



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

Τεχνητή Νοημοσύνη

2η Εργασία

Μανούσος Λιναρδάκης, it22064

Ερώτηση 1:

Ο pacman ακολουθεί την στρατηγική minimax στην οποία θεωρεί ότι ο αντίπαλος παίζει βέλτιστα. Κάποιες φορές ο pacman μπορεί να “χάνει” με τον minimax επειδή αποφάσισε ότι έτσι θα κερδίσει το μεγαλύτερο score (αν είναι παγιδευμένος από “βέλτιστα” φαντάσματα ή τον πιέζει ο χρόνος). Επίσης, ο pacman περιμένει τα φαντάσματα να τον πλησιάσουν στον minimax, καθώς όταν τον ακολουθούν (από πίσω του), ο pacman μπορεί να συνεχίσει να παίζει και να μην παγιδεύεται από αυτά.

Ερώτηση 2:

Εφόσον ο pacman βρίσκεται σε μία κατάσταση στην οποία δεν μπορεί να κερδίσει επιπλέον πόντους (είναι “παγιδευμένος” από τους αντιπάλους του και πιστεύει ότι παίζουν βέλτιστα) αποφασίζει ότι η καλύτερη κίνηση είναι να “χάσει” το συντομότερο δυνατόν, έτσι ώστε να μην χάσει και άλλους πόντους εξαιτίας του χρόνου. Η στρατηγική Minimax είναι λάθος όταν οι αντίπαλοι δεν παίζουν βέλτιστα, καθώς μπορεί να οδηγήσει σε περιπτώσεις όπου ο pacman κερδίζει περισσότερους πόντους.

Για παράδειγμα, αν ο pacman θεωρεί ότι οι αντίπαλοι παίζουν βέλτιστα (minimax) αυτό είναι το αποτέλεσμα:



Ο pacman προσπάθησε να χάσει το συντομότερο δυνατόν!

Η στρατηγική αυτή είναι λάθος όταν για παράδειγμα το μπλε φάντασμα παίζει τυχαία, καθώς ο pacman μπορεί να κερδίσει περισσότερους πόντους, όπως φαίνεται και παρακάτω:



Ερώτηση 3:

Η συνάρτηση αξιολόγησης που υλοποίησα λαμβάνει υπόψη όσα attributes κάνουν καλή ή χειρότερη μία κατάσταση. Αυτά τα attributes που σκέφτηκα είναι το score, καθώς ένα ψηλότερο score σημαίνει καλύτερη κατάσταση, την απόσταση από την κοντινότερη κουκίδα, αφού όσο πιο κοντά βρίσκεται ο pacman σε μία κουκίδα τόσο το καλύτερο, την απόσταση (του pacman) από τα φαντάσματα και τον αριθμό των εναπομείναντων φαγητών και capsules. Όλα αυτά τα attributes προστίθενται σε ένα τελικό άθροισμα, το οποίο αποτελεί και το evaluation της κατάστασης. Φυσικά, κάποια μετράνε αρνητικά στο τελικό evaluation της κατάστασης, όπως το πόσα φαγητά και capsules έχουν μείνει, για αυτό και αφαιρούμε τα attributes αυτά (καθώς αν είναι πολλά, σημαίνει ότι ο pacman χρειάζεται περισσότερο χρόνο για να τελειώσει το παιχνίδι, άρα βρίσκεται σε χειρότερη κατάσταση). Είναι αξιοσημείωτο ότι στο τελικό άθροισμα προσθέτουμε το αντίστροφο του closest_food ($1 / \text{closest_food}$) καθώς έτσι όσο πιο μικρή είναι η απόσταση του closest_food από τον pacman, τόσο το καλύτερο είναι το evaluation – άθροισμα. Επίσης, όταν τα φαντάσματα είναι πολύ κοντά στο pacman (σε απόσταση manhattan μικρότερη ή ίση του 2 – το νούμερο αυτό βγήκε πειραματικά καθώς είχε το καλύτερο score), τότε **δεν** συγκεντρωνόμαστε στο κοντινότερο φαγητό (για αυτό το θέτουμε ως άπειρο, με αποτέλεσμα το $1 / \text{closest_food}$ που προστίθεται στο τελικό evaluation να είναι σχεδόν 0).

Εκτελώντας την εντολή:

```
python pacman.py -p AlphaBetaAgent -a evalFn=better,depth=2 -l  
smallClassic -k 2
```

και συγκρίνοντας το αποτέλεσμα της με αυτό της άσκησης 1 (minimax), τρέχοντας:

```
python pacman.py -p AlphaBetaAgent -a depth=2 -l smallClassic -k 2
```

παρατηρούμε ότι ο πράκτορας είναι πιο “αισιόδοξος” και δεν μένει τόσο πολύ σε ένα σημείο χρησιμοποιώντας το better evaluation function. Η συμπεριφορά αυτή εξηγείται καθώς λαμβάνει υπόψη πιο δυναμικά στοιχεία (όπως το score και την απόσταση από το κοντινότερο φαγητό), σε αντίθεση με τον minimax που απλώς βασίζεται στον αντίπαλο να παίζει τέλεια (με αποτέλεσμα να μην κάνει πολλές κινήσεις). Έτσι με την χρήση του better evaluation function κερδίζουμε μεγαλύτερο score από τον minimax. Είναι επίσης αξιοσημείωτο, ότι οι αντίπαλοι δεν είναι βέλτιστοι (κάνουν τυχαίες κινήσεις) με αποτέλεσμα ο minimax να μην είναι βέλτιστος.