθε παίχτη. Αν 2 δείχνουν ωστόσο ως σχεδιαστική επιλογή στο ίδιο αντικείμενο. Η οπτική του χρήστη ανθρώπου είναι αυτή που χρησιμοποιείται ια το υπόλοιπο της λύσης. Το παραπάνω θεωρήθηκε η βέλτιστη επιλογή από άποψης μνήμης και πολυπλοκότητας υλοποίησης.

Στην εφαρμογή θα υπάρχει ένας παίκτης που παίζει με αντίπαλο τον υπολογιστή. Η γραφική διεπαφή που θα υλοποιηθεί στη συνέχεια θα παρουσιάζει τα δύο ταμπλό του παίκτη, ενώ για τον αντίπαλο (υπολογιστής) δεν θα υπάρχει γραφική απεικόνιση των δικών του ταμπλό. Επιπλέον, θεωρούμε ότι τα ταμπλό που μοντελοποιούν το χώρο της ναυμαχίας αντιστοιχτούν σε ένα πλέγμα 10 γραμμών και 10 στηλών.

Πράγματι εμφανίζονται μόνο 2 ταμπλό από την οπτική του παίχτη των 10 επί 10 γραμμών.

Πριν ξεκινήσει το παιχνίδι, έχουμε την τοποθέτηση των πλοίων στο βασικό ταμπλό κάθε παίκτη. Κάθε πλοίο καταλαμβάνει έναν αριθμό από συνεχόμενα κελία ενώ τα πλοία τοποθετούνται είτε κάθετα είτε οριζόντια και δεν μπορούν να καταλαμβάνουν διαγώνια κελία στο ταμπλό. Το πλήθος των κελιών που καταλαμβάνει κάθε πλοίο εξαρτάται από τον τύπο του. Κάθε κελί μπορεί να καταλαμβάνεται το πολύ από ένα πλοίο ενώ κάθε παίκτης έχει στη διάθεση του ένα πλοίο από την κάθε κατηγορία. Θεωρούμε 5 διαφορετικούς τύπους πλοίων τα χαρακτηριστικά των οποίων συνοψίζονται στον παρακάτω πίνακα.

Πίνακας 1 Χαρακτηριστικά διαθέσιμων πλοίων

Τύπος πλοίου	Πλήθος θέσεων	Πόντοι εύστοχης βολής	Bonus βύθισης
Carrier	5	350	1000

Battleship	4	250	500
Cruiser	3	100	250
Submarine	3	100	0
Destroyer	2	50	0

Πράγματι οι υλοποιήσεις των παραπάνω πλοίων έγιναν κατά την διάρκεια εκκίνησης του παιχνιδιού με απουσία διαγώνιων πλοίων. Για λεπτομέρειες του πως μοντελοποιήθηκε το παραπάνω ταμπλό, εσκεμένα υπάρχει Documentation της σχετικής κλάσης απαντώντας έτσι και στο ερώτημα Β.3.1

Κάθε πλοίο βρίσκεται σε μια από τις παρακάτω καταστάσεις:
- Ανέπαφο: Δεν έχει δεχθεί απολύτως καμία βολή.
- Χτυπημένο: Έχει δεχθεί τουλάχιστον μια βολή αλλά δεν έχει ακόμη χτυπηθεί σε όλες τις θέσεις και παραμένει στο παιχνίδι.
- Βυθισμένο: Έχει χτυπηθεί σε όλες τις θέσεις που καταλαμβάνει στο ταπλό, σε αυτή την περίπτωση βγαίνει εντελώς εκτός παιχνιδιού.

Η πληροφορία αυτή η εξάγεται έμμεσα από δεδομένα που υπάρουν στην κλάση που παρουσιάζεται στο documentation.

Η εφαρμογή θα πρέπει να σχεδιαστεί με γενικό τρόπο ώστε να μπορεί να υποστηρίζει διαφορετικές περιγραφές για την διάταξη των πλοίων του παίκτη και του υπολογιστή. Θεωρούμε πώς υπάρχει ένας προκαθορισμένος φάκελος ***"medialab"*** που περιλαμβάνει μια σειρά από αρχεία περιγραφής που παρέχουν τα απαραίτητα στοιχεία. Το αρχείο με την περιγραφή της διάταξης των πλοίων για το παίκτη θα πρέπει να ονομάζεται ***"player_SCENARIO-ID.txt"*** και το αντίστοιχο αρχείο για τον υπολογιστή ***"enemy_SCENARIO-ID.txt".***

ΠΑΡΑΔΟΧΗ 1: Υπάρχει φάκελο medialab εντός του Project ωστόσο ως σύστημα ονοματοδοσίας χρημοποιήθηκε αντίστοιχο με αυτό το examples.rar . Δηλαδή αρχεία της μορφη: USER_default_RESTOFNAME.txt

Όπου USER:{player , enemy} και RESTOFNAME οτιδήποτε έφκυρο αποτελούμενο αποκλειστικά από χρακτήρες αγγλικούς που τα Windows 8 δέχονται ως έγκυρο όνομα.

ΠΑΡΑΔΟΧΗ 2: Στα αρχεία και στο Παιχνίδι η αρίθμηση στειλώ είναι από 0 έως 9 ωστόσο στην Διεπαφή ως πιο άνετο και ρεαλιστικό για τον χρήστη επιλέχθηκε απεικόνηση 1 εως 10.

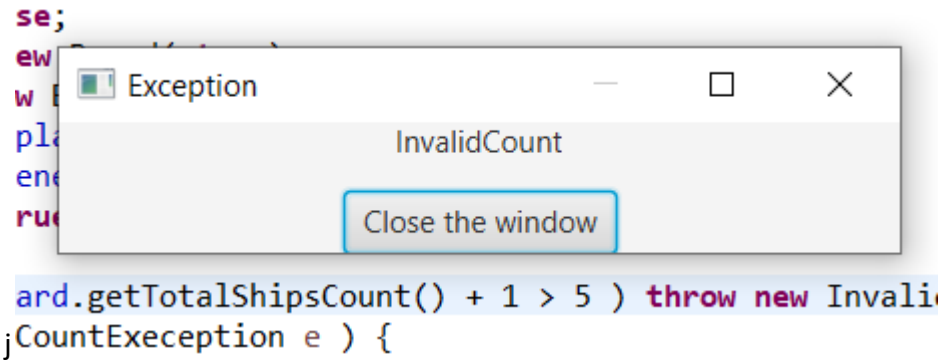
Θα πρέπει επίσης να κατασκευαστούν μέθοδοι που ελέγχουν αν η τοποθέτηση τηρεί τους παρακάτω περιορισμούς:

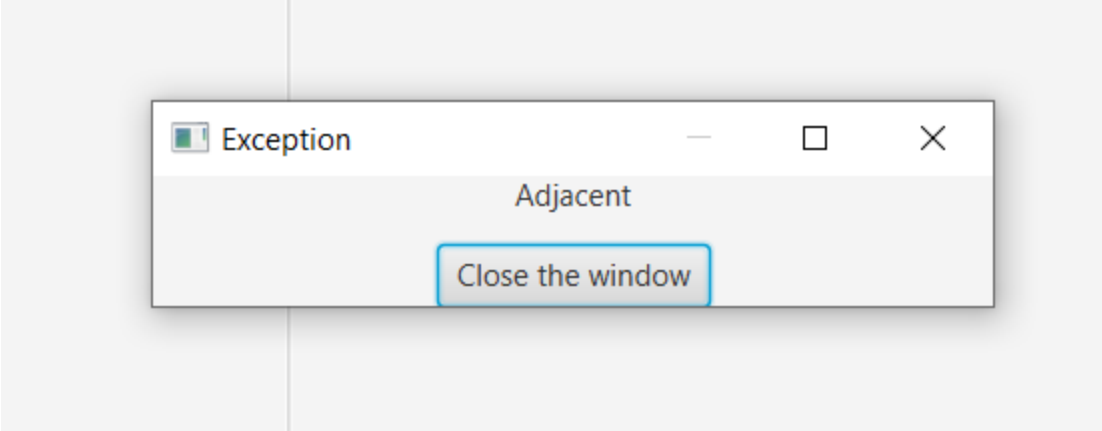
- Ένα πλοίο δεν μπορεί να βγαίνει εκτός των ορίων του ταμπλό.
- Ένα πλοίο δεν μπορεί να τοποθετηθεί σε κελί που ήδη έχει άλλο πλοίο.
- Ένα πλοίο δεν μπορεί να εφάπτεται κάθετα ή οριζόντια με κανένα άλλο πλοίο, έστω και για ένα κελί. Αυτό σημαίνει ότι τα πλοία έχουν μεταξύ τους απόσταση τουλάχιστον ένα ελεύθερο κελί.
- Δεν μπορούν να υπάρχουν περισσότερα από ένα πλοία για κάθε τύπο.

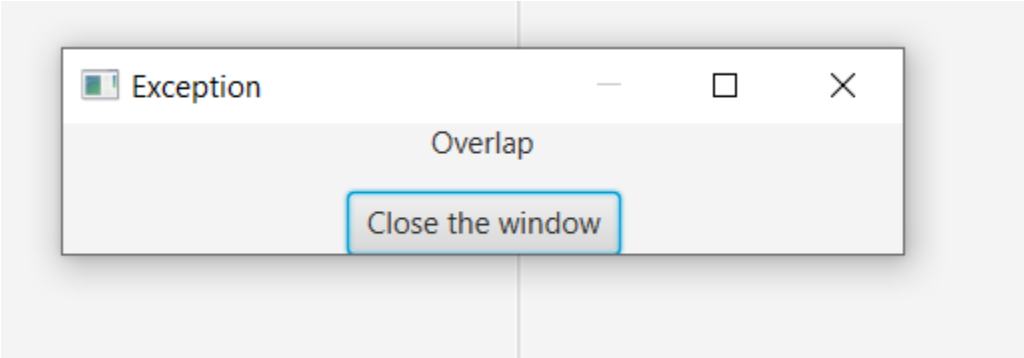
Για κάθε μία από τις παραπάνω περιπτώσεις μπορείτε να ορίσετε τον αντίστοιχο τύπο εξαίρεσης: ***OversizeException, OverlapTilesException, AdjacentTilesException, InvalidCountExeception***.

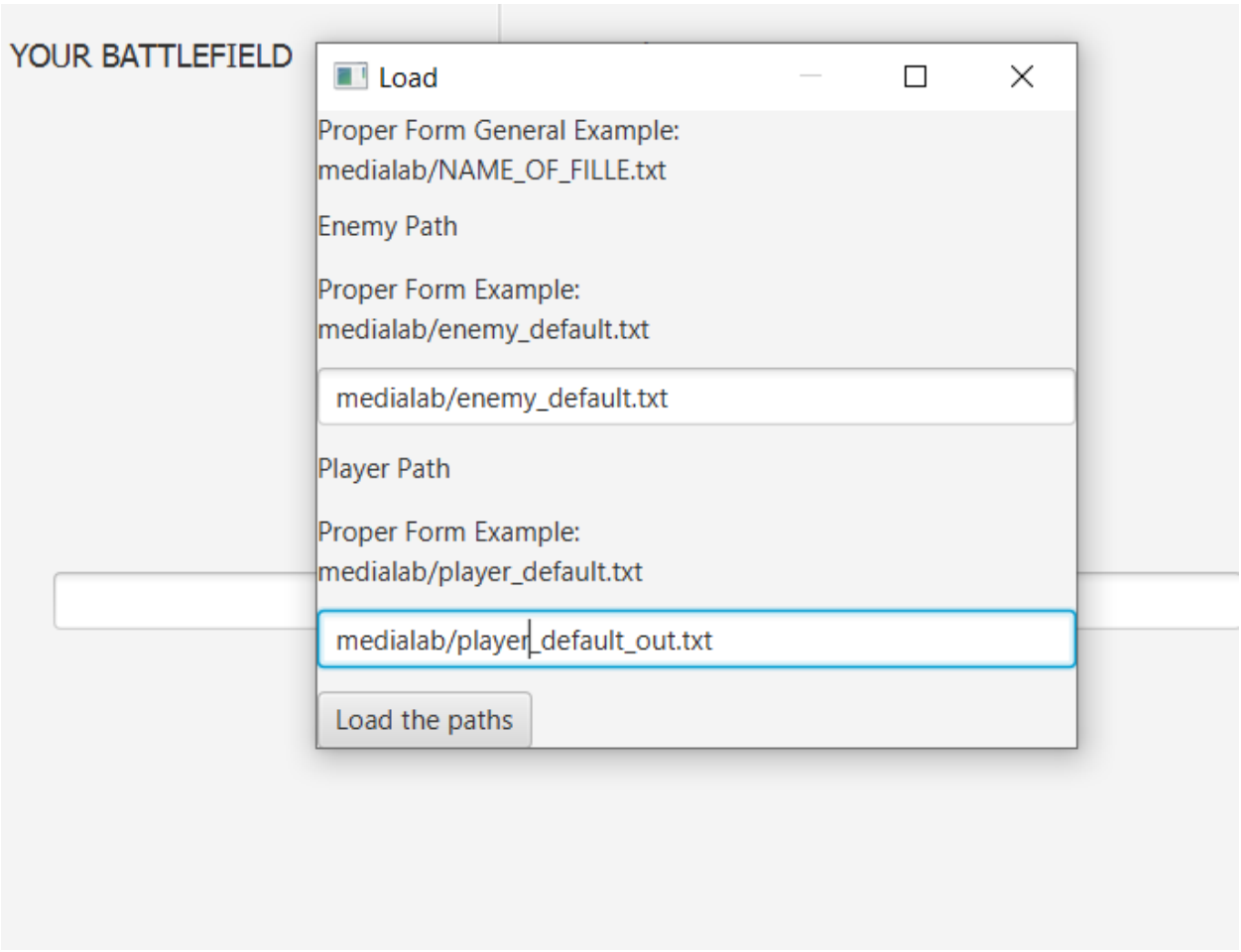
Στη σελίδα του εργαστηρίου μαζί με την εκφώνηση μπορείτε να βρείτε το αρχείο *"**examples.rar**"* που περιλαμβάνει ολοκληρωμένα παραδείγματα περιγραφής, καθώς και σημειώσεις για την σύνταξη των σχετικών αρχείων περιγραφής.

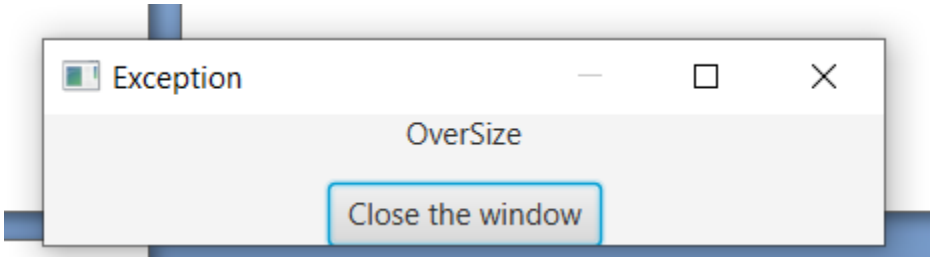
Υλοποιήθηκαν και τα 4 Exceptions με ειδικά κατασκευασμένες κλάσεις. Προφανώς η InvalidCount δεν γίνεται ποτέ εμφανής στο Runtime, διότι λόγο των δεδομένων της άσκησης ο αριθμός των πλοίων δεν ορίζεται στην Διεπαφή. Ως προς την ενημέρωση του χρήστη στο Runtime για τις υπόλοιπες τρεις, χρησιμοποιήθηκαν static συναρτήσεις της κλάσης που χειρίζεται ζητήματα που αφορούν Pop up windows, και ζητήθηκαν ούτως ή άλλως για τις ανάγκες του κομματιού 6 της Διεπαφής. Ακολουθούν παραδείγματα.




Exception
Adjacent
Close the window


Exception
Overlap
Close the window





Στον φάκελο medialab τα ερχεία _out, _adj, _over είναι παραδείγματα που οδηγούν σε Exceptions της προαναφερόμενης μορφής.

Μετά την αρχικοποίηση με την κατάλληλη τοποθέτηση των πλοίων το παιχνίδι συνεχίζεται με μια σειρά από γύρους. Αρχικά, η εφαρμογή θα επιλέγει τυχαία αν την πρώτη κίνηση σε κάθε γύρο την κάνει ο παίκτης ή ο υπολογιστής. Η επιλογή γίνεται πριν το πρώτο γύρο και διατηρείται για όλη την διάρκεια του παιχνιδιού, ενώ θα πρέπει να ενημερώνεται κατάλληλα ο χρήστης με κάποιο μήνυμα στην γραφική διεπαφή.

Υλοποιήθηκε.

Ο παίκτης θα ορίζει το κελί «στόχο» μέσω της γραφικής διεπαφής. Η εφαρμογή με βάση την διάταξη των πλοίων του αντιπάλου, που θα διατηρεί εσωτερικά, θα αναγνωρίζει αν η βολή ήταν επιτυχής ή όχι. Στην περίπτωση επιτυχούς βολής ο χρήστης κερδίζει τους πόντους εύστοχης βολής που αντιστοιχούν στο τύπο του πλοίου ενώ αν η βολή έχει ως αποτέλεσμα τη βύθιση του πλοίου τότε επιπρόσθετα κερδίζει και τους πόντους που αντιστοιχούν στο bonus βύθισης. Ακόμη, μετά από κάθε βολή θα πρέπει να ενημερώνεται κατάλληλα η γραφική διεπαφή.

Υλοποιήθηκε.

ΠΑΡΑΔΟΧΗ 3: Το σύστημα πόντων εσωτερικά ανανεώνεται μετά από κάθε κίνηση έκαστου παίχτης. Ωστόσο γίνεται έλεγχος νίκης αφού παίξει έκαστος παίχτης και όχι αφού παίξουν και οι 2 παίχτες, ως προς την βύθιση όλων των πλοίων. Για την $2^{η}$ συνθήκη νίκης ισχύει το ανάποδο πρέπει να έχουν παίξει και οι 2 παίχτες.

Ο υπολογιστής αρχικά θα πραγματοποιεί τυχαίες βολές αλλά όταν έχει κάποια εύστοχη θα προσπαθεί να κάνει βολές σε γειτονικά κελιά, όπως θα έκανε ο χρήστης. Μετά από κάθε βολή του υπολογιστή θα πρέπει να ανανεώνεται κατάλληλα η τρέχουσα κατάσταση και η γραφική διεπαφή.

Υλοποιήθηκε. Η παραπάνω απαίτηση για τα γειτονικά κελία δουλεύει με την μοντελοποίηση ενός αυτομάτου καταστάσεων, που περιγράφει τον παρακάτω αλγόριθμο.

ΑΛΓΟΡΙΘΟΣ: Αφού ο υπολογιστής τυχαία πετύχει πλοίο, και επειδή το πλοία απέχουν τουλάχιστον 1 κελί μεταξύ τους, έχει το πολύ 4 επιλογές για την επόμενη θέση του πλοίου.

Μετά την δεύτερη επιτυχή βολή γνωρίζει πλέον την κατεύθυνση του πλοίου. Έτσι μέχρι το πλοίο να πεθάνει επιτελεί βολές στην ίδια γραμμή για οριζόντια διάταξη και στήλη για την κατακόρυφη πέριξ των 2 πρώτων σημείων.

ΠΑΡΑΔΟΧΗ 4: Αφού βρει την κατεύθυνση, δεν βαρά εναλλάξ εκατέρωθεν τον 2 κελιών, σε αποδεκτές πάντα θέσεις. Ανταυτό βαρά όλες τις πιθανές θέσεις από την μια πλευρά και μετά όλες από την άλλη μέχρι το πλοίο να είναι επίσημα νεκρό. Αυτό μπορεί να του δόσει μειωνέκτημα σε σχέση ε τον άνθρωπο που υλοποιεί ποιο ευφάνταστα τον παραπάνω αλγόρθμο, καθώς δεν θα συνεχίζει προς μια κατεύθυνση με το που δει το πρώτο miss. Επειδή η ακριβής μοντελοποίηση του ανθρώπου κρίθηκε εξεραιτικά πολύπλοκη και πέραν των αναγκάιων απαιτήσεων, για να αντισταθμιστεί αυτό το μειωνέκτημα μετά την δεύτερη επιτυχή βολή ο υπολογιστής ενημερώνεται για το μήκους του πλοίου που βαρά.

Αν και όχι βέλτιστη λύση είναι κατά πολύ αποτελεσματικότερη της τυχαίας επιλογής στόχων

Τέλος, θεωρούμε πως κάθε παίκτης έχει στη διαθέση του μέχρι 40 βολές. Το παιχνίδι ολοκληρώνεται είτε όταν έχουν βυθιστεί όλα τα πλοία ενός παίκτη είτε όταν έχουν συμπληρωθεί όλες οι διαθέσιμες βολές για τους παίκτες. Στη πρώτη περίπτωση, νικητής είναι ο παίκτης που κατάφερε να βυθίσει όλα τα πλοία του αντιπάλου. Στη δεύτερη περίπτωση, νικητής είναι ο παίκτης που έχει συγκεντρώσιε την μεγαλύτερη βαθμολογία.

Υλοποιήθηκε.

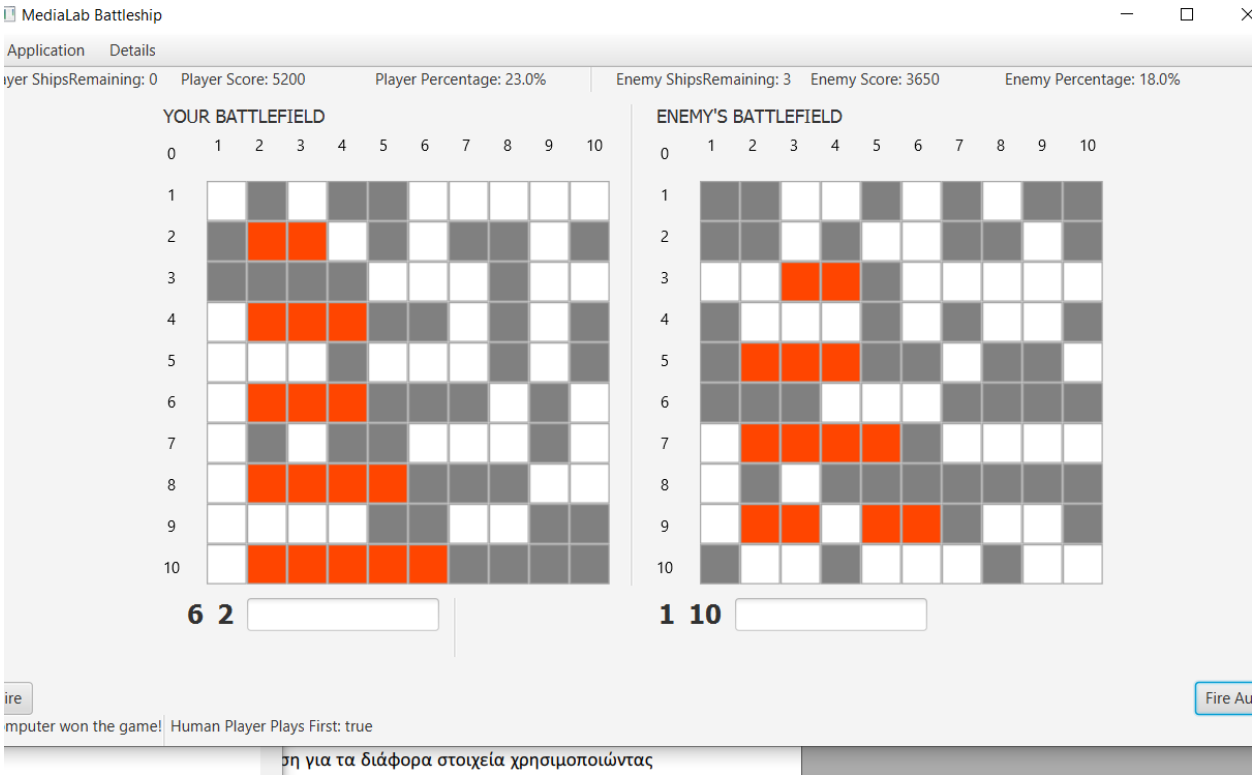
ΠΑΡΑΔΟΧΗ 5: Η ισοπαλία πόντων οδηγεί σε ήττα του χρήστη.

Β.2. Δημιουργία γραφικής διεπαφής (25%)

Θα πρέπει να σχεδιάσετε και να υλοποιήσετε την κατάλληλη γραφική διεπαφή χρήστη (Graphical User Interface - GUI) χρησιμοποιώντας το πλαίσιο JavaFX [1][2].

Σημείωση: Στη συνέχεια παρουσιάζονται οι βασικές προδιαγραφές για την γραφική διεπαφή, για όλες τις λεπτομέρειες της τελικής υλοποίησης μπορείτε να κάνετε όποιες επιλογές θέλετε σχετικά με την εμφάνιση και τη γενικότερη αλληλεπίδραση του χρήστη με την εφαρμογή, χωρίς καμία επίπτωση στην τελική βαθμολογία. Για παράδειγμα, μπορείτε να επιλέξετε μια απλή απεικόνιση για τα διάφορα στοιχεία χρησιμοποιώντας διαφορετικά χρώματα και κείμενο ή να συνδυάσετε εικόνες με διάφορα χαρακτηριστικά από το JavaFX ώστε να δημιουργήσετε ένα αποτέλεσμα που να αντιστοιχεί σε μια σύγχρονη εφαρμογή. Σε κάθε περίπτωση, δεν υπάρχει λόγος να κάνετε υπερβολικά πολύπλοκο το συγκεκριμένο μέρος της εργασίας.

Παρακάτω φαίνεται το αποτέλεσμα τις διεπαφής κατά την διάρκεια ενός παιχνιδιού(χωρίς το όριο των 40 γύρων).



Για την δημιουργία του γραφικού περιβάλλοντος της εφαρμογής θα πρέπει να ακολουθήσετε τις παρακάτω γενικές οδηγίες:

1. Δημιουργήστε το κεντρικό «παράθυρο» της εφαρμογής με τίτλο “MediaLab Battleship” και ορίστε τις κατάλληλες διαστάσεις.

Υλοποιήθηκε.

2. Χωρίστε το παράθυρο σε τρία βασικά μέρη.

Υλοποιήθηκε.

ΠΑΡΑΔΟΧΗ 6: Το ζητούμενο έγινε μέσω VBox στο άνω μέρος Boarder Pane και όχι με χρήση των θέσεων του Boarder Pane, τόσο για αισθητικούς όσο και για προγραμματιστικούς λόγους.

3. Στο πάνω μέρος της οθόνης θα εμφανίζονται συγκεντρωτικές πληροφορίες:

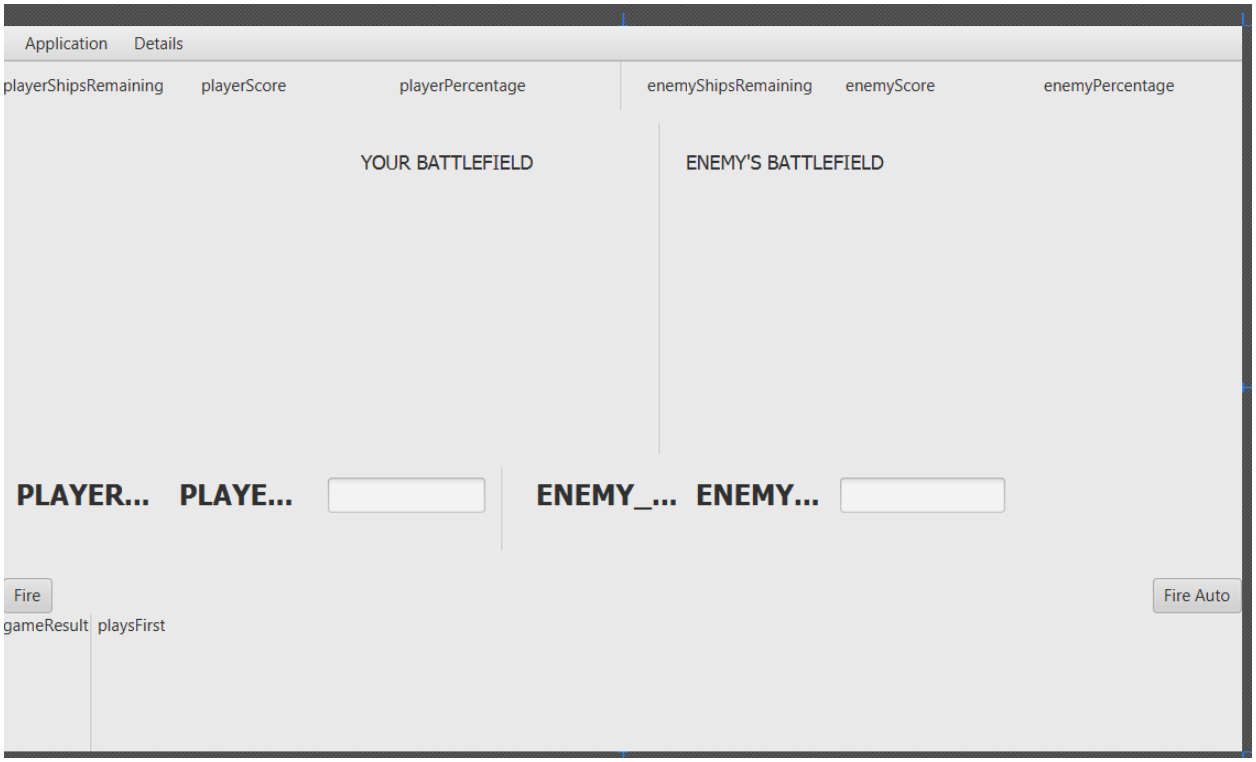
ΠΑΡΑΔΟΧΗ 7: Έχουμε 6 κελιά 3 ανά παίχτης από αριστερά προς δεξιά πρώτα του χρήστη και έπειτα του υπολογιστή.

a. Πλήθος ενεργών πλοίων κάθε παίκτη

Υλοποιήθηκε.

b. Συνολικοί πόντοι κάθε χρήστη
Υλοποιήθηκε.

c. Ποσοστό επιτυχών βολών κάθε χρήστη
Υλοποιήθηκε.



4. Στο μεσαίο τμήμα που θα αποτελείται από δύο επιμέρους τμήματα:
a. Το πρώτο, που θα βρίσκεται αριστερά, θα αντιστοιχεί στη γραφική παρουσίαση του ταμπλό του παίκτη που καταγράφονται οι θέσεις που βρίσκονται τα πλοία του και οι θέσεις που έχει κάνει βολές ο αντίπαλος. Θα πρέπει να περιλαμβάνει την αρίθμηση των γραμμών και στηλών και να παρουσιάζει με ευδιάκριτο τρόπο τα σημεία που έχει κάνει βολές ο αντίπαλος και το αποτέλεσμα της κάθε βολής. Υπάρχει πλήρης ελευθερία για τον τρόπο παρουσίασης των συγκεκριμένων πληροφοριών καθώς και πιθανών επιπρόσθετων πληροφοριών της επιλογής σας. Τα περιεχόμενα θα πρέπει να ενημερώνονται αυτόματα μετά από κάθε βολή του αντιπάλου και ανάλογα με το αποτέλεσμα της.
Υλοποιήθηκε.

b. Το δεύτερο, που θα βρίσκεται δεξιά, θα αντιστοιχεί στη γραφική παρουσίαση του ταμπλό του παίκτη που καταγράφει τις βολές του. Θα πρέπει να περιλαμβάνει την αρίθμηση των γραμμών και στηλών και να απεικονίζει με ευδιάκριτο τρόπο τα σημεία που ο παίκτης έχει επιχειρήσει βολές και το αποτέλεσμα της κάθε βολής. Υπάρχει πλήρης ελευθερία για τον τρόπο παρουσίας των πληροφοριών. Τα περιεχόμενα θα πρέπει να ενημερώνονται αυτόματα μετά από κάθε βολή του χρήστη και ανάλογα με το αποτέλεσμα της.
Υλοποιήθηκε.

5. Στο κάτω μέρος της οθόνης:
a. Θα υπάρχει η κατάλληλη φόρμα που θα επιτρέπει στον παίκτη να ορίζει το στόχο για την επόμενη βολή του και να την εκτελεί. Μετά την εκτέλεση της βολής θα πρέπει να ανανεώνονται τα κατάλληλα πεδία της γραφικής διεπαφής.
Υλοποιήθηκε.

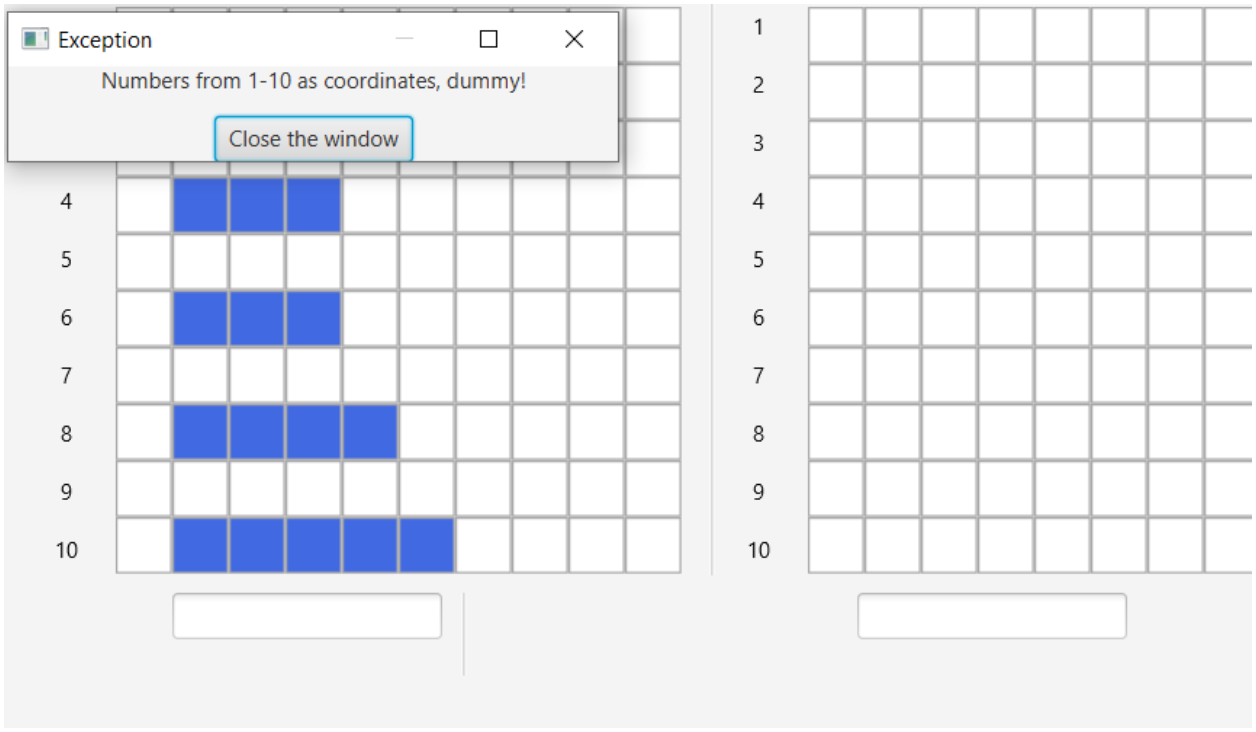
2 BUTTONS: Πέρα από το κουμπί Fire που εκτελεί την βολή που ο παίχτης έχει προκαθορίσει με κυτία υλοποιήθηκε και ένα κουμπί το FireAuto που επιλέγει τυχαία τις συντεταγμένες βολής του παίχτης. Το Fire δεν λειτουγεί αν δοθούν έγκυρες σε μορφή συντεταγμένες που έχουν ήδη χτυπηθεί. Επίσης δέχεται μη ακέραιο αριθμό, τον μετρέπει σε ακέραιο απευθείας.

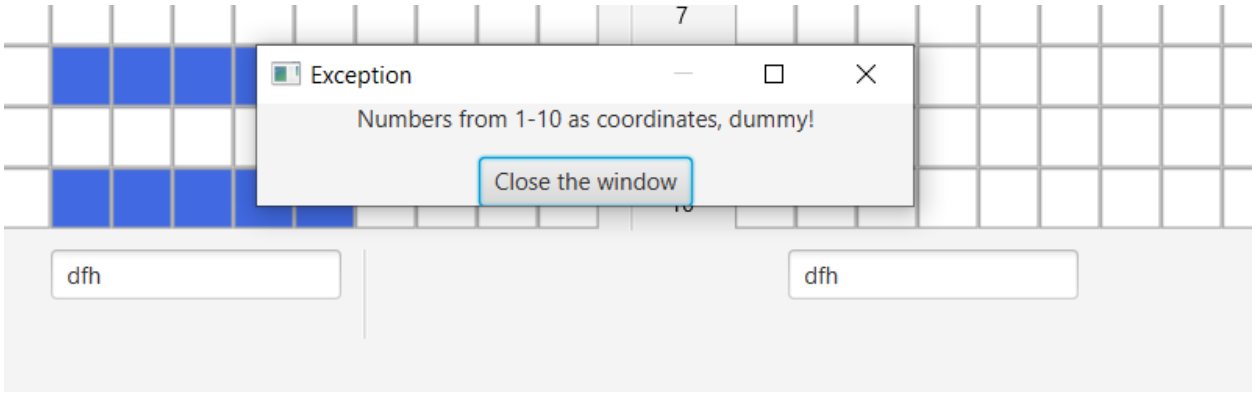
Το FireAuto κάνει τυχαίες επιλογές μέχρι να εντοπίση την πρόώτη που δεν έχει χτυπηθεί.

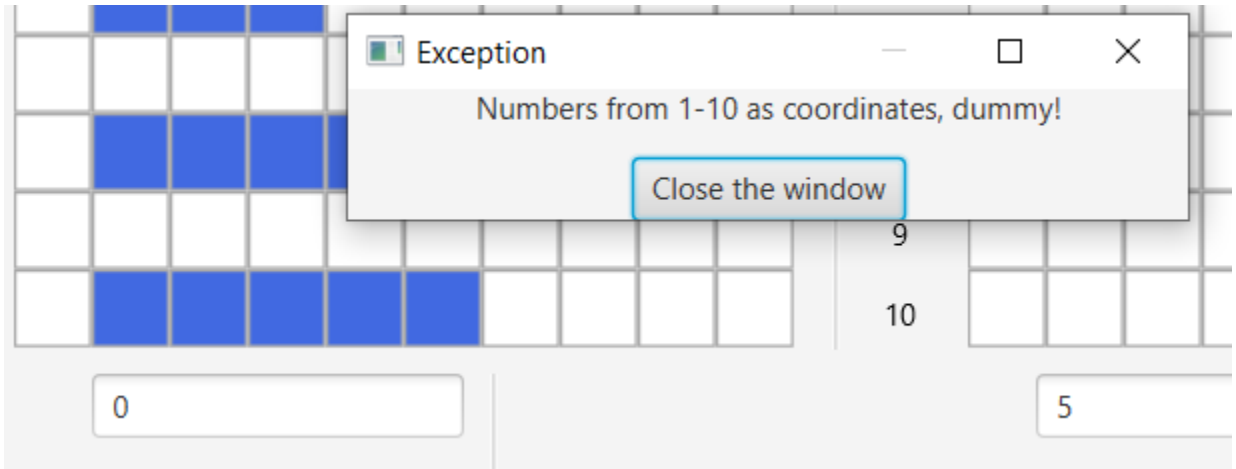
EXCEPTION HANDLING:

Έχουμε τρεις περιπτώεις, αυτή κενών κυτίων πρώτη σε προτεραιώτητα διαχείρησης.

Έπειτα αυτή τον κυτίων με μη αριθμιτικό περιεχόμενο. Τέλος αυτή με ακεραίους που αντιπροσωπεύουν στόχους εκτός ορίων του ταμπλό.


Exception
Numbers from 1-10 as coordinates, dummy!
Close the window
1
2
3
4
5
6
7
8
9
10


Exception
Numbers from 1-10 as coordinates, dummy!
Close the window
dfh
dfh

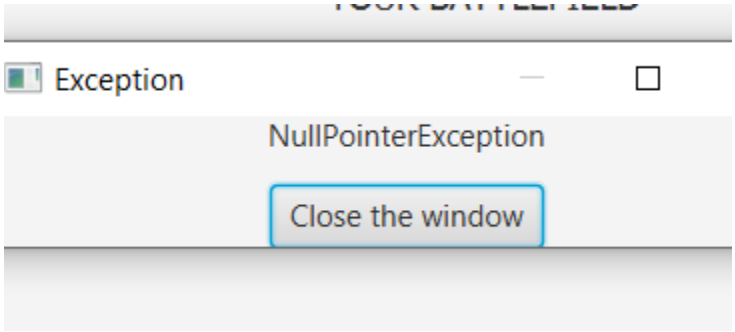

Exception
Numbers from 1-10 as coordinates, dummy!
Close the window
9
10
0
5

6. Προσθέστε ένα menu bar που θα περιλαμβάνει:
a. Menu “Application” με επιλογές:
i. Start: Έναρξη νέας εκτέλεσης με βάση το επιλεγμένο σενάριο. Αν υπάρχει κάποια ενεργή εκτέλεση θα πρέπει να διακόπτεται και να αρχίζει μια νέα με κατάλληλη αρχικοποίηση όλων των παραμέτρων και των πληροφοριών που υπάρχουν στην γραφική διεπαφή.
Υλοποιήθηκε.

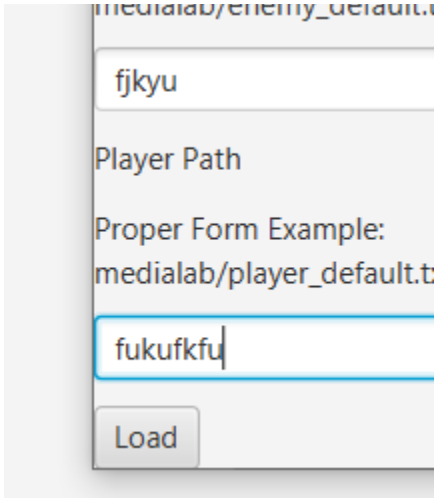
ii. Load: Μέσω ενός popup παραθύρου ο χρήστης θα προσδιορίζει το “SCENARIO-ID”. Στη συνέχεια η εφαρμογή θα αναζητά τα αντίστοιχα αρχεία (player_SCENARIO-ID.txt, enemy_SCENARIO-ID.txt). Αν οι περιγραφές είναι σωστές (δεν προκύπτουν εξαιρέσεις) θα γίνεται η κατάλληλη αρχικοποίηση της εφαρμογής, διαφορετικά θα εμφανίζεται το κατάλληλο μήνυμα σφάλματος.
Υλοποιήθηκε.

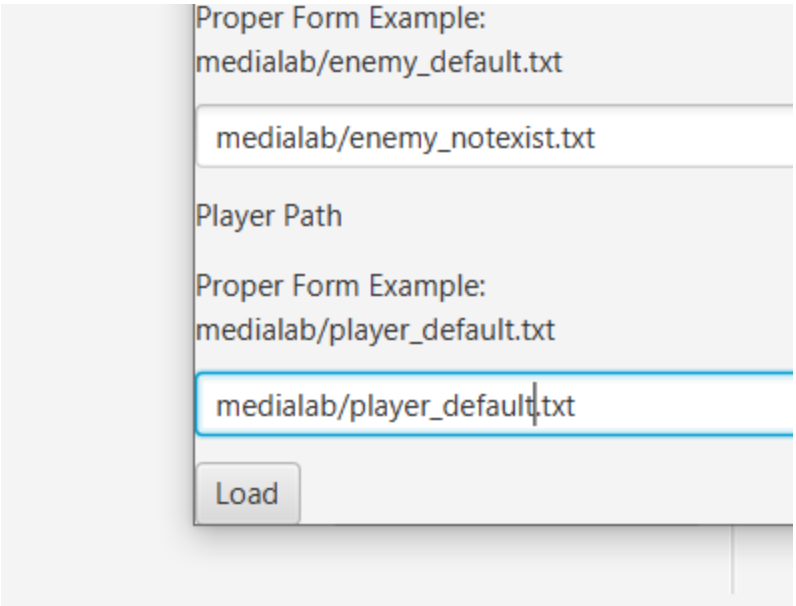
LOAD EXCEPTION HANDLING

Έστω ότι αφήσουμε κενά τα κελιά και πατηθεί το X του παραθύρου. Τότε θα εμφανιστεί:

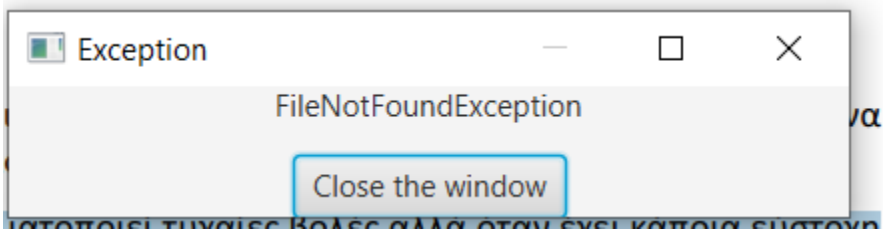


Έστω ότι έχουμε τις 2 μη έγκαιρες αποδόσεις path name.



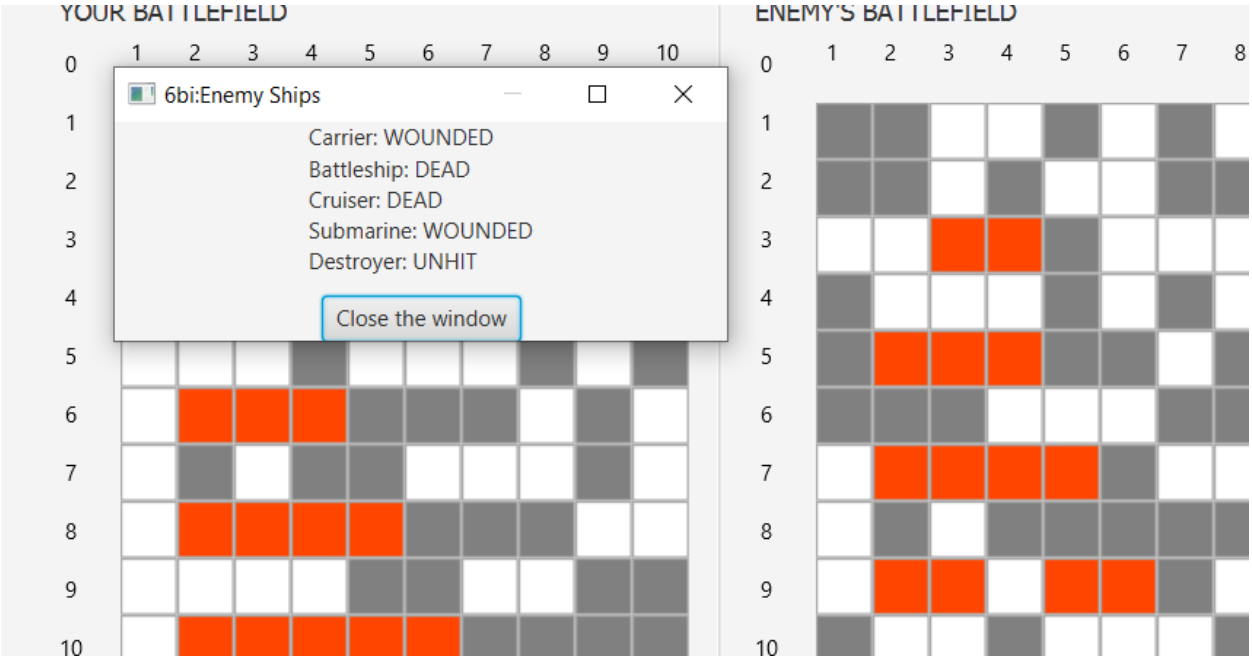


Τότε θα εμφανιστεί:



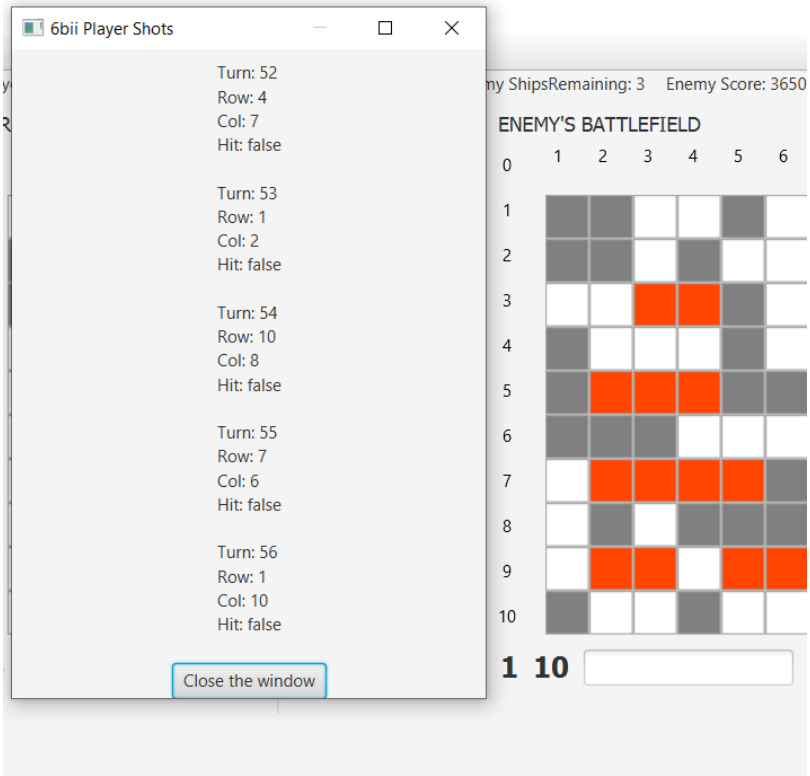
iii. Exit: Τερματισμός εφαρμογής.
Υλοποιήθηκε.

b. Menu “Details” με επιλογές:
i. Enemy Ships: Μέσω ενός popup παραθύρου θα παρουσιάζει την κατάσταση για όλα τα πλοία του αντιπάλου.
Υλοποιήθηκε.

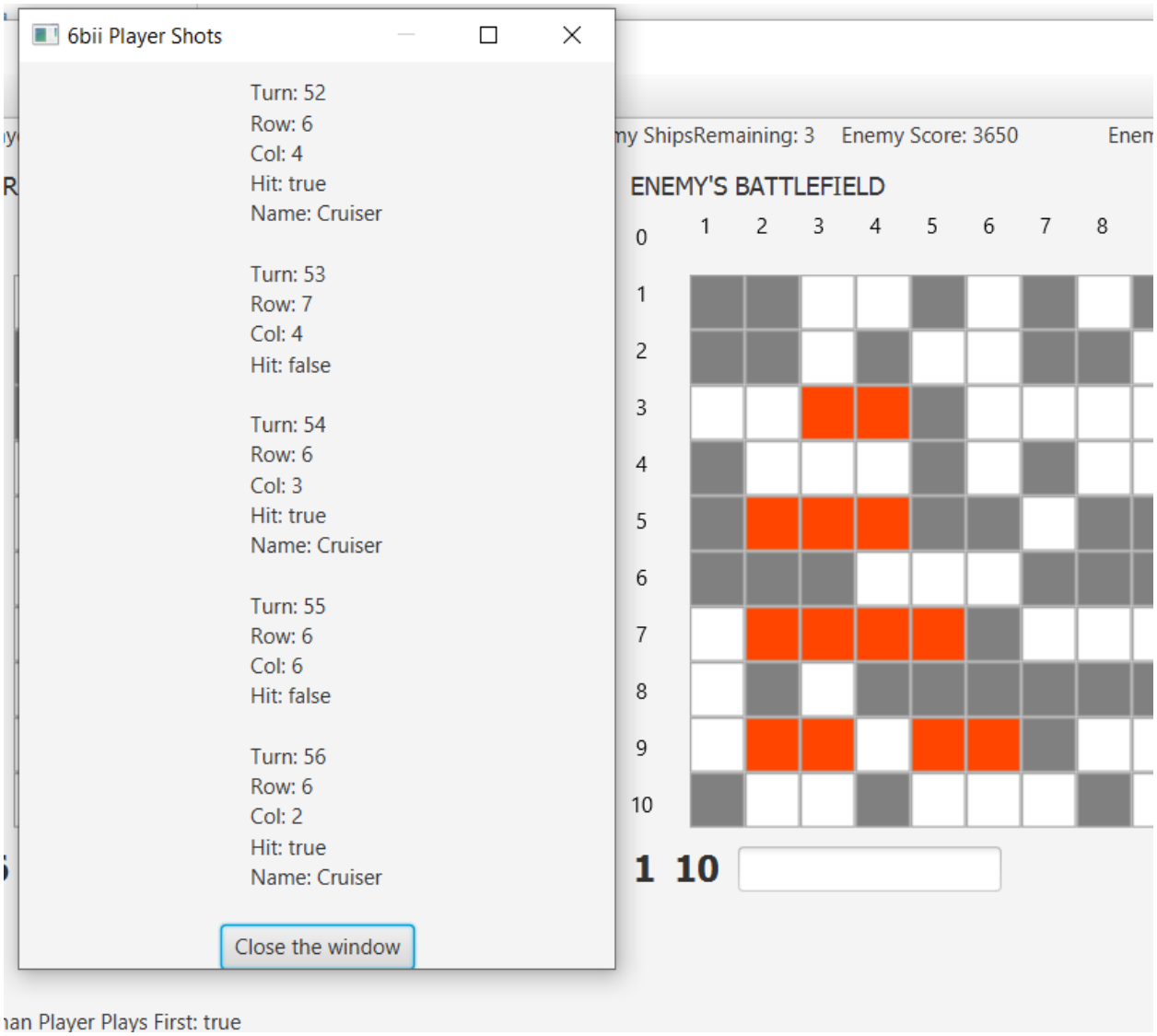


ii. Player Shots: Μέσω ενός popup παραθύρου θα παρουσιάζει για τις 5 τελευταίες βολές του χρήση τις παρακάτω πληροφορίες: συντεταγμένες στόχου, αποτέλεσμα βολής και τύπο πλοίου σε περίπτωση εύστοχης βολής.

Υλοποιήθηκε.



iii. Enemy Shots: Μέσω ενός popup παραθύρου θα παρουσιάζει για τις 5 τελευταίες βολές του αντιπάλου τις παρακάτω πληροφορίες: συντεταγμένες στόχου, αποτέλεσμα βολής και τύπο πλοίου σε περίπτωση εύστοχης βολής.
Υλοποιήθηκε.



Β.3. Λοιπές απαιτήσεις (20%)

• Η υλοποίηση θα πρέπει να ακολουθεί τις αρχές σχεδίασης του αντικειμενοστραφούς προγραμματισμού (OOP design principles).

Υλοποιήθηκε. Το πρόβλημα επιλύθηκε από ένα σύνολο κλάσεων με συγκεκριμένη ιεραρχία. Το Public χρησιμοποιήθηκε το κατά δυνοτόν λιγότερο. Ορίστικαν όπου χρειάστηκε Getters and Setters. Χρησιμοποι΄θηκε εμφολευμένη κλάση. Έγινε χρήση του static σε μεθόδους της κλάσης PopUpWindow ώστε να μην χρειάζεται κατασκευή αντικειμένου. Το παράδειγμα των Boards δείχνει την χρήση έξυπνης απογυγής κατασκυής περιττών ανικειμένων. Η αρχικοποίηση το διαφόρων πινάκων γίνεται με τρόπο ώστε να μην παράγονται παρά μόνο τα πλέον αναγκαία αντικείμενα. Οι τοπικές μεταβλητές των μεθόδων δεν ορίζονται ούτε ως Private ούτε ως public. Σχετικά παραδείγματα υπάρχουν διάσπαρτα σε όλον τον κώδικα.

;

• Σε μια κλάση της επιλογής σας θα πρέπει κάθε public μέθοδος που περιέχει να είναι τεκμηριωμένη σύμφωνα με τις προδιαγραφές του εργαλείου javadoc [3].

Τεκμηριώθηκε πλήρως η κλάσης Battleship , και μόνο το όνομα του συγγραφέα στις υπόλοιπες.

Τα παρακάτω είναι τα VM arguments όπως χρησιμοποιήθηκαν στο Eclipse.

--module-path C:/Users/Thanasis/Documents/COMPUTER_SCIENSE/Java/javafx-sdk-11.0.2/lib --add-modules javafx.controls,javafx.fxml