ΥΣ02 ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ-ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2021-2022

ΕΡΓΑΣΙΑ ΔΕΥΤΕΡΗ

ΠΑΝΑΓΙΩΤΗΣ ΚΟΝΤΟΕΙΔΗΣ

ΑΜ 1115201900266

Q1.

Για την υλοποίηση του της συνάρτησης αξιολόγησης του πρώτου ερωτήματος, αρχικά βρέθηκαν οι θέσεις των φαντασμάτων και υπολογίστηκε η απόσταση Μανχάταν του πακμαν από το καθένα από αυτά. Οι αποστάσεις αποθηκεύτηκαν στην λίστα ghost_dist. Στην συνέχεια με τον ίδιο τρόπο υπολογίστηκε η απόσταση Μανχάταν του πακμαν από την κάθε τροφή και οι αποστάσεις τοποθετήθηκαν στην λίστα food_dist. Ο σκοπός της συνάρτησης αξιολόγησης είναι να επιστρέφει υψηλότερες τιμές για ευνοϊκότερες καταστάσεις του πακμαν. Σκεπτόμενος από την εργασία το πρόβλημα των τεσσάρων γωνιών πειραματίστηκα αρκετά με τις μέγιστες-ελάχιστες αποστάσεις των τροφών και των φαντασμάτων και κατέληξα σε αυτή την παράσταση:

```
return childGameState.getScore()-max(food_dist)-
0.01*min(food_dist)+min(ghost_dist)
```

Από αυτήν την παράσταση παρατήρησα ότι όταν αυξάνεται ο συντελεστής του min(ghost_dist) το πακμαν κερδίζει σε κάθε περίπτωση, όμως με πολύ χαμηλό σκορ, αφού οι κινήσεις του εξαρτούνται κυρίως από τις κινήσεις των φαντασμάτων. Λόγω αυτού, πέρα από την αφαίρεση του max(food_dist), αφαιρείται και το min(food_dist), με έναν πολύ χαμηλό συντελεστή, για την βελτίστοποίηση των κινήσεων χωρίς να επεμβαίνει τόσο πολύ στην απόσταση του πακμαν από τα φαντάσματα.

Εκτός αυτού όταν τα φαντάσματα βρίσκονται σε τρομαγμένη κατάσταση δεν προσμετράται η απόσταση του πακμαν από αυτά.

Q2.

Για την υλοποίηση του δεύτερου ερωτήματος χρησιμοποιήθηκε εξ' ολοκλήρου ο αλγόριθμος του βιβλίου των Stuart Russell και Peter Norvig, <<Τεχνητή Νοημοσύνη Μια σύγχρονη προσέγγιση>> 4η έκδοση, σελ.177 εικόνα 5.3 .

```
function MINIMAX-SEARCH(παιχνίδι, κατάσταση) returns μια ενέργεια
  παίκτης ← παιχνίδι.TO-MOVE(κατάσταση)
  αξία, κίνηση ← MAX-VALUE(παιχνίδι, κατάσταση)
  return κίνηση

function MAX-VALUE(παιχνίδι, κατάσταση) returns ένα ζεύγος (χρησιμότητα, κίνηση)
  if παιχνίδι.IS-TERMINAL(κατάσταση) then return παιχνίδι.UTILITY(κατάσταση, παίκτης), null
  v ← −∞
  for each a in παιχνίδι.ACTIONS(κατάσταση) do
    v2, a2 ← MIN-VALUE(παιχνίδι, παιχνίδι.RESULT(κατάσταση, a))
    if v2 > v then
      v, κίνηση ← v2, a
  return v, κίνηση

function MIN-VALUE(παιχνίδι, κατάσταση) returns ένα ζεύγος (χρησιμότητα, κίνηση)
  if παιχνίδι.IS-TERMINAL(κατάσταση) then return παιχνίδι.UTILITY(κατάσταση, παίκτης), null
  v ← +∞
  for each a in παιχνίδι.ACTIONS(κατάσταση) do
    v2, a2 ← MAX-VALUE(παιχνίδι, παιχνίδι.RESULT(κατάσταση, a))
    if v2 < v then
      v, κίνηση ← v2, a
  return v, κίνηση
```

Η minimax τιμή επιστρέφεται από την getAction που χρησιμοποιεί την max_value η οποία δέχεται σαν ορίσματα την κατάσταση του παιχνιδιού(gameState), τον αριθμό του Πράκτορα(AgentIndex) και το βάθος(depth). Η getAction καλεί την max_value με agentIndex=0 και depth=0, αφού στην ρίζα του δέντρου πάντα έχουμε έναν κόμβο μαξ που είναι ο πακ μαν και το βάθος ισούται με 0. Στην συνέχεια, οι δύο συναρτήσεις min_value max_value καλούνται αναδρομικά μέχρι να διατρέξουν όλο το δέντρο που έχει σχηματιστεί και να γυρίσουν την τιμή minimax. Περαιτέρω σχεδιαστικές επιλογές είναι ότι στην max_value εφόσον το agentIndex είναι πάντα 0, καλείται η min_value αυξάνοντας το agentIndex αλλά διατηρώντας το βάθος. Στην min_value ελέγχεται αν το φάντασμα που εξετάζεται είναι το τελευταίο φάντασμα έτσι ώστε μετά να κληθεί η max_value και να αυξηθεί το βάθος κατά 1. Εάν όμως υπάρχουν και άλλα φαντάσματα που πρέπει να εξεταστούν τότε καλείται η min_value ξανά, το βάθος παραμένει σταθερό και αυξάνεται κατά 1 το agentIndex για το επόμενο φάντασμα.

Q3.

Για την υλοποίηση του τρίτου ερωτήματος χρησιμοποιήθηκαν οι συναρτήσεις getAction, max_value, min_value του δεύτερου ερωτήματος. Ακολουθώντας τον αλγόριθμο από τις διαφάνειες του φροντιστηρίου έγιναν κάποιες προσθήκες στις συναρτήσεις αυτές, για την εφαρμογή του κλαδέματος. Συγκεκριμένα προστέθηκαν σαν ορίσματα τα α,β στις συναρτήσεις max_value,min_value, καθώς και οι απαραίτητοι έλεγχοι για τα α,β στις max_value και min_value.

Q4.

Για την υλοποίηση του τέταρτου ερωτήματος χρησιμοποιήθηκαν οι συναρτήσεις getAction και max_value του δευτέρου ερωτήματος. Σε αυτές τις δύο συναρτήσεις δεν πραγματοποιήθηκε καμία αλλαγή. Για την μοντελοποίηση της τυχαιότητας των κινήσεων των φαντασμάτων προστέθηκε μια νέα συνάρτηση η chance_value, η οποία υπολογίζει την πιθανότητα p της εκάστοτε κίνησης του φαντάσματος και επιστρέφει το άθροισμα των τιμών που προκύπτουν από τις διαθέσιμες κινήσεις επί την πιθανότητα της κάθεμιας, p, να προκύψει.

Q5.

Για την υλοποίηση του πέμπτου ερωτήματος χρησιμοποιήθηκε η συνάρτηση αξιολόγησης του πρώτου ερωτήματος. Παρόλο που, η συνάρτηση του πρώτου ερωτήματος έπαιρνε στο πέμπτο ερώτημα μέσο σκορ 1038 στον autograder και 6/6 μονάδες με την προσθήκη μίας βελτιστοποίησης ανέβηκε το μέσο σκορ στο 1202,6. Η προσθήκη που έγινε ήταν η καταμέτρηση της τιμής newScaredTimes, όπου όταν η τιμή αυτή δεν είναι μηδενική, προστίθεται στην υπόλοιπη παράσταση με έναν χαμηλό συντελεστή(0.5). Αυτό προκύπτει από το γεγονός ότι όταν η τιμή newScaredTimes δεν είναι μηδενική σημαίνει ότι τα φαντάσματα δεν μπορούν να σκοτώσουν το πακμαν, επομένως το πακμαν έχει ένα πλεονέκτημα σε αυτήν την κατάσταση το οποίο και εκμεταλλεύεται.