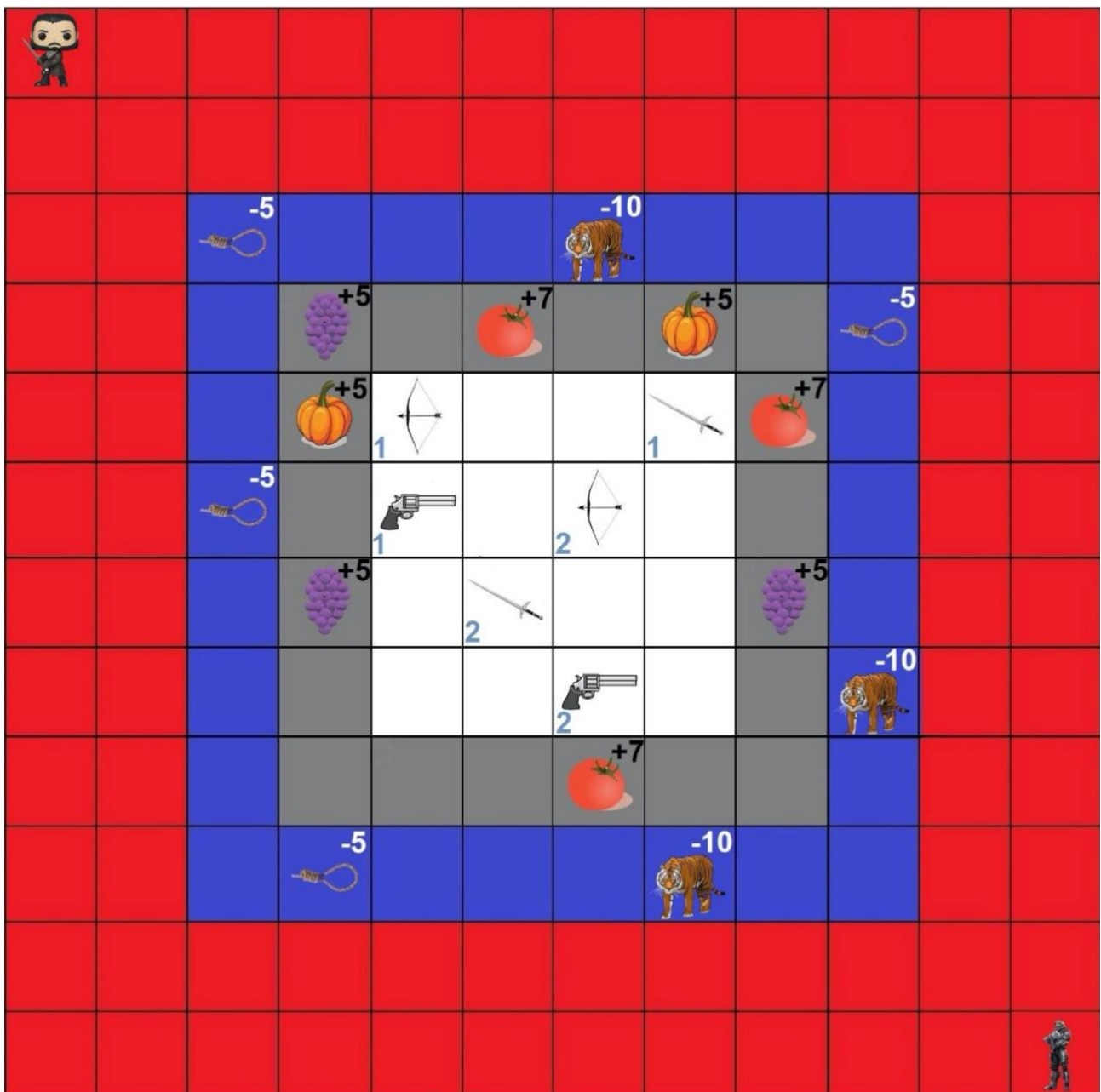


HUNGER GAMES

1ο Παραδοτέο Εργασίας στις Δομές Δεδομένων 2019/20.

ΟΜΑΔΑ 11:

Καλαμάρας Κωνσταντίνος 9716
Καϊμακαμίδης Ανέστης 9627



ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ:

Ο συνολικός στόχος της εργασίας αυτής είναι η δημιουργία μιας προσομοίωσης του παιχνιδιού Hunger Games σε περιβάλλον JAVA. Το παιχνίδι παίζεται από δυο παίκτες , οι οποίοι καλούνται να αποφύγουν τα εμπόδια που παρουσιάζονται στον δρόμο τους , συλλέγοντας παράλληλα εφόδια και όπλα, στοιχεία απαραίτητα για την επιβίωση τους και την εξόντωση του αντιπάλου αντίστοιχα. Κάθε 3 γύρους , όμως , τα περιθώρια για αυτούς στενεύουν , καθώς το ταμπλό του παιχνιδιού μικραίνει και νιώθουν όλο και πιο “καυτή” την ανάσα του αντιπάλου τους. Νικητής κρίνεται εκείνος που θα σκοτώσει τον αντίπαλό του ή (εφόσον δεν προλάβει κάποιος από τους δυο να το κάνει) εκείνος που έχει συλλέξει τους περισσότερους πόντους ,έως ότου το ταμπλό του παιχνιδιού να έχει διαστάσεις 4x4.

Στο συγκεκριμένο παραδοτέο της εργασίας ζητείται η δημιουργία του ταμπλό του παιχνιδιού , ο εμπλουτισμός του με όπλα , εφόδια και παγίδες σε συγκεκριμένα σημεία , καθώς και η ανάπτυξη μιας στρατηγικής τυχαίας κίνησης ενός παίκτη πάνω σε αυτό.

Οι κλάσεις που ζητείται να δημιουργηθούν είναι οι : Weapon , Food , Trap , Board , Player , Game και θα αναλυθεί εκτενώς το περιεχόμενο και η λειτουργία τους στην συνέχεια της αναφοράς αυτής.

ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ-ΑΛΓΟΡΙΘΜΟΥ-ΔΙΑΔΙΚΑΣΙΩΝ:

Κλάση Weapon

Η κλάση Weapon αντιπροσωπεύει το όπλο και έχει τις εξής μεταβλητές:

int id: το id του όπλου.

int x: η συντεταγμένη x του πλακιδίου του ταμπλό όπου βρίσκεται το όπλο.

int y: η συντεταγμένη y του πλακιδίου του ταμπλό όπου βρίσκεται το όπλο.

int playerId: το id του παίκτη στον οποίο ανήκει το όπλο.

String type: το είδους του όπλου (pistol , bow ή sword).

Οι μέθοδοι που υλοποιούνται σε αυτήν την κλάση είναι :

Δύο **constructor** που αρχικοποιούν τις μεταβλητές της κλάσης . Ο πρώτος δέχεται ως όρισμα την κάθε μεταβλητή ξεχωριστά , ενώ ο δεύτερος ένα στιγμιότυπο αντικειμένου τύπου Weapon.

Οι **getters** και οι **setters** όλων των μεταβλητών , έτσι ώστε να μπορεί ο χρήστης να τις προσπελάσει και εκτός της κλάσης τους.

Κλάση Food

Η κλάση **Food** αντιπροσωπεύει το εφόδιο και έχει τις εξής μεταβλητές:

int id: το id του εφοδίου.

int x: η συντεταγμένη x του πλακιδίου του ταμπλό όπου βρίσκεται το εφόδιο.

int y: η συντεταγμένη y του πλακιδίου του ταμπλό όπου βρίσκεται το εφόδιο.

int points: οι πόντοι που κερδίζει ο παίκτης που τρώει το εφόδιο.

Οι μέθοδοι που υλοποιούνται σε αυτήν την κλάση είναι :

Δυο **constructor** που αρχικοποιούν τις μεταβλητές της κλάσης . Ο πρώτος δέχεται ως όρισμα την κάθε μεταβλητή ξεχωριστά , ενώ ο δεύτερος ένα στιγμιότυπο αντικειμένου τύπου Food.

Οι **getters** και οι **setters** όλων των μεταβλητών , έτσι ώστε να μπορεί ο χρήστης να τις προσπελάσει και εκτός της κλάσης τους.

Κλάση Trap

Η κλάση **Trap** αντιπροσωπεύει την παγίδα και έχει τις εξής μεταβλητές:

int id: το id της παγίδας.

int x: η συντεταγμένη x του πλακιδίου του ταμπλό όπου βρίσκεται η παγίδα.

int y: η συντεταγμένη y του πλακιδίου του ταμπλό όπου βρίσκεται η παγίδα.

String type: το είδος της παγίδας (ropes ή animals).

int points: οι πόντοι που χάνει ο παίκτης όταν πέσει στην παγίδα και δεν έχει το κατάλληλο όπλο για να την αντιμετωπίσει.

Οι μέθοδοι που υλοποιούνται σε αυτήν την κλάση είναι :

Δυο **constructor** που αρχικοποιούν τις μεταβλητές της κλάσης . Ο πρώτος δέχεται ως όρισμα την κάθε μεταβλητή ξεχωριστά , ενώ ο δεύτερος ένα στιγμιότυπο αντικειμένου τύπου Trap.

Οι **getters** και οι **setters** όλων των μεταβλητών , έτσι ώστε να μπορεί ο χρήστης να τις προσπελάσει και εκτός της κλάσης τους.

Κλάση Board

Η κλάση Board αντιπροσωπεύει το ταμπλό του παιχνιδιού και έχει τις εξής μεταβλητές:

int N, M: οι διαστάσεις του ταμπλό NxM

int W, F, T: ο αριθμός των όπλων, των εφοδίων και των παγίδων που υπάρχουν στο ταμπλό.

int[4][2] weaponAreaLimits: ένας δισδιάστατος πίνακας 4x2 που περιέχει τα (x,y) όρια της περιοχής του ταμπλό όπου υπάρχουν τα όπλα.

int[4][2] foodAreaLimits: ένας δισδιάστατος πίνακας 4x2 που περιέχει τις συντεταγμένες (x,y) των γωνιακών πλακιδίων της περιοχής του ταμπλό όπου υπάρχουν τα εφόδια.

int[4][2] trapAreaLimits: ένας δισδιάστατος πίνακας 4x2 που περιέχει τις συντεταγμένες (x,y) των γωνιακών πλακιδίων της περιοχής του ταμπλό όπου υπάρχουν οι παγίδες.

Weapon[] weapons: ένας πίνακας με αντικείμενα τύπου Weapon.

Food[] food: ένας πίνακας με αντικείμενα τύπου Food.

Trap[] traps: ένας πίνακας με αντικείμενα τύπου Trap.

Οι μέθοδοι που υλοποιούνται σε αυτήν την κλάση είναι :

Οι getters και οι setters όλων των μεταβλητών , έτσι ώστε να μπορεί ο χρήστης να τις προσπελάσει και εκτός της κλάσης τους.

Τρεις constructor που αρχικοποιούν τις μεταβλητές της κλάσης . Ο πρώτος δεν δέχεται ορίσματα και αρχικοποιεί όλες τις μεταβλητές με την τιμή 0, ο δεύτερος δέχεται ως όρισμα τις μεταβλητές N,M,T,F,W με απαραίτητη προϋπόθεση το N να ισούται με το M, ενώ ο τρίτος ένα στιγμιότυπο αντικειμένου τύπου Board.

Η συνάρτηση void() createRandomWeapon :

Περιέχει τις μεταβλητές :

String type1: το είδος του όπλου που δημιουργείται

int x,y: οι τυχαίες συντεταγμένες του όπλου μέσα στα όρια τοποθέτησης όπλων

int[][] oldxy: ένας πίνακας με τις συντεταγμένες των προηγούμενων όπλων που έχουν δημιουργηθεί, έτσι ώστε να μην υπάρχουν περισσότερα από ένα όπλα στην ίδια θέση

boolean ok: ελέγχει την do-while εντολή που χρησιμοποιείται για την αποφυγή ύπαρξης πολλών όπλων στην ίδια θέση

int randtype1,2,3: η μεταβλητή που ελέγχει ποιο είδος θα δοθεί στο κάθε όπλο

Η λειτουργία της με λίγα λόγια είναι η εξής:

Αρχικά, αρχικοποιεί όλα τα στοιχεία του πίνακα `oldxy` με την τιμή 0 και ξεκινάει να δίνει τυχαίες τιμές στις συντεταγμένες του πρώτου όπλου που δημιουργούνται, με τις τιμές αυτές να ανήκουν ,φυσικά, στα επιτρεπόμενα όρια. Στην συνέχεια, ελέγχει αν οι συντεταγμένες αυτές έχουν δοθεί σε άλλο όπλο και εφόσον αυτό δεν συμβαίνει, τις αποθηκεύουν στον πίνακα `oldxy`. Η διαδικασία αυτή επαναλαμβάνεται για κάθε παραγόμενο όπλο. Τέλος, μεταβιβάζει σε κάθε παίκτη όπλα και από τα τρία διαθέσιμα είδη με τέτοιο τρόπο, ώστε να εξασφαλίζεται η ύπαρξη όπλων όλων των ειδών και η ισομερής κατανομή τους και τοποθετεί τα όπλα στον πίνακα `Weapon[] weapons`.

Η συνάρτηση **void() createRandomTrap** :

Περιέχει τις μεταβλητές :

String type1: το είδος της παγίδας που δημιουργείται

int x,y: οι τυχαίες συντεταγμένες της παγίδας μέσα στα όρια τοποθέτησης παγίδων

int[][] oldxy: ένας πίνακας με τις συντεταγμένες των προηγούμενων παγιδών που έχουν δημιουργηθεί, έτσι ώστε να μην υπάρχουν περισσότερες από μια παγίδες στην ίδια θέση

boolean ok: ελέγχει την do-while εντολή που χρησιμοποιείται για την αποφυγή ύπαρξης πολλών παγιδών στην ίδια θέση.

int point: οι πόντοι που χάνει κάθε παίκτης που “πέφτει” στην συγκεκριμένη παγίδα

int randtype: η μεταβλητή που ελέγχει ποιο είδος θα δοθεί στην κάθε παγίδα

Η λειτουργία της με λίγα λόγια είναι η εξής:

Αρχικά, δίνει τυχαίες τιμές από -10 έως -1 στην μεταβλητή `point`, αρχικοποιεί όλα τα στοιχεία του πίνακα `oldxy` με την τιμή 0 και ξεκινάει να δίνει τυχαίες τιμές στις συντεταγμένες της πρώτης παγίδας που δημιουργείται, με τις τιμές αυτές να ανήκουν ,φυσικά, στα επιτρεπόμενα όρια. Στην συνέχεια, ελέγχει αν οι συντεταγμένες αυτές έχουν δοθεί σε άλλη παγίδα και εφόσον αυτό δεν συμβαίνει, τις αποθηκεύουν στον πίνακα `oldxy`. Η διαδικασία αυτή επαναλαμβάνεται για κάθε παραγόμενη παγίδα. Τέλος, μεταβιβάζει εναλλάξ καθένα από τα δυο διαθέσιμα είδη στις παγίδες(με τυχαία επιλογή του πιο είδος θα δοθεί πρώτο, αλλά εξασφαλισμένη την ισομερή παρουσία αυτών) και τοποθετεί τις παγίδες στον πίνακα `Trap[] traps`.

Η συνάρτηση **void() createRandomFood** :

Περιέχει τις μεταβλητές :

String type1: το είδος του εφοδίου που δημιουργείται

int x,y: οι τυχαίες συντεταγμένες του εφοδίου μέσα στα όρια τοποθέτησης εφοδίων

int[][] oldxy: ένας πίνακας με τις συντεταγμένες των προηγούμενων εφοδίων που έχουν δημιουργηθεί, έτσι ώστε να μην υπάρχουν περισσότερα από ένα εφόδια στην ίδια

θέση

boolean ok: ελέγχει την do-while εντολή που χρησιμοποιείται για την αποφυγή ύπαρξης πολλών εφοδίων στην ίδια θέση

int points: οι πόντοι που κερδίζει κάθε παίκτης που “πέφτει” στο συγκεκριμένο εφόδιο

Η λειτουργία της με λίγα λόγια είναι η εξής:

Αρχικά, δίνει τυχαίες τιμές από 1 έως 10 στην μεταβλητή points ,αρχικοποιεί όλα τα στοιχεία του πίνακα oldxy με την τιμή 0 και ξεκινάει να δίνει τυχαίες τιμές στις συντεταγμένες του πρώτου εφοδίου που δημιουργείται , με τις τιμές αυτές να ανήκουν ,φυσικά, στα επιτρεπόμενα όρια. Στην συνέχεια, ελέγχει αν οι συντεταγμένες αυτές έχουν δοθεί σε άλλο εφόδιο και εφόσον αυτό δεν συμβαίνει, τις αποθηκεύουν στον πίνακα oldxy. Η διαδικασία αυτή επαναλαμβάνεται για κάθε παραγόμενο εφόδιο. Τέλος , τοποθετεί τα εφόδια στον πίνακα Food[] foods.

Η συνάρτηση **void() createBoard** :

Η συνάρτηση αυτή καλεί τις συναρτήσεις createRandomWeapon, createRandomTrap και createRandomFood.

Η συνάρτηση **void() resizeBoard(Player p1,Player p2):**

Η συνάρτηση αυτή δέχεται ως ορίσματα δυο στιγμιότυπα παικτών (p1 , p2) και αλλάζει το μέγεθος του Board αφαιρώντας το εξωτερικό του τετράγωνο , με την προϋπόθεση ότι κανένας από τους δυο παίκτες δεν βρίσκεται σε αυτό.

Η συνάρτηση **String[][] getStringReprasantation():**

Περιέχει τις μεταβλητές :

String[][] board: το String που επιστρέφεται και περιέχει τον πίνακα

boolean ok: μεταβλητή που ελέγχει τον βρόγχο της for και δεν εμφανίζει “___” όταν σε εκείνο το σημείο υπάρχει ένα αντικείμενο ένας χαρακτήρας

int x,y: μετατρέπουν τις καρτεσιανές συντεταγμένες σε αυτές του Board

Η λειτουργία της με λίγα λόγια είναι η εξής:

Η συνάρτηση αυτή, δημιουργεί το String που παρουσιάζει το board του παιχνιδιού σε κάθε γύρο, μετατρέπει τις καρτεσιανές συντεταγμένες σε συντεταγμένες του Board(προσθέτοντας στα x και y τα M/2 και N/2 λόγω της “στροφής 90 μοιρών” που ουσιαστικά παρατηρείται και αφαιρώντας 1 για τα θετικά x και y , επειδή έχουμε αγνοήσει τις ευθείες x=0 και y=0). Επιπλέον , στα σημεία του Board που υπάρχουν τα όπλα εμφανίζει “WAB”,όπου τα A και B είναι το playerId του ιδιοκτήτη του συγκεκριμένου όπλου και το id

του όπλου αντίστοιχα. Παράλληλα , για τα εφόδια και τις παγίδες εμφανίζει τα γράμματα “F” και “T” αντίστοιχα, συνοδευόμενα από το id του κάθε στοιχείου. Τέλος , ελέγχει σε ποια σημεία του Board δεν έχει τοποθετηθεί κανένα item ή κανέναν παίκτη και εμφανίζει σε αυτά “___”.

Κλάση Player

Η κλάση Player θα αντιπροσωπεύει τον παίκτη και θα έχει τις εξής μεταβλητές:

int id: ο κωδικός του παίκτη.

String name: το όνομα του παίκτη.

Board board: το ταμπλό του παιχνιδιού.

int score: το σκορ του παίκτη, το οποίο καθορίζεται από τους πόντους που του δίνουν τα εφόδια και του αφαιρούν οι παγίδες σε περίπτωση που δεν έχει το κατάλληλο όπλο για να τις αντιμετωπίσει.

int x: η συντεταγμένη x του πλακιδίου όπου βρίσκεται ο παίκτης

int y: η συντεταγμένη y του πλακιδίου όπου βρίσκεται ο παίκτης

Weapon bow: το όπλο τύπου bow του παίκτη. Σε περίπτωση που ο παίκτης δεν διαθέτει το συγκεκριμένο όπλο, η τιμή του είναι null.

Weapon pistol: το όπλο τύπου pistol του παίκτη. Σε περίπτωση που ο παίκτης δεν διαθέτει το συγκεκριμένο όπλο, η τιμή του είναι null.

Weapon sword: το όπλο τύπου sword του παίκτη. Σε περίπτωση που ο παίκτης δεν διαθέτει το συγκεκριμένο όπλο, η τιμή του είναι null.

Οι μέθοδοι που υλοποιούνται σε αυτήν την κλάση είναι :

Δυο **constructor** που αρχικοποιούν τις μεταβλητές της κλάσης . Ο πρώτος δεν δέχεται ορίσματα και αρχικοποιεί όλες τις μεταβλητές με την τιμή 0, ενώ ο δεύτερος δέχεται ως όρισμα όλες τις μεταβλητές της κλάσης.

Οι **getters** και οι **setters** όλων των μεταβλητών , έτσι ώστε να μπορεί ο χρήστης να τις προσπελάσει και εκτός της κλάσης τους.

Η συνάρτηση **int[2] getRandomMove()**:

Περιέχει τις μεταβλητές :

int[2] move: ο πίνακας δυο ακεραίων που επιστρέφεται από αυτήν την συνάρτηση και περιέχει τις συντεταγμένες της νέας θέσης του παίκτη μετά την μετακίνησή του

int a,b: το βήμα του παίκτη στους άξονες x και y αντίστοιχα

int c: παίζει τον ρόλο του “ζαριού” και παίρνει τυχαία τιμή από 1 έως 8

Η λειτουργία της με λίγα λόγια είναι η εξής:

Αρχικά, η μεταβλητή c παίρνει μια τυχαία τιμή και ανάλογα με αυτήν οι a και b παίρνουν έναν συνδυασμό τιμών που ανήκουν στο $\{-1,0,1\}$, χωρίς όμως να παίρνουν ταυτόχρονα την τιμή 0. Ωστόσο, δίνεται ιδιαίτερη προσοχή και στην περίπτωση που τα x,y του παίκτη ισούνται με -1 ή 1 και πρέπει να αποφύγει τις ευθείες $x=0$ και $y=0$, όποτε και προστίθεται ή αφαιρείται 2 αντί για 1 στην αντίστοιχη μεταβλητή που κινδυνεύει να μηδενιστεί. Η διαδικασία αυτή επαναλαμβάνεται, έως ότου η νέα θέση του παίκτη $(x+a,y+b)$ να ανήκει στο Board του παιχνιδιού. Τότε, οι συνιστώσες της θέσης αυτής αποθηκεύονται στα δυο “κελιά” του πίνακα `move` και επιστρέφονται.

Η συνάρτηση **int[] move()**:

Περιέχει τις μεταβλητές :

int[5] numbers: ο πίνακας που επιστρέφεται από την συνάρτηση και περιέχει τις νέες συντεταγμένες του παίκτη, τα όπλα και τα εφόδια που μάζεψε, καθώς και τις παγίδες στις οποίες “έπεσε”

int[2] newMove: οι τιμές της νέας θέσης του παίκτη, όπως επιστρέφονται από την καλούμενη συνάρτηση `getRandomMove()`

int wc,fc,tc: μετρητές όπλων, εφοδίων και παγίδων

Η λειτουργία της με λίγα λόγια είναι η εξής:

Αρχικά, καλείται η συνάρτηση `getRandomMove()` και η τιμή της αποθηκεύεται στον πίνακα `newMove`. Εν συνεχεία, ελέγχεται για κάθε όπλο του ταμπλό εάν η θέση του συμπίπτει με την νέα θέση του παίκτη και το `id` του ιδιοκτήτη του είναι ίσο με αυτό του παίκτη. Σε αυτήν την περίπτωση αυξάνεται κατά 1 ο μετρητής `wc`, εκτυπώνεται το είδος του όπλου και διαγράφεται από το Board. Μια παρόμοια διαδικασία πραγματοποιείται και για τα εφόδια, με τις διαφορές ότι στο `score` του παίκτη χρειάζεται να προστεθούν οι πόντοι του κάθε εφοδίου, ο μετρητής που αυξάνεται είναι ο `fc` και δεν απαιτείται έλεγχος `id` του παίκτη. Συμπληρωματικά, λαμβάνει χώρα και έλεγχος των παγίδων στις οποίες έχει “πέσει” ο παίκτης, ο `tc` αυξάνεται κατά 1 και όταν ο παίκτης δεν έχει στην κατοχή του σπαθί ή τόξο για να αντιμετωπίσει τα σχοινιά ή τα ζώα αντίστοιχα, αφαιρούνται από το `score` του οι πόντοι ποινής της κάθε παγίδας. Τέλος, οι `wc`, `fc`, `tc` και οι νέα θέση του παίκτη: `newMove[0]`, `newMove[1]` αποθηκεύονται στον `numbers` και επιστρέφονται.

Κλάση Game

Η κλάση `Game` αντιπροσωπεύει το παιχνίδι και έχει την εξής μεταβλητή:

int round: ο τρέχον γύρος του παιχνιδιού.

Οι μέθοδοι που υλοποιούνται σε αυτήν την κλάση είναι :

Δυο **constructor** που αρχικοποιούν την μεταβλητή της κλάσης . Ο πρώτος δεν δέχεται ορίσματα και αρχικοποιεί την round με την τιμή 0, ενώ ο δεύτερος δέχεται την μεταβλητή αυτή ως όρισμα.

Οι **getters** και οι **setters** όλων της μεταβλητής round , έτσι ώστε να μπορεί ο χρήστης να την προσπελάσει και εκτός της κλάσης της.

Η συνάρτηση **public static void main()**:

Σε αυτήν την συνάρτηση εκτελείται η προσομοίωση του παιχνιδιού που έχουμε δημιουργήσει σε ολόκληρο το παραδοτέο . Αρχικά , δημιουργούνται στιγμιότυπα των κλάσεων Game (με αρχική τιμή round = 1) και Board (με μέγεθος 20x20). Έπειτα , περιορίζουμε την περιοχή τοποθέτησης όπλων στον 4x4 υποπίνακα του εσωτερικού του Board με τιμές συντεταγμένων : $\{-2 \leq x \leq 2, -2 \leq y \leq 2, x, y \neq 0\}$. Αντίστοιχα , τα εφόδια τοποθετούνται στο περίγραμμα του προηγούμενου πίνακα που φράσσεται από τα σημεία : (-3,3) , (-3,-3) , (3,3) , (3,-3) και οι παγίδες στο περίγραμμα των εφοδίων που φράσσεται από τα σημεία : (-4,4) , (-4,-4) , (4,4) , (4,-4).

Παράλληλα , δημιουργούνται και αρχικοποιούνται δυο στιγμιότυπα παικτών , που αγωνίζονται έως ότου το ταμπλό να γίνει 4x4. Κάθε γύρος αποτελείται από μια κίνηση του κάθε παίκτη μέσω της συνάρτησης move() (της κλάσης Player) και συνοδεύεται από την εκτύπωση του Board και του score του κάθε παίκτη. Κάθε 3 γύρους , όμως , καλείται η συνάρτηση resizeBoard() για σμίκρυνση του ταμπλό. Τελικός νικητής ανακηρύσσεται εκείνος με το μεγαλύτερο τελικό σκορ!

ΤΕΛΟΣ ΕΡΓΑΣΙΑΣ!

