

Δομές δεδομένων

Εργασία Β



Σιμίτσης Βασίλης ΑΕΜ: 8458 Email: [vsimitsis@gmail.com](mailto:vsimitsis@gmail.com)

Κωστινούδης Ευάγγελος ΑΕΜ: 8708 Email: ekostinou@auth.gr

Contents

[1 Περιγραφή προβλήματος 2](#_Toc25944161)

[2 Κλάση HeuristicPlayer 2](#_Toc25944162)

[2.1 Μεταβλητές 2](#_Toc25944163)

[2.2 Μέθοδοι 2](#_Toc25944164)

[3 Κλάση Game 3](#_Toc25944165)

[3.1 Μεταβλητές 3](#_Toc25944166)

[3.2 Μέθοδοι 3](#_Toc25944167)

# Περιγραφή προβλήματος

Στην εργασία αυτή ζητήθηκε η υλοποίηση σε java μίας παραλλαγής τους παιχνιδιού hunger games. Υπάρχουν δύο παίκτες που μπορούν να κινηθούν μία θέση σε κάθε γύρο. Οι παίκτες μπορούν να μαζέψουν όπλα που υπάρχουν στο ταμπλό και εφόδιο. Ακόμα, υπάρχουν παγίδες που μπορούν να πέσουν οι παίκτες. Το ταμπλό μικραίνει μέχρις να γίνει 4x4 όπου τελειώνει το παιχνίδι. Ακόμα, ένας παίκτης μπορεί να σκοτώσει τον αντίπαλο του όπου και πάλι το παιχνίδι τελειώνει.

# Κλάση HeuristicPlayer

Η κλάση HeuristicPlayer αντιπροσωπεύει ένα παίκτη που παίζει με στρατηγική. Κληρονομεί την κλάση Player.

## Μεταβλητές

* private ArrayList<Integer[]> path: Πληροφορίες για την κάθε κίνηση του παίκτη. Σε κάθε γύρο αποθηκεύονται οι τιμές του ζαριού, των πόντων που κερδίζει-χάνει ο παίκτης αν πήρε πιστόλι, τόξο ή σπαθί, αν μάζεψε εφόδιο και αν έπεσε σε παγίδα ζώων ή σκοινιών.
* private static int r: Η απόσταση που μπορεί να δει κάθε παίκτης.

Για κάθε μεταβλητή υπάρχουν setter και getter μέθοδοι.

## Μέθοδοι

**Setters**:

* void setR(int r)
* void setPath(ArrayList<Integer[]> path)

**Getters**:

* int getR()
* ArrayList<Integer[]> getPath()

**Constructors**:

* HeuristicPlayer(): Κενός constructor.
* HeuristicPlayer(ArrayList<Integer[]> path): Constructor με μοναδικό όρισμα το path.
* HeuristicPlayer(int id, String name, Board board, int score, int x, int y): Constructor με ορίσματα όλες τις τιμές της κλάσης Player εκτός των όπλων.
* HeuristicPlayer(int id, String name, Board board, int score, int x, int y, ArrayList<Integer[]> path): Constructor με ορίσματα όλες τις τιμές της κλάσης Player εκτός των όπλων και του αρχικού path.
* HeuristicPlayer (int id, String name, Board board, int score, int x, int y, Weapon bow, Weapon pistol, Weapon sword): Constructor με ορίσματα όλες τις μεταβλητές της κλάσης Player.
* HeuristicPlayer(int id, String name, Board board, int score, int x, int y, Weapon bow, Weapon pistol, Weapon sword, ArrayList<Integer[]> path): Constructor με ορίσματα όλες τις μεταβλητές της κλάσης Player και του αρχικού path.
* HeuristicPlayer(Player player): Constructor που δέχεται ένα αντικείμενο Player ως είσοδο και αντιγράφει όλες τις τιμές του στο νέο.
* HeuristicPlayer(Player player, ArrayList<Integer[]> path): Constructor που δέχεται αντικείμενο Player και το αρχικό path.
* HeuristicPlayer(HeuristicPlayer heuristicPlayer): Constructor που δέχεται ως όρισμα αντικείμενο HeuristicPlayer και το αντιγράφει.

**Μέθοδοι**:

* float playersDistance(Player p): Επιστρέφει την ευκλείδεια απόσταση του παίκτη από τον αντίπαλο p, αν ο αντίπαλος είναι στο οπτικό πεδίο, αλλιώς επιστρέφει -1.
* double evaluate(int dice, Player p): Αξιολογεί την κίνηση του παίκτη για ζάρι dice. Συγκεριμένα, αν ο παίκτης έχει πιστόλι και ο αντίπαλος είναι στο οπτικό πεδίο κερδίζει (100 / απόσταση από αντίπαλο) πόντους. Αν μαζέψει πιστόλι 10 πόντους, 3 πόντους για σπαθί ή τόξο και τους πόντους που κερδίζει-χάνει αν μαζέψει ή πέσει σε παγίδα.
* static boolean kill(Player player1, Player player2, float d): Αν η απόσταση των παικτών είναι μικρότερη του d επιστρέφει true.
* int[] move(Player p): Αξιολογεί και επιλέγει την καλύτερη διαθέσιμη κίνηση βάση της evaluate (αν υπάρχουν παραπάνω από μία καλύτερη επιλέγει μία από αυτές με τυχαίο τρόπο). Ακόμα, μετακινεί τον παίκτη στη νέα θέση προσθέτει τη κίνηση του στη μεταβλητή path. Τέλος, επιστρέφει τη νέα θέση του παίκτη.
* void statistics(): Τυπώνει τη τελευταία κίνηση του παίκτη. Συγκεκριμένα, τυπώνει το ζάρι που επέλεξε ο παίκτης, ακόμα αν μάζεψε κάποιο όπλο ή κάποιο εφόδιο ή αν έπεσε σε κάποια παγίδα τυπώνει κατάλληλο μήνυμα.

# Κλάση Game

Η κλάση Game αντιπροσωπεύει το παιχνίδι.

## Μεταβλητές

* int round: Ο γύρος του παιχνιδιού.

Για κάθε μεταβλητή υπάρχουν setter και getter μέθοδοι.

## Μέθοδοι

**Setters**:

* void setRound(int round)

**Getters**:

* int getRound()

**Constructor**:

* Game(): Θέτει το γύρο τη τιμή 0.

**Main**:

Δημιουργεί το παιχνίδι, το ταμπλό και τους δύο παίκτες (αντικείμενα Game, Board, Player, HeuristicPlayer αντίστοιχα). Οι αρχικές θέσεις των παικτών είναι η πάνω αριστερά και κάτω δεξιά γωνία του ταμπλό. Ακόμα το αρχικό σκορ τους είναι 0. Ορίζει την απόσταση που βλέπει ο παίκτης σε 3. Θέτει τα όρια των περιοχών των όπλων, εφοδίων και παγίδων. Για κάθε γύρο, μετακινεί του παίκτες και εμφανίζει τον γύρο, το ταμπλό και τις κινήσεις των παικτών. Σταματάει όταν το ταμπλό έχει μέγεθος 4x4 ή όταν ένας παίκτης σκοτωθεί. Τέλος, εμφανίζει τους πόντους κάθε παίκτη και το αποτέλεσμα του παιχνιδιού.