# Δεύτερη Εργασία Συστήματα Ευφυών Πρακτόρων - Αναφορά

Θεοφάνης Τριανταφύλλης
E17150
fanis_30fillis@outlook.com

Φεβρουάριος 2022

## Περίληψη Αρχιτεκτονικής

Η αρχιτεκτονική της εργασία που υλοποίησα έχει τέσσερις βασικές οντότητες:

- Πλέγμα - Grid
  Παίζει τον ρόλο του περιβάλλοντος. Αρχικοποιεί και έχει την λίστα με τα πλακίδια-θέσεις του περιβάλλοντος. Επίσης κάνει τις δημοπρασίες για την κατανομή των πελατών στους πράκτορες και για την επίλυση των συγκρούσεων.
- Πλακίδιο - Tile
  Έχει συντεταγμένες που δείχνουν την θέση του στον κόσμο.
- Πράκτορας - Agent
  Ο πράκτορας - ταξί που μεταφέρει πελάτες εκεί που πρέπει. Χρησιμοποιεί τον αλγόριθμο A* για τον σχεδιασμό των κινήσεων και συνεργάζεται με τον άλλο πράκτορα ώστε να μεταφέρουν όσο περισσότερους πελάτες γίνεται.

### Περίληψη Λειτουργίας

Αρχικά το περιβάλλον βρίσκει τις θέσεις των πρακτόρων στο πλέγμα και τους τις αποστέλλει, έπειτα δημιουργεί πέντε πελάτες και ενημερώνει τους πράκτορες ώστε να γίνουν οι δημοπρασίες για ανάληψη των πελατών.

Έπειτα ξεκινά η λειτουργία του περιβάλλοντος και των πρακτόρων, το περιβάλλον δημιουργεί έναν πελάτη αν υπάρχουν λιγότεροι από οχτώ ενεργοί πελάτες και ένας από τους πράκτορες δεν έχει πελάτη, αν έχουν περάσει πέντε προσπάθειες και υπάρχουν λιγότεροι από οχτώ ενεργοί πελάτες ή αν υπάρχουν λιγότεροι από πέντε ενεργοί πελάτες. Το περιβάλλον δίνει εξήντα κινήσεις στους πράκτορες και μετά τερματίζει.

## Λεπτομερής Περιγραφή

Σε αυτή την ενότητα θα περιγραφεί αναλυτικά η υλοποίηση της κάθε οντότητας. Αναφέρονται τα πεδία και οι μέθοδοι καθώς και λεπτομέρειες σχετικά με την λειτουργία τους όπου χρειάζεται.

### Πλακίδια

Το πλακίδιο είναι ουσιαστικά μια θέση στο πλέγμα και αποτελούν το περιβάλλον.

### Πεδία

Τα πεδία του πλακιδίου είναι:

- coordinates
  Κρατάει τις καρτεσιανές συντεταγμένες του πλακιδίου στο πλέγμα
- fullCost
  Κρατάει το πλήρες κόστος, δηλαδή το κόστος μέχρι το πλακίδιο και από το πλακίδιο στον στόχο. Χρησιμοποιείται από τον αλγόριθμο A*.
- costUntilHere
  Κρατάει το κόστος μέχρι εδώ.
- prev
  Κράταει το προηγούμενο πλακίδιο για να βρεθεί το μονοπάτι που πρέπει να ακολουθήσει ο πράκτορας για να φτάσει στον στόχο. Χρησιμοποιείται από τον αλγόριθμο A*.
- neighbors
  Κρατάει τους γείτονες του πλακιδίου.

### Μεθόδοι

Οι μέθοδοι του πλακιδίου είναι

- Tile
  Κατασκευαστής του πλακιδίου. Δέχεται μόνο τις συντεταγμένες του πλακιδίου.
- addNeighbor
  Προσθέτει έναν γείτονα στο πλακίδιο που δίνεται ως όρισμα.
- getHeuristic
  Παίρνει τον στόχο ως όρισμα και επιστρέφει την απόσταση του στόχου.
- euclDist
  Δέχεται συντεταγμένες του στόχου και επιστρέφει την Ευκλείδεια απόσταση από τον στόχο.
- compareTo
  Χρειάζεται για τις PriorityQueues που θα χρησιμοποιηθούν στον αλγόριθμο A*.

## Πράκτορας

Ο πράκτορας κάνει όλη την δουλειά, δηλαδή πρέπει να πάει στον πράκτορα για να τον παραλάβει και να τον παραδώσει εκεί που πρέπει.

### Αρχιτεκτονική

Ο πράκτορας είναι ένας κυρίως προληπτικός πράκτορας καθώς σχεδιάζει πλήρως το μονοπάτι που θα πάρει πριν ξεκινήσει και αφότου ξεκίνησε αλλάζει διαδρομή μόνο αν υπάρξει σύγκρουση με άλλον πράκτορα. Στην περίπτωση σύγκρουσης απλά περιμένει ώστε ο άλλος πράκτορας να ολοκληρώσει την κίνηση του και μετά κινείτε αυτός.

Οι πράκτορες δεν επικοινωνούν ποτέ μεταξύ τους απ' ευθείας αλλά για την κατανομή των πελατών και την επίλυση συγκρούσεων βασίζονται στην διαμεσολάβηση του περιβάλλοντος.

### Πεδία

Τα πεδία της οντότητας είναι:

- lastAction
  Δείχνει την ενέργεια που δεν έκανε ο πράκτορας την προηγούμενη φορά. Αν η ενέργεια του πράκτορα πραγματοποιήθηκε παίρνει τιμή κενής συμβολοσειράς.
- clientQueue
  Είναι μια ουρά που περιέχει τις θέσεις των πρακτόρων που έχει αναλάβει ο πράκτορας.
- clientTargetQueue
  Είναι μια ουρά που περιέχει τους στόχους των πρακτόρων που έχει αναλάβει ο πράκτορας.
- coordinates
  Καρτεσιανές συντεταγμένες που δείχνουν την τωρινή θέση του πράκτορα.
- currentTile
  Κρατάει το τωρινό πλακίδιο.
- target
  Καρτεσιανές συντεταγμένες που δείχνουν τον στόχο του πράκτορα στο περιβάλλον.
- targetTile
  Κρατάει το πλακίδιο που έχει ως στόχο ο πράκτορας.
- carrying
  Αληθοτιμή που δείχνει αν ο πράκτορας κουβαλάει τον πελάτη.
- tiles
  Κρατάει τα πλακίδια του περιβάλλοντος.
- path
  Κρατάει το μονοπάτι που πρέπει να ακολουθήσει ο πράκτορας.
- finalCoords
  Κρατάει τις συντεταγμένες που θα καταλήξει ο πράκτορας όταν τελειώσει την διαδρομή του. Ουσιαστικά είναι οι συντεταγμένες στόχου του τελευταίου πελάτη.

### Μεθόδοι

- resetPrev
  Αυτή η μέθοδος αφορά τον αλγόριθμο A* που χρησιμοποιεί το πεδίο prev των πλακιδίων για να βρει το μονοπάτι. Αυτή η συνάρτηση αφαιρεί από όλα τα πλακίδια την τιμή του πεδίου prev βάζοντας ένα null.
- eucledianDistance
  Επιστρέφει την Ευκλείδεια απόσταση δύο συντεταγμένων.
- aStar
  Υλοποιεί τον αλγόριθμο A*. Δέχεται ένα αρχικό πλακίδιο και το τελικό πλακίδιο. Ο αλγόριθμος θα αναλυθεί μετέπειτα.
- getPath
  Μετατρέπει τις συντεταγμένες σε ενέργειες. Χρησιμοποιεί μια Στοίβα ώστε να αποθηκεύσει τις συντεταγμένες των πλακιδίων και μετά μετατρέπει τις συντεταγμένες σε ενέργειες.

- chooseNextMove
  Επιστρέφει το την επόμενη κίνηση που πρέπει να κάνει ο πράκτορας.

- getClient
  Ενημερώνει το περιβάλλον ποιον πελάτη θέλει να αναλάβει.

- cooperate
  Συμμετέχει σε δημοπρασία με το περιβάλλον για να ανατεθεί ο πελάτης.

- handleConflict
  Συμμετέχει σε δημοπρασία με το περιβάλλον για να επιλυθεί η σύγκρουση θέσης.

### Δομές Περιβάλλοντος

Το πλέγμα του περιβάλλοντος αναπαριστάτε μόνο στους πράκτορες και είναι ένας γράφος όπου το κάθε πλακίδιο είναι ένας κόμβος με ακμές σε όλα του τα γειτονικά πλακίδια. Οι τοίχοι του περιβάλλοντος είναι υλοποιημένοι ως έλλειψη ακμών, κατά συνέπεια είναι αδύνατο ένας πράκτορας να περάσει μέσα από τοίχο.

### Αλληλεπίδραση με το Περιβάλλον

Ο πράκτορας αλληλεπιδρά με το περιβάλλον για να επιβεβαιώσει τις κινήσεις του, για να επιλυθούν οι τυχών συγκρούσεις μεταξύ των πρακτόρων και για να κατανεμηθούν οι πελάτες.

Στην περίπτωση της επιβεβαίωσης όταν ο πράκτορας κάνει μια νέα κίνηση δέχεται την νέα θέση του και αν κάνει προσπάθεια να σηκώσει ή να αφήσει κάποιον πελάτη δέχεται ανατροφοδότηση για το αν τα κατάφερε ή όχι. Επίσης ως ανατροφοδότηση ενδέχεται να σταλεί στον πράκτορα ότι υπάρχει Σύγκρουση, αυτή η περίπτωση θα εξεταστεί παρακάτω.

Στην περίπτωση σύγκρουσης μεταξύ πρακτόρων το περιβάλλον ενημερώνει τους πράκτορες και ο κάθε πράκτορας πρέπει να αποστείλει αν κουβαλάει πελάτη και το κόστος του μέχρι το τέλος της διαδρομής. Το περιβάλλον ενημερώνει τον κάθε πράκτορα για το αποτέλεσμα της δημοπρασίας.

Για το ενδεχόμενο ανάληψης πελάτη ο κάθε πράκτορας μόλις λάβει επόμενο πελάτη στέλνει τον αριθμό των τωρινών πελατών που έχουν αναλάβει, μετά περιμένει το επόμενο μήνυμα το οποίο είτε θα είναι για να δώσει το κόστος του είτε θα ανακοινώνει ότι δεν έλαβε τον πελάτη. Στην περίπτωση που του ζητήθηκε να στείλει το κόστος του το στέλνει και περιμένει τα αποτελέσματα της δημοπρασίας.

## Πλέγμα - Grid

Το πλέγμα είναι μια απλή οντότητα που πραγματοποιεί τις ενέργειες του πράκτορα και αρχικοποιεί τις θέσεις του πράκτορα και του πελάτη.

### Πεδία

Τα πεδία του πράκτορα είναι:

- carrying
  Πίνακας αληθοτιμών που δείχνει αν κουβαλάει ο κάθε πράκτορας.

- numberOfClients
  Κρατάει τον αριθμό των ενεργών πελατών.

- numberOfDeliveredClients
  Κρατάει τον αριθμό των πελατών που έχουν παραδοθεί.

- maxNumberOfClients
  Κρατάει τον μέγιστο αριθμό ταυτόχρονων πελατών.

- clients
  Είναι ένα λεξικό (Dictionary) που περιλαμβάνει όλους τους πράκτορες. Στο κλειδί υπάρχει ένας ακέραιος αριθμός που είναι μοναδικός για κάθε πελάτη και ως τιμή έχουμε έναν δισδιάστατο πίνακα με την τωρινή θέση και τον στόχο του πελάτη.

- assigned
  Είναι ένα λεξικό που κρατάει τους ανατεθειμένους πελάτες ανά πράκτορα. Ως κλειδί υπάρχει το αναγνωριστικό του πράκτορα και ως τιμή υπάρχει μια ουρά (Queue) που περιέχει όλους τους ανατεθειμένους πελάτες ανά πράκτορα.

- assignedClient
  Είναι ένας πίνακας που δείχνει ποιον πελάτη έχει αναλάβει τώρα ο πράκτορας.

- agentPos
  Δείχνει στην θέση του κάθε πράκτορα με καρτεσιανές συντεταγμένες.

- agents
  Κρατάει τα ονόματα των πρακτόρων.

- agentScore
  Κρατάει τους πόντους των πρακτόρων.

- clientId
  Κρατάει το επόμενο αναγνωριστικό του πελάτη.

Όταν ένας πράκτορας αναλαμβάνει έναν πελάτη μπαίνει στην ουρά πελατών που πρέπει αν ακολουθήσει, αυτός ο πελάτης θα είναι ο τελευταίος που θα παραδοθεί.

Στην μονοπρακτορική υλοποίηση χρησιμοποιήθηκαν δύο ArrayLists για να αποθηκεύονται οι θέσεις και οι στόχοι των πελατών, αυτό όμως δεν κλιμακώνει για τρεις πράκτορες καθώς το περιβάλλον πρέπει να μπορεί να κρατάει το ποιος πράκτορας έχει αναλάβει ποιόν πελάτη και αν τον έχει παραδώσει στην σωστή θέση. Αυτό διευκολύνεται πολύ με την χρήση ενός λεξικού και αναγνωριστικών για τους πελάτες.

### Μέθοδοι

Οι μέθοδοι που έχει η κλάση Grid είναι:

- sendClientToAgent
  Στέλνει τον πελάτη που δημιουργήθηκε στον πράκτορα.

- spawnClient
  Δημιουργεί έναν νέο πελάτη και ξεκινά την δημοπρασία για την κατανομή του σε κάποιον πράκτορα. Η διαδικασία της δημοπρασίας θα αναλυθεί αργότερα.

- pickUpClient
  Καλείτε όταν πρέπει να σηκώσει έναν πελάτη ένας από τους πράκτορες. Ενημερώνει τον πράκτορα για το αποτέλεσμα και ενημερώνει τους βαθμούς του πράκτορα.

- putDownClient
  Καλείτε όταν ο πράκτορας θέλει να αφήσει τον πελάτη σε ένα σημείο. Όπως και με το pickUpClient ενημερώνει τον πελάτη σχετικά με το αποτέλεσμα και τους πόντους.

- findStartingLocation
  Βρίσκει την αρχική θέση ενός πράκτορα.

- sendAgentPosition
  Στέλνει την θέση του πράκτορα.

- sendPickedUpClient
  Στέλνει μήνυμα στον πράκτορα ότι σήκωσε τον πελάτη επιτυχώς.

- sendNotPickedUpClient
  Στέλνει μήνυμα στον πράκτορα ότι δεν κατάφερε να σηκώσει τον πελάτη.

- moveAgent
  Δέχεται ένα όρισμα και μετακινεί τον πράκτορα ανάλογα με την κατεύθυνση που θέλει να μετακινηθεί.

- printGrid
  Εκτυπώνει το πλέγμα. Ενα κενό κελί σημαίνει ότι δεν υπάρχει εκεί πράκτορας ή πελάτης, το A σε ένα κελί σημαίνει ότι είναι ο πράκτορας, το C σημαίνει ότι είναι ο πελάτης και το B σημαίνει ότι είναι πελάτης και πράκτορας στο σημείο.

- getNewPositions
  Παίρνει τις επόμενες θέσεις των πρακτόρων με βάση τις κινήσεις που κάνουν.

- getConflictingAgents
  Καλείτε μετά από την getNewPositions και επιστρέφει μια λίστα από πράκτορες που έχουν σύγκρουση.

- auctionConflict
  Παίρνει τις ενέργειες των πρακτόρων, ελέγχει αν υπάρχει σύγκρουση μεταξύ τους και αν υπάρχει την επιλύει μέσω μιας δημοπρασίας.

## Αλγόριθμος A*

Ο αλγόριθμος υλοποιείται ως μέθοδος του πράκτορα. Χρησιμοποιεί δύο PriorityQueues για την λίστα με τα ανοιχτά πλακίδια, δηλαδή για τα πλακίδια που θα ελεγχθούν, και για την λίστα με κλειστά πλακίδια, δηλαδή πλακίδια που έχουν ελεγχθεί.

Αρχικά ο αλγόριθμος προσθέτει το αρχικό πλακίδιο στη λίστα με τα ανοιχτά πλακίδια, Μετά αφαιρεί το πρώτο πλακίδιο από την λίστα ανοιχτών πλακιδίων και ελέγχει αν είναι ο στόχος, αν δεν είναι τότε για κάθε από τους γείτονες του αν δεν υπάρχει σε κάποια από τις λίστες τότε θέτει ως προηγούμενο του, το πλακίδιο που επιλέχθηκε, παίρνει την ευριστική συνάρτηση κόστους του και τον προσθέτει στη λίστα με τα ανοιχτά πλακίδια. Μετά προσθέτει το πλακίδιο που ελέγχθηκε στη λίστα με τα κλειστά πλακίδια.

Αφότου ολοκληρωθεί ο αλγόριθμος πρέπει να παρθούν οι ενέργειες που πρέπει να γίνουν για να φτάσει ο πράκτορας στον στόχο του. Αυτό γίνεται με μια άλλη συνάρτηση που περνάει το μονοπάτι που έκανε ο πράκτορας και αποθηκεύει τις συντεταγμένες σε μια στοίβα ώστε να αντιστραφεί η σειρά τους. Μετά από αυτό παίρνει το πρώτο στοιχείο της στοίβας ως τις τωρινές συντεταγμένες του πράκτορα. Για κάθε μια από τις επόμενες αποθηκεύει την κίνηση που πρέπει να κάνει για να φτάσει σε αυτή σε μια ArrayList.

# Συνεργασία Πρακτόρων

Η συνεργασία των πρακτόρων γίνεται εξ' ολοκλήρου από το περιβάλλον, δηλαδή επικοινωνεί με τους πράκτορες και κανονίζει την συνεργασία τους ως μεσάζοντας.

## Κατανομή Πελατών

Η κατανομή των πελατών στους πράκτορες γίνεται μέσω μιας τροποποιημένης Αγγλικής δημοπρασίας που κάνει το περιβάλλον. Η δημοπρασία είναι τροποποιημένη καθώς η τελική τιμή της προσφοράς υπολογίζεται από το περιβάλλον αφότου λάβει κάποιες πληροφορίες από τον πράκτορα.

Αρχικά το περιβάλλον παίρνει από τους πράκτορες τον αριθμό των πελατών που έχει. Όσοι δεν έχουν κάποιον πελάτη αυτόματα περνούν στον επόμενο γύρο. Αν έχουν όλοι κάποιον πελάτη τότε όλοι προχωρούν στην επόμενη φάση της δημοπρασίας.

Αποστέλλει σε όλους τους πράκτορες που έχουν περάσει το πρώτο στάδιο την θέση και τον στόχο του πελάτη, αφότου οι πράκτορες υπολογίζουν το κόστος της τελευταίας θέσης που θα πάρουν οι πράκτορες, αυτή θα είναι η τωρινή τους θέση αν δεν έχουν πελάτη ή ο στόχος του τελευταίου πελάτη που έχουν αναλάβει. Αφότου υπολογίσουν το κόστος τους από τον πελάτη το αποστέλλουν στο περιβάλλον που δίνει τον πελάτη στον πράκτορα με το λιγότερο κόστος. Το περιβάλλον υποθέτει ότι όλοι οι πράκτορες είναι ειλικρινής και κατά συνέπεια δεν ελέγχει το κόστος.

## Επίλυση Συγκρούσεων

Αν δύο πράκτορες θέλουν να προσπελάσουν το ίδιο πλακίδιο τότε έχουμε σύγκρουση μεταξύ τους η οποία πρέπει να λυθεί από το περιβάλλον. Στην υλοποίηση του περιβάλλοντος αν κάποιος πράκτορας κάνει φόρτωση ή αποφόρτωση κάποιου πελάτη αυτόματα νικάει την δημοπρασία, η δημοπρασία γίνεται αλλά είναι στημένη ώστε να νικήσει ο πράκτορας αυτός. Επίσης υπάρχει το ενδεχόμενο να υπάρχει μόνο μια επίλυση της σύγκρουσης, σε αυτή την περίπτωση το περιβάλλον την αντιληφθεί και θα στήσει την δημοπρασία ώστε να επικρατήσει αυτή η λύση. Κατά τη διάρκεια της δημοπρασίας το περιβάλλον δέχεται το πόσους πελάτες έχει ο κάθε πράκτορας και το κόστος από τον προορισμό ώστε να βρει ποιος θα αναλάβει τον πελάτη.

# Αποτελέσματα Αρχιτεκτονικής

Τα αποτελέσματα της αρχιτεκτονικής είναι:

Σε δέκα επαναλήψεις η αρχιτεκτονική είχε τα αποτελέσματα:

| Πελάτες | 15 | 14 | 15 | 11 | 13 | 12 | 15 | 11 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|
| Πόντοι | 120 | 100 | 120 | 50 | 80 | 60 | 120 | 40 | 80 | 100 |

Όπως φαίνεται υπάρχει μεγαλύτερη διασπορά στα αποτελέσματα από την διασπορά του μονοπρακτορικού συστήματος. Πέρα από αυτή την παρατήρηση τα αποτελέσματα είναι περίπου τριπλάσια από τα αποτελέσματα της μονοπρακτορικής αρχιτεκτονικής. Αυτό σημαίνει ότι δεν υπάρχει μεγάλη απώλεια από την συνεργασία των πρακτόρων.