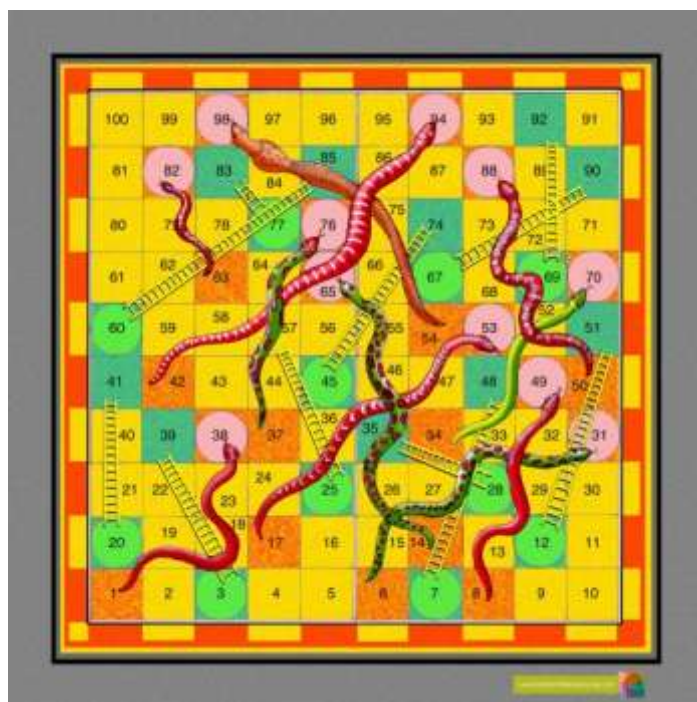


Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Δομές Δεδομένων

3ο Εξάμηνο Σπουδών

Εργασία Β – Heuristic Player



Ηλίας Κωφοκώτσος ΑΕΜ 9380

Σκαπέτης Χρήστος ΑΕΜ 9378

Χειμερινό εξάμηνο 2018 -2019

Εργασία B – Heuristic Player

Περιγραφή του Προβλήματος

Στο δεύτερο παραδοτέο του μαθήματος Δομές Δεδομένων, ζητήθηκε από τους φοιτητές να υλοποιήσουν δυο κλάσεις, οι οποίες θα ήταν επιπρόσθετες για την υλοποίηση που ζητήθηκε από τους φοιτητές εντός του πρώτου παραδοτέου του μαθήματος. Επιπλέον λοιπόν των κλάσεων *Apple*, *Ladder*, *Snake*, *Board* και *Game*, οι οποίες χρησιμεύουν για την υλοποίηση του παιχνιδιού στην βασική του μορφή, ζητήθηκε μια εξ ολοκλήρου νέα κλάση, η *HeuristicPlayer* και μια αλλαγή στην υλοποίηση της κλάσης *Game*.

Η κλάση *HeuristicPlayer* πρέπει να υλοποιεί έναν παίκτη ο οποίος θα παίζει με ένα επιπλέον επίπεδο ελέγχου στο παιχνίδι. Θα μπορεί δηλαδή να βλέπει μπροστά του στο ταμπλό του παιχνιδιού και να ζυγίζει τις διαφορετικές πιθανότητες που υπάρχουν, ανάλογα με το αποτέλεσμα του ζαριού. Στην συνέχεια, ο παίκτης αυτός θα επιλέγει την καλύτερη δυνατή κίνηση που του προσφέρεται με βάση μια διαδικασία αξιολόγησης, προκειμένου να είναι πάντα ο ίδιος ο νικητής του παιχνιδιού. Αυτή η υλοποίηση, σε προγραμματιστικό επίπεδο απαιτεί η κλάση αυτή να κληρονομεί την κλάση *Player* και να έχει τρεις τουλάχιστον συναρτήσεις, σύμφωνα με την εκφώνηση.

Η κλάση *Game* πρέπει να υλοποιεί μια παρτίδα του παιχνιδιού στην οποία θα παίζουν δυο μόνο παίκτες, ένας φυσιολογικός και ένας με αυξημένες δυνατότητες (αντικείμενα της κλάσης *Player* και *HeuristicPlayer* αντίστοιχα). Πρώτος πρέπει να παίζει αυτός ο οποίος φέρνει το μικρότερο ζάρι και στην συνέχεια να παίζουν οι υπόλοιποι παίκτες με αύξουσα σειρά ζαριού. Τέλος, απαιτείται να εκτυπώνονται ένας αριθμός από πληροφορίες του παιχνιδιού. Σε προγραμματιστικό επίπεδο, αυτό απαιτεί η κλάση αυτή να περιέχει τουλάχιστον δυο συναρτήσεις σύμφωνα με την εκφώνηση, εκ των οποίων η μια να είναι η *main* συνάρτηση του προγράμματος.

Περιγραφή του Αλγορίθμου και των Διαδικασιών

Για την υλοποίηση του δεύτερου παραδοτέου χρησιμοποιήθηκε ο κώδικας ο οποίος είχε αναπτυχθεί για το πρώτο παραδοτέο και όχι ο πρότυπος κώδικας που ανέβηκε στο *elearning*.

Εφόσον δεν έχουν μεταβληθεί τα περιεχόμενα των κλάσεων *Ladder*, *Snake*, *Apple*, *Board* και *Player* δεν θα γίνει ανάλυσή τους στην παρούσα αναφορά θεωρώντας ότι επαρκούν οι επεξηγήσεις που έγιναν κατά την παράδοση της πρώτης εργασίας καθώς και τα σχόλια στον πηγαίο κώδικα.

Στην παρούσα αναφορά θα περιγραφούν παρακάτω οι κλάσεις *HeuristicPlayer* και *Game* οι οποίες αποτελούσαν το ζητούμενο της δεύτερης άσκησης.

Εργασία B – Heuristic Player

Κλάση *HeuristicPlayer*

Η κλάση αυτή κληρονομεί όλα τα στοιχεία της κλάσης *Player* και περιλαμβάνει επιπλέον μια μεταβλητή, το *path*, τύπου *ArrayList*, και 3 συναρτήσεις πέραν από τους *constructors* της κλάσης. Αυτές είναι οι *evaluate*, *getNextMove* και *statistics*.

Συνάρτηση *evaluate*

Η συνάρτηση αυτή έχει επιστρεφόμενη τιμή τύπου *double* και λαμβάνει δυο ορίσματα τύπου *int*, το *currentPos* και το *dice*. Στην συνάρτηση αυτή δεδομένης της θέσης του παίκτη και του ζαριού που έφερε, δηλαδή του *currentPos* και του *dice* αντίστοιχα, υπολογίζονται η τελική θέση που θα φτάσει ο παίκτης καθώς και οι πόντοι που θα λάβει. Στην συνέχεια πραγματοποιείται μια αξιολόγηση της κίνησης αναφορικά με τα βήματα και τους πόντους που προσφέρει, σύμφωνα με μια συνάρτηση βάρους που αποφασίστηκε μετά από μια σειρά δοκιμών πως θα είναι η εξής:

$$f(steps, points) = 0,75 * steps + 0,25 * points$$

Τέλος επιστρέφεται αυτή η αξιολόγηση στην καλούσα συνάρτηση.

Συνάρτηση *getNextMove*

Η συνάρτηση αυτή έχει επιστρεφόμενη τιμή τύπου *int* και δέχεται ένα όρισμα τύπου *int*, το *currentPos*, το οποίο είναι η τρέχουσα θέση του παίκτη. Η συνάρτηση αυτή καλεί 6 φορές την συνάρτηση *evaluate*, μία για κάθε πιθανή τιμή του ζαριού, έπειτα βρίσκει την κίνηση με την καλύτερη αξιολόγηση και στην συνέχεια καλεί την συνάρτηση *move* της κλάσης *player* προκειμένου να πραγματοποιηθεί αυτή η κίνηση. Τέλος, προσθέτει στο *ArrayList*, τύπου *<int[]>*, *path* τα δεδομένα που αφορούν αυτή την κίνηση, δηλαδή το ζάρι που έριξε ο παίκτης, την αξιολόγηση την κίνησης, την τελική θέση που έφτασε ο παίκτης και τον αριθμό των φιδιών, σκαλών και μήλων που χρησιμοποίησε ο παίκτης κατά την κίνησή του.

Συνάρτηση *statistics*

Η συνάρτηση αυτή είναι τύπου *void* χωρίς ορίσματα. ο σκοπός της συνάρτησης αυτής είναι να εκτυπώνει όλες τις κινήσεις του *heuristicPlayer* κατά την διάρκεια του παιχνιδιού. Εμφανίζει δεδομένα όπως τον γύρο, το ζάρι, την αξιολόγηση της κίνησης και αν συνάντησε ο παίκτης κάποιο αντικείμενο στον δρόμο του. Τέλος, κάνει έναν τελικό απολογισμό του παιχνιδιού για τον παίκτη αυτόν.

Κλάση *Game*

Εργασία B – Heuristic Player

Η κλάση *game* αντιπροσωπεύει την παρτίδα του παιχνιδιού. Περιέχει μία μόνο μεταβλητή τύπου *int*, την *round*, η οποία αντιπροσωπεύει τον γύρο στον οποίο βρισκόμαστε, και δύο συναρτήσεις πέρα από τους *constructors* και τους *setters/getters*, την *setTurns*, και την *main*.

Συνάρτηση *setTurns*

Η συνάρτηση *setTurns* επιστρέφει ένα *Map* τύπου $\langle Integer, Integer \rangle$ και δέχεται ένα όρισμα *ArrayList* τύπου *Object* το *players*, που περιέχει όλους τους παίκτες που παίζουν το παιχνίδι. Ο σκοπός αυτής της συνάρτησης είναι να επιστρέψει έναν *map* ο οποίος θα περιέχει τους παίκτες ταξινομημένους με αύξουσα σειρά ανάλογα με το ζάρι που έριξαν ώστε να καθοριστεί η σειρά με την οποία θα παίζουν. Υπάρχει περιορισμός στον αριθμό των παικτών έτσι ώστε να είναι το πολύ έξι. Επιπλέον εάν δύο παίκτες έχουν ρίξει το ίδιο αρχικό ζάρι ο ένας από τους δύο ξαναρίχνει το ζάρι έως ότου να έχει διαφορετικό ζάρι από όλους τους υπόλοιπους παίκτες.

Συνάρτηση *main*

Η συνάρτηση αυτή είναι η *main* συνάρτηση και εντός αυτής υλοποιείται όλο το παιχνίδι. Σε αυτή δημιουργούνται τα αντικείμενα των παικτών, του ταμπλό και της παρτίδας. Καλείται η συνάρτηση *setTurns* για να εξακριβωθεί η σειρά που θα παίζουν οι παίκτες. Κατόπιν, εντός επαναληπτικής διαδικασίας παίζουν οι παίκτες με την προηγουμένως καθορισμένη σειρά μέχρις ότου ένας από αυτούς φτάσει στον τερματισμό. Τέλος, εκτυπώνεται κατάλληλο μήνυμα στην κονσόλα που περιέχει πληροφορίες για τις κινήσεις που πραγματοποίησε ο *HeuristicPlayer*, το τελικό σκορ των παικτών, όπως αυτό προκύπτει από την συνάρτηση βάρους που χρησιμοποιήθηκε και στην συνάρτηση *evaluate* της κλάσης *HeuristicPlayer*, και προκηρύσσεται με βάση αυτό το σκορ ο νικητής του παιχνιδιού.