Pacman Project 2

Q1

Σε αυτό το ερώτημα το evaluation function που υλοποίησα λαμβάνει υπόψην τους εξής παράγοντες (με σειρά προτεραιότητας):

- Πόσο κοντά σε τρομαγμένο (scared) φάντασμα πηγαίνει ο pacman
- Πόσο κοντά σε φάντασμα πηγαίνει ο pacman
- Πόσο κοντά σε κάποιο φαγητό πηγαίνει ο pacman
- Αν στην επόμενη κίνηση τρώει μεγάλο food dot
- Αν στην επόμενη κίνηση είναι πάνω σε φάντασμα (χάνει)

Αρχικά παίρνουμε ως base το score του pacman (eval). Για το κοντινότερο food dot, προσθέτω στο eval 4/(την απόσταση αυτή). Έτσι όσο μικραίνει η απόσταση προς το food dot, μεγαλώνει το evaluation.

Αντίστοιχα κάνω το ίδιο με την απόσταση προς κάθε scared φάντασμα, (1/(την απόσταση)), καθώς και με την απόσταση προς κάθε κανονικό φάντασμα (3/(την απόσταση)).
Με αυτόν τον τρόπο λαμβάνει μεγαλύτερη προτεραιότητα το φαγητό, εκτός και αν υπάρχουν πολλά φαντάσματα σε ένα σημείο, όπου και θα λάβει προτεραιότητα το να τα κυνηγήσει ή να τα αποφύγει αντίστοιχα.

Επιπλέον, δίνω τεράστια προτεραιότητα στο αν στην επόμενη κίνηση τα scaredtimes αυξάνονται (δηλαδή να τρώει πάντα την μεγάλη τελεία, γιαυτό προσθέτω 200 στο evaluation, πολύ μεγαλύτερο αριθμό απ'τους παραπάνω).

Τέλος, ύψιστη σημασία δίνεται στο αν στην επόμενη κίνηση πέφτει πάνω σε κανονικό φάντασμα, όπου και προσθέτω έναν τεράστιο αριθμό στο eval.

Η επιλογή των αριθμών (200, 4, 1, 3) που εμπλέκονται στον υπολογισμό του eval έγινε μετά από αρκετό testing πολλών τιμών και έγιναν tuned για να περνάνε τα test του autograder.

Q2

Σε αυτό το ερώτημα υλοποιήθηκε ο αλγόριθμος minimax όπως ακριβώς περιγράφεται στις διαφάνειες, με την προσθήκη μιας μεταβλητής depth όπου κρατάει το επίπεδο που βρισκόμαστε για να τερματίσουμε την αναζήτηση.

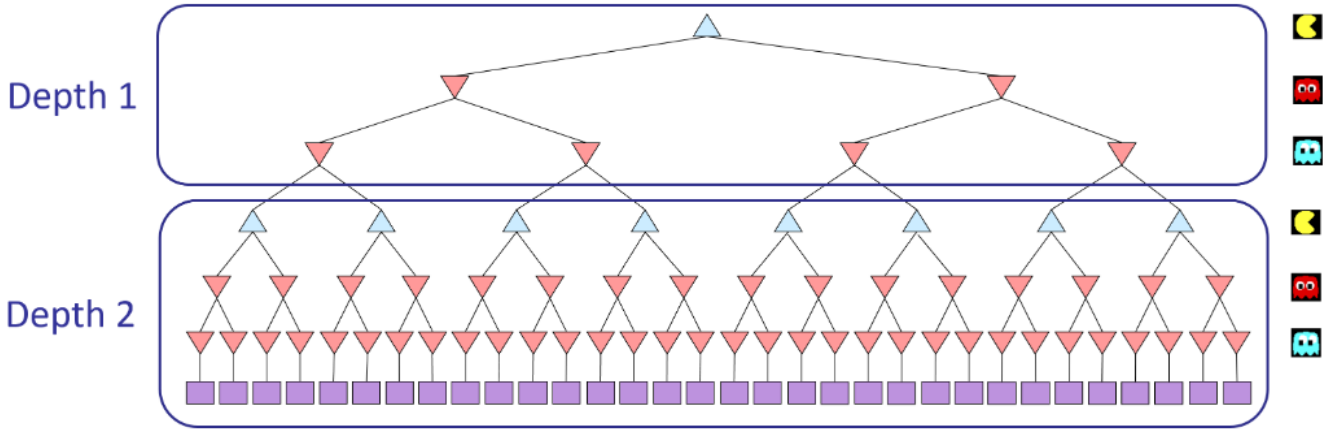
Τιμές Minimax

- Έτσι έχουμε την ακόλουθη ισότητα:

$$\text{MINIMAX}(s) = \begin{cases} \text{UTILITY}(s) \text{ if } \text{TERMINAL} - \text{TEST}(s) \\ \max_{a \in \text{ACTIONS}(s)} \text{MINIMAX}\big(\text{RESULT}(s,a)\big) \text{ if } \text{PLAYER}(s) = \text{MAX} \\ \min_{a \in \text{ACTIONS}(s)} \text{MINIMAX}\big(\text{RESULT}(s,a)\big) \text{ if } \text{PLAYER}(s) = \text{MIN} \end{cases}$$

- Μπορούμε να εφαρμόσουμε την παραπάνω ισότητα στο προηγούμενο δένδρο παιχνιδιού και να υπολογίσουμε τις minimax τιμές για κάθε κατάσταση.

Το επίπεδο αυξάνεται όταν βρισκόμαστε στον τελευταίο MIN, όπως φαίνεται και στο site των pacman project.



Επιπλέον σε κάθε αναδρομική κλήση από MIN παίκτη χρησιμοποιούμε το `(agentIndex + 1)%gameState.getNumAgents()` για να κάνουμε loop όλα τα indices. (αν είναι 0 ξέρουμε το επόμενο index, οπότε βάζουμε την παράμετρο agentIndex κατευθείαν 1).

Τέλος, αφού βρήκαμε την minimax τιμή στην ρίζα, επιστρέφουμε με lambda το action που αντιστοιχεί σε αυτήν την τιμή.

`return max(gameState.getLegalActions(0), key=lambda action: minimax(gameState.generateSuccessor(0, action), 0, 1))`.

Q3

Εδώ υλοποιήθηκε ο αλγόριθμος alpha-beta search. Ο κώδικας είναι παρόμοιος με τις διαφάνειες, απλά οι συναρτήσεις min-value και max-value έγιναν "hard-coded" στην συνάρτηση. Η λογική του incrementation του agentIndex και του depth είναι η ίδια με τον minimax.

Τέλος, επιστρέφουμε το action που αντιστοιχεί στο alpha-beta value που τρέξαμε στην ρίζα. (εδώ δεν μπορούμε με lambda γιατί χρειαζόμαστε και το alpha από το πρώτο branch όταν εξετάζουμε το 2ο κ.ο.κ οπότε το κάνουμε με loop)

Q4

Εδώ και πάλι υλοποιήθηκε ο αλγόριθμος αναζήτησης expectimax ακριβώς όπως περιγράφεται στις διαφάνειες.

Expectimax Τιμές

- Η μαθηματική παράσταση για τις τιμές **expectimax** είναι:

$$\text{EXPECTIMAX}(s) = \begin{cases} \text{UTILITY}(s) \text{ if } \text{TERMINAL} - \text{TEST}(s) \\ \max_a \text{EXPECTIMAX}(\text{RESULT}(s,a)) \text{ if } \text{PLAYER}(s) = \text{MAX} \\ \sum_r P(r)\text{EXPECTIMAX}(\text{RESULT}(s,r)) \text{ if } \text{PLAYER}(s) = \text{CHANCE} \end{cases}$$

όπου το r συμβολίζει το αποτέλεσμα μια ζαριάς (ή ενός άλλου τυχαίου γεγονότος), $P(r)$ είναι η πιθανότητα του r και $\text{RESULT}(s,r)$ είναι η ίδια κατάσταση όπως η s με την επιπλέον συνθήκη ότι το αποτέλεσμα της ζαριάς είναι r.

- Ο αλγόριθμος που αντιστοιχεί στην παραπάνω παράσταση αφήνεται σαν άσκηση.

Ο κώδικας είναι πανομοιότυπος με τον minimax, απλά εδώ θεωρούμε τα min nodes ως κόμβους τύχης και έτσι παίρνουν τιμή το average των τιμών των παιδιών τους.

Q5

Αυτό το ερώτημα είναι ακριβώς το ίδιο με το ερώτημα Q1.
Κάνουμε το ίδιο ακριβώς evaluation, απλά για το current state και όχι το successor state.