

# MULTIMEDIA PROJECT 2023- MINESWEEPER

**Γεωργία Μπουσμπουκέα- el19059**

Η υλοποίηση της εφαρμογής έχει γίνει ως εξής:

- Η κλάση main, όπου ορίζεται το βασικό scene της εφαρμογής, game, στο οποίο έχουν εφαρμοστεί εντολές css, του αρχείου application.css. Εδώ ακόμη ορίζεται ένα BorderPane, root, που δημιουργείται από το menu.fxml, μέσω του SceneBuilder, και εμφανίζεται στο game.
- Η κλάση Tile, όπου ορίζονται τα χαρακτηριστικά και οι λειτουργίες ενός tile του ταμπλό, δηλαδή ενός τετραγώνου. Συγκεκριμένα, κάθε αντικείμενο αυτής της κλάσης κρατάει πληροφορία για τις σχετικές συντεταγμένες του (0,0 είναι το πάνω αριστερά κοκ), για το αν έχει νάρκη, υπερνάρκη, αν είναι ανοιχτό, αν είναι flagged και για το περιεχόμενό του(text) (X σε περίπτωση υπερνάρκης, αριθμός γειτόνων με νάρκη, σε περίπτωση που δεν έχει νάρκη, εικόνα σημαίας ή βομβας). Σε κάθε κλικ πάνω στο tile αυξάνεται κατά ένα το πλήθος των κινήσεων (το χρειάζομαι για τον έλεγχο για υπερνάρκη). Ανάλογα με το είδος του κλικ επιτελούνται διαφορετικές λειτουργίες. Συγκεκριμένα, αν είναι αριστερό κλικ (open()), αυξάνεται το πλήθος των επιτυχημένων προσπαθειών (το χρειάζομαι για το score), το τετράγωνο ανοίγει και γίνονται οι απαραίτητοι έλεγχοι για κάθε περίπτωση. Αν πρόκειται για δεξί κλικ (setFlag()), γίνονται πάλι οι διάφοροι έλεγχοι και μπαίνει σημαία. Αν το πλήθος των κινήσεων είναι μικρότερο ή ίσο του 4 καλείται η openSuper() για όλα τα tiles της ίδιας γραμμής και στήλης, που είναι ίδια με την open(), με την διαφορά ότι τώρα δεν γίνεται αναδρομική αποκάλυψη των τετραγώνων. Τέλος, αν προκύψει από κάποιον από τους παραπάνω ελέγχους ότι ο παίκτης κέρδισε ή έχασε καλείται η finish() με πρώτη παράμετρο true ή false αντίστοιχα και δεύτερη false (είναι true όταν ο χρήστης τελειώσει το παιχνίδι βλέποντας την λύση), η οποία σταματάει το χρονόμετρο, κάνει save το score με τα απαραίτητα στοιχεία, διαγράφει το mines.txt και εμφανίζεται το κατάλληλο μήνυμα.
- Η κλάση Minesweeper.java, όπου ορίζονται τα συνολικά χαρακτηριστικά ενός παιχνιδιού. Συγκεκριμένα ορίζεται το ταμπλό ως ένας 2D πίνακας τετραγώνων, grid[ ][ ], ορίζεται η συνάρτηση time(), που αποτελεί το χρονόμετρο η συνάρτηση createContent(), που αρχικοποιεί κατάλληλα το ταμπλό, ανάλογα με τις διαστάσεις του. Συγκεκριμένα, επιλέγονται τυχαία οι θέσεις των ναρκών και τίθεται κατάλληλα το text κάθε τετραγώνου (για τις νάρκες χρησιμοποιείται το unicode σύμβολο).
- Η κλάση InvalidDescriptionException, που ελέγχει αν δίνονται 4 ορίσματα (μη κενά).
- Η κλάση InvalidValueException, που ελέγχει αν τα ορίσματα είναι έγκυρα, σύμφωνα με το επίπεδο δυσκολίας του δεδομένου παιχνιδιού.
- Η κλάση Score, που αποθηκεύει σε έναν πίνακα 5 θέσεων τα ζητούμενα στοιχεία για τα 5 τελευταία παιχνίδια.
- Η κλάση SampleController, που περιέχει τις συναρτήσεις που καλούνται με το πάτημα κουμπιών της διεπαφής (ορισμένα στα fxml αρχεία).
  - Συγκεκριμένα, αν από το αρχικό menu της εφαρμογής επιλεγεί το “Create a new scenario”, καλείται η createWindow(), που δημιουργεί ένα pop-up window, μέσω του create.fxml. Σε αυτό το παράθυρο εισάγει ο χρήστης τα στοιχεία που επιθυμεί να έχει

το νέο σενάριο και, πατώντας create, καλείται η `createScenario()`, που αποθηκεύει το σενάριο στον φάκελο medialab.

- Αν από το αρχικό menu της εφαρμογής επιλεγεί το “Load a scenario”, καλείται η `loadWindow()`, η οποία δημιουργεί ένα pop-up window (πιο απλο, δεν χρησιμοποιώ το SceneBuilder). Ο χρήστης καλείται να διαλέξει ένα από τα υπάρχοντα σενάρια και ελέγχεται για `invalid value/ description exceptions`. Αν το αρχείο δεν πετάει κάποια εξαίρεση δηλώνεται ότι το `fileGame` είναι αυτό το αρχείο (`fileGame` είναι το αρχείο που θα περιγράφει το παιχνίδι που θα φορτωθεί την επόμενη φορά που θα ξεκινήσει νέο παιχνίδι).
- Αν από το αρχικό menu επιλεγεί το “Start playing” τότε αρχικοποιείται η εφαρμογή σύμφωνα με τις τιμές του `fileGame`, μέσω της συνάρτησης `launch()`. Ακόμη, ορίζονται οι ετικέτες που θα δείχνουν το πλήθος των ναρκών και των σημαιών και τον υπολειπόμενο χρόνο. Οι ετικέτες μαζί με το ταμπλό, που δημιουργείται με την κλήση της `createContent()` εμφανίζονται, θέτοντας το center του `BorderPane root`. Αν υπάρχει υπερνάρκη στην περιγραφή του παιχνιδιού, επιλέγεται τυχαία μία από τις νάρκες για υπερνάρκη και γράφονται όλες οι νάρκες στο `mines.txt`, που δημιουργείται. Ξεκινάει το χρονόμετρο.
- Αν από το αρχικό menu επιλεγεί το “Show score”, καλείται η `scoreWindow()`, η οποία δημιουργεί ένα pop-up window, και με χρήση ετικετών δείχνει τα δεδομένα που έχουν αποθηκευτεί στον πίνακα της Score.
- Αν επιλεγεί το “Show solution”, σταματάει το χρονόμετρο και ανοίγουν όλα τα τετράγωνα του grid. Καλείται η `Tile.finish()` με τις δύο παραμέτρους `false`, καταχωρώντας το παιχνίδι ως χαμένο.

Για διευκρίνιση: κάθε φορά που γίνεται start αρχικοποιείται το παιχνίδι σύμφωνα με το αρχείο που έγινε load την τελευταία φορά. Ο έλεγχος για τα exceptions γίνεται στο load. Γίνεται, δηλαδή, ενώ ο χρήστης παίζει έναν γύρο να κάνει load ένα άλλο αρχείο, χωρίς να αλλάξει κάτι στον γύρο του. Το αρχείο θα φορτωθεί όταν πατήσει start. Ακόμη, αν ολοκληρωθεί ο γύρος που παίζει ο χρήστης και πατήσει start χωρίς να κάνει κάποιο load εκ νέου, παίζει το ίδιο σενάριο με την τελευταία φορά (βέβαια, καθώς οι θέσεις των ναρκών επιλέγονται τυχαία, ουσιαστικά το παιχνίδι είναι διαφορετικό).

(Προσοχή: κάποιες φορές λαγκάρει το eclipse και το mines.txt δεν διαγράφεται)