**1η Προαιρετική Εργασία στο Μάθημα της Τεχνητής Νοημοσύνης**

**Ονοματεπώνυμο: Βίκτωρ Κυρτσούδης**

**ΑΕΜ: 4143**

Αντί για Βόρεια-Νότια χρησιμοποιούνται οι κατευθύνσεις Πάνω-Κάτω και αντί για Ανατολικά-Δυτικά οι κατευθύνσεις Δεξιά-Αριστερά.

Επίσης το πλέγμα έχει την αρχή των αξόνων πάνω δεξιά αντί για κάτω αριστερά. (Οπότε η μετακίνηση προς τα πάνω γίνεται με μείωση του y κατά 1)

Για την επίλυση του προβλήματος δημιουργήθηκαν 4 αρχεία:

* Car.h
* Puzzle.h
* Puzzle.cpp
* main.cpp

Στο Car.h είναι η υλοποίηση της κλάσης Car που αναπαριστά ένα αυτοκίνητο του προβλήματος αποθηκεύοντας τις x και y συντεταγμένες του στο πλέγμα αλλά και τον προσανατολισμό του με μία bool μεταβλητή horizontal (με προεπιλεγμένη τιμή true). Η μέθοδος getKey() υπολογίζει και επιστρέφει την συμβολοσειρά που αντιπροσωπεύει το αμάξι την ορισμένη στιγμή. Επίσης υλοποιήθηκε η υπερφόρτωση του τελεστή «<» με βάση την getKey() και του τελεστή της ανάθεσης που αντιγράφει τα δεδομένα του αντικειμένου Car δεξιά του τελεστή στο αντικείμενο αριστερά.

Στο Puzzle.h είναι η δήλωση της κλάσης Puzzle που αναπαριστά μία συγκεκριμένη κατάσταση του προβλήματος και στο Puzzle.cpp η υλοποίηση των μεθόδων. Για την αναπαράσταση μίας κατάστασης χρησιμοποιούνται:

* vector<Car> cars (που αποθηκεύει όλα τα αμάξια που βρίσκονται ακόμα στο πλέγμα)
* string key (που χρησιμοποιείται για την αντιστοίχιση της κατάστασης σε ένα unordered\_map και για την σύγκριση καταστάσεων με τον τελεστή «==»)
* bool \*\*free (που αποθηκεύει τις ελεύθερες και τις κατειλημμένες θέσεις του πλέγματος)

Για το μονοπάτι από την ρίζα έχουμε ένα δείκτη στην προηγούμενη κατάσταση του προβλήματος (Puzzle \*previous) που χρησιμοποιείται στην μέθοδο getDepth().

Η επέκταση της κατάστασης γίνεται με την μέθοδο expand που για κάθε αμάξι της τρέχουσας κατάστασης ελέγχει αν είναι δυνατή η κίνηση του προς τις δύο κατευθύνσεις που του αντιστοιχούν και αντίστοιχα φτιάχνει καινούριες καταστάσεις τις οποίες εισάγει στο vector<Puzzle \*> children και το επιστρέφει κατά την έξοδο της μεθόδου. Αν το αμάξι έχει οριζόντιο προσανατολισμό μπορεί να κατευθυνθεί δεξιά με την goRight() και αριστερά με την goLeft() ενώ αν έχει κάθετο προσανατολισμό μπορεί να κατευθυνθεί πάνω με την goUp() και κάτω με την goDown(). Επίσης για να μπορεί να μετακινηθεί προς οποιαδήποτε κατεύθυνση θα πρέπει η επιθυμητή θέση να μην καλύπτεται από ένα αντικείμενο ή ένα άλλο αυτοκίνητο.

Στο main.cpp υλοποιείται η συνάρτηση εισόδου στην οποία δημιουργείται η αρχική κατάσταση, καλείται η συνάρτηση επίλυσης και εκτυπώνονται στην οθόνη τα στατιστικά της επίλυσης.

Ο αλγόριθμος αναζήτησης που χρησιμοποιείται είναι ο Breadth First Search με τις καταστάσεις που έχουν ήδη εξεταστεί να αποθηκεύονται στο unordered\_map<string, Puzzle\*> closed και τις καταστάσεις που αναμένουν εξέταση στο μέτωπο αναζήτησης που υλοποιείται με ουρά. Ξεκινώντας με το μέτωπο να περιέχει μόνο την αρχική κατάσταση επαναλαμβάνω μέχρι να βρω την τελική ή να μην έχω άλλες διαθέσιμες καταστάσεις: παίρνω την παλαιότερη κατάσταση που μπήκε στο μέτωπο και αν είναι τελική την επιστρέφω. Διαφορετικά ελέγχω αν ανήκει στο κλειστό σύνολο και αν όχι την επεκτείνω και προσθέτω στο μέτωπο όσα παιδιά της δεν ανήκουν στο κλειστό. Η μεταβλητή mem μετράει τον μέγιστο συνολικό αριθμό καταστάσεων ταυτόχρονα στο μέτωπο και στο κλειστό σύνολο και η μεταβλητή examined τον συνολικό αριθμό καταστάσεων που έχουν επεκταθεί.

Για την αξιολόγηση του αλγορίθμου δημιουργήθηκαν τρεις αρχικές καταστάσεις αυξανόμενης δυσκολίας.

Με **Ο** σημειώνονται τα εμπόδια, με ↔ τα αμάξια οριζόντιου προσανατολισμού και με ↕ τα αμάξια κάθετου προσανατολισμού.

**1**ο επίπεδο:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** |
| **0** |  |  | **↔** | **O** |
| **1** | **O** | **↔** | **↕** |  |
| **2** |  | **↕** | **O** |  |
| **3** |  | **O** |  |  |

Οι αξιολογημένες καταστάσεις είναι 132 ενώ στην μνήμη υπάρχουν ταυτόχρονα μέχρι 171. Η λύση απαιτεί 11 κινήσεις.

**2**ο επίπεδο:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** |
| **0** |  | **O** |  |  |  |
| **1** |  | **↕** |  | **O** | **↔** |
| **2** |  |  |  |  |  |
| **3** | **O** | **↔** |  | **↕** | **↕** |
| **4** |  |  |  |  | **O** |

Οι αξιολογημένες καταστάσεις είναι 596 ενώ στην μνήμη υπάρχουν ταυτόχρονα μέχρι 707. Η λύση απαιτεί 15 κινήσεις.

**3**ο επίπεδο:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** |
| **0** |  |  |  |  |  |  |
| **1** |  | **O** |  |  | **↕** | **↔** |
| **2** |  |  |  |  |  |  |
| **3** | **↕** |  |  |  |  | **O** |
| **4** | **O** |  | **↔** | **↔** |  |  |
| **5** |  |  |  |  | **↕** |  |

Οι αξιολογημένες καταστάσεις είναι 7356 ενώ στην μνήμη υπάρχουν ταυτόχρονα μέχρι 9177. Η λύση απαιτεί 15 κινήσεις.