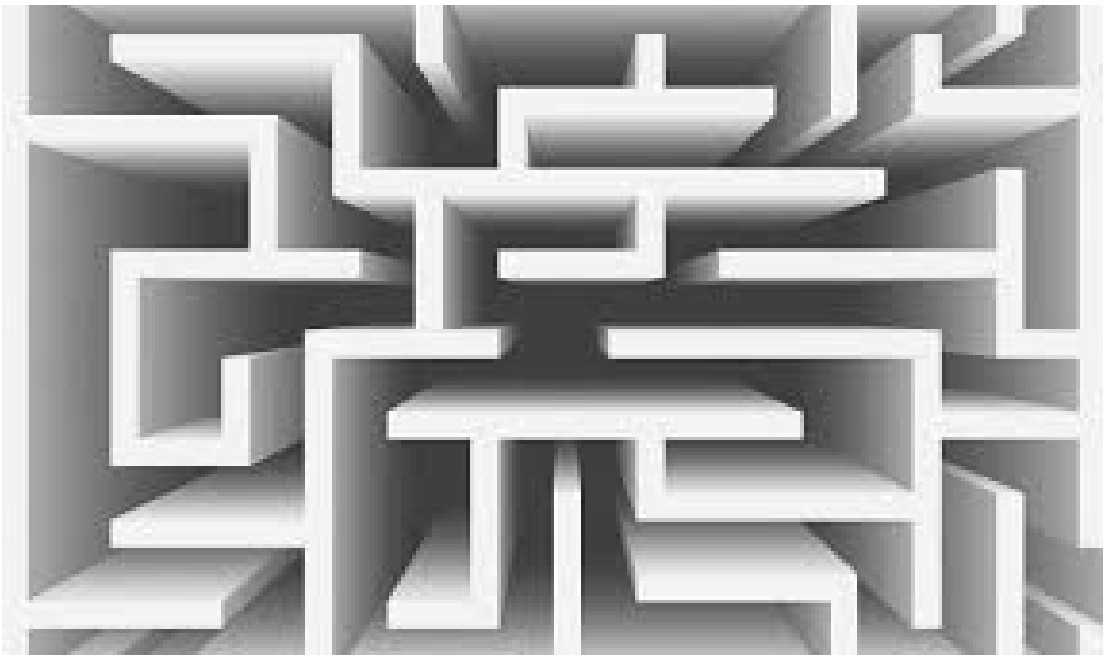
ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Λαβύρινθος

Ο Θησέας και ο Μινώταυρος



Φοιτητές (ομάδα 128):

Βοζίκης Γεώργιος, 9838, vozikisg@ece.auth.gr

Δούμου Ευγενία, 10178, evgedoum@ece.auth.gr

Σιμιτσάκη Ελένη, 10211, simielen@ece.auth.gr

Λίγα λόγια για το παιχνίδι:

Ο Θησέας και ο Μινώταυρος είναι ένα παιχνίδι που παίζεται με δύο παίκτες εκ των οποίων ένας είναι ο Θησέας και ένας ο Μινώταυρος. Σκοπός του Θησέα είναι να καταφέρει να μαζέψει όλα τα λάφυρα μέσα από τον λαβύρινθο ξεφεύγοντας από τον Μινώταυρο που έχει ως στόχο να τον σκοτώσει.

Ο αλγόριθμος του προγράμματος στο πρώτο παραδοτέο αναλύεται σε πέντε βασικές κλάσεις σύμφωνα την εκφώνηση.

Κλάση Supply

Η κλάση αυτή αντιπροσωπεύει τα εφόδια που υπάρχουν στο παιχνίδι. Παίρνει ως ορίσματα τις ακέραιες μεταβλητές: supplyId που είναι το id του εφόδου, τις συντεταγμένες x και y του εφόδου και την supplyTileId που αντιστοιχεί στο id του πλακιδίου στο οποίο βρίσκεται το εφόδιο στο ταμπλό . Τέλος, σ' αυτή την κλάση φτιάχνουμε τους constructors, τους setters και τους getters των μεταβλητών αυτών, ώστε να αρχικοποιούμε, να τροποποιούμε και να επιστρέφουμε τις τιμές τους.

Κλάση Tile

Η κλάση αυτή αντιπροσωπεύει τα πλακίδια που υπάρχουν στο ταμπλό. Παίρνει ως ορίσματα τις ακέραιρες μεταβλητές: tileId όπου είναι το id του πλακιδίου, τις συντεταγμένες x και y του πλακιδίου και τέσσερις boolean μεταβλητές up, down, left και right, οι οποίες δείχνουν εάν υπάρχει τοίχος στην αντίστοιχη πλευρά του πλακιδίου. Τέλος, στην κλάση αυτή φτιάχνουμε τους constructors, τους setters και τους getters των μεταβλητών αυτών, ώστε να αρχικοποιούμε, να τροποποιούμε και να επιστρέφουμε τις τιμές τους.

Κλάση Board

Η κλάση Board αντιπροσωπεύει το ταμπλό του παιχνιδιού. Παίρνει ως ορίσματα τις εξής ακέραιες μεταβλητές: N, που είναι ο αριθμός για τις διαστάσεις του ταμπλό (NxN) , S, που είναι ο αριθμός των συνολικών εφοδίων που υπάρχουν στο ταμπλό και W, που είναι ο συνολικός αριθμός των τειχών που μπορούν να προστεθούν στον λαβύρινθο συνολικά. Ακόμα, δηλώνουμε έναν πίνακα tiles με αντικείμενα τύπου Tile και έναν πίνακα supplies με αντικείμενα τύπου Supply. Επιπλέον, φτιάχνουμε τους constructors, τους setters και τους getters των μεταβλητών αυτών, ώστε να αρχικοποιούμε, να τροποποιούμε και να επιστρέφουμε τις τιμές τους. Σε αυτήν την κλάση έχουμε τέσσερις ακόμη μεθόδους:

- ***Συνάρτηση void createTile():***

Αρχικά, σε αυτή τη μέθοδο αρχικοποιούμε τον πίνακα tiles (tileId,x,y,up,down,left,right), όπου:

- τα ids παίρνουν τιμές από 0 έως N*N-1 όπως φαίνεται και στην παρακάτω εικόνα.

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

- το x παίρνει την τιμή id/N

- το y παίρνει την τιμή id%N και

- οι up, down, left, right αρχικοποιούνται ως false.

Επίσης, με την χρήση τεσσάρων διαδοχικών επαναλήψεων *for* δημιουργούμε το περίγραμμα του ταμπλό (σειρά 66 έως 85 στον κώδικα).

Έπειτα, ξεκινάμε την τοποθέτηση των τειχών στο εσωτερικό του ταμπλό με ψευδό-τυχαίο τρόπο. Για αρχή, χρησιμοποιούμε μια μεταβλητή (counter), η οποία αυξάνεται κάθε φορά, που στο πλακίδιο με id (= i) υπάρχει τοίχος. Στη συνέχεια, χρησιμοποιούμε μία δεύτερη μεταβλητή (count), η οποία παίρνει τυχαίες τιμές (από 1 έως 2). Κατόπιν, με την εντολή if ελέγχουμε αν η μεταβλητή count είναι μικρότερη του 2, ώστε να πάρουμε τον περιορισμό των μέγιστων τειχών σε ένα πλακίδιο (max = 2) και με μία εντολή switch τοποθετούμε με τυχαίο τρόπο τους υπόλοιπους τοίχους σε ένα πλακίδιο, ενώ ταυτόχρονα, με τις μεταβλητές, counter2 και counter3, και διαδοχικές if ελέγχουμε (στα αντίστοιχα cases) εάν το γειτονικό πλακίδιο, από αυτό που θέλουμε να τοποθετήσουμε τον τοίχο, ικανοποιεί και αυτό τον περιοριοσμό των λιγότερων απο 2 τειχών. Το πλακίδιο με id (=i), πέρα από αυτόν τον περιορισμό, ελέγχουμε αν υπάρχει ήδη τοίχος στο σημείο που θέλουμε να τον τοποθετήσουμε. Αυτή η διαδικασία επαναλαμβάνεται μέχρι ο αριθμός των τειχών (μεταβλητή numOfWalls) να γίνει ίσος με το 0, καθώς κάθε φορά που τοποθετείται ένα τοίχος σε ένα πλακίδιο και αντίστοιχα ο τοίχος στο γειτονικό πλακίδιο τότε η μεταβλητη αυτή μειώνεται κατά δύο μονάδες.

- ***void createSupply():***

Σε αυτή τη συνάρτηση δημιουργούμε τα εφόδια στο ταμπλό. Χρησιμοποιώντας μια for (σειρά 173 του κώδικα) αρχικοποιούμε τον πίνακα των supplies (σειρά 207). Αρχικά δηλώνουμε τις συντεταγμένες του εφοδίου (x και y) και στη

συνέχεια τις αρχικοποιούμε με τυχαίες τιμές (σειρά 182 και 182). Τη μεταβλητή k τη θέτουμε ίση με το id του εφοδίου που αντιστοιχεί στο τωρινό i και με μια δεύτερη for ελέγχουμε κάθε φορά αν το εφόδιο με το τυχαίο id (=k) ταυτίζεται με κάποιο άλλο id. Η διαδικασία αυτή επαναλαμβάνεται με τη βοήθεια μιας boolean μεταβλητής (=b) η οποία τερματίζει τον βρόγχο επανάληψης do...while.

- ***void createBoard():***

  Σε αυτή τη συνάρτηση δημιουργούμε το ταμπλό του παιχνιδιού με ψευδό-τυχαίο τρόπο, καλώντας τις δύο παραπάνω συναρτήσεις (createTile και createSupply).

- ***String[][] getStringRepresentation(int theseusTile, int minotaurTile):***

Η συνάρτηση αυτή είναι υπεύθυνη για την απεικόνιση του ταμπλό, των εφοδίων και των παικτών. Αρχικά γίνεται η δέσμευση ενός πίνακα με τύπο επιστροφής string και διαστάσεις (2*N+1)xN. Οι δύο πρώτες for (σειρές 225-240 του κώδικα) δημιουργούν το κάτω και πάνω όρισμα του ταμπλό αντίστοιχα.. Στη συνέχεια παίρνουμε μια διπλή for η οποία τρέχει σειριακά μεσα στον πίνακα και τοποθετεί τα δεδομένα μας.Αρχικά γίνεται διάκριση στις άρτιες και περιττές γραμμές του πίνακα. Στις άρτιες γραμμές (σειρά 245 εως 261) γίνεται αρχικά μια διάκριση των στηλών αυτή τη φορά, στην τελευταία στήλη και σε όλες τις υπόλοιπες. Στην περίπτωση που δεν βρισκόμαστε στην τελευταία στήλη, ελέγχουμε σειριακά κάθε πλακίδιο (σημείωση: βρέθηκε η αντιστοιχία του id των πλακιδίων της εικόνας 4 να αντιστοιχεί κάθε φορά στο (i/2)*N+j της κάθε επανάληψης) αν έχει τοίνο στην κάτω πλευρά του και το τοποθετούμε (συμβολικά "+---"). Η αντίστοιχη διαδικασία επαναλαμβάνεται για την τελευταία στήλη με την μόνη διαφορά πως σε αυτή την περίπτωση τοποθετούμε "+---+". Από τη σειρά 262 και εξής περνάμε στις περιττές γραμμές όπου ελέγχουμε την ύπαρξη πλάγιων τοιχών, εφοδίων και παικτών. Και σε αυτή τη περίπτωση γίνεται διαχωρισμός της τελευταίας στήλης με τις υπόλοιπες. Η διαδικασία που επαναλαμβάνεται και στις δυο αυτές περιπτώσεις είναι η ίδια και διαφέρει μόνο η εκτύπωση, που στην περίπτωση της τελευταιας στήλης τυπώνεται στο τέλος ένα επιπλέον "|". Με διαδοχικές *if* ελέγχονται όλες οι πιθανές περιπτώσεις εκτύπωσεις και τυπώνονται τα αντίστοιχα σύμβολα στην οθόνη. Τέλος (σειρές 350 έως 355) τυπώνεται ολόκληρο το ταμπλό με τους τοίχους, τους παίκτες και τα εφόδια ενώ στην σειρά 357 επιστρέφεται ο πίνακας που δημιουργήσαμε.

Κλάση Player

Η κλάση αυτή αντιπροσωπεύει τον κάθε παίκτη του παιχνιδιού. Παίρνει ως ορίσματα τις εξής ακέραιες μεταβλητές: playerId που είναι το id του παίκτη, score που είναι το συνολικό σκορ του κάθε παίκτη, το οποίο αυξάνεται κατά την εύρεση εφοδίων και οι συντεταγμένες x και y. Επίσης, δηλώνεται μια μεταβλητή name τύπου String, που αντιστοιχεί στο όνομα του κάθε παίκτη και μια μεταβλητή board τύπου Board, που αντιστοιχεί στο ταμπλό του παιχνιδιού. Όπως και στις άλλες κλάσεις έτσι και εδώ έχουμε

τους constructors, setters και getters των μεταβλητών. Επίσης σε αυτή την κλάση έχουμε μια ακόμη μέθοδο:

- ***int[] move(int id):***

  Η συνάρτηση αυτή ελέγχει τις κινήσεις των παικτών. Αρχικά δημιουργεί 2 πίνακες τεσσάρων θέσεων και αρχικοποιεί των έναν από αυτούς (τον πίνακα moves) με 1,3,5,7 που αντιστοιχούν στις πιθανές κινήσεις πάνω, δεξιά, κατώ και αριστερά αντίστοιχα. Στη συνέχεια με τη δομή επανάληψης for, την αρχικοποίηση μιας μεταβλητής (randMoves) με τυχαίες τιμές από το 0 εως το 4 και μία switch κινεί τους παίκτες με ψευδο-τυχαίο τρόπο. Τοποθετεί τους τυχαίους αριθμούς στον πίνακα moves και τοποθετεί την αντίστοιχη τιμή που επιστρέφεται (η οποία όπως είπαμε και πριν αντιστοιχεί σε μια κίνηση. Η τιμή 1 αντιστοιχεί στην κίνηση κατά πάνω, η 3 στην κίνηση δεξιά, η 5 στην κίνηση προς τα κάτω και η 7 στην κίνηση κατά αριστερά) στην switch. Μέσα σε κάθε case της *switch* ελέγχεται αρχικά υπάρχει τοίχος που αντιτίθεται στην αντίστοιχη κίνηση του παίκτη. Σε κάθε περίπτωση βγάζει αντίστοιχο μήνυμα για το αν είναι εφικτή ή όχι η κίνηση του παίκτη ενώ στην περίπτωση που είναι εφικτή, τυπώνονται οι καινούργιες συντεταγμένες της θέσης του παίκτη. Οι 3 πρώτες θέσεις του πίνακα array αρχικοποιούνται με τις νέες συντεταγμένες του παίκτη X και Y και με το νέο Id του αντίστοιχα. Τέλος μέσα σε κάθε case ελέγχεται (με τη χρήση της δομής επανάληψης for) αν στο νέο πλακίδιο που βρέθηκε ο παίκτης υπάρχει εφόδιο και αν ο παίκτης που βρέιθηκε εκεί είναι ο Θησέας έτσι ώστε να το συλλέξει.. Το Id αυτού του εφοδίου αποθηκεύεται στην 4η θέση του πίνακα array και στη συνέχεια το εφόδιο εξαφανίζεται από το ταμπλό, αφού του εφαρμόζεται τιμή έξω από τα όρια του πίνακα (πχ σειρά 121 board.supplies[j].setSupplyTileId(-1);). Στο τέλος επιστρέφεται ο πίνακας array. Η ίδια διαδικασία εφαρμόζεται για όλα τα cases.

Κλάση Game:

Σε αυτήν την τελευταία κλάση του προγράμματος, έχουμε ως όρισμα μία ακέραια μεταβλητή μόνο, τη round, που αντιστοιχεί στον τρέχον γύρο του παιχνιδιού. Όπως και στις προηγούμενες κλάσεις έτσι και εδώ έχουμε τους constructors, setters και getters των μεταβλητών. Τέλος, σε αυτή την κλάση έχουμε και τη βασική συνάρτηση του προγράμματος, main, η οποία είναι υπεύθυνη για τη δημιουργία, την ορθή εξέλιξη και την λήξη του παιχνιδιού.

- ***public static void main(String[] args):***

  Αρχικά δηλώνει μεταβλητές και τις αρχικοποιεί σύμφωνα με τα ζητούμενα των οδηγιών (N=15 η διάσταση του ταμπλό, S=4 τα εφόδια, W=(3*N*N+1)/2 τα συνολικά τείχη, n=100 για τις ζαριές των παικτών οι οποίες είναι 2n και μια Boolean μεταβλητή η οποία τερματίζει τον βρόγχο do...while που

χρησιμοποιούμε. Στη συνέχεια δημιουργείται το ταμπλό με τα αντίστοιχα ορίσματα N,S,W και οι 2 παίκτες. Γίνεται μια εκτύπωση του ταμπλό για την αρχική του κατάσταση, πριν δηλαδή οι παίκτες αρχίσουν να ρίχνουν τα ζάρια και να κινούνται πάνω στο ταμπλό.. Η μεταβλητή i αρχικοποιείται με 0 και μεσα στη δομή επανάληψης αυξάνεται κατά ένα έτσι ώστε να αντιπροσωπεύει τον αντίστοιχο γύρο του παιχνιδιού. Επιπλέον σε κάθε επανάληψη (άρα και σε κάθε γύρο) εκτυπώνονται ο τρέχον γύρος, οι κίνηση του κάθε παίκτη, το ταμπλό με τις θέσεις των παικτών και των εφοδίων ενώ στο τέλος του παιχνιδιού υπάρχει εκτύπωση που δηλώνει τον παίκτη που νίκησε ή την ισοπαλία στην περίπτωση που έχουν παρέλθει 2n ζαρίες και δεν έχει νικήσει κανένας από τους 2!