

# Τεχνητή Νοημοσύνη

**Αναγνώστου Πάυλος 5440**

**Νικόλαος Ντερμάρης 5477**

## Άσκηση 1

Εισαγωγικά:

Συνολικά χρησιμοποιήσαμε 6 κλάσεις. Από αυτές οι βασικές είναι 4. Η Main, που βρίσκεται η μέθοδος main, η Utilities, που περιέχει τις μεθόδους που υλοποιούν UCS και A\* και άλλες βοηθητικές μεθόδους, η Cell που περιγράφει το κάθε κελί του λαβυρίνθου, και η Maze που διαχειρίζεται τον λαβύρινθο.

UCS:

Για την UCS χρειαστήκαμε μία βοηθητική κλάση την PathNode, που μοιάζει με την Cell, ωστόσο περιέχει επιπλέον πληροφορίες που χρειάζονται στην UCS μέθοδο, όπως το κόστος και το μονοπάτι από την αρχή έως τον εκάστοτε κόμβο. Η σύγκριση μεταξύ των PathNodes γίνεται με βάση το κόστος τους.

A\*:

Για την A\* χρησιμοποιήσαμε την κλάση AStarNode που είναι ανάλογη με τη PathNode η οποία χρησιμοποιεί αντί για 1 κόστος 2, το συνολικό εκτιμώμενο κόστος και το πραγματικό (έως τώρα).

Ευρετική Συνάρτηση:

Η ευρετική επινοήθηκε με βάση ορισμένους παράγοντες. Μια καλή ευρετική συνάρτηση δεν υπερεκτιμά ποτέ το κόστος δηλαδή  $\text{εκτιμώμενο κόστος} \leq \text{πραγματικό κόστος}$ . Οπότε η ευρετική υπολογίζει 3 διαφορετικές αποστάσεις. Την απόσταση αρχής με τέλους, που την ορίζουμε ως το ελάχιστο μεταξύ των διαφορών οριζόντιας και κάθετης απόστασης, της απόστασης αρχή - κατω αριστερή γωνία - πανω δεξιά γωνία - στόχος, και της απόστασης αρχή - πανω δεξιά γωνία - κατω αριστερή γωνία - στόχος. Επιλέγεται το κόστος που είναι μικρότερο από αυτές τις 3 διαδρομές. Η ιδέα αυτή προέκυψε από τα 2 διαφορετικά σενάρια που περιγράφουν κάθε λαβύρινθο και την επίλυσή του, είτε

κίνηση διαγώνια από την αρχή προς το τέλος (παρακάμπτοντας τα εμπόδια) ή χρήση του portal, δηλαδή πέρασμα από κάτω αριστερά πάνω δεξιά ή αντίστροφα. Με αυτό τον τρόπο μπορούμε να έχουμε μια αρκετά καλή ευρετική συνάρτηση.

Συμπεράσματα:

Το πρόγραμμα ξεκινά με τη δημιουργία του λαβυρίνθου με βάση τις υποδείξεις της εκφώνησης. Έπειτα τρέχει την ucs τυπώνει το μονοπάτι και το πλήθος των κόμβων που επισκέφθηκε. Το ίδιο γίνεται και στη συνέχεια με την A.. Μετά από πολλά πειράματα πάντα και οι 2 αλγόριθμοι έβρισκαν τη μικρότερη διαδρομή και είχαν το ίδιο κόστος. Ωστόσο, παρατηρούμε ότι πάντα σχεδόν η A\* επισκέπτεται πολύ λιγότερους κόμβους, λόγω της χρήσης μίας καλής ευρετικής συνάρτησης. Μάλιστα όσο αυξάνεται το μέγεθος του λαβυρίνθου τόσο αυξάνεται και το χάσμα μεταξύ των επαναλήψεων που εκτελεί η ucs με την A\*. Για μέγεθος = 20, η ucs έκανε περίπου 2,5 φορές περισσότερες επεκτάσεις κόμβων, ενώ για μέγεθος = 150, 10 φορές περισσότερες επεκτάσεις.