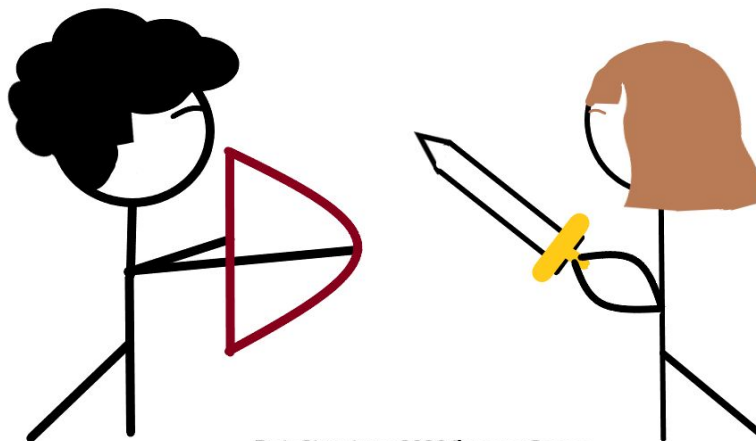


HUNGER GAMES

ΜΑΣΤΟΡΑΣ ΝΕΚΤΑΡΙΟΣ(9808) - ΜΙΧΑΛΤΣΗ ΕΙΡΗΝΗ(9736)

Παραδοτέο 1ο



DataStructures2020/hungerGames

Δομές Δεδομένων 017

3ο εξάμηνο 2019-2020

Νεκτάριος Μάστορας (9808) - Ειρήνη Μιχάλτση (9736)

Τμήμα Ηλεκτρολόγων Μηχανικών/ Μηχανικών Υπολογιστών

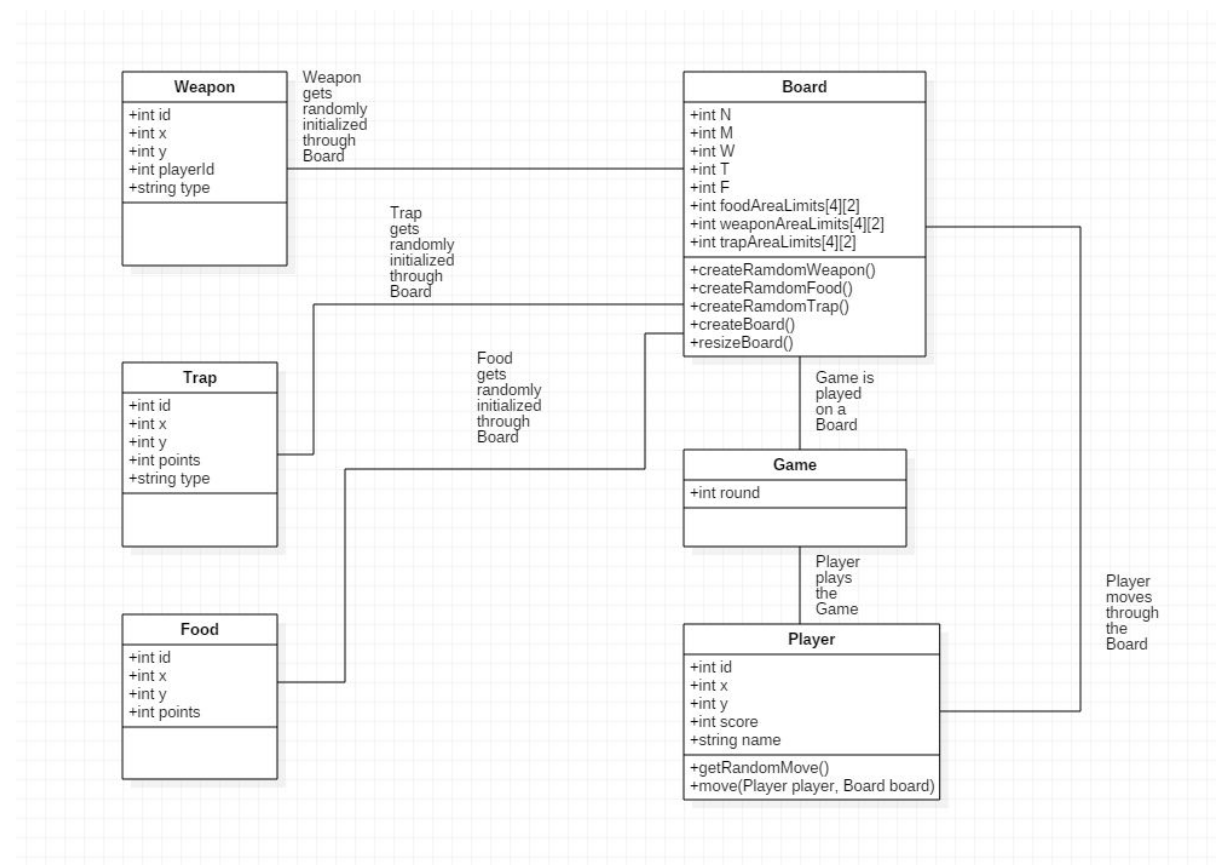
Παραδοτέο 1ο

Ημερομηνία λήξης προθεσμίας 08/11/2019

Πρόβλημα

Κληθήκαμε να δημιουργήσουμε μια απλουστευμένη αναπαράσταση του παιχνιδιού Hunger Games. Το παιχνίδι αποτελείται από ένα ζεύγος παικτών οι οποίοι προσπαθούν να επιβιώσουν ο ένας έναντι του άλλου. Συγκεντρώνουν εφόδια και όπλα και πέφτουν σε παγίδες που βρίσκονται σε συγκεκριμένες περιοχές του ταμπλό, στο οποίο κινούνται (τυχαία).

Σκοπός μας είναι να δημιουργήσουμε την αλληλεπίδραση των παικτών, του ταμπλό και των αντικειμένων τα οποία εμπλέκονται ώστε να προσομοιωθεί το παιχνίδι μέσω ενός προγράμματος σε Java. Το συγκεκριμένο πρόγραμμα περιέχει τις κλάσεις: Weapon, Food, Trap, Board, Player και Game όπως φαίνονται και στο ακόλουθο UML διάγραμμα.



Κάθε μια από αυτές αποτελείται από μεταβλητές και από μεθόδους, οι οποίες της επιτρέπουν να αλληλεπιδρά με τα άλλα αντικείμενα.

Κλάσεις

Weapon

Η κλάση Weapon περιέχει τις μεταβλητές όπως δίνονται στις οδηγίες της εργασίας.

Η κλάση Weapon περιέχει επίσης τις μεθόδους, που εξυπηρετούν αντίστοιχα:

οι constructors της κλάσης αυτής που μας επιτρέπουν να αρχικοποιούμε αντικείμενα,

οι getters της κλάσης που μας επιτρέπουν να έχουμε πρόσβαση στις μεταβλητές της,

οι setters της κλάσης που μας επιτρέπουν να επεξεργαζόμαστε τις μεταβλητές της

Food

Η κλάση Food περιέχει τις μεταβλητές όπως δίνονται στις οδηγίες της εργασίας.

Η κλάση Food περιέχει επίσης τις μεθόδους, που εξυπηρετούν αντίστοιχα:

οι constructors της κλάσης αυτής που μας επιτρέπουν να αρχικοποιούμε αντικείμενα,

οι getters της κλάσης που μας επιτρέπουν να έχουμε πρόσβαση στις μεταβλητές της,

οι setters της κλάσης που μας επιτρέπουν να επεξεργαζόμαστε τις μεταβλητές της

Trap

Η κλάση Trap περιέχει τις μεταβλητές όπως δίνονται στις οδηγίες της εργασίας.

Η κλάση Trap περιέχει επίσης τις μεθόδους, που εξυπηρετούν αντίστοιχα:

οι constructors της κλάσης αυτής που μας επιτρέπουν να αρχικοποιούμε αντικείμενα,

οι getters της κλάσης που μας επιτρέπουν να έχουμε πρόσβαση στις μεταβλητές της,

οι setters της κλάσης που μας επιτρέπουν να επεξεργαζόμαστε τις μεταβλητές της

Player

Η κλάση Player περιέχει τις μεταβλητές όπως δίνονται στις οδηγίες της εργασίας.

Η κλάση Player περιέχει επίσης τις μεθόδους, που εξυπηρετούν αντίστοιχα:

οι constructors της κλάσης αυτής που μας επιτρέπουν να αρχικοποιούμε αντικείμενα,

οι getters της κλάσης που μας επιτρέπουν να έχουμε πρόσβαση στις μεταβλητές της,

οι setters της κλάσης που μας επιτρέπουν να επεξεργαζόμαστε τις μεταβλητές της,

`int[] getRandomMove()`, συνάρτηση η οποία επιλέγει τυχαία μια από τις 8 διαθέσιμες κινήσεις, ελέγχει αν ο παίκτης εκτελώντας την βρίσκεται εντός του ταμπλό και επιστρέφει τη μετατόπιση σε κάθε άξονα (a κατά x, b κατά y) και τον “κωδικό” της κίνησης (c, 1..8) σε ένα πίνακα `arr`.

`int[] move(Player player, Board board)`, συνάρτηση η οποία μεταβάλλει τη θέση του παίκτη καλώντας την `getRandomMove()`, ώστε να λάβει τις αντίστοιχες μετατοπίσεις. Έπειτα εκτελεί τους απαραίτητους ελέγχους μέσα από διαδοχικές `if` ώστε να εξασφαλιστεί ότι θα επιστραφεί η σωστή ως προς τους δοθέντες άξονες θέση (επειδή τα σημεία (0, y) και (x, 0) δεν αναπαριστώνται). Χρησιμοποιώντας το τρίτο στοιχείο του πίνακα (c) αποκωδικοποιεί την κατεύθυνση κίνησης και εκτυπώνει το αντίστοιχο μήνυμα. Ακολουθώς, ελέγχει αν η θέση μετακίνησης του παίκτη εμπεριέχει κάποιο αντικείμενο ενδιαφέροντος όπως όπλα, εφόδια ή παγίδες (συγκρίνοντας τις θέσεις τους και αν το όπλο προορίζεται για το συγκεκριμένο παίκτη). Πόντοι προστίθενται αν πρόκειται για εφόδια ή αφαιρούνται αν πρόκειται για παγίδες και ο παίκτης δεν έχει το κατάλληλο όπλο να τις

αντιμετωπίσει. Αν κάποιος παίκτης αλληλοεπιδράσει με κάποιο από τα αντικείμενα, αυτό μετακινείται αυτόματα στην αρχή των αξόνων, δηλαδή εξαφανίζεται από το ταμπλό. Αριθμούνται με 3 μετρητές (`weaponCounter`, `foodCounter`, `trapCounter`) τα 3 αντικείμενα τα οποία θα μπορούσε να συναντήσει ο παίκτης και εκτυπώνεται το αντίστοιχο μήνυμα. Οι νέες συντεταγμένες της θέσης του παίκτη και οι 3 μετρητές επιστρέφονται μέσω ενός πίνακα `arr`.

Board

Η κλάση `Board` περιέχει τις μεταβλητές όπως δίνονται στις οδηγίες της εργασίας.

Η κλάση `Board` περιέχει επίσης τις μεθόδους, που εξυπηρετούν αντίστοιχα:

οι `constructors` της κλάσης αυτής που μας επιτρέπουν να αρχικοποιούμε αντικείμενα,

οι `getters` της κλάσης που μας επιτρέπουν να έχουμε πρόσβαση στις μεταβλητές της,

οι `setters` της κλάσης που μας επιτρέπουν να επεξεργαζόμαστε τις μεταβλητές της,

`createRandomWeapon()`, συνάρτηση η οποία αρχικοποιεί τυχαία τα αντικείμενα του πίνακα των `weapons` εντός των ορίων τους. Μέσω της μεταβλητής `check` ελέγχεται αν η θέση του όπλου έχει ήδη καταληφθεί από κάποιο άλλο και επαναλαμβάνεται η διαδικασία μέχρι κάθε όπλο να βρίσκεται μόνο του σε κάποια θέση. Με βάση τη θέση του μετρητή στον πίνακα καθορίζεται ο παίκτης στον οποίο ανήκει και το είδος του. Δηλαδή, με τη γραμμή: `this.weapons[i].setplayerId(i % 2)` εξασφαλίζεται ότι θα δημιουργούνται εναλλάξ όπλα για κάθε παίκτη (αφού υπάρχουν μόνο 2 παίκτες) και με την εντολή:

`switch (i % 3)` εξασφαλίζεται ότι δημιουργούνται 3 όπλα κάθε τύπου. Τα όπλα όπως θα εμφανίζονται στο ταμπλό (`W + player_id+weapon_id`) θα δημιουργηθούν με τη σειρά:

`W00`, `W11`, `W02`, `W10`, `W01`, `W12`. Έχουμε θέσει τα `id` των αντικειμένων τύπου `Weapon` να αντιστοιχούν σε: 0 - Pistol, 1 - Sword, 2 - Bow.

`createRandomFood()`, συνάρτηση η οποία με ανάλογο τρόπο αρχικοποιεί τον πίνακα της τροφής, χρησιμοποιώντας μια αντίστοιχη μεταβλητή `check`. Ελέγχεται δηλαδή αν τα αντικείμενα τύπου `Food` είναι μέσα στην περιοχή τους και ότι οι συντεταγμένες που δημιουργήθηκαν τυχαία δεν ανήκουν σε άλλο αντικείμενο. Τυχαία επίσης επιλέγονται και οι θετικοί πόντοι κάθε αντικειμένου `[1, 10]`.

`createRandomTrap()`, συνάρτηση που αντίστοιχα αρχικοποιεί τον πίνακα αντικειμένων των παγίδων, χρησιμοποιώντας την ανάλογη μεταβλητή `check` για να ελεγχθεί αν το κάθε αντικείμενο τύπου `Trap` έχει συντεταγμένες εντός ορίων της περιοχής που έχει οριστεί και αν δεν ανήκουν οι συντεταγμένες του σε άλλο αντικείμενο. Επιλέγονται τυχαία οι πόντοι ποινής της παγίδας `[-10, -1]` καθώς και ο τύπος της (υπάρχουν 2 τύποι γι' αυτό και χρησιμοποιείται η εντολή `type = rand.nextInt(2)`)

`createBoard()`, συνάρτηση η οποία καλεί τις 3 προηγούμενες και δημιουργεί ένα ταμπλό αρχικοποιώντας τους 3 πίνακες τυχαία

`resizeBoard(Player p1, Player p2)`, συνάρτηση η οποία χρησιμοποιώντας τις συντεταγμένες των δύο παικτών ελέγχει αν δε βρίσκεται κανείς τους στα όρια του ταμπλό και αν ο έλεγχος αυτός είναι θετικός και οι δύο παίκτες είναι σε θέση που δε θα διαγραφεί με συρρίκνωση του ταμπλό τότε μικραίνει το ταμπλό κατά ένα πόντο περιμετρικά, δηλαδή το `N` και το `M` μειώνονται κατά 2. Επίσης ελέγχει όταν οι διαστάσεις του ταμπλό φτάσουν να είναι

8x8 ή 6x6 αν υπάρχουν παγίδες ή εφόδια στην περίμετρο του ταμπλό και τα μετακινεί στο (0, 0) αφού με τη συρρίκνωση του ταμπλό οι συντεταγμένες τους θα είναι εκτός ορίων.

`getStringRepresentation()`, συνάρτηση η οποία επιστρέφει ένα δισδιάστατο πίνακα που αναπαριστά το ταμπλό με τις θέσεις των αντικειμένων πάνω σε αυτό. Κάθε τεταρτημόριο του ταμπλό ορίζεται ανεξάρτητα στον πίνακα με ένα δικό του βρόχο `for` λόγω της ιδιαιτερότητας του συστήματος συντεταγμένων που δίνεται στις οδηγίες. Ελέγχεται αν στη συγκεκριμένη θέση βρίσκεται κάποιο αντικείμενο και αν βρίσκεται εμφανίζεται στην αντίστοιχη θέση. Λόγω της ιδιόρρυθμης διάταξης του ταμπλό για κάθε τεταρτημόριο είναι διαφορετικός ο έλεγχος και ο τρόπος αναπαράστασης του.

Game

Η κλάση `Game` περιέχει μόνο τη μεταβλητή `round` η οποία αναπαριστά το γύρο στον οποίο βρίσκεται το παιχνίδι.

Περιέχει τις μεθόδους, που εξυπηρετούν αντίστοιχα:

οι `constructors` της κλάσης αυτής που μας επιτρέπουν να αρχικοποιούμε αντικείμενα,

ο `getter` της κλάσης που μας επιτρέπει να έχουμε πρόσβαση στη μεταβλητή της,

ο `setter` της κλάσης που μας επιτρέπει να επεξεργαζόμαστε τη μεταβλητή της,

`main(String[] args)`, η κύρια συνάρτηση του προγράμματος. Το παιχνίδι ξεκινά δημιουργώντας ένα αντικείμενο τύπου `Game` και θέτοντας το γύρο `round = 1`. Δημιουργείται ένα ταμπλό καλώντας την αντίστοιχη συνάρτηση (`Board board = new Board(20, 20, 6, 10, 8)`). Ως ορίσματα του `constructor` θέτονται τα δοθέντα από την εκφώνηση $M \times N = 20 \times 20$, 6 όπλα, 10 τροφές και 8 παγίδες. Έπειτα δημιουργούνται οι 2 παίκτες του παιχνιδιού οι οποίοι αρχικά βρίσκονται στα άκρα της διαγωνίου του ταμπλό (`Player1: (-10, -10)`, `Player2: (10,10)`). Το παιχνίδι ξεκινά με κάποια εναρκτήρια μηνύματα. Όσο το ταμπλό είναι μεγαλύτερο του 4x4 το παιχνίδι εξελίσσεται. Εμφανίζεται αρχικά ο γύρος στον οποίο βρίσκεται το παιχνίδι, οι παίκτες κινούνται κι έπειτα ελέγχεται αν χρειάζεται να συρρικνωθεί το ταμπλό (κάθε 3 γύρους δηλαδή). Ακολούθως εκτυπώνεται το ταμπλό μέσω της συνάρτησης `board.getStringRepresentation()`. Ο γύρος του παιχνιδιού αυξάνεται ενώ εμφανίζονται μηνύματα που πληροφορούν για τους πόντους κάθε παίκτη. Μόλις το ταμπλό αποκτήσει τις διαστάσεις 4x4 το παιχνίδι τερματίζεται εμφανίζοντας το αντίστοιχο μήνυμα. Η έκβαση του παιχνιδιού ανακοινώνεται μέσω ενός μηνύματος που πληροφορεί για το νικητή ή την ισοπαλία των 2 παικτών, βάσει των πόντων τους, όπου και τερματίζεται το πρόγραμμα.