

## Εργασία 2 ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

### Παραδοχές και Σχολία στον κώδικα

#### Ασκηση 1

Για την πρώτη άσκηση απλά δημιουργήσα μια απλή EvaluationFunction η οποία παρακίνηει τον pacman να κατευθυνθεί προς τα φαγητά του map όσο δεν τον διατρεχει ο κίνδυνος. Επίσης χρησιμοποιήσα έναν έλεγχο για το ποσο κοντα είναι τα φαντάσματα επιστρέφοντας μια πολυ μικρη τιμη για να παρακινήσει τον pacman οπωσδήποτε να μετακινηθεί.

#### Ασκηση 2

Για την άσκηση 2 απλά χρησιμοποιήσα τις διαφανειες την θεωρίας και υλοποίησα έναν αλγοριθμο minimax που ουσιαστικά είναι αναδρομικός με προσθήκη καποιων ελεγχων για να γνωρίζουμε ποτε παιζει ο pacman και ποτε τα φαντάσματα για την αναλογη κληση της maxvalue ή της minvalue αλλά και για να ξερούμε σε τι βαθος βρίσκομαστε στο δεντρο (deepness).

#### Ασκηση 3

Για την άσκηση 3 και για την υλοποίηση του κλαδεματος απλά προσθέσα έναν έλεγχο στις συναρτησεις maxvalue και minvalue καθώς και μια ενημερωση των ορίων των διαστημάτων [a, b] όπως ακριβώς ελεγε και η εκφωνηση της ασκησης αλλά και οι διαφανειες της θεωρίας.

#### Ασκηση 4

Για την άσκηση 4 και για την υλοποίηση ενός αλγοριθμου expectimax δημιουργηθηκε η expectiValue η οποία ουσιαστικά είναι ο κομβος των φαντασμάτων. Επομενως, ο pacman είναι ο παίκτης MAX και τα φαντάσματα είναι κόμβοι τύχης CHANCE. Η πιθανοτητα επιλογης καποιας κινήσης είναι ισοπιθανη με όλες τις άλλες οποτε  $probability = 1.0 / len(ghostActions)$  και τέλος το value που θα επιστραφει πρέπει ουσιαστικά να είναι μια μεση τιμη των πιθανοτητων για όλα τα δυνατα αποτελεσματα του κομβου τυχης.

πχ.  $Value = 1/3 * 20 + 1/3 * 10 + 1/3 * 7$  (εχοντας 3 κινήσεις για τον κομβο CHANCE)

#### Ασκηση 5

Για την άσκηση 5 και για την υλοποίηση μιας καλύτερης EvaluationFunction ουσιαστικά χρησιμοποιηθηκαν οι παραδοχες, τα σχολια και ο κωδικας την ασκησης 1 με την διαφορα ωστοσο οτι πλέον υπολογίζονται και ο αριθμος των φαγητων που υπολειπονται καθώς και η συναρτηση getScore() του καθε currentGameState. Αυτα χρησιμοποιηθηκαν γιατι πλέον η καινουργια μας συναρτηση αποτιμησης πρέπει να κανει evaluate gamestates και οχι actions όπως στην ασκηση 1. Επομενως η συναρτηση επιστρεφοντας την currentGameState.getScore() σε συνδυασμο με το πληθος των φαγητων που υπολειπονται αλλά και την απόσταση του κοντινότερου φαγητου παρακινεί τον pacman να δώσει εμφαση στην τροφη και να κινηθει προς το επόμενο φαγητο όσο ο ιδιος δεν διατρεχει καποιο κίνδυνο.

$1.0 / closestFood$  χρησιμοποιειται γιατι όσο πλησιαζει ο pacman τόσο πιο μεγалος γινεται αυτος ο αριθμος τείνοντας στην μοναδα οποτε και τον προσθέτω, ενώ η μεταβλητη capsulesNum είναι με προσημο αρνητικο γιατι όσο πιο πολλα φαγητα τρωει ο pacman τόσο πιο μεγалος θα γινεται ο αριθμος οποτε θα αφαιρείτε μικροτερη ποσοτητα απο την τιμη που επιστρεφει η evaluationFunction.