



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ

ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνολογία Πολυμέσων

Τελική εργασία

Δημήτριος Γεωργούσης, 03119005

Η εργασία έγινε με χρήση των παρακάτω εργαλείων και συστήματος:

- Windows 11 Pro (64 bit)
- IntelliJ IDEA 2022.3.3 (Ultimate Edition) Build #IU-223.8836.41, built on March 10, 2023
- Java SDK: 19 java version "19.0.2"
- Η JavaFX για το project είναι ό,τι χρησιμοποίησε το Maven.
- Η JavaFX που χρησιμοποίησα για το JavaDoc είναι javafx-sdk-19.0.2.1

Σχεδιασμός και υλοποίηση λογικής:

Αρχεία περιγραφής: έχουν τη δομή που περιγράφεται στην εκφώνηση. Ο τύπος **InvalidDescriptionException** χρησιμοποιείται αν δεν υπάρχουν οι 4 γραμμές που θέλουμε, ενώ ο τύπος **InvalidValueException** χρησιμοποιείται αν οι τιμές κάποιων από τις γραμμές δεν είναι αποδεκτές (π.χ. δεν είναι αριθμοί ή αριθμοί εκτός των αποδεκτών ορίων).

Αρχικοποίηση: όπως στην εκφώνηση με μία διαφορά. Αν η πρώτη ενέργεια αποκάλυψης τετραγώνων του χρήστη είναι αριστερό κλικ πάνω σε νάρκη, τότε οι νάρκες ξανά – ανακατεύονται στο πλέγμα, ξαναγράφεται το “mines.txt” και υπάρχει εγγύηση ότι το τετράγωνο που πατήθηκε δεν θα είναι πάλι νάρκη (ούτε τα γειτονικά του). Έτσι ο παίκτης δεν μπορεί να χάσει από τη πρώτη του προσπάθεια. Αν ο παίκτης είχε μαρκάρει πριν το ξανα – ανακάτεμα το τετράγωνο στο οποίο βρέθηκε η υπερνάρκη μετά από το ανακάτεμα, τότε η λειτουργία της δεν θα ενεργοποιηθεί (θα πρέπει να αφαιρέσει το μαρκάρισμα και να το ξαναβάλει), αν και είναι αρκετά άσκοπο παίκτης να παίζει με αυτόν τον τρόπο.

Αποκάλυψη τετραγώνου: Έχει υλοποιηθεί η κλασσική αναδρομή του minesweeper. Ένα τετράγωνο αποκαλύπτει τους γείτονές του μόνο αν κανένας τους δεν είναι νάρκη. Επίσης, ένας γείτονας αποκαλύπτει τους δικούς του γείτονες μόνο αν κανένας από αυτούς δεν είναι νάρκη, αλλιώς δεν αποκαλύπτει κανέναν. Η αναδρομή έχει υλοποιηθεί με τρόπο τέτοιο ώστε να μπορεί να «υπερπηδήσει» τετράγωνο που έχουν αποκαλυφθεί από την λειτουργία της υπερνάρκης.

Μαρκάρισμα νάρκης: υπάρχουν μερικές διαφορές με την εκφώνηση. Ο παίκτης δεν μπορεί να αποκαλύψει με αριστερό κλικ μαρκαρισμένο τετράγωνο, αν θέλει να το αποκαλύψει πρέπει πρώτα να αφαιρέσει το μαρκάρισμα, έτσι, προστατεύεται από το να χάσει κατά λάθος πατώντας τετράγωνο που θεωρεί ότι είναι νάρκη. Αν έχουμε αποκαλύψει νάρκες (π.χ. μέσω της υπερνάρκης) αυτές «καταναλώνουν» μία από τις διαθέσιμες σημαίες.

Ολοκλήρωση παιχνιδιού: όπως στην εκφώνηση. Στο τέλος αποκαλύπτονται πάντα οι νάρκες χρωματισμένες κατάλληλα σε περίπτωση νίκης ή ήττας.

Δημιουργία γραφικής διεπαφής: Οι οδηγίες 1 – 5 έχουν ακολουθηθεί. Για την οδηγία 6 έχουμε:

Menu “Applications”:

- i. **Create:** όπως περιγράφεται στην εκφώνηση, ο χρήστης δίνει κείμενο στα αντίστοιχα πεδία
- ii. **Load:** όπως περιγράφεται στην εκφώνηση, δηλαδή, αν ο χρήστης αναζητά το “test.txt” θα δώσει στο πεδίο του αναδυνόμενου παραθύρου το “test”. Αν η περιγραφή είναι έγκυρη φορτώνεται σε μία μεταβλητή του προγράμματος που τη διατηρεί στη μνήμη. Δεν επηρεάζει το τρέχον παιχνίδι.
- iii. **Start:** Αν υπάρχει τρέχον παιχνίδι το διακόπτει και ξεκινά νέο με βάση την φορτωμένη περιγραφή.
- iv. **Exit:** Αν υπάρχει τρέχον παιχνίδι το διακόπτει και τερματίζει την εφαρμογή.

Menu “Details”:

- i. **Rounds:** Παρουσιάζει τις πληροφορίες για τα 5 τελευταία παιχνίδια σε έναν πίνακα. Το πιο πρόσφατο βρίσκεται πάνω πάνω.
- ii. **Solution:** Εκτελεί ενέργειες μόνο αν υπάρχει τρέχον παιχνίδι το οποίο ακόμα δεν έχει λήξει. Δείχνει στον παίκτη τις θέσεις των ναρκών και καταχωρεί το παιχνίδι ως ήττα του παίκτη.

Χρησιμοποιείται μια τριγωνική εικόνα για τη σημαία με μπλε φόντο. Μία διαφανής νάρκη για την απλή νάρκη και μία μαύρη νάρκη για την υπερνάρκη. Το κόκκινο φόντο δηλώνει την επισφαλή αποκάλυψη νάρκης (που οδηγεί σε ήττα του παίκτη). Αν δεν φορτωθούν οι εικόνες τότε ‘F’ είναι η σημαία, ‘X’ είναι η νάρκη και ‘H’ είναι η υπερνάρκη.

Οποιαδήποτε είσοδος/έξοδος από/σε αρχεία αλλά και αρχικοποίηση (π.χ. λειτουργία “Load”) οδηγεί σε σφάλματα εμφανίζοντας κατάλληλο μήνυμα στον χρήστη της εφαρμογής. Επίσης, εμφανίζονται και άλλα ενημερωτικά μηνύματα (π.χ. νίκη ή ήττα, επιτυχής ή ανεπιτυχής αποθήκευσης αποτελέσματος παιχνιδιού κ.λπ.).

Ο φάκελος “medialab” βρίσκεται στη διαδρομή “src/main/resources/medialab”.

Ενσωμάτωση αρχών σχεδίασης αντικειμενοστρεφούς προγραμματισμού:

Στο κομμάτι του παιχνιδιού υπάρχει διαχωρισμός των γραφικών στοιχείων με τα στοιχεία/κλάσεις που υλοποιούν την λογική του παιχνιδιού. Είσοδο, έξοδο και μηνύματα προς τον χρήστη αναλαμβάνουν να εμφανίσουν τα οι κλάσεις που υλοποιούν γραφικά.

Στο κομμάτι των μενού η υλοποίηση είναι πιο απλή, οπότε έχουν ενσωματωθεί τα περισσότερα σε μία κλάση. Στη λειτουργία της Load που υπάρχει μεγαλύτερη πολυπλοκότητα, η λογική των ελέγχων έχει διαχωριστεί.

Γενικά, έχω προσπαθήσει να ακολουθήσω την λογική ότι κάθε κλάση εκτελεί ενέργειες στον εαυτό της έχοντας όμως μερικές κλάσεις με παρενέργειες σε άλλες, όπου κρίνεται σκόπιμο (π.χ. στη κλάση Tile μέσα στην Game).

Τεκμηρίωση: Υπάρχει τεκμηρίωση της κλάσης Game και των public μεθόδων της στον φάκελο “documentation”. Τρέξτε το “index.html” που υπάρχει στον φάκελο και πλοηγηθείτε στο:

Module ntua.multimedia.minesweeper
Package ntua.multimedia.minesweeper.game

Οδηγίες:

Το project έχει δημιουργηθεί ως JavaFX project στο IntelliJ με την επιλογή Build System: Maven.

Όταν το ανοίξετε θα χρειαστεί να δώσετε κατάλληλη τιμή στη μεταβλητή “PATH_TO_FX”
File -> Settings -> Appearance & Behaviour -> αν δεν υπάρχει η “PATH_TO_FX” δημιουργήστε την με τιμή το μονοπάτι όπου έχετε αποθηκεύσει τον φάκελο “lib” του javafx-sdk σας. Π.χ. “C:\Program Files\JavaFX\javafx-sdk-19.0.2.1\lib”.

Στο Project Structure -> Libraries θα πρέπει στην JavaFXForJavaDoc να δώσετε το ίδιο μονοπάτι.

Αυτά χρειάστηκαν για να λειτουργήσει το JavaDoc εργαλείο σωστά.

Μιας και το project χρησιμοποιεί Maven dependencies μπορείτε να το τρέξετε κάνοντας “run MinesweeperApp” μέσα από το IntelliJ.

Τέλος, το project συμπεριλαμβάνεται ως jar στο κατάλληλο περιβάλλον για να μπορέσετε να το τρέξετε κατευθείαν (παράχθηκε jar artifact μέσω του IntelliJ). Κατευθυνθείτε στον φάκελο “out\artifacts\minesweeper.jar” και εκτελέστε σε τερματικό (π.χ. PowerShell) την εντολή:
java --module-path . --add-modules javafx.controls -jar minesweeper.jar

Στο PowerShell των Windows 11 μπορούμε να τρέξουμε την εντολή:
java --module-path \$Env:PATH_TO_FX --add-modules javafx.controls -jar minesweeper.jar

Όπου “PATH_TO_FX” είναι μεταβλητή περιβάλλοντος με την ίδια διαδρομή της “PATH_TO_FX” που αναφέραμε παραπάνω. Αν δεν τρέξει δοκιμάστε στα modules να φορτώσετε και άλλα εκτός του javafx.controls (π.χ. το javafx.base κ.λπ) αλλά σε εμένα λειτουργεί κανονικά και έτσι.

Πληροφορίες για τις εντολές αυτές: [Getting Started with JavaFX \(openjfx.io\)](https://openjfx.io/GettingStartedwithJavaFX)