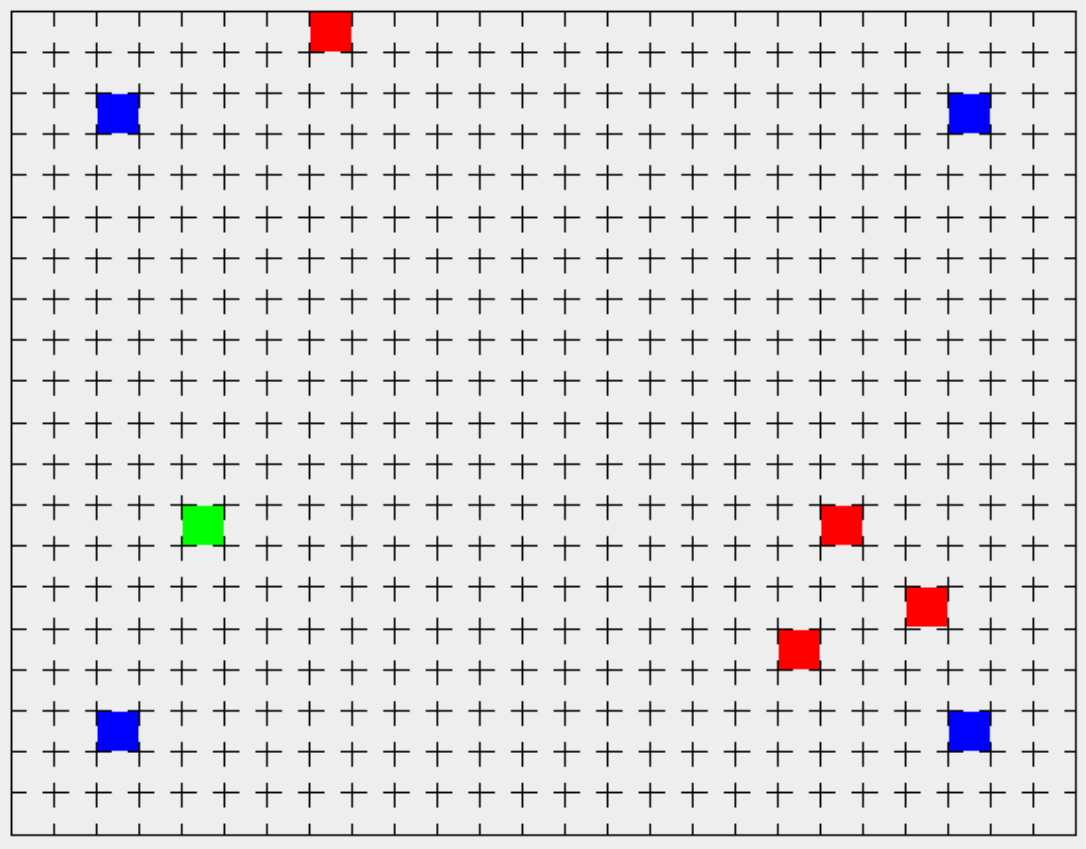
Εργαστήριο Επεξεργασίας Πληροφορίας και Υπολογισμών
Τομέας Ηλεκτρονικής και Υπολογιστών **3ο/5ο Εξάμηνο**
ΤΗΜΜΥ Α.Π.Θ **2017-2018**
Δευτέρα 22 / 11 / 2017

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

DS – Pac-Man – Part 2

Κίνηση Pacman (0,75 βαθμοί)

Στο δεύτερο παραδοτέο καλείστε να υλοποιήσετε την κίνηση του Pacman. Για να γίνει το παιχνίδι περισσότερο ενδιαφέρον, μέσα στην πλατφόρμα τοποθετήθηκαν 4 σημαίες. Σκοπός σας είναι να δημιουργήσετε μια δομή δεδομένων που θα συγκεντρώνει τις διαθέσιμες κινήσεις του Pacman στον επόμενο γύρο, καθώς και την υλοποίηση μιας συνάρτησης αξιολόγησης για την εύρεση της βέλτιστης επόμενης κίνησης με βάση διάφορα κριτήρια.



Εικόνα 1: Το νέο περιβάλλον του παιχνιδιού PacMan για το δεύτερο παραδοτέο.

Οι μονάδες μπορούν να κινούνται μέσα στο χώρο μόνο καθέτως και οριζοντίως (όχι διαγωνίως). Σκοπός του Pac-Man είναι να μαζέψει όλες τις σημαίες ενώ ταυτόχρονα αποφύγει τα φαντάσματα. Τα φαντάσματα από την άλλη προσπαθούν να πιάσουν τον Pac-Man. Θεωρούμε ότι ένα φάντασμα έχει πιάσει τον Pac-Man όταν οι δύο τους καταλήξουν στο ίδιο κελί ή όταν ανταλλάξουν θέσεις.

Ο νέος κώδικας της πλατφόρμας περιέχει μια υλοποίηση για τα φαντάσματα, η οποία θα είναι ίδια για όλους.

ΠΡΟΣΟΧΗ!!!

ΤΑ ΦΑΝΤΑΣΜΑΤΑ ΔΕΝ ΜΠΟΡΟΥΝ ΝΑ ΠΕΡΑΣΟΥΝ ΠΑΝΩ ΑΠΟ ΤΙΣ ΣΗΜΑΙΕΣ ΟΠΩΣ ΚΑΝΕΙ Ο PACMAN, ΠΡΕΠΕΙ ΝΑ ΚΑΝΟΥΝ ΤΟΝ ΚΥΚΛΟ.

ΜΗΝ ΑΝΤΙΚΑΤΑΣΤΗΣΕΤΕ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΩΝ ΦΑΝΤΑΣΜΑΤΩΝ ΠΟΥ ΕΧΟΥΜΕ ΕΝΣΩΜΑΤΩΣΕΙ ΕΜΕΙΣ ΜΕ ΑΥΤΗ ΠΟΥ ΥΛΟΠΟΙΗΣΑΤΕ ΣΤΟ ΠΡΟΗΓΟΥΜΕΝΟ ΠΑΡΑΔΟΤΕΟ. ΘΕΛΟΥΜΕ ΟΛΟΙ ΝΑ ΕΧΟΥΝ ΤΗΝ ΙΔΙΑ ΠΛΑΤΦΟΡΜΑ.

Χρήσιμες Μεταβλητές και Συναρτήσεις

Για την κλάση Room:

- `isGhost()`:Επιστρέφει true, αν στο κελί με συντεταγμένες [i, j] βρίσκεται κάποιο φάντασμα.
- `isFlag()`:Επιστρέφει true, αν στο κελί με συντεταγμένες [i, j] βρίσκεται κάποια σημαία.
- `isCapturedFlag()`:Επιστρέφει true, αν στο κελί με συντεταγμένες [i, j] βρίσκεται κάποια σημαία που έχει ήδη αποκτηθεί από τον Pacman.
- `walls[k]`:Ο μονοδιάστατος πίνακας walls με μέγεθος τέσσερα επιστρέφει την τιμή **μηδέν,** αν το κελί [i,j] **έχει** τοίχο στην κατεύθυνση που ορίζεται από το k. Αν ο τοίχος δεν υπάρχει, η τιμή είναι 1. Σκεφτείτε το ως μια μεταβλητή που σας λέει αν μπορείτε να κινηθείτε προς εκείνη την κατεύθυνση (τιμή 1) ή δεν μπορείτε να κινηθείτε προς εκείνη την κατεύθυνση (τιμή 0).

Άλλες σταθερές που θα σας φανούν χρήσιμες είναι οι εξής:

- `int PacmanUtilities.numberOfRows:` Επιστρέφει τον αριθμό των γραμμών της πίστας.
- `int PacmanUtilities.numberOfColumns:` Επιστρέφει τον αριθμό των στηλών της πίστας.
- `int PacmanUtilities.NumberOfGhosts:` Επιστρέφει τον αριθμό των φαντασμάτων.
- `int PacmanUtilities.stepLimit:` Επιστρέφει το όριο βημάτων του παιχνιδιού.

Για την υλοποίηση αυτής της εργασίας, θα χρειαστείτε οπωσδήποτε την κλάση ArrayList του πακέτου java.util. Η κλάση αυτή αναπαριστά έναν δυναμικό πίνακα, του οποίου το μέγεθος δεν είναι καθορισμένο από την αρχή, αλλά μπορείτε να προσθαφαιρείτε αντικείμενα κατά βούληση. Μπορείτε να δηλώσετε τι τύπου αντικείμενα θα περιέχονται. Για παράδειγμα ένα ArrayList με αντικείμενα τύπου String θα αρχικοποιηθεί ως εξής:

ArrayList<String> thisArrayList = new ArrayList <String>();

Οι μέθοδοι που προτείνεται να χρησιμοποιήσετε είναι οι εξής:

- `void add(Object o):` Προσθέτει το αντικείμενο που δίνεται ως όρισμα στην ArrayList, το οποίο θα πρέπει να είναι της αντίστοιχη κλάση που έχετε δηλώσει σαν έμμεσο όρισμα στην δήλωση της ArrayList (π.χ. String παραπάνω).

- `Object get(int pos)`: Επιστρέφει το αντικείμενο που βρίσκεται στην θέση που δώσαμε ως όρισμα. Η ArrayList θα πρέπει να έχει αρχικοποιηθεί για να παίρνει αντικείμενα ενός συγκεκριμένου τύπου, οπότε επιστρέφει τον συγκεκριμένο τύπο και όχι ένα απλό Object.
- `int size()`: Επιστρέφει το πλήθος των αντικειμένων που είναι αποθηκευμένα στην ArrayList.

Κλάση Αποθήκευσης Διαθέσιμων Κινήσεων

Για την υλοποίηση των διαθέσιμων κινήσεων προτείνεται η δημιουργία μιας δικής σας κλάσης με το όνομα NodeAEM1AEM2[1], η οποία θα χρησιμοποιηθεί αυτούσια και στην Τρίτη εργασία.

Η κλάση αυτή θα έχει ως μεταβλητές:

1. Την θέση του Pacman για την κίνηση αυτή. Αυτό υλοποιείται σαν δύο ξεχωριστές μεταβλητές (**nodeX, nodeY**).
2. **nodeMove**: Την κατεύθυνση που αντιπροσωπεύει αυτή η κίνηση με τιμές:
$$\text{West} \rightarrow 0$$
$$\text{South} \rightarrow 1$$
$$\text{East} \rightarrow 2$$
$$\text{North} \rightarrow 3$$
3. **nodeEvaluation**: Την τιμή της συνάρτησης αξιολόγησης που έχετε δημιουργήσει για αυτή τη κίνηση.
4. **currentGhostPos**: Η μεταβλητή αυτή θα αποθηκεύει την παρούσα θέση των φαντασμάτων, με παρόμοιο τρόπο με την currentPos της προηγούμενης εργασίας, δηλαδή ένας διδιάστατος πίνακας ακεραίων όπου
    a. Η πρώτη διάσταση θα είναι ο αριθμός του φαντάσματος (0 – 3).
    b. Η δεύτερη διάσταση θα περιλαμβάνει την τετμημένη (x, θέση 0) και την τεταγμένη (y, θέση 1) της τοποθεσίας του φαντάσματος.
5. **flagPos**: Οι σημαίες τοποθετούνται στην πλατφόρμα κατά την εκκίνηση και μένουν στην ίδια θέση καθ' όλη την διάρκεια του γύρου. Η μεταβλητή αυτή θα είναι ένας διδιάστατος πίνακας ακεραίων όπου
    a. Η πρώτη διάσταση θα είναι ο αριθμός της σημαίας (0 – 3).
    b. Η δεύτερη διάσταση θα περιλαμβάνει την τετμημένη (x, θέση 0) και την τεταγμένη (y, θέση 1) της τοποθεσίας της σημαίας.
6. **currentFlagStatus**: είναι ένας πίνακας boolean που περιλαμβάνει την τρέχουσα κατάσταση των σημαίων με δείκτη τον αριθμό της σημαίας και τιμή
    a. false, αν είναι ακόμα ενεργή και ο Pacman πρέπει να την αποκτήσει.
    b. true, αν ο Pacman την έχει αποκτήσει ήδη.

Οι συναρτήσεις της κλάσης που θα πρέπει να υλοποιηθούν είναι:

1. **Node():** Ένας ή περισσότεροι constructors για την κλάση σας, με διαφορετικά ορίσματα.

[1] Όπου AEM1 και AEM2 ο αριθμός μητρώου των δύο ατόμων κάθε ομάδας.

2. **int[][] findGhosts():** Η συνάρτηση αυτή θα πρέπει να βρίσκει την θέση των φαντασμάτων στην παρούσα κατάσταση του Maze.
3. **int[][] findFlags():** Η συνάρτηση αυτή θα πρέπει να βρίσκει την θέση των σημαιών στην πλατφόρμα.
4. **Boolean[] checkFlags():** Η συνάρτηση αυτή θα πρέπει να επιστρέφει την κατάσταση της κάθε σημαίας που βρίσκεται στην πλατφόρμα.
5. **double evaluate():** η συνάρτηση αυτή θα υπολογίζει το πόσο αξιόλογη είναι η συγκεκριμένη κίνηση.

Μπορείτε να φτιάξετε και άλλες μεταβλητές ή συναρτήσεις για την κλάση σας, αν το θεωρείτε απαραίτητο.

Στο package gr.auth.ee.dsproject.node βρίσκεται μια πρότυπη κλάση με το όνομα Node. Θα πρέπει να την μετονομάσετε (δεξί κλικ → Refactor → Rename) σύμφωνα με την εκφώνηση και να υλοποιήσετε τις κατάλληλες συναρτήσεις.

Εύρεση και Αξιολόγηση Διαθέσιμων Κινήσεων

Η εύρεση των διαθέσιμων κινήσεων του Pacman είναι ευκολότερη από αυτή των φαντασμάτων καθώς χρειάζεται να ελέγξετε μόνο δύο πράγματα για μη επιτρεπτές κινήσεις

1. Αν ο Pacman πέφτει πάνω σε τοίχο (το όποιο είναι απαγορευμένη κίνηση).
2. Αν ο Pacman κινείται στην θέση που είναι ή που μπορεί να καταλήξει ένα φάντασμα, κάτι το οποίο οδηγεί μεν στο τέλος του παιχνιδιού, μπορεί όμως να είναι μονόδρομος σε κάποιες περιπτώσεις (πχ περικύκλωση από φαντάσματα).

Το **ζητούμενο** είναι να δημιουργήσετε ένα ArrayList<NodeAEM1AEM2> μέσα στην συνάρτηση calculateNextPacmanPosition της κλάσης Creature το οποίο να αποθηκεύει πριν από κάθε γύρο παιχνιδιού που είναι η σειρά του Pacman τις διαθέσιμες κινήσεις του, και να επιστρέφετε την κίνηση που έχει την καλύτερη απόδοση, σύμφωνα με την αξιολόγηση των κινήσεων που είναι διαθέσιμες.

Η αξιολόγηση των κινήσεων μπορεί να γίνει με διάφορα κριτήρια. Κάποιες ιδέες είναι:

1) Η απόσταση του Pacman από μια σημαία που πρέπει να καταλάβει.
2) Η απόσταση του Pacman από τα φαντάσματα που τον κυνηγούν.
3) Απόσταση φαντασμάτων από την σημαία που θέλει να καταλάβει ο Pacman
4) Αν είναι προς το κέντρο ή τα άκρα του πίνακα.
5) Αν έχει τοίχους γύρω του.

Η συνάρτηση που θα φτιάξετε θα πρέπει να επιστρέφει μια τιμή double μέσα σε κάποια λογικά όρια (πχ -100 έως 100), η οποία θα αντιστοιχεί στην αξιολόγηση της τρέχουσας κατάστασης του παιχνιδιού. Οι αρνητικοί αριθμοί αντιστοιχούν σε καταστάσεις που ο αντίπαλος είναι σε καλύτερη θέση, ενώ οι θετικοί αντιστοιχούν σε καταστάσεις, όπου εσείς βρισκόσαστε σε καλύτερη θέση. Όσο καλύτερη κρίνεται η κατάσταση του ταμπλό, τόσο μεγαλύτερος πρέπει να είναι ο αριθμός που επιστρέφει η συνάρτηση αξιολόγησης.

Η υλοποίηση αυτής της μεθόδου είναι ελεύθερη, αλλά μπορείτε να βασιστείτε στις γενικές κατευθύνσεις που σας δίνονται παραπάνω. Όμως, ακριβώς επειδή η υλοποίηση είναι ελεύθερη, θα πρέπει να δώσετε ιδιαίτερη βαρύτητα στον σχολιασμό του κώδικα, όσο και στην λεπτομερή ανάλυση του σκεπτικού σας στην γραπτή αναφορά.

Οδηγίες

Τα προγράμματα θα πρέπει να υλοποιηθούν σε Java, με πλήρη τεκμηρίωση του κώδικα. Το πρόγραμμά σας πρέπει να περιέχει επικεφαλίδα σε μορφή σχολίων με τα στοιχεία σας (ονοματεπώνυμα, ΑΕΜ, τηλέφωνα και ηλεκτρονικές διευθύνσεις). Επίσης, πριν από κάθε κλάση ή μέθοδο θα υπάρχει επικεφαλίδα σε μορφή σχολίων με σύντομη περιγραφή της λειτουργικότητας του κώδικα. Στην περίπτωση των μεθόδων, πρέπει να περιγράφονται και οι μεταβλητές τους.

Είναι δική σας ευθύνη η απόδειξη καλής λειτουργίας του προγράμματος.

Παραδοτέα για κάθε μέρος της εργασίας

1. Ηλεκτρονική αναφορά που θα περιέχει: εξώφυλλο, περιγραφή του προβλήματος, του αλγορίθμου και των διαδικασιών που υλοποιήσατε και τυχόν ανάλυσή τους. Σε καμία περίπτωση να μην αντιγράφεται ολόκληρος ο κώδικας μέσα στην αναφορά (εννοείται ότι εξαιρούνται τμήματα κώδικα τα οποία έχουν ως στόχο τη διευκρίνιση του αλγορίθμου)
Προσοχή: Ορθογραφικά και συντακτικά λάθη πληρώνονται.

2. Ένα αρχείο σε μορφή .zip με όνομα "AEM1_AEM2_PartB.zip", το οποίο θα περιέχει **όλο το project σας** στον eclipse καθώς και το αρχείο της γραπτής αναφοράς σε pdf (**αυστηρά**). Το αρχείο .zip θα γίνεται upload στο site του μαθήματος. Τα ονόματα των αρχείων να είναι με **λατινικούς χαρακτήρες**.

Προθεσμία υποβολής

Κώδικας και αναφορά Παρασκευή 15 Δεκεμβρίου, 23:59 (ηλεκτρονικά)
Δε θα υπάρξει καμία παρέκκλιση από την παραπάνω προθεσμία.