

Ο Θησέας και ο Μινώταυρος



Τζαμτζής Μάριος 10038 6949214612 tzamtzis@ece.auth.gr

Τσίπης Παντελής 10224 6947548593 panttsip@ece.auth.gr

Το πρόγραμμα έχει χωριστεί σε 5 κλάσεις:

A) Supply

B) Tile

Γ) Board

Δ) Player

E) Game

Κλάση Supply: Αρχικά ορίζουμε τις 4 μεταβλητές(τύπου int). Έπειτα δημιουργούμε τους 3 constructors, ο πρώτος είναι ο default, στον οποίο αρχικοποιούνται όλες οι μεταβλητές με την μηδενική τιμή, ο δεύτερος είναι αυτός που λαμβάνει ως ορίσματα τις μεταβλητές και τις αρχικοποιεί αντιστοίχως. Ο τρίτος είναι ο copy constructor, ο οποίος λαμβάνει ως όρισμα ένα αντικείμενο τύπου Supply και αντιγράφει τις μεταβλητές του αντικειμένου που έχει δεχτεί ως όρισμα στις μεταβλητές του καινούριου αντικειμένου. Τέλος, ορίζουμε τους setters και τους getters όλων των μεταβλητών.

Κλάση Tile: Αρχικά ορίζουμε τις 7 μεταβλητές(3 τύπου int και 4 τύπου Boolean). Έπειτα δημιουργούμε τους 3 constructors, ο πρώτος είναι ο default, στον οποίο αρχικοποιούνται οι μεταβλητές τύπου int με την μηδενική τιμή και οι μεταβλητές τύπου Boolean με false, ο δεύτερος είναι αυτός που λαμβάνει ως ορίσματα τις μεταβλητές και τις αρχικοποιεί αντιστοίχως. Ο τρίτος είναι ο copy constructor, ο οποίος λαμβάνει ως όρισμα ένα αντικείμενο τύπου Tile και αντιγράφει τις μεταβλητές του αντικειμένου που έχει δεχτεί ως όρισμα στις μεταβλητές του καινούριου αντικειμένου. Τέλος, ορίζουμε τους setters και τους getters όλων των μεταβλητών.

Κλάση Board: Αρχικά ορίζουμε 3 μεταβλητές(τύπου int), έναν πίνακα τύπου Tile και έναν πίνακα τύπου Supply. Έπειτα δημιουργούμε τους 3 constructors, ο πρώτος είναι ο default, στον οποίο αρχικοποιούνται οι μεταβλητές τύπου int με την μηδενική τιμή, ο δεύτερος είναι αυτός που λαμβάνει ως ορίσματα τις μεταβλητές τύπου int και τις αρχικοποιεί αντιστοίχως, ακόμη δημιουργούμε έναν πίνακα τύπου Supply μεγέθους S και έναν πίνακα τύπου Tile μεγέθους $N*N$, ο οποίος αρχικοποιείται. Ο τρίτος είναι ο copy constructor, ο οποίος λαμβάνει ως όρισμα ένα αντικείμενο τύπου Board και αντιγράφει τις μεταβλητές και τους πίνακες του αντικειμένου που έχει δεχτεί ως όρισμα στις μεταβλητές και τους πίνακες του καινούριου αντικειμένου αντίστοιχα. Επιπλέον, ορίζουμε τους setters και τους getters όλων των μεταβλητών και των πινάκων.

Στη συνέχεια ορίζουμε την συνάρτηση **createTile()** με την οποία αρχικοποιούνται τα αντικείμενα του πίνακα tiles με τυχαίο τρόπο. Κατ'αρχάς αρχικοποιείται η μεταβλητή w η οποία προκύπτει αν αφαιρέσουμε από τα συνολικά τείχη(W) τα εξωτερικά($4*N$ =περίμετρος) και την μεσοτοιχία την παίρνουμε ως ένα τείχος, ακόμη έχουμε τις 4 κατευθύνσεις(up=0,down=1,left=2,right=3) και μία θα επιλέγεται τυχαία μέσω της Random και έχουμε και το RandomId για να επιλέγεται ένα τυχαίο TileId. Μετά στην πρώτη σειρά βάζουμε σε όλα τα tiles κάτω τείχος, στην τελευταία σειρά βάζουμε σε όλα τα tiles πάνω τείχος, στην πρώτη στήλη βάζουμε σε όλα τα tiles αριστερό τείχος και στην τελευταία στήλη βάζουμε σε όλα τα tiles δεξιό τείχος. Στη συνέχεια δημιουργούμε μια for για $i=0$ μέχρι w αφού θέλουμε να μπουν στην board το μέγιστο W τείχη. Μέσα στην for επιλέγουμε ένα τυχαίο id και μια τυχαία κατεύθυνση, αν είμαστε στο πρώτο tile ή στο τελευταίο tile της πρώτης σειράς ή στο πρώτο tile της τελευταίας σειράς ή στο τελευταίο tile, τα οποία έχουν ήδη 2 τείχη συνεχίζουμε στο επόμενο $i(continue)$. Αλλιώς ελέγχουμε με τις 4 συνεχόμενες if και με την βοήθεια της μεταβλητής temp αν στο tile που είμαστε έχουμε δυο τείχη, αν ναι κάνουμε continue. Αν όχι ελέγχουμε ποια κατεύθυνση επιλέχτηκε τυχαία για να βάλουμε το τείχος. Αν επιλέχτηκε τυχαία η πάνω κατεύθυνση ελέγχουμε αν το tile είναι στην τελευταία σειρά κι αν ναι continue αφού υπάρχει ήδη τείχος πάνω, αλλιώς αν το από πάνω tile έχει τείχος κάτω βάζουμε στο tile που είμαστε ένα τείχος πάνω και continue, αλλιώς με τις 4 διαδοχικές if και με την βοήθεια της μεταβλητής temp2 ελέγχουμε αν το από πάνω tile έχει δύο τείχη(τα τείχη που μπορεί να έχει το από πάνω είναι δεξιό, αριστερό, πάνω), αν ναι continue αφού αν βάλουμε τείχος το από πάνω θα έχει 3, αλλιώς

βάζουμε στο tile που είμαστε πάνω τείχος και στο από πάνω, κάτω τείχος. Η λογική και για τις άλλες 3 κατευθύνσεις είναι η ίδια απλά θα τις καταγράψουμε για να είναι πιο ολοκληρωμένη η αναφορά. Αν επιλέχτηκε τυχαία η κάτω κατεύθυνση ελέγχουμε αν το tile είναι στην πρώτη σειρά κι αν ναι continue αφού υπάρχει ήδη τείχος κάτω, αλλιώς αν το από κάτω tile έχει τείχος πάνω βάζουμε στο tile που είμαστε ένα τείχος κάτω και continue, αλλιώς με τις 4 διαδοχικές if και με την βοήθεια της μεταβλητής temp2 ελέγχουμε αν το από κάτω tile έχει δύο τείχη(τα τείχη που μπορεί να έχει το από κάτω είναι δεξιό, αριστερό, κάτω), αν ναι continue αφού αν βάλουμε τείχος το από κάτω θα έχει 3, αλλιώς βάζουμε στο tile που είμαστε κάτω τείχος και στο από κάτω, πάνω τείχος. Αν επιλέχτηκε τυχαία η αριστερή κατεύθυνση ελέγχουμε αν το tile είναι στην πρώτη στήλη κι αν ναι continue αφού υπάρχει ήδη τείχος αριστερά, αλλιώς αν το προηγούμενο tile έχει τείχος δεξιά βάζουμε στο tile που είμαστε ένα τείχος αριστερά και continue, αλλιώς με τις 4 διαδοχικές if και με την βοήθεια της μεταβλητής temp2 ελέγχουμε αν το προηγούμενο tile έχει δύο τείχη(τα τείχη που μπορεί να έχει το προηγούμενο είναι πάνω, αριστερό, κάτω), αν ναι continue αφού αν βάλουμε τείχος το προηγούμενο θα έχει 3, αλλιώς βάζουμε στο tile που είμαστε αριστερό τείχος και στο προηγούμενο δεξιό τείχος. Αν επιλέχτηκε τυχαία η δεξιά κατεύθυνση ελέγχουμε αν το tile είναι στην τελευταία στήλη κι αν ναι continue αφού υπάρχει ήδη τείχος δεξιά, αλλιώς αν το επόμενο tile έχει τείχος αριστερά βάζουμε στο tile που είμαστε ένα τείχος δεξιά και continue, αλλιώς με τις 4 διαδοχικές if και με την βοήθεια της μεταβλητής temp2 ελέγχουμε αν το επόμενο tile έχει δύο τείχη(τα τείχη που μπορεί να έχει το επόμενο είναι πάνω, δεξιό, κάτω), αν ναι continue αφού αν βάλουμε τείχος το επόμενο θα έχει 3, αλλιώς βάζουμε στο tile που είμαστε δεξιό τείχος και στο επόμενο αριστερό τείχος.

Η επόμενη συνάρτηση που ορίζουμε είναι η **createSupply()** με την οποία αρχικοποιούνται τα αντικείμενα του πίνακα supplies με τυχαίο τρόπο. Αρχικά δημιουργούμε έναν πίνακα τύπου int μεγέθους S. Μπαίνουμε μέσα στην for για $i=0$ μέχρι S, βάζουμε τυχαία τιμή(από το 1 έως και το $N*N-1$) στην θέση του πίνακα με δείκτη i, στη συνέχεια ελέγχουμε εάν η τιμή που δώσαμε είναι ίση με την αρχική θέση του μινώταυρου($N*N/2$), αν ναι ξαναδίνουμε μια τυχαία τιμή μέχρις ότου να μην είναι ίση με το $N*N/2$. Ύστερα ελέγχουμε αν το στοιχείο στο οποίο είμαστε έχει την ίδια τιμή με κάποιο άλλο στοιχείο του πίνακα counter. Τέλος κι αφού έχουμε δώσει τις κατάλληλες τιμές στον

πίνακα counter αρχικοποιείται κατάλληλα ο πίνακας supplies(εύκολα προκύπτει ότι $x=i \div N$ και $y= i \bmod N$).

Η επόμενη συνάρτηση είναι η **createBoard()**, δημιουργεί το ταμπλό του παιχνιδιού με ψευδό-τυχαίο τρόπο, χρησιμοποιώντας τις παραπάνω συναρτήσεις(**createTile()** και **createSupply()**).

Η τελευταία συνάρτηση της κλάσης Board είναι η **getStringRepresentation**, η οποία δημιουργεί και επιστρέφει έναν πίνακα διάστασης $(2*N + 1) \times N$. Αρχικά δημιουργούμε έναν δισδιάστατο πίνακα(array) τύπου String μεγέθους $(2*N + 1) \times N$. Γενικά, στις γραμμές του array, οι οποίες αντιστοιχούν σε ζυγό αριθμό βάζουμε είτε τα οριζόντια τείχη είτε τα «οριζόντια» κενά, ενώ στις γραμμές του array, οι οποίες αντιστοιχούν σε περιττό αριθμό βάζουμε είτε τα κάθετα τείχη είτε τα σύμβολα για τα λάφυρα είτε το σύμβολο για τον Θησέα είτε το σύμβολο για τον Μινώταυρο είτε τα «κάθετα» κενά. Δημιουργούμε την τελευταία γραμμή του board, και ελέγχουμε αν είμαστε στο τελευταίο tile της γραμμής αυτής ή όχι και βάζουμε ανάλογα τα οριζόντια τείχη. Μπαίνουμε στην διπλή for, αρχικά ελέγχουμε αν το tile έχει κάτω τείχος, αν ναι βάζουμε την κατάλληλη συμβολοσειρά για το οριζόντιο τείχος στον array, αλλιώς βάζουμε την κατάλληλη συμβολοσειρά της μη ύπαρξης τείχους σε αυτήν την θέση. Εκεί τελειώνει η υλοποίηση των σειρών, οι οποίες αντιστοιχούν σε ζυγούς αριθμούς και αρχίζει η υλοποίηση των σειρών, οι οποίες αντιστοιχούν σε περιττούς αριθμούς. Αν το tile είναι στην τελευταία στήλη, ελέγχουμε αν υπάρχει τείχος στα αριστερά, αν ναι βάζουμε την κατάλληλη συμβολοσειρά για το κάθετο τείχος στον array, αλλιώς αφήνουμε κενό. Μετά δίνουμε την κατάλληλη τιμή στην μεταβλητή tempSupply, η οποία χρειάζεται για να βάλουμε τα supplies στον array και έπειτα αν στο tile είναι ο Μινώταυρος και ένα λάφυρο βάζουμε στον array την κατάλληλη συμβολοσειρά για τον Μινώταυρο και το λάφυρο, αν στο tile είναι ο Μινώταυρος βάζουμε στον array την κατάλληλη συμβολοσειρά για τον μινώταυρο, αν είναι ο Θησέας βάζουμε στον array την κατάλληλη συμβολοσειρά για τον Θησέα, αν είναι ένα λάφυρο βάζουμε στον array την κατάλληλη συμβολοσειρά για το λάφυρο(με την βοήθεια του tempSupply), αλλιώς αν είναι κενό το tile βάζουμε την κατάλληλη συμβολοσειρά για το κενό. Στο τέλος και επειδή είμαστε στην τελευταία στήλη βάζουμε και την κατάλληλη συμβολοσειρά για ένα ακόμη κάθετο τείχος. Διαφορετικά, αν το tile δεν είναι στην τελευταία στήλη ακολουθούμε την ίδια διαδικασία απλώς στο τέλος δεν βάζουμε ένα ακόμη κάθετο τείχος εφόσον δεν είμαστε στην τελευταία στήλη. Τέλος, ελέγχουμε αν είμαστε στην αρχή του παιχνιδιού(ο Θησέας είναι στην θέση με tileId=0 και ο

μινώταυρος στην θέση με $\text{tileId} = N * N / 2$) κι αν ναι αφήνουμε κενό το κάτω μέρος του πρώτου tile σαν να μην υπάρχει τείχος. Στο τέλος επιστρέφουμε τον array.

Κλάση Player: Αρχικά ορίζουμε τις 6 μεταβλητές (4 τύπου int, 1 τύπου String και 1 τύπου Board). Έπειτα δημιουργούμε τους 3 constructors, ο πρώτος είναι ο default, στον οποίο αρχικοποιούνται οι μεταβλητές τύπου int με την μηδενική τιμή και η μεταβλητή τύπου String με μία κενή συμβολοσειρά, ο δεύτερος είναι αυτός που λαμβάνει ως όρισμα τις μεταβλητές και τις αρχικοποιεί αντιστοίχως. Ο τρίτος είναι ο copy constructor, ο οποίος λαμβάνει ως όρισμα ένα αντικείμενο τύπου Player και αντιγράφει τις μεταβλητές του αντικειμένου που έχει δεχτεί ως όρισμα στις μεταβλητές του καινούριου αντικειμένου. Επίσης, ορίζουμε τους setters και τους getters όλων των μεταβλητών.

Τέλος, ορίζουμε την συνάρτηση **move**, η οποία πραγματοποιεί την πιθανή κίνηση του παίκτη. Αρχικά, δημιουργούμε έναν πίνακα τύπου int και μεγέθους 4, ο οποίος στο τέλος θα επιστραφεί. Στη συνέχεια ορίζουμε τις 4 κατευθύνσεις ($\text{down}=0, \text{left}=1, \text{up}=2, \text{right}=3$). Ύστερα ελέγχουμε αν επιλέχτηκε τυχαία η κάτω κατεύθυνση, αν ναι ελέγχουμε αν υπάρχει κάτω τείχος κι αν ναι δεν μπορεί να κάνει αυτήν την κίνηση και εμφανίζει στην οθόνη το αντίστοιχο μήνυμα. Αλλιώς κάνει την κίνηση προς τα κάτω, εμφανίζει στην οθόνη το αντίστοιχο μήνυμα και βάζουμε τις καινούριες τιμές (TileId, x, y) στις αντίστοιχες θέσεις του array. Έπειτα ελέγχουμε αν παίζει ο Θησέας και αν υπάρχει κάποιο λάφυρο στην καινούρια θέση, αν ναι εμφανίζει στην οθόνη το αντίστοιχο μήνυμα, το λάφυρο αυτό εξαφανίζεται από το ταμπλό, αυξάνεται κατά ένα το σκορ του Θησέα και βάζουμε στην τέταρτη θέση του array το αναγνωριστικό του λαφύρου το οποίο συλλέχτηκε. Αν δεν υπάρχει λάφυρο στην καινούρια θέση, εμφανίζεται το κατάλληλο μήνυμα. Η λογική και για τις άλλες 3 κατευθύνσεις είναι η ίδια απλά θα τις καταγράψουμε για να είναι πιο ολοκληρωμένη η αναφορά. Ελέγχουμε αν επιλέχτηκε τυχαία η αριστερή κατεύθυνση, αν ναι ελέγχουμε αν υπάρχει αριστερό τείχος κι αν ναι δεν μπορεί να κάνει αυτήν την κίνηση και εμφανίζει στην οθόνη το αντίστοιχο μήνυμα. Αλλιώς κάνει την κίνηση προς τα αριστερά, εμφανίζει στην οθόνη το αντίστοιχο μήνυμα και βάζουμε τις καινούριες τιμές (TileId, x, y) στις αντίστοιχες θέσεις του array. Έπειτα ελέγχουμε αν παίζει ο Θησέας και αν υπάρχει κάποιο λάφυρο στην καινούρια θέση, αν ναι εμφανίζει στην οθόνη το αντίστοιχο μήνυμα, το λάφυρο αυτό εξαφανίζεται από το ταμπλό, αυξάνεται κατά ένα το σκορ του Θησέα και βάζουμε στην τέταρτη θέση του array το

αναγνωριστικό του λαφύρου το οποίο συλλέχτηκε. Αν δεν υπάρχει λάφυρο στην καινούρια θέση, εμφανίζεται το κατάλληλο μήνυμα. Ελέγχουμε αν επιλέχτηκε τυχαία η πάνω κατεύθυνση, αν ναι ελέγχουμε αν υπάρχει πάνω τείχος κι αν ναι δεν μπορεί να κάνει αυτήν την κίνηση και εμφανίζει στην οθόνη το αντίστοιχο μήνυμα. Αλλιώς κάνει την κίνηση προς τα πάνω, εμφανίζει στην οθόνη το αντίστοιχο μήνυμα και βάζουμε τις καινούριες τιμές(TileId,x,y) στις αντίστοιχες θέσεις του array. Έπειτα ελέγχουμε αν παίζει ο Θησέας και αν υπάρχει κάποιο λάφυρο στην καινούρια θέση, αν ναι εμφανίζει στην οθόνη το αντίστοιχο μήνυμα, το λάφυρο αυτό εξαφανίζεται από το ταμπλό, αυξάνεται κατά ένα το σκορ του Θησέα και βάζουμε στην τέταρτη θέση του array το αναγνωριστικό του λαφύρου το οποίο συλλέχτηκε. Αν δεν υπάρχει λάφυρο στην καινούρια θέση, εμφανίζεται το κατάλληλο μήνυμα. Τέλος, ελέγχουμε αν επιλέχτηκε τυχαία η δεξιά κατεύθυνση, αν ναι ελέγχουμε αν υπάρχει δεξιό τείχος κι αν ναι δεν μπορεί να κάνει αυτήν την κίνηση και εμφανίζει στην οθόνη το αντίστοιχο μήνυμα. Αλλιώς κάνει την κίνηση προς τα δεξιά, εμφανίζει στην οθόνη το αντίστοιχο μήνυμα και βάζουμε τις καινούριες τιμές(TileId,x,y) στις αντίστοιχες θέσεις του array. Έπειτα ελέγχουμε αν παίζει ο Θησέας και αν υπάρχει κάποιο λάφυρο στην καινούρια θέση, αν ναι εμφανίζει στην οθόνη το αντίστοιχο μήνυμα, το λάφυρο αυτό εξαφανίζεται από το ταμπλό, αυξάνεται κατά ένα το σκορ του Θησέα και βάζουμε στην τέταρτη θέση του array το αναγνωριστικό του λαφύρου το οποίο συλλέχτηκε. Αν δεν υπάρχει λάφυρο στην καινούρια θέση, εμφανίζεται το κατάλληλο μήνυμα.

Κλάση Game: Αρχικά ορίζουμε την μία και μοναδική μεταβλητή(τύπου int). Έπειτα δημιουργούμε τους 3 constructors, ο πρώτος είναι ο default, στον οποίο αρχικοποιείται η μεταβλητή με την μηδενική τιμή, ο δεύτερος είναι αυτός που λαμβάνει ως όρισμα την μεταβλητή και την αρχικοποιεί. Ο τρίτος είναι ο copy constructor, ο οποίος λαμβάνει ως όρισμα ένα αντικείμενο τύπου Game και αντιγράφει την μεταβλητή του αντικειμένου που έχει δεχτεί ως όρισμα στην μεταβλητή του καινούριου αντικειμένου. Ακόμη, ορίζουμε τον setter και τον getter της μεταβλητής.

Τέλος, ορίζουμε την συνάρτηση **main**, η οποία εκκινεί το παιχνίδι. Αρχικά, δημιουργούμε ένα αντικείμενο τύπου Game(αρχικοποιείται το round σε 1). Στη συνέχεια δημιουργούμε ένα ταμπλό διάστασης 15X15, εισάγουμε 4 εφόδια σε αυτό και τα συνολικά τείχη που μπορούν να δημιουργηθούν είναι $(15*15*3+1)/2=338$. Μετά δημιουργούμε τους 2 παίκτες, ο πρώτος είναι ο Θησέας ο οποίος ξεκινάει από το tileId=0 και ο δεύτερος είναι ο Μινώταυρος

ο οποίος ξεκινάει από το $\text{tileId}=15*15/2=112$. Έπειτα μπαίνουμε στην for η οποία τερματίζει όταν ένας από τους 2 παίκτες κερδίσει ή παρέλθουν οι 200 ζαριές(100 γύροι). Μέσα στην for εκτυπώνουμε τον γύρο στον οποίον είναι το παιχνίδι, το ταμπλό του παιχνιδιού με τους παίκτες και τα λάφυρα και μετά τις κινήσεις του κάθε παίκτη. Πρώτος παίζει ο Θησέας(με την **move**), ελέγχεται αν μετά την κίνηση του Θησέα, ο Θησέας και ο Μινώταυρος βρίσκονται στο ίδιο tile κι αν ναι βγαίνει το κατάλληλο μήνυμα ο Μινώταυρος σκοτώνει τον Θησέα και κερδίζει άρα το παιχνίδι τελειώνει, αλλιώς αν ο Θησέας έχει μαζέψει και τα 4 λάφυρα τότε βγαίνει το κατάλληλο μήνυμα ανοίγει μια καταπακτή, ο Μινώταυρος πέφτει μέσα και ο Θησέας κερδίζει άρα τελειώνει το παιχνίδι. Αλλιώς δεύτερος παίζει ο Μινώταυρος(με την **move**). Αυξάνουμε τον αριθμό των γύρων κατά 1. Ελέγχεται αν μετά την κίνηση του Μινώταυρου, ο Θησέας και ο Μινώταυρος βρίσκονται στο ίδιο tile κι αν ναι βγαίνει το κατάλληλο μήνυμα ο Μινώταυρος σκοτώνει τον Θησέα και κερδίζει άρα το παιχνίδι τελειώνει, αλλιώς αν το παιχνίδι βρίσκεται στον τελευταίο γύρο(γύρος 100) βγαίνει το κατάλληλο μήνυμα και το παιχνίδι λήγει ισόπαλο και στο τέλος σε κάθε περίπτωση(νίκη Θησέα, νίκη Μινώταυρου, ισοπαλία) εκτυπώνεται η τελική κατάσταση του ταμπλό. Αλλιώς συνεχίζεται ο βρόγχος.

Τζαμτζής Μάριος 10038 6949214612 tzamtzis@ece.auth.gr

Τσίπης Παντελής 10224 6947548593 panttsip@ece.auth.gr

