10/11/2020

Ηλεκτρονική αναφορά

Part A

Ονοματεπώνυμο	ΑΕΜ	Τηλέφωνο	Email
Κωνσταντίνος Μαργαρίτης	10061	6989850683	knmargar@ece.auth.gr
Κωνσταντίνος Μυλωνάς	10027	6980369600	kmylonas@ece.auth.gr
Φίλιππος Χαραλαμπίδης	9959	6945147108	fcharala@ece.auth.gr

Πίνακας περιεχομένων

Περιγραφή του προβλήματος

Η συγκεκριμένη εργασία έχει στόχο την δημιουργία ενός παιχνιδιού που στηρίζεται στη γνωστή ιστορία από τη μυθολογία με κύριους ήρωες τον Θησέα και τον Μινώταυρο, και περιλαμβάνει στοιχεία από την ταινία «Μια νύχτα στο μουσείο». Ο Θησέας επιθυμώντας να σκοτώσει τον Μινώταυρο εισέρχεται στο μουσείο - λαβύρινθο με στόχο να μαζέψει όλα τα απαραίτητα εφόδια, ανοίγοντας έτσι μια καταπακτή στην οποία θα πέσει ο Μινώταυρος. Αντίθετα, ο Μινώταυρος κερδίζει αν βρει τον Θησέα προτού αυτός καταφέρει να μαζέψει όλα τα εφόδια. Στο πρώτο παραδοτέο καλούμασταν να υλοποιήσουμε τη δημιουργία του ταμπλό του παιχνιδιού και την τυχαία κίνηση των παικτών πάνω σε αυτό.

Πιο αναλυτικά, χρειάστηκε να δημιουργήσουμε 5 κλάσεις σε Java, κάθε μία από τις οποίες περιλάμβανε μεταβλητές και μεθόδους. Οι πιο πολλές από αυτές μας δίνονταν σε σχετικό pdf με ανάλογες οδηγίες, ωστόσο υλοποιήσαμε και μερικές ακόμα που θεωρήσαμε χρήσιμες για την ολοκλήρωση της εργασίας. Κάθε κλάση είχε προφανώς getters και setters ώστε να μπορούμε να διαβάζουμε τη τιμή της κάθε μεταβλητής της κλάσης, αλλά να μπορούμε και να δώσουμε τιμές σε αυτή.

Περιγραφή και ανάλυση του αλγορίθμου και των διαδικασιών

Κλάση Supply

Η κλάση αυτή αφορά τα εφόδια που προσπαθεί να συγκεντρώσει ο Θησέας. Περιλαμβάνει 4 μεταβλητές οι οποίες δηλώνονται στην αρχή της κλάσης. Στην συνέχεια, κατασκευάζονται οι constructors της κλάσης Supply, στους οποίους αρχικοποιούνται οι μεταβλητές, καθώς και οι getters και setters της κλάσης, μέσω των οποίων αργότερα διαβάζουμε τις τιμές των μεταβλητών και αναθέτουμε τιμές σε αυτές.

Κλάση Tile

Η κλάση αυτή αφορά τα πλακίδια του ταμπλό και περιλαμβάνει 7 μεταβλητές, 3 ακέραιες και 4 boolean. Οι ακέραιες αφορούν στο id και στις συντεταγμένες του πλακιδίου, ενώ οι boolean δείχνουν αν το πλακίδιο διαθέτει τοίχο στις 4 πλευρές του (βόρεια/πάνω, νότια/κάτω, ανατολικά/δεξιά, δυτικά/αριστερά). Παρόμοια με την κλάση Supply, κατασκευάζονται οι constructors, οι getters και οι setters της κλάσης Tile, που επιτελούν ίδιες λειτουργίες για τις αντίστοιχες μεταβλητές. Επίσης, υλοποιήσαμε μια μέθοδο, η οποία λέγεται numberOfWalls, η οποία επιστρέφει πόσα τείχη περικλείουν ένα συγκεκριμένο πλακίδιο (Tile).

Κλάση Board

Πρόκειται για το ταμπλό του παιχνιδιού, το οποίο περιλαμβάνει ως μεταβλητές τις διαστάσεις του, τον αριθμό των εφοδίων που υπάρχουν σε αυτό, τον αριθμό των τειχών μεταξύ των πλακιδίων, έναν πίνακα αντικειμένων τύπου Tile και έναν αντίστοιχο τύπου Supply. Αφού κατασκευάστηκαν οι constructors, στους οποίους αρχικοποιούνται οι πίνακες tiles και supplies με άδεια Tiles και Supplies αντίστοιχα, οι getters και οι setters της κλάσης, χρειάστηκε να υλοποιηθούν ορισμένες μέθοδοι για την αρχικοποίηση και αναπαράσταση του ταμπλό.

Μέθοδος void createTile()

Σε αυτή τη μέθοδο αρχικοποιούνται τα αντικείμενα του πίνακα Tile με τυχαίο τρόπο μέσα στο ταμπλό. Η μεταβλητή wallsToBeAdded κρατάει πόσα τείχη θα μπουν σε ένα πλακίδιο. Η μεταβλητή directionOfWall κρατάει σε ποια κατεύθυνση του πλακιδίου θα μπει το τείχος. Οι κατευθύνσεις είναι οι εξής: 1=πάνω, 3=δεξιά, 5=κάτω, 7=αριστερά. Αρχικά, με τη πρώτη λούπα αρχικοποιούνται τα εξωτερικά τείχη του Board, με την εξής λογική: αν το πλακίδιο βρίσκεται στην πάνω σειρά του πίνακα, τότε μπαίνει τοίχος στη βόρεια/πάνω πλευρά του, αν το πλακίδιο βρίσκεται στην κάτω σειρά του πίνακα, τότε μπαίνει τοίχος στη νότια/κάτω πλευρά του, αν το πλακίδιο βρίσκεται στην δεξιά στήλη του πίνακα, τότε μπαίνει τοίχος στη ανατολική/δεξιά πλευρά του, αν το πλακίδιο βρίσκεται στην αριστερή στήλη του πίνακα, τότε μπαίνει τοίχος στη δυτική/αριστερή πλευρά του. Στη συνέχεια, δηλώνεται μια νέα μεταβλητή, η wallsLeft, η οποία κρατάει πόσα τείχη μένουν συνολικά για να μπουν στο Board. Μέσα στη λούπα που ακολουθεί ελέγχονται οι περιορισμοί που έχουν τεθεί, δηλαδή να μην μπει τείχος εκεί που ήδη υπάρχει, να μην μπουν παραπάνω από δύο τείχη σε ένα tile, ελέγχοντας κάθε φορά το tile προς τη κατεύθυνση που έχουμε αποφασίσει με τυχαίο τρόπο να μπει τείχος. Αν ικανοποιούνται αυτές οι συνθήκες, μπαίνει

τείχος στην συγκεκριμένη κατεύθυνση. Ακόμη, το διπλανό tile παίρνει τείχος στην πλευρά που εφάπτεται στο tile που βρισκόμαστε ήδη. Το πρόγραμμα σταματάει όταν δεν υπάρχουν πλέον τείχη διαθέσιμα για να μπουν στο board.

Μέθοδος void createSupply()

Σε αυτή τη μέθοδο αρχικοποιούνται τα αντικείμενα του πίνακα Supply με τυχαίο τρόπο. Η λογική που ακολουθήσαμε είναι η ακόλουθη: η μεταβλητή j κρατάει το id του supply, το οποίο υπολογίζεται με τυχαίο τρόπο, μέσω της Math.random. Αρχικά, λοιπόν, βρίσκουμε ένα τυχαίο tile στο οποίο επιθυμούμε να τοποθετήσουμε το εφόδιο, και στη συνέχεια ελέγχουμε αν στο tile αυτό υπάρχει ήδη εφόδιο. Επίσης, ελέγχουμε εάν στο tile αυτό βρίσκεται ήδη ο Θησέας ή ο Μινώταυρος, και εάν κάτι από αυτά ισχύει, με τη χρήση ενός counter, καταφέρνουμε το εφόδιο να μην τοποθετηθεί σε αυτό το πλακίδιο, αλλά να επαναληφθεί η λούπα έως ότου βρεθεί διαθέσιμο πλακίδιο και φυσικά τελειώσουν τα εφόδια που είναι διαθέσιμα για να μπουν στο board. Όταν βρεθεί διαθέσιμο tile, μέσω των setters, θέτουμε το supplyTileId, το supplyId, το x και το y. Για τα x και y αξίζει να σημειωθεί ο τρόπος με τον οποίο λαμβάνουν τις τιμές τους. Το x υπολογίζεται από το πηλίκο της ακέραιας διαίρεσης του supplyTileId με το μήκος του board και αναπαριστά σε ποια σειρά βρίσκεται το supply. Ομοίως, το y υπολογίζεται από το υπόλοιπο της ακέραιας διαίρεσης του supplyTileId με το μήκος του board και αναπαριστά σε ποια στήλη βρίσκεται το supply.

Μέθοδος void createBoard()

Αυτή η μέθοδος απλώς χρησιμοποιεί τις δύο παραπάνω ώστε να δημιουργήσει το ταμπλό του παιχνιδιού με ψευδο-τυχαίο τρόπο, όπως αναφέρουν οι οδηγίες. Καλούνται, επομένως, αυτές οι μέθοδοι και δημιουργείται το ταμπλό.

Μέθοδος String[][] getStringRepresentation(int theseusTile, int minotaurTile)

Η μέθοδος αυτή αναπαριστά το ταμπλό του παιχνιδιού με τη χρήση διάφορων strings. Αρχικά, δηλώνεται και αρχικοποιείται σε κενά string ο πίνακας table. Στη συνέχεια, αναπαριστούμε τα οριζόντια τείχη με παύλες μέσω μίας λούπας, η οποία έχει βήμα=2, καθώς τα οριζόντια τείχη διατηρούνται στην πρώτη γραμμή από τις δύο που απαιτούνται για να αναπαρασταθεί κάθε tile, ελέγχοντας κάθε φορά εάν το πλακίδιο έχει τείχος στην αντίστοιχη πλευρά του, μέσω των getters της κλάσης tile. Με την επόμενη λούπα, η οποία έχει πάλι βήμα=2, όμως ξεκινάει από τη δεύτερη σειρά του πίνακα table, αναπαριστούμε όλα τα κάθετα τείχη με μια κάθετο, ελέγχοντας πάλι εάν το πλακίδιο έχει τείχος στην αντίστοιχη πλευρά του, μέσω των getters της κλάσης tile. Επίσης, αναπαριστούμε τον Μινώταυρο και τον Θησέα στα αντίστοιχα tiles, καθώς και τα εφόδια στα αντίστοιχα tiles που έχουν μπει ήδη από τη μέθοδο createSupply. Τέλος, τοποθετούμε τείχη στην τελευταία σειρά του πίνακα table, τα τείχη που βρίσκονται στη νότια/κάτω πλευρά της τελευταίας σειράς πλακιδίων του πίνακα board.

Κλάση Player

Η κλάση αυτή αφορά τους παίκτες του παιχνιδιού. Περιλαμβάνει 7 μεταβλητές οι οποίες δηλώνονται στην αρχή της κλάσης. Στην συνέχεια, κατασκευάζονται οι constructors της κλάσης, στους οποίους αρχικοποιούνται οι μεταβλητές, καθώς και οι getters και setters της κλάσης, μέσω των οποίων αργότερα διαβάζουμε τις τιμές των μεταβλητών και αναθέτουμε τιμές σε αυτές. Ο πίνακας Board αρχικοποιείται με αρκετό χώρο στη μνήμη.

Μέθοδος int[] move(int id)

Στη μέθοδο αυτή αποφασίζεται η κίνηση του κάθε παίκτη με τυχαίο τρόπο. Πιο συγκεκριμένα, μέσω μιας μεταβλητής j, η οποία λαμβάνει μια τυχαία τιμή μεταξύ των 1, 3, 5, 7, και ανάλογα με αυτή τη τιμή προσπαθεί να κινηθεί προς την αντίθετη κατεύθυνση (1=πάνω, 3=δεξιά, 5=κάτω, 7=αριστερά). Ελέγχεται, στην συνέχεια, εάν ο παίκτης μπορεί να κινηθεί προς την αντίστοιχη κατεύθυνση ή αν υπάρχει τείχος εκεί. Αν δεν μπορεί να κινηθεί προς αυτή την κατεύθυνση, εμφανίζεται αντίστοιχο μήνυμα. Αν ο παίκτης (μόνο ο Θησέας) βρει εφόδιο, εμφανίζεται αντίστοιχο μήνυμα και το εφόδιο αφαιρείται από το table και το board (μετακινείται σε συντεταγμένες εκτός των πινάκων), ώστε να βλέπει ο παίκτης ότι δεν υπάρχει. Κάτι αντίστοιχο υλοποιείται αργότερα στην main για να το βλέπει το ταμπλό. Τελικά, επιστέφεται ένας πίνακας 4 ακεραίων, που περιλαμβάνει όσα ζητούνται στις οδηγίες.

Κλάση Game

Η κλάση αυτή αφορά το ίδιο το παιχνίδι. Περιλαμβάνει τη βασική μέθοδο main. Αρχικά, εκτός της main δηλώνονται και αρχικοποιούνται οι μεταβλητές της κλάσης, και δημιουργούνται οι constructors, getters και setters, που επιτελούν αντίστοιχες λειτουργίες με αυτούς των παραπάνω κλάσεων.

Μέθοδος public static void main()

Στην αρχή δηλώνεται ένα αντικείμενο τύπου Game και στη συνέχεια δηλώνεται το μέγεθος του ταμπλό, ο αριθμός των εφοδίων και ο αριθμός των τειχών. Επίσης, δηλώνονται δύο πίνακες που αντιπροσωπεύουν τις κινήσεις του Θησέα και του Μινώταυρου. Ακόμη δηλώνονται τα name, id και tileId του Μινώταυρου και του Θησέα και δηλώνεται και αρχικοποιείται το ταμπλό. Ξεκινάει το παιχνίδι, που διαρκεί 2n γύρους, όπου n=100, εκτός εάν ένας από τους δύο παίκτες κερδίσει σε λιγότερους γύρους. Μέσα στη λούπα, τυπώνεται ο αριθμός του γύρου, η κίνηση του κάθε παίκτη, το σκορ του κάθε παίκτη και ο νικητής, εάν υπάρχει. Στην περίπτωση που παρέλθουν 2n γύροι χωρίς νικητή, τυπώνεται αντίστοιχο μήνυμα. Ακόμη μέσα στην λούπα της μεθόδου αυτής, αν ο παίκτης έχει μαζέψει εφόδιο, αυτό διαγράφεται από το ταμπλό, όπως αναφέρθηκε νωρίτερα.

Γενικά σχόλια

Η ανάπτυξη του κώδικα έγινε σε περιβάλλον IntelliJ και τελικά μεταφέρθηκε το project στον eclipse. Για να γίνει compile απαιτείται η χρήση jdk.