

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΕΧΝΟΛΟΓΙΑ ΠΟΛΥΜΕΣΩΝ

Battleship Game

Παπαγεωργίου Νικολέττα Ευσταθία

A.M.: 03116648

Εισαγωγή

Για την υλοποίηση της παρούσας εργασίας έγινε χρήση του eclipse , jdk 13 , javafx 11. Για να τρέξουν ίσως χρειαστεί η παρακάτω εντολή για το VM:

```
--module-path      "path-to-javafx\javafx-sdk-11.0.2\lib"      --add-modules  
javafx.controls,javafx.fxml
```

Διευκρινίσεις

Το παιχνίδι υλοποιήθηκε σύμφωνα με την εκφώνηση που μας δόθηκε.Πιο συγκεκριμένα:

Αρχικά το παιχνίδι είναι “κλειδωμένο” (ακόμα και να βάλει συντεταγμένες και να πατήσει το κουμπί “Shoot” δε θα συμβεί τίποτα.) παρόλο που εμφανίζει δύο μπλε αριθμημένους πίνακες και περιμένει από τον παίκτη να πατήσει στο μενού το “application” και έπειτα ένα από τα “start”, “load”:

- “Start”: εμφανίζει ένα παράθυρο ειδοποίησης ότι το παιχνίδι θα ξεκινήσει ξανά φορτώνοντας τα scenario-id που είχε από τον προηγούμενο γύρο. Στην περίπτωση που είναι πρώτη φορά που παίζει ο χρήστης και δεν έχει επιλέξει scenario-id έχει ως default να πάρει το scenario-id =0.
- “Load”: Εμφανίζει ένα pop up παράθυρο και περιμένει από τον χρήστη να πληκτρολογήσει το νούμερο του σεναρίου που θέλει να παίζει.(θεωρούμε ότι ο χρήστης θα πατήσει μόνο αριθμό). Στην περίπτωση που πατήσει αριθμό και δεν υπάρχει το αντίστοιχο σενάριο μέσα στον φάκελο medialab θα βγάλει pop up window με την αντίστοιχη ένδειξη(FileNotFoundException) και θα παραμείνει κλειδωμένο όπως ήταν και στην αρχή.
- “Exit” : Τερματίζει την εφαρμογή.

Φορτώνεται το αντίστοιχο scenario-id και γίνονται οι απαραίτητοι έλεγχοι ότι τα στοιχεία που δόθηκαν είναι τα σωστά. Σημειώνεται εδώ ότι θεωρούμε ότι ο χρήστης θα δώσει κάθε φορά 5 πλοία και στην περίπτωση που δώσει παραπάνω απλώς δεν τα λαμβάνουμε υπόψη. Διαβάζονται τα αντίστοιχα αρχεία και ανάλογα το πρόβλημα που θα υπάρξει πετάει ένα από τα παρακάτω:

- **OversizeException**:Ένα πλοίο δεν μπορεί να βγαίνει εκτός των ορίων του ταμπλό.

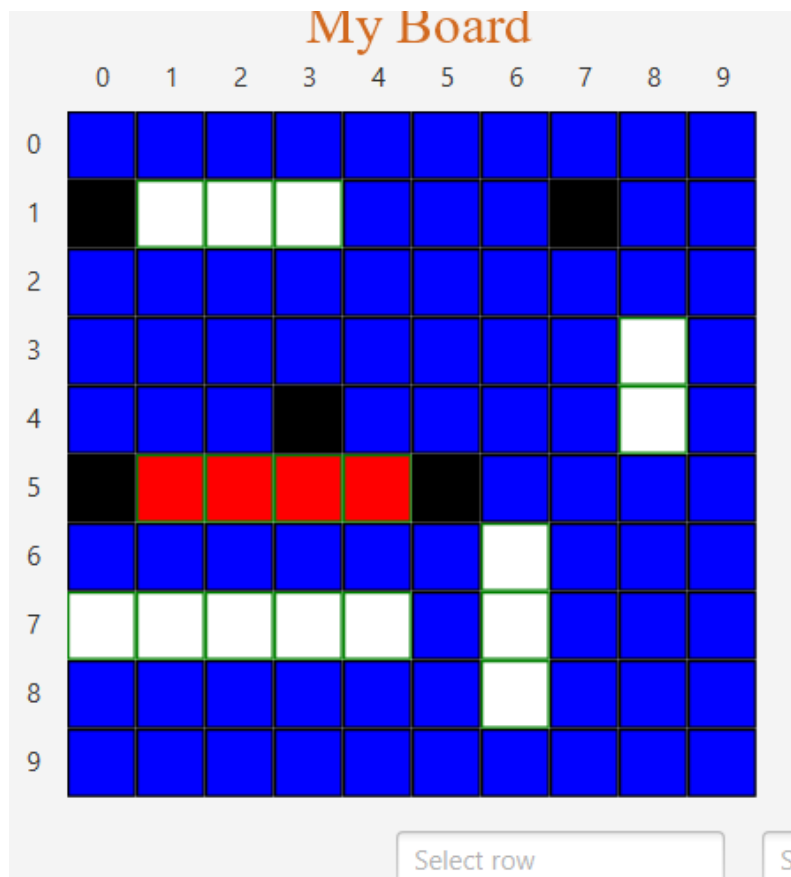
- `OverlapTilesException`: Ένα πλοίο δεν μπορεί να τοποθετηθεί σε κελί που ήδη έχει άλλο πλοίο.
- `AdjacentTilesException`: Ένα πλοίο δεν μπορεί να εφάπτεται κάθετα ή οριζόντια με κανένα άλλο πλοίο, έστω και για ένα κελί. Αυτό σημαίνει ότι τα πλοία έχουν μεταξύ τους απόσταση τουλάχιστον ένα ελεύθερο κελί.
- `InvalidCountException`: Δεν μπορούν να υπάρχουν περισσότερα από ένα πλοία για κάθε τύπο
- `InvalidException`: Ο χρήστης έδωσε για τύπο πλοίου τιμή που δεν ανήκει στο [1,2,...,5].

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 και γνωστοποιείται στον παίκτη μέσω ενός pop up παραθύρου.

Ο υπολογιστής παίζει σύμφωνα με αλγόριθμο : μέχρι να πετύχει κάποιο πλοίο επιλέγει τυχαία τους στόχους του. Απο τη στιγμή που πετύχει κάποιο πλοίο προσπαθεί να βρει την διεύθυνση του και εφόσον τη βρεί χτυπάει και προς τις δύο πλευρές(αν απαιτείται) μέχρι να το βυθίσει. Η διαδικασία είναι η ίδια κάθε φορά. Αφού χτυπήσει τυχαία ένα πλοίο μετά έχει σαν πιθανούς στόχους τους γείτονες του. Χωρίς να μας επηρεάζει κάτι παίρνουμε ότι ο υπολογιστής εξετάζει πάντα με την ίδια σειρά τους γείτονες.(πανω,δεξιά, κάτω, αριστερά). Ελέγχει αυτούς τους 4 γείτονες και ανάλογα με το ποιος από αυτούς κρύβει το πλοίο πράττει αντίστοιχα χωρίς να χρειάζεται να ψαξει ξανά όλους τους γείτονες του γείτονα αφού πια γνωρίζει τη διεύθυνση. (Σημειώνουμε ότι επειδή πάντα στο πρώτο χτυπημένο κελί κοιτάμε με τη σειρά πανω, δεξιά, κάτω, αριστερά αν φτάσουμε στο κάτω ή στο αριστερά δεν υπάρχει λόγος να ελέγξουμε και τις αντίθετες πλευρες για πλοία αφού αυτές έχουν ήδη ελεγχθεί.) πχ:

Στο παρακάτω σχήμα ο αλγόριθμος χτύπησε το κελί

(5,3) ->(4,3)->(5,4)->(5,5)->(5,2)->(5,1)->(5,0)



Ο χρήστης πληκτρολογεί στα κουτάκια που βρίσκονται κάτω από τους πίνακες τις συντεταγμένες του κελιού που θέλει να χτυπήσει. (Θεωρούμε ότι ο χρήστης πληκτρολογεί αριθμούς 0-9 και όχι κάποιον μεγαλύτερο ή κάποιο γράμμα). Αν ο χρήστης πληκτρολογήσει ξανά τα ίδια νούμερα δε γίνεται κάποια αλλαγή και περιμένουμε από αυτόν να πληκτρολογήσει νέα.

Σημειώνουμε ότι με :

- λευκό: απεικονίζονται τα πλοία μας.
- μαύρο: οι βολές που δε βρήκαν κάποιο στόχο.
- κόκκινο: οι βολές που βρήκαν σε πλοίο.

Το παιχνίδι ολοκληρώνεται είτε όταν έχουν βυθιστεί όλα τα πλοία ενός παίκτη είτε όταν έχουν συμπληρωθεί όλες οι διαθέσιμες βολές (40) 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 συγκεντρώσει την μεγαλύτερη βαθμολογία. Στην περίπτωση που τα σκόρ είναι τα ίδια έχουμε ισοβαθμία.

Επιπλέον στο μενού υπάρχουν οι επιλογές:

- Enemy Ships: Μέσω ενός popup παραθύρου παρουσιάζει την κατάσταση για όλα τα πλοία του αντιπάλου.
- Player Shots: Μέσω ενός popup παραθύρου παρουσιάζει τις 5 τελευταίες βολές του χρήστη τις παρακάτω πληροφορίες: συντεταγμένες στόχου, αποτέλεσμα βολής και τύπο πλοίου σε περίπτωση εύστοχης βολής.
- Enemy Shots: Μέσω ενός popup παραθύρου παρουσιάζει τις 5 τελευταίες βολές του αντιπάλου τις παρακάτω πληροφορίες: συντεταγμένες στόχου, αποτέλεσμα βολής και τύπο πλοίου σε περίπτωση εύστοχης βολής.

Σημειώνουμε ότι στα Enemy-Player Shots οι νεότερες βολές εμφανίζονται ψηλότερα.

Τέλος καθόλη τη διάρκεια του παιχνιδιού εμφανίζονται στατιστικά στοιχεία που ανανεώνονται αυτόματα μετά από κάθε βολή και έχει:

- Πλήθος ενεργών πλοίων κάθε παίκτη
- Συνολικοί πόντοι κάθε χρήστη
- Ποσοστό επιτυχών βολών κάθε χρήστη

Σημειώνεται ότι η υλοποίηση ακολουθεί τις αρχές σχεδίασης του αντικειμενοστραφούς προγραμματισμού (OOP design principles).

Η κλάση που τεκμηριώθηκε σύμφωνα με τις προδιαγραφές του εργαλείου javadoc είναι η Shots.