

Αναφορά για την Εξαμηνιαία Εργασία “Medialab Minesweeper” για το μάθημα Τεχνολογία Πολυμέσων

Αριστοτέλης Γρίβας

A.M : 03119889

Η σχεδίαση του παιχνιδιού βασίζεται στην κλάση **Element** η οποία χρησιμοποιείται για τη δημιουργία των στοιχείων(κουτιών) του παιχνιδιού. Κάθε τέτοιο στοιχείο είναι ένα τετράγωνο(rectangle).Ο constructor ενός τέτοιου στοιχείου ορίζει την οριζόντια και κατακόρυφη θέση του(x,y), καθώς και αν περιέχει νάρκη(hasBomb). Με χρήση της κλάσης αυτής μπορούμε να σχεδιάσουμε την κύρια σκηνή του παιχνιδιού που κατασκευάζεται μέσω της μεθόδου “CreateGame” ,η οποία επιστρέφει έναν κόμβο Parent όπου προσθέτουμε τα Elements ως παιδιά. Η μέθοδος αυτή χρησιμοποιεί τα εξής:

i) Τη μέθοδο getneighbors() η οποία γυρνά μία λίστα από τα στοιχεία - γείτονες του εκάστοτε στοιχείου. Με χρήση αυτής μπορούμε να θέσουμε το κείμενο κάθε στοιχείου εξετάζοντας πόσοι από τους γείτονές του έχουν νάρκη.

ii) Τον 2 διαστάσεων πίνακα minetable ο οποίος περιέχει την πληροφορία της ύπαρξης νάρκης στο σημείο x,y (0 → δεν υπάρχει νάρκη, 1 → υπάρχει νάρκη,2 → υπάρχει υπερνάρκη).

Με βάση τα δύο παραπάνω θέτουμε το κείμενο στο κάθε στοιχείο(X για βόμβα,αριθμός/κενό διαφορετικά),και το αποκρύβουμε έως ότου κάνουμε αριστερό κλικ πάνω του,καλώντας τη μέθοδο open() της κλάσης Element,ή δεξί κλικ καλώντας τη μέθοδο flagit()(μαρκάρισμα κουτιού → το χρώμα του γίνεται πράσινο). Ο πίνακας GameArray[][] περιέχει όλα τα στοιχεία που προστέθηκαν στη σκηνή με τις αντίστοιχες θέσεις τους.

Για τις υπόλοιπες σκηνές γίνεται χρήση των fxml αρχείων sample1.fxml και sample2.fxml,οι λειτουργίες των οποίων ελέγχονται από τους SampleController και Sample2Controller αντίστοιχα.

Η σκηνή που κατασκευάζει το αρχείο sample1.fxml είναι η σκηνή που βλέπουμε όταν τρέχει αρχικά ο κώδικας. Η σκηνή αυτή περιέχει το κύριο Menu με τις λειτουργίες που αναφέρονται στην εκφώνηση(Create,Load,Start etc.) καθώς και τα μηνύματα σχετικά με τα μαρκαρισμένα στοιχεία,το πλήθος ναρκών και τον χρόνο που απομένει. Η σκηνή αυτή είναι διαφορετική από τη σκηνή που απεικονίζει το παιχνίδι καθαυτό,την οποία περιγράψαμε παραπάνω και εμφανίζεται όταν πατάμε StartGame.

Η τρίτη και τελευταία σκηνή που περιγράφεται από το αρχείο sample2.fxml παρουσιάζει το pop up παράθυρο όπου μπορούμε να ορίσουμε τις ρυθμίσεις δυσκολίας του παιχνιδιού. Εμφανίζεται όταν πατάμε CreateGame καθώς καλείται η μέθοδος GiveNumbers του SampleController,και αποτελείται από 4 γραμμές όπου ορίζουμε τις μεταβλητές χρόνου /δυσκολίας/ναρκών/υπερνάρκης , μέσω πληκτρολογίου, και τις υποβάλλουμε με το Submit Settings button. Όλα τα buttons της σκηνής αυτής ρυθμίζονται από το Sample2Controller. Αν η περιγραφή μας δεν είναι όπως αναμένεται,οι αντίστοιχες εξαιρέσεις θα αποδοθούν στην έξοδο και ο χρήστης θα πρέπει να ορίσει εκ νέου όποια μεταβλητή είναι εκτός ορίων. Αφού ,λοιπόν, ορίσουμε τις μεταβλητές σε επιτρεπόμενες τιμές,το button SubmitSettings θα

δημιουργήσει το αρχείο “SCENARIO-ID.txt” μέσω της συνάρτησης CreateFile() του SampleController, γράφοντας την κάθε μεταβλητή σε ξεχωριστή γραμμή όπως αναφέρεται στην εκφώνηση.

Στη συνέχεια με το LoadGame button καλείται η μέθοδος Load() του SampleController και η κάθε τιμή που δώσαμε προηγουμένως περνά στις μεταβλητές του κυρίου προγράμματος. Στο σημείο αυτό μπορούμε να αρχίσουμε το παιχνίδι μέσω του κουμπιού StartGame, το οποίο καλεί τη μέθοδο StartGame και μέσω αυτής τη begin του Main προγράμματος, ανοίγει τη σκηνή με τον αντίστοιχο αριθμό στοιχείων και ρυθμίζει έναν μετρητή (timer) ο οποίος μετρά αντίστροφα.

Τέλος όσον αφορά στον SampleController, υπάρχουν και οι συναρτήσεις για τα κουμπιά Details/Solve/ExitGame οι οποίες τηρούν τις προβλεπόμενες λειτουργίες. Συγκεκριμένα για τη λειτουργία Details, τα αποτελέσματα κάθε παιχνιδιού γράφονται στο προκαθορισμένο αρχείο details.txt, σε νέα γραμμή, και σε κάθε πάτημα διαβάζονται οι 5 τελευταίες γραμμές του αρχείου (5 τελευταία παιχνίδια, με το πιο πρόσφατο να εμφανίζεται στην τελευταία γραμμή).

Παραδοχές:

- Επειδή δεν ήταν κατανοητή η λειτουργία του αρχείου SCENARIO-ID, υποθέσαμε ότι με κάθε νέο παιχνίδι το αρχείο αυτό δημιουργείται εκ νέου (στο φάκελο medialab) και εγγράφονται σε αυτό οι νέες τιμές. Δηλαδή η εφαρμογή υποστηρίζει διαφορετικές περιγραφές, αλλά κάθε φορά πρέπει να ορίζεται η επιθυμητή. Αυτό κάνει απίθανο το ενδεχόμενο εξαίρεσης InvalidDescriptionException (καθώς γράφουμε πάντα 4 αριθμούς σε 4 γραμμές), ωστόσο εάν θέλουμε να δούμε τη λειτουργία της εξαίρεσης αυτής μπορούμε να αλλάξουμε manually το αρχείο SCENARIO-ID και να επιχειρήσουμε τα Load/Start Game. Εκεί ελέγχουμε εάν το αρχείο έχει ακριβώς 4 γραμμές και ακριβώς 4 αριθμούς και αν χρειαστεί, η εξαίρεση θα εμφανιστεί.

- Τα αρχεία details.txt και mines.txt βρίσκονται στο φάκελο medialab, ωστόσο το mines.txt δημιουργείται εκ νέου με κάθε νέο παιχνίδι, ενώ το details.txt απλά ανανεώνεται.

- Ο φάκελος medialab βρίσκεται στο φάκελο του project, και περιέχει τα παραπάνω αρχεία καθώς και όποιες εικόνες χρησιμοποιούνται στη διεπαφή.

- Η υπερνάρκη εμφανίζεται στα 4 πρώτα μαρκαρίσματα.

Μη υλοποιημένη λειτουργικότητα : Το Menu και το MainGame δεν είναι στην ίδια σκηνή.

Άλλες παρατηρήσεις:

- Κάθε φορά που θέλουμε να αρχίσουμε ένα νέο παιχνίδι, πρέπει να κάνουμε load υποχρεωτικά το σενάριο που θέλουμε, διαφορετικά δεν θα αρχίσει το παιχνίδι και θα αποδοθεί ανάλογο μήνυμα.

- Οι μέθοδοι StaticBufferSize και countLineJava χρησιμοποιούνται για το διάβασμα των 5 τελευταίων γραμμών για τη λειτουργία Details.

- Έχουν προστεθεί διάφορα μηνύματα για διευκόλυνση και καλύτερη επικοινωνία με το χρήστη .

- Αν ο παίκτης κερδίσει, χάσει, επιλέξει να δει τη λύση του προβλήματος ή τελειώσει ο χρόνος του, τυπώνεται στην έξοδο Defeat/Victory και η σκηνή “παγώνει” .

- Η εφαρμογή υλοποιήθηκε στο EclipseIDE σε συνδιασμό με τη χρήση της εφαρμογής SceneBuilder.