

Project 2 - Τεχνητή Νοημοσύνη

Παναγιώτα Γύφτου , Α.Μ.: 1115201900318

Δεκέμβριος 2022

Περίληψη

Θέμα εργασίας: Αναζήτηση με Αντιπαλότητα.

Readme προγραμματιστικών προβλημάτων *Pacman*.

Question1

Για την αξιολόγηση της κίνησης, λαμβάνονται υπόψη κάποιες καταστάσεις με τις οποίες έρχεται αντιμέτωπος ο *pacman*. Κάποιες είναι θετικές ως προς τον *pacman* και κάποιες άλλες είναι αρνητικές. Αρχικά ελέγχεται εάν η απόσταση μεταξύ του *pacman* και κάποιου δυνατού φαντάσματος είναι μικρότερη ή ίση της μονάδας. Εάν βρίσκεται ο *pacman* σε αυτή την κατάσταση τότε επιστρέφεται τιμή μείον άπειρο, ώστε να αποφύγει αυτή την κίνηση ο *pacman*, διότι θα ηττηθεί. Στην συνέχεια ελέγχεται αν η κατεύθυνση του *pacman* που επιλέχθηκε για να βρεθεί στην καινούργια θέση είναι η *stop*. Γνωρίζουμε ότι όταν ο *pacman* είναι στάσιμος, ο χρόνος που σπαταλιέται άσκοπα, δρά αρνητικά στο σκορ, με αποτέλεσμα να χάνει μονάδες, για τον λόγο αυτό ο μετρητής του σκορ μειώνεται κατά 10. Ο τρίτος και ο τέταρτος έλεγχος δίνουν μπόνους στον *pacman*. Πιο συγκεκριμένα:

3ος έλεγχος: Αν ο *pacman* βρίσκεται σε απόσταση 0 από ένα φοβισμένο φάντασμα, λαμβάνει 100 μπόνους πόντους ή εάν βρίσκεται σε απόσταση μήκους 1 παίρνει το 20% του μπόνους της απόστασης μήκους 0, δηλαδή +20.

4ος έλεγχος: Τέταρτος έλεγχος: Αν ο *pacman* βρίσκεται σε απόσταση 0 από ένα χάπι δύναμης, λαμβάνει 50 μπόνους πόντους ή εάν βρίσκεται σε απόσταση μήκους 1 παίρνει το 20% του μπόνους της απόστασης μήκους 0, δηλαδή +10.

Τέλος συγκρίνεται σε ποιά από τις δύο θέσεις (προηγούμενη ή καινούρια) βρίσκεται το κοντινότερο φαγητό. Εάν το κοντινότερο φαγητό είναι ως προς την προηγούμενη θέση πιο κοντά τότε ο *pacman* δέχεται μία μείωση του σκορ κατά 10, διαφορετικά παίρνει ένα μικρό μπόνους ανάλογα με το μήκος απόστασης του κοντινότερου φαγητού. Η συνάρτηση επιστρέφει το σκορ του διαδόχου συν το σκορ με τα μπόνους και τις μειώσεις.

Question2

Η getAction:

Στο σώμα της ελέγχονται όλες οι κινήσεις του *pacman* στην αρχή του παιχνιδιού, και σκοπός είναι να βρεθεί εκείνη η κίνηση με την μεγαλύτερη αξία. Για κάθε κίνηση του κόμβου ρίζα καλείται η συνάρτηση *MiniMax_Search* με ορίσματα την καινούρια κατάσταση, το *index* του *pacman*, και το βάθος της τελευταίας στρώσης. Η *MiniMax_Search* επιστρέφει την αξία της κίνησης που της ζητήθηκε να υπολογίσει. Σε ένα *dictionary* με όνομα *ActionsValues* αποθηκεύονται οι αρχικές κινήσεις με τις αξίες τους. Αφού έχουν ελεγχθεί όλες οι αρχικές κινήσεις του *pacman* παίρνουμε την κίνηση με την μέγιστη αξία που έχει αποθηκευτεί στο λεξικό και επιστρέφεται ως η καλύτερη κίνηση.

Η Is_Terminal:

Η συνάρτηση αυτή επιστρέφει μια *boolean* τιμή δείχνοντας εάν η τρέχουσα κατάσταση είναι τελική ή όχι.

Η MiniMax_Search:

Η συνάρτηση *MiniMax_Search* αρχικά βρίσκει τον επόμενο πράκτορα προς αναζήτηση. Έπειτα ελέγχει εάν το μέγεθος της στρώσης (*ply*) στην επομενή επανάληψη μηδενιστεί, αυτό σημαίνει ότι βρισκόμαστε στα φύλλα και πρέπει να κληθεί η συνάρτηση *evaluationFunction*, για να επιστρέψει την τιμή αξιολόγησης του φύλλου. Εάν η στρώση *ply* – 1 είναι διάφορη του μηδενός συνεχίζουμε τις αναζητήσεις για την εύρεση της αξίας της κίνησης που είναι για έλεγχο. Εάν ο πράκτορας είναι ο *pacman*, τότε επιστρέφεται η τιμή της συνάρτησης *Max_Value*, διαφορετικά αν είναι ο πρακτορας φάντασμα τότε καλείται η *Min_Value* συνάρτηση. Οι συναρτήσεις *Max_Value* και *Min_Value*, διατρέχουν όλο το δέντρο παιχνιδιού μέχρι τα φύλλα.

Η *Max_Value*:

Επιστρέφει εκείνη την κίνηση με την μέγιστη αξία των διαδόχων του *pacman*.

Η *Min_Value*:

Επιστρέφει εκείνη την κίνηση με την ελάχιστη αξία των διαδόχων του φαντάσματος.

Οι συναρτήσεις *MiniMax_Search*, *Max_Value* και *Min_Value* είναι γραμμένες με την βοήθεια των διαφανειών των διαλέξεων, του εργαστηρίου και του βιβλίου Σελ.177

Question3

Η λογική υλοποίησης είναι η ίδια με αυτή του *Question2*. Οι αλλαγές είναι λίγες.
Πιο συγκεκριμένα:

1. Η ενημέρωση της μεταβλητής *maxVal* (είναι η μέγιστη αξία κίνησης) στην συνάρτηση *getAction* γίνεται εντός του βρόχου ελέγχου και εύρεσης της κίνησης, διότι τώρα θέλουμε να ενημερώσουμε το α , με βάση το οποίο θα γίνει η επιλογή της κίνησης. Όταν τελειώσουμε και αποθηκευτούν οι τιμές στο λεξικό, θα ψάξουμε στο *dictionary* για την κίνηση με αξία α , η οποία και θα επιστραφεί ως η καλύτερη κίνηση.
2. Η συνάρτηση *Alpha_Beta_Search* έχει ακριβώς την ίδια λειτουργία με την *MiniMax_Search*.
3. Η συνάρτηση *Max_Value* με την ενημέρωση της *maxVal* ενημερώνει και την μεταβλητή α , λαμβάνοντας την μεγαλύτερη τιμή αξίας κίνησης που έχει ένας διάδοχος, αντίστοιχα στην *Min_Value* με την ενημέρωση της *minVal* ενημερώνει και την μεταβλητή β , λαμβάνοντας την μικρότερη τιμή αξίας κίνησης που έχει ένας διάδοχος.

Οι συναρτήσεις *Alpha_Beta_Search*, *Max_Value* και *Min_Value* είναι γραμμένες με την βοήθεια των διαφανειών των διαλέξεων, του εργαστηρίου και του βιβλίου Σελ.182

Question4

Η λογική υλοποίησης είναι η ίδια με αυτή του *Question2*. Οι αλλαγές είναι λίγες.
Πιο συγκεκριμένα:

1. Η συνάρτηση *ExpectiMiniMax_Search* έχει ακριβώς την ίδια λειτουργία με την *MiniMax_Search*.
2. Η συνάρτηση *Expected_Value* είναι παρόμοια με την *Min_Value*, με την διαφορά ότι δεν υπολογίζουμε την ελάχιστη αξία που μπορεί να έχει ένας διάδοχος αλλά των μέσο όρο των αξιών των κινήσεων των διαδόχων.

Οι συναρτήσεις *ExpectiMiniMax_Search*, *Max_Value* και *Expected_Value* είναι γραμμένες με την βοήθεια των διαφανειών των διαλέξεων, του εργαστηρίου και του βιβλίου Σελ. 177,195,196

Question5

Η λογική υλοποίησης είναι η ίδια με αυτή του *Question1*. Οι αλλαγές είναι λίγες.
Πιο συγκεκριμένα:

1. Πλεον δεν επιστρέφεται τιμή μείον άπειρο όταν ο *pacman* έρχεται κοντά με ένα δυνατό φάντασμα αλλά αντιθέτως του δίνεται μεγάλη τιμή, ώστε να τον ωθίσουμε να φάει τα κοντινότερα μπιλάκια και να αποφύγει έτσι την συνάντηση με το φάντασμα.
2. Έχει αφαιρεθεί ο έλεγχος προσανατολισμού του *pacman*, διότι τώρα ενδιαφερόμαστε για την τρέχουσα κατάσταση.
3. Για τον ίδιο λόγο έχει αφαιρεθεί ο και ο έλεγχος κοντινότερης απόστασης φαγητού από την προηγούμενη ή την νέα θέση.