

Αναφορά Τελικής Εργασίας μαθήματος Τεχνολογίας Πολυμέσων

Το project της εργασίας βρίσκεται στο αρχείο Minesweeper.zip. Η ανάπτυξή του έγινε με τη χρήση του IDE IntelliJ, και συνίσταται η χρήση του για το τρέξιμο της εφαρμογής (μέσω του MinesweeperApp.java) και την περιήγηση στα αρχεία του project. Μπορείτε να βρείτε τον κώδικα και να δείτε την πρόοδο ολοκλήρωσης της εφαρμογής και σε [repository στο github](#).

Περιγραφή Υλοποίησης – Λογική Παιχνιδιού

Η υλοποίηση της λογικής του παιχνιδιού βρίσκεται στο πακέτο gameLogic του project. Αυτό περιέχει 2 βασικές κλάσεις που υλοποιούν τα τμήματα που αφορούν το σενάριο του παιχνιδιού (scenario.java), και τη διαχείριση του παιχνιδιού (Game.java), καθώς και δύο βοηθητικές κλάσεις (Board και Mine) εντός της τελευταίας για τη διαχείριση των ναρκών και του ταμπλό του παιχνιδιού. Παρακάτω θα περιγραφεί ο σχεδιασμός τους.

Scenario.java:

Τα instances αυτής της κλάσης περιέχουν τις πληροφορίες του σεναρίου του παιχνιδιού. Μπορούν να δημιουργηθούν, όπως ζητήθηκε, διαβάζοντας το περιεχόμενο κάποιου αρχείου με τη χρήση της στατικής μεθόδου readFromFile, η οποία επίσης επικυρώνει ότι οι παράμετροι του σεναρίου είναι εντός των ορίων της εκφώνησης. Η δημιουργία scenario με κλήση του constructor δεν εκτελεί επικύρωση. Οι custom εξαιρέσεις που εγείρει η readFromFile ορίζονται στο πακέτο customExceptions του project.

Game.java:

Τα αντικείμενα αυτής της κλάσης χειρίζονται τις παρτίδες του παιχνιδιού. Δημιουργούνται με βάση κάποιο scenario, από το οποίο παίρνουν τις παραμέτρους που θα έχει η παρτίδα, και χειρίζονται τα δεδομένα του παιχνιδιού που δεν αφορούν το ταμπλό, όπως πόσες προσπάθειες έχει κάνει ο χρήστης, πόσες σημαίες και πόσος χρόνος του μένουν, καθώς και αν η παρτίδα έχει τελειώσει και αν ο χρήστης κέρδισε ή όχι. Με την κλήση απευθείας αυτής της κλάσης από ένα terminal, γίνεται να παιχτεί κανονικά το παιχνίδι στο terminal (μαζί με το όνομα του σεναρίου, πχ java Game.class SCENARIO-01.txt), δίνοντας εντολές για αποκάλυψη ή μαρκάρισμα ενός τετραγώνου γράφοντας r ή f αντίστοιχα μαζί με τις συντεταγμένες του τετραγώνου (πχ r 1 2 για αποκάλυψη του τετραγώνου στην 1η σειρά, 2η στήλη).

Η βοηθητική κλάση Board αναλαμβάνει το μεγαλύτερο μέρος της λογικής του παιχνιδιού, μιας και χειρίζεται το ταμπλό του και συνεπώς την αποκάλυψη και μαρκάρισμα τετραγώνων, όπως και τις λειτουργίες της υπερ-νάρκης. Μεγαλύτερη σημασία έχουν οι δύο πίνακες mineBoard και revealedBoard, που περιλαμβάνουν τα τετράγωνα με νάρκες και τους χαρακτήρες που βλέπει ο χρήστης στο κάθε τετράγωνο αντίστοιχα. Η αντιστοιχία των χαρακτήρων αυτών με τα αντικείμενα του παιχνιδιού είναι ως εξής:

Αντικείμενο	Χαρακτήρας
Κρυμμένο Τετράγωνο	'\u0000' (default χαρακτήρας)
Φανερωμένο τετράγωνο, χωρίς διπλανές νάρκες	' ', κενό
Φανερωμένο τετράγωνο, με διπλανές νάρκες	'1' έως '8', ανάλογα με τον αριθμό
Τετράγωνο με σημαία	'F'
Νάρκη (μετά το τέλος του παιχνιδιού)	'M'
Νάρκη (που "πάτησε" ο παίκτης)	'T'
Υπερ-Νάρκη που βρήκε ο παίκτης	'U'

Η αποκάλυψη τετραγώνων γίνεται επίσης από την κλάση Board, με την πρώτη αποκάλυψη (αυτή που κάνει ο χρήστης) να θεωρείται unsafe (οδηγεί σε τέλος παιχνιδιού αν βρει νάρκη) ενώ όλες προκύπτουν αναδρομικά από τετράγωνα χωρίς διπλανές νάρκες είναι “safe”, δεν αποκαλύπτουν νάρκες και δεν οδηγούν σε τέλος παιχνιδιού. Η αποκάλυψη της Υπερ-Νάρκης επίσης γίνεται σε αυτήν την κλάση, όπως και το μαρκάρισμα τετραγώνων.

Η βοηθητική κλάση Mine αφορά τις νάρκες του παιχνιδιού (συντεταγμένες και flag Υπερ-Νάρκης), την εκτύπωσή τους σε κατάλληλη μορφή ώστε να αποθηκευτούν στο αρχείο mines.txt, καθώς και τη δημιουργία τους σε τυχαία σημεία του ταμπλό (αλλά η κλάση Board αναλαμβάνει να φροντίσει να μην δημιουργηθούν δύο στο ίδιο σημείο).

Περιγραφή Υλοποίησης – Γραφική Διεπαφή Χρήστη (GUI)

Η διεπαφή οργανώνεται σε 4 βασικά αρχεία, που περιλαμβάνουν τους controller για τα αντίστοιχα fxml αρχεία που απαρτίζουν τη διεπαφή :

- MinesweeperApp.java, απ’ όπου ξεκινάει το GUI του παιχνιδιού,
- MainPageController.java για την διαχείριση της κύριας οθόνης του παιχνιδιού και της παρτίδας (χρόνος, έναρξη/λήξη, άνοιγμα άλλων παραθύρων), που αφορά το base.fxml. Εδώ γίνεται η δημιουργία των τετραγώνων όπου κάνει κλικ ο χρήστης για να παίξει το παιχνίδι.
- CreateScenarioController.java για τη δημιουργία σεναρίων και την εμφάνιση των αντίστοιχων warning στο χρήστη αν κάτι είναι εκτός παραμέτρων, για το scenarioCreation.fxml. Εδώ προστέθηκε κουμπί που καθαρίζει τις τιμές των πεδίων των παραμέτρων του παιχνιδιού και δίνει ένα τυχαίο όνομα στο σενάριο,
- LoadScenarioController.java, όπου γίνεται φόρτωση σεναρίων από το χρήστη, για το scenarioLoad.fxml. Προστέθηκε επίσης ένα checkbox που όταν επιλεγεί κλείνει το παράθυρο μετά την επιτυχή φόρτωση του σεναρίου,
- scoreController.java, για την εμφάνιση των τελευταίων 5 παιχνιδιών στο χρήστη, για το results.fxml.

Στο ταμπλό του παιχνιδιού εμφανίζονται οι ίδιοι χαρακτήρες που χρησιμοποιήθηκαν στην κλάση Board για την περιγραφή των αντικειμένων του παιχνιδιού, με διαφορετικά χρώματα και μεγέθη κειμένου ανάλογα με το αντικείμενο (πχ μπλε για τα ‘F’ των σημαιών, κόκκινο για τις νάρκες κλπ).

Στις διάφορες περιπτώσεις όπου εμφανίζονται ειδοποιήσεις στο χρήστη (πχ λανθασμένες ρυθμίσεις σεναρίου, απουσία φορτωμένου σεναρίου κλπ) ανοίγει ξεχωριστό Alert Window με πληροφορίες.

Περιγραφή Υλοποίησης – Λοιπές Απαιτήσεις

Μεγάλο μέρος της εργασίας δεν ακολουθεί πλήρως τις αρχές SOLID του αντικειμενοστραφούς προγραμματισμού, μιας και λόγω χρονικών περιορισμών αναγκάστηκα να ξεκινήσω την εργασία πριν τις διδαχθούμε στο μάθημα. Αυτό αφορά κυρίως το Open-Closed Principle και το Dependency Inversion Principle, μιας και γίνεται ελάχιστη χρήση getters-setters στις κλάσεις που υλοποιήθηκαν νωρίς κατά την εκπόνηση της εργασίας, και γίνεται μηδαμινή χρήση abstract κλάσεων. Ωστόσο, η λογική της υλοποίησης παραμένει αυτή του αντικειμενοστραφούς προγραμματισμού.

Η κλάση που περιέχει τεκμηρίωση σύμφωνα με τις προδιαγραφές του javadoc είναι η scenario.java, η οποία έχει περιγραφεί παραπάνω.