ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΕΧΝΟΛΟΓΙΑ ΠΟΛΥΜΕΣΩΝ

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ

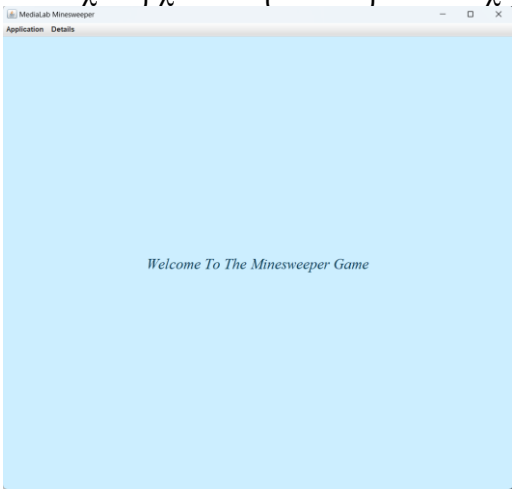
Ειρήνη Δόντη
ΑΜ: 03119839

7ο εξάμηνο

Αθήνα 2023

Περιγραφή Εφαρμογής

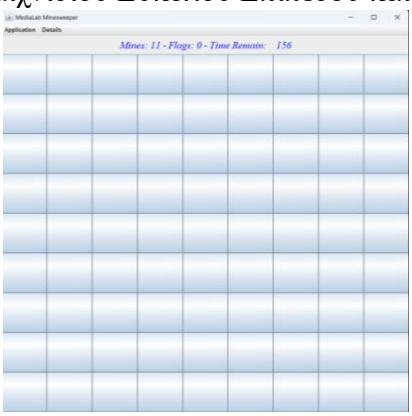
Στην ***κλάση InitWindow***, δημιουργούμε ένα παράθυρο το οποίο καλωσορίζει τον χρήστη στο παιχνίδι. Θεωρούμε ένα menubar, τοποθετημένο στο πάνω μέρος της οθόνης, στο οποίο υπάρχει το Menu "Application" με submenu τις ζητούμενες επιλογές "Create", "Load", "Start", "Exit" και το Menu "Details" με submenu τις ζητούμενες επιλογές "Rounds", "Solution". Τα παραπάνω Menu και submenu είναι υλοποιημένα με JMenu και JMenuItem αντίστοιχα και με τη βοήθεια της addActionListener(), όταν ο χρήστης επιλέγει το εκάστοτε MenuItem, εκτελούνται οι ζητούμενες ενέργειες. Στα MenuItem "Create", "Load", "Rounds" χρησιμοποιούνται Option Pane τα οποία δημιουργούν ένα pop up παράθυρο στο οποίο ο χρήστης μπορεί να συμπληρώσει με το πληκτρολόγιο τα ζητούμενα πεδία για να δημιουργήσει ένα αρχείο περιγραφής, να φορτώσει το αρχείο περιγραφής και να δει τις πέντε πρώτες νίκες του αντίστοιχα. Το MenuItem "Start" εκτελείται αν και μόνο αν έχει οριστεί έγκυρο αρχείο περιγραφής στο "Load". Αν πατηθεί το submenu "Exit", η εφαρμογή τερματίζεται ενώ όταν πατηθεί submenu "Solution", εμφανίζεται το μήνυμα "No Initialized Game", γιατί δεν έχει αρχικοποιηθεί το πρώτο παιχνίδι.

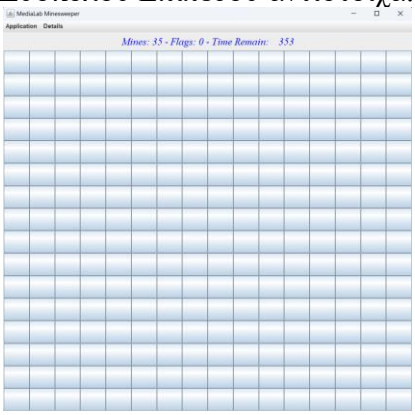


Το πρώτο παιχνίδι ξεκινά αφού φορτώσουμε ένα έγκυρο αρχείο περιγραφής και καλέσουμε την ***κλάση Game*** με την οποία αναθέτουμε, σε μεταβλητές, τα στοιχεία που δίνονται στο αρχείο SCENARIO-ID. Αν το αρχείο περιγραφής δεν πληροί τα ζητούμενα κριτήρια, προκύπτουν εξαιρέσεις. Αντίθετα, ορίζεται η διάσταση πλέγματος βάσει του επιπέδου δυσκολίας που καταχωρείται στο αρχείο (1: επίπεδο των αρχαρίων και 2: επίπεδο των προχωρημένων). Τέλος, καλείται η κλάση Window και αρχικοποιείται, ώστε να εμφανίσει το παράθυρο με το παιχνίδι (αν είναι έγκυρο). Στην ***κλάση FileService***, διαβάζουμε κάθε γραμμή του SCENARIO-ID.txt αρχείου και αναθέτουμε τις τιμές σε έναν πίνακα ακεραίων input[]. Αν λείπει κάποια γραμμή από το αρχείο, προκύπτει μία εξαίρεση τύπου InvalidDescriptionException (ορίζεται στην ***κλάση InvalidDescriptionException***) και τυπώνεται το μήνυμα "Wrong Description!". Επίσης, αν κάποια από τις γραμμές δεν περιέχουν τις κατάλληλες τιμές, όπως δίνονται στην εκφώνηση, τότε προκύπτει μία εξαίρεση τύπου InvalidValueException (ορίζεται στην ***κλάση InvalidValueException***)και τυπώνεται το μήνυμα "Wrong Value(s)!". Εφόσον δεν έχει βρεθεί το αρχείο περιγραφής, τότε προκύπτει εξαίρεση τύπου FileNotFoundException και τυπώνεται το μήνυμα "Error Occurred". Σε περίπτωση που δεν έχει προκύψει εξαίρεση, τότε καταχωρούνται σε μεταβλητές στις αντίστοιχες μεθόδους στην ***κλάση FileDetails***. Το αρχείο

περιγραφής πρέπει να βρίσκεται στον φάκελο medialab, ο οποίος περιλαμβάνει μία σειρά από αρχεία περιγραφής (απαραίτητα πρέπει να έχει οριστεί το αρχείο περιγραφής SCENARIO-ID) που περιέχουν τις απαραίτητες πληροφορίες.
Στην ***κλάση Cell***, ορίζουμε το τύπο, τη θέση, αν έχει αποκαλυφθεί, αν είναι σημειωμένο με σημαία και τον εκάστοτε Handler. Επίσης, καθορίζεται μέσω MouseListener() τι συμβαίνει το εκάστοτε Cell (με τη βοήθεια της κλάσης Handler) αν πατηθεί πάνω του αριστερό ή δεξί κλικ.
Στην ***κλάση Grid***, στη μέθοδο createCells(), δημιουργούμε το αρχείο mines.txt στον προκαθορισμένο φάκελο, ώστε να αποθηκεύουμε μέσω FileWriter τις θέσεις που τοποθετήσαμε τις νάρκες και την τιμή 1 αν η νάρκη αποτελεί υπερνάρκη ή την τιμή 0 αν δεν αποτελεί. Για κάθε νάρκη επιλέγουμε μία τυχαία θέση στο πλέγμα και την αποθηκεύουμε σε μία λίστα mines και διατρέχοντας όλες τις θέσεις των Cell στο πλέγμα, καθορίζουμε τις συντεταγμένες στις οποίες βρίσκεται η κάθε νάρκη και τη θέση της υπερνάρκης. Έπειτα, αποθηκεύουμε τα ζητούμενα του αρχείου σε ένα String str το οποίο αποθηκεύεται με τη σειρά του στο αρχείο mines.txt. Επίσης, δημιουργούμε μία λίστα cellGrid τύπου Cell στην οποία θα αποθηκεύεται το εκάστοτε Cell παίρνοντας περιπτώσεις και για τα άκρα του πλέγματος. Αν η θέση καταλαμβάνεται από νάρκη, τότε στη λίστα αποθηκεύεται ένα Cell τύπου 1, ενώ αν στις γειτονικές περιοχές υπάρχουν νάρκες, τότε στη λίστα αποθηκεύεται ένα Cell τύπου 2, αλλιώς αποθηκεύεται ένα Cell τύπου 0. Κάθε φορά που ανανεώνεται το παιχνίδι, ακόμη και με την ίδια περιγραφή, οι θέσεις που αντιστοιχούν σε νάρκες διαφέρουν. Οι βοηθητικές μέθοδοι addCells() και removeCells() προσθέτουν στο πλέγμα και αφαιρούν τα στοιχεία από τη λίστα cellGrid αντίστοιχα.
Στην ***κλάση Handler***, υλοποιούμε τη συμπεριφορά της εφαρμογής σε περίπτωση που ο χρήστης πατήσει δεξί ή αριστερό κλικ σε κάποιο Cell. Αν το Cell δεν έχει επισημανθεί με σημαία, τότε, με αριστερό κλικ, ελέγχουμε τι τύπος Cell πατιέται. Αρχικά, αν το Cell είναι κενό, τότε ελέγχουμε όλες τις περιπτώσεις ώστε να αποθηκεύσουμε τις θέσεις στις οποίες πρέπει να αποκαλυφθούν αναδρομικά τα Cell με τη βοήθεια μίας λίστας queue. Σε περίπτωση που το Cell περιέχει αριθμό με τον οποίο υποδεικνύει ότι υπάρχουν γειτονικές νάρκες, τότε, για κάθε περίπτωση, μετριέται ο αριθμός αυτός. Αν το Cell περιέχει νάρκη, τότε αποκαλύπτονται όλες οι θέσεις των ναρκών επισημαίνοντας τες με «X» εκτός από εκείνη που πατήθηκε εκείνη τη στιγμή, καθώς επισημαίνεται με «MINE». Σε περίπτωση νίκης, αποκαλύπτονται οι θέσεις των ναρκών με «X» και στα υπόλοιπα Cell αναγράφεται «WIN». Επίσης, καταγράφεται η νίκη (αν ανήκει στις πέντε πρώτες) στο μενού με τα Rounds με τη βοήθεια πινάκων.
Σε περίπτωση που γίνει δεξί κλικ, αν το Cell δεν έχει επισημανθεί ήδη με σημαία ή έχει αποκαλυφθεί επιτυχώς, ελέγχουμε αν οι σημαίες δεν ξεπερνούν τον αριθμό των ναρκών του παιχνιδιού. Επίσης, αν οι προσπάθειες είναι ίσες ή μικρότερες από τέσσερις, το παιχνίδι είναι δύσκολου επιπέδου και η θέση του Cell είναι εκείνη της υπερνάρκης, τότε επισημαίνεται ως σημαία με την επιγραφή «H» και αποκαλύπτονται όλα τα Cell που βρίσκονται στην ίδια γραμμή και στήλη με την υπερνάρκη. Αν σε αυτά τα Cell, περιέχεται νάρκη, τότε αποκαλύπτεται με την επιγραφή «M» (τα οποία επισημαίνονται ως σημαίες), αν είναι κενά τα Cell αποκαλύπτονται χωρίς την αναδρομική αποκάλυψη και αν περιέχουν τον αριθμό των ναρκών στα γειτονικά Cell να αποκαλύπτονται με τον εκάστοτε αριθμό. Στις υπόλοιπες περιπτώσεις, το Cell επισημαίνεται με σημαία με την επιγραφή «F». Αν ξαναπατηθεί το δεξί κλικ, τότε μειώνεται ο αριθμός των επισημασμένων σημαιών, το Cell δε θεωρείται με σημαία και ο παίκτης μπορεί να το αποκαλύψει με αριστερό κλικ.

Στην ***κλάση Main***, ορίζουμε τη μέθοδο TimerStart() στην οποία χρησιμοποιείται Task και TimerTask ώστε να γίνεται η αντίστροφη μέτρηση του διαθέσιμου χρόνου που ορίζεται από κάποιο έγκυρο αρχείο περιγραφής. Ουσιαστικά, μειώνεται ο χρόνος στο πάνω μέρος του κεντρικού παραθύρου μέχρι να τελειώσει ο διαθέσιμος χρόνος ή να πατηθεί με αριστερό κλικ σε Cell με νάρκη ή όταν ο παίκτης κερδίσει. Αν τελειώσει ο διαθέσιμος χρόνος χωρίς να κερδίσει ο παίκτης, τότε αποκαλύπτονται όλες οι θέσεις των ναρκών επισημαίνοντας τες με «X».
Δημιουργούμε την ***κλάση Window*** στην οποία θα σχεδιάσουμε τη μορφή που θα έχει το παιχνίδι στο χρήστη. Αρχικά, θεωρούμε ως τίτλο παραθύρου «MediaLab Minesweeper» και δημιουργούμε ένα frame για το εκάστοτε παιχνίδι. Θεωρούμε ένα menubar το οποίο είναι παρόμοια υλοποιημένο με εκείνο της κλάσης InitWindow. Την πρώτη φορά που έχει δοθεί έγκυρο αρχείο περιγραφής, αρχικοποιούμε τον timer για να αρχίσει το παιχνίδι και αρχικοποιούμε το frame για το παιχνίδι. Φορτώνοντας ένα καινούριο παιχνίδι, ακυρώνεται η παλιά αντίστροφη μέτρηση και ξεκινά νέα αντίστροφη μέτρηση βάσει της καινούριας έγκυρης περιγραφής. Σε περίπτωση που το αρχείο περιγραφής δεν είναι έγκυρο, προκύπτει εξαίρεση και τυπώνεται το κατάλληλο μήνυμα. Στο MenuItem “Solution”, ο χρήστης μπορεί να αποκαλύψει τις θέσεις των ναρκών (αν ακόμη το παιχνίδι είναι ενεργό) και να σταματήσει το χρονόμετρο. Τα πεδία του πάνω μέρους της γραφικής διεπαφής ανανεώνονται με τη βοήθεια των μεθόδων update() (ανανέωση του πάνω μέρους του παραθύρου πλην του χρόνου) και timeupdate() (ανανέωση χρόνου κατά τη διάρκεια του εκάστοτε παιχνιδιού).
Μορφή Παιχνιδιού Εύκολου Επιπέδου και Δύσκολου Επιπέδου αντίστοιχα:





Στην ***κλάση CreateFile***, δημιουργείται ένα νέο αρχείο περιγραφής, βάσει της εισαγωγής στοιχείων στο MenuItem “Create”. Σε περίπτωση που ο χρήστης δεν πληκτρολογήσει ακέραιο αριθμό ή πληκτρολογήσει κάποια συμβολοσειρά πέρα από το πεδίο εισαγωγής για το SCENARIO-ID, δε θα καταγράφεται το πεδίο στο αρχείο περιγραφής.
Η ***κλάση CopyFile*** υλοποιείται ώστε να αντιγράφεται το αρχείο περιγραφής που φορτώνει ο χρήστης στο SCENARIO-ID.txt σε περίπτωση που έχει δημιουργηθεί.

Η κλάση CopyFile είναι τεκμηριωμένη σύμφωνα με τις προδιαγραφές του εργαλείου Javadoc. Συγκεκριμένα, η μέθοδος copyContent της κλάσης CopyFile παίρνει δύο File objects ως παραμέτρους και αντιγράφει το περιεχόμενο του πρώτου αρχείου στο δεύτερο αρχείο. Το Javadoc σχόλιο ξεκινά με μία περιγραφή του τι κάνει η μέθοδος. Ακολουθείται από το tag @param για κάθε παράμετρο, το οποίο περιγράφει τι παρουσιάζει η παράμετρος και έπειτα από το tag @throws που περιγράφει την εξαίρεση που προκαλείται από τη μέθοδο.