

DS-Pacman.PartA

Βαρβαρίγγος Ιωάννης AEM: 8782
Βελέντζας Ιάσων-Γεώργιος AEM: 8785

5ο Εξάμηνο



Περιεχόμενα

1	Περιγραφή Παιχνιδιού	3
2	Περιγραφή Προβλήματος	3
3	Σχεδίαση Αλγορίθμου	4
4	Ανάλυση Αλγορίθμου	4
4.1	Αρχικοποίηση tempdir	5
4.2	Αρχικοποίηση newdir	5
4.3	Δημιουργία Βρόχου Αρχικοποίησης	5
4.4	Υλοποίηση Ελέγχων	6
4.4.1	Σύγκρουση με Τοίχο	6
4.4.2	Σύγκρουση με Άλλο Φάντασμα	6
4.5	Ανάθεση τιμής στην newdir	7
4.6	Δημιουργία Βρόχου Ελέγχων	8
4.7	Επιστροφή Πίνακα Τιμών	8

1 Περιγραφή Παιχνιδιού

Το παιχνίδι του Pacman είναι λίγο έως πολύ γνωστό σε όλους. Σ' αυτήν την εργασία ασχολούμαστε με μια απλουστευμένη εκδοχή του, στην οποία ο Pacman προσπαθεί απλώς να αποφύγει τα φαντάσματα, ενώ τα φαντάσματα προσπαθούν με τη σειρά τους να πιάσουν τον Pacman. Ο "ήρωάς" μας νικάει μόνο αν ολοκληρώσει έναν προκαθορισμένο αριθμό κινήσεων πριν τον πιάσει κάποιο φάντασμα. Διαφορετικά, νικητές θεωρούνται τα φαντάσματα.

Όσον αφορά την προσομοίωση του παιχνιδιού, εκτυλίσσεται σε ένα δωμάτιο - λαβύρινθο που περιβάλλεται περιμετρικά από τοίχους και είναι χωρισμένο σε τετραγωνικά κελιά. Ο Pacman αναπαρίστανται ως ένα πράσινο τετραγωνάκι, ενώ τα φαντάσματα ως κόκκινα.

2 Περιγραφή Προβλήματος

Ζητούμενο της παρούσας εργασίας είναι η σχεδίαση ενός αλγορίθμου, ο οποίος να καθορίζει την κίνηση των φαντασμάτων, καθώς και η υλοποίηση αυτού του αλγορίθμου σε Java. Για το σκοπό αυτό θα τροποποιήσουμε τη συνάρτηση:

```
int [ ] calculateNextGhostPosition( Room[ ][ ] Maze, int[ ][ ] currentPos)
```

Οι κανόνες που διέπουν την κίνηση των φαντασμάτων είναι οι εξής:

- Είναι τυχαία
- Δεν οδηγεί το φάντασμα σε κελί που έχει τοίχο (σύγκρουση με τοίχο)
- Δεν οδηγεί το φάντασμα σε κελί που έχει άλλο φάντασμα ή σε κελί προς το οποίο κινείται άλλο φάντασμα (σύγκρουση με φάντασμα)

Αν η κίνηση του φαντάσματος οδηγεί σε σύγκρουση, τότε ακυρώνεται η κίνηση αυτή και το φάντασμα παραμένει στη θέση του. Ο καθορισμός της κίνησης γίνεται μέσω της απόδοσης νέας κατεύθυνσης σε κάθε φάντασμα.

Οι τιμές κατεύθυνσης είναι:

West	=	0
South	=	1
East	=	2
North	=	3

3 Σχεδίαση Αλγορίθμου

Για την επίλυση του προβλήματος σχεδιάστηκε ο παρακάτω αλγόριθμος: ¹

1. Για κάθε φάντασμα θέσε μια τυχαία *tempdir*.
2. Θέσε μια μη έγκυρη *newdir*.
3. Επανάλαβε τα βήματα 1-2 για κάθε φάντασμα.
4. Όσο η *newdir* ενός φαντάσματος είναι μη έγκυρη, έλεγχε αν η *tempdir* οδηγεί σε έγκυρη κίνηση.
5. Αν η κίνηση που ορίζει η *tempdir* είναι έγκυρη, θέσε ως *newdir* την *tempdir*. Αλλιώς, θέσε μια άλλη *tempdir*.
6. Επανάλαβε τα βήματα 4-5 για κάθε φάντασμα.
7. Επίστρεψε τις τιμές *newdir* όλων των φαντασμάτων.

4 Ανάλυση Αλγορίθμου

Για την υλοποίηση του αλγορίθμου αποθηκεύουμε τις τιμές των *tempdir* και *newdir* σε δύο μονοδιάστατους πίνακες, των οποίων το μέγεθος είναι ίσο με το πλήθος των φαντασμάτων.

Το πλήθος των φαντασμάτων με τη σειρά του καθορίζεται από τους κανόνες του παιχνιδιού και ανακτάται μέσω της σταθεράς: ²

int PacmanUtilities.numberOfGhosts

¹ **Σημείωση:** Για να γίνει πιο κατανοητή η περιγραφή του αλγορίθμου θα ονομάσουμε την κατεύθυνση που δίνεται τυχαία στο φάντασμα (πριν ελεγχθεί αν οδηγεί σε έγκυρη κίνηση) *tempdir* και την τελική κατεύθυνση που θα δοθεί για κάθε φάντασμα *newdir*.

²Στα πλαίσια του παιχνιδιού έχει αποδοθεί η τιμή 4 μέσω της εντολής:

public static final int numberOfGhosts = 4 ;

4.1 Αρχικοποίηση tempdir

Θέλουμε να αρχικοποιήσουμε τυχαία τη μεταβλητή tempdir με έναν ακέραιο αριθμό στο διάστημα $[0, 3]$. Γι' αυτό χρησιμοποιούμε τη συνάρτηση Math.random() που δεν δέχεται ορίσματα και επιστρέφει έναν αριθμό τύπου double στο διάστημα $[0, 1)$. Επομένως, αν πολλαπλασιάσουμε με 4 τον αριθμό που επιστρέφει η Math.random(), παίρνουμε έναν double στο διάστημα $[0, 4)$, τον οποίο μπορούμε να μετατρέψουμε σε int εφαρμόζοντας type-casting. Εν τέλει καταλήγουμε σε μια έγκυρη τιμή κατεύθυνσης, δηλαδή έναν ακέραιο στο διάστημα $[0, 3]$.

Η γραμμή κώδικα που υλοποιεί την εκχώρηση τιμής που μόλις περιγράψαμε είναι:

```
tempGhostDirection[i] = (int) (Math.random() * 4);
```

4.2 Αρχικοποίηση newdir

Δίνεται στη μεταβλητή newGhostDirection[i] μια μη έγκυρη τιμή μέσω μιας απλής εκχώρησης τιμής:

```
newGhostDirection[i] = -1;
```

Θυμίζουμε ότι έγκυρες τιμές θεωρούνται οι: 0,1,2,3. Ο λόγος της αρχικοποίησης αυτής εξηγείται στο 4ο βήμα.

4.3 Δημιουργία Βρόχου Αρχικοποίησης

Τα βήματα 1 και 2 επαναλαμβάνονται για κάθε φάντασμα, όπως ορίζει το βήμα 3. Επομένως υλοποιούνται εντός ενός βρόχου επανάληψης με:

- αρχική συνθήκη: μετρητής = 0
- συνθήκη ελέγχου: μετρητής < PacmanUtilities.numberOfGhosts
- βήμα αύξησης : +1

Code Implementation:

```
for (int i = 0; i < PacmanUtilities.numberOfGhosts; i++)
```

Ο μετρητής συμβολίζεται με τη μεταβλητή int i.

4.4 Υλοποίηση Ελέγχων

Δημιουργούμε ένα βρόχο επανάληψης ώστε να διεξάγουμε τους ελέγχους για σύγκρουση. Συνθήκη ελέγχου του βρόχου είναι η εγκυρότητα της `newdir`. Σε περίπτωση που η `newdir` έχει έγκυρη τιμή δεν θα εκτελεστεί ο βρόχος. Γι' αυτό αρχικοποιήσαμε τη μεταβλητή `newdir` με μη έγκυρη τιμή, ώστε να εκτελεστούν τουλάχιστον μια φορά οι έλεγχοι σύγκρουσης.

4.4.1 Σύγκρουση με Τοίχο

Για τον έλεγχο σύγκρουσης του *i*-οστού φαντάσματος με τοίχο χρησιμοποιούμε τον πίνακα

$$\text{int Maze}[x][y]. \text{ walls}[k]$$

Ο ακέραιος που επιστρέφεται είναι 0 αν υπάρχει τοίχος και 1 αν δεν υπάρχει.

Πρέπει να εκφράσουμε τις παραμέτρους με βάση ήδη γνωστά δεδομένα και δη τον δισδιάστατο πίνακα `currentPos`, ο οποίος σε κάθε γραμμή έχει το *i* -οστό φάντασμα και στις στήλες έχει τετμημένη και τεταγμένη αντίστοιχα άρα:

- *x*: αριθμός γραμμής στην οποία βρίσκεται το φάντασμα (τετμημένη)

$$x[i] = \text{currentPos}[i][0];$$

- *y*: αριθμός στήλης στην οποία βρίσκεται το φάντασμα (τεταγμένη)

$$y[i] = \text{currentPos}[i][1];$$

- *k*: κατεύθυνση προς την οποία πρόκειται να κινηθεί το φάντασμα

$$k[i] = \text{tempGhostDirection}[i];$$

Επομένως για να μην υπάρχει τοίχος στην κατεύθυνση *k* από το κελί που βρίσκεται το φάντασμά μας πρέπει:

Code Implementation:

$$\text{Maze} [\text{currentPos}[i][0]] [\text{currentPos}[i][1]].\text{walls}[\text{tempGhostDirection}[i]] == 1$$

4.4.2 Σύγκρουση με Άλλο Φάντασμα

Για τον έλεγχο αυτό χρησιμοποιούμε τη συνάρτηση:

$$\text{boolean}[] \text{ checkCollision}(\text{int}[] \text{ moves}, \text{int}[] [] \text{ currentPos})$$

Η συνάρτηση αυτή δέχεται ως ορίσματα την *tempdir* του φαντάσματος και τη θέση του στο λαβύρινθο και επιστρέφει *true* αν η κατεύθυνση του φαντάσματος οδηγεί σε σύγκρουση ή *false* αν η κατεύθυνση οδηγεί σε έγκυρη κίνηση. Επομένως, χρειαζόμαστε έναν πίνακα ο οποίος θα αποθηκεύει τις τιμές που επιστρέφει η *checkCollision*. Αυτός είναι ο:

boolean[] ghostCollisionStatus

που υλοποιείται:

Code Implementation:

ghostCollisionStatus[i] = checkCollision(tempGhostDirection , currentPos)

Άρα η συνθήκη μας για να μην υπάρχει σύγκρουση με άλλο φάντασμα είναι:

Code Implementation:

! ghostCollisionStatus[i]

Η τελική συνθήκη ελέγχου δίνεται από το λογικό ΚΑΙ των 2 συνθηκών

Code Implementation:

if (Maze[currentPos[i][0]] [currentPos[i][1]].walls[tempGhostDirection[i]] == 1
&&
!ghostCollisionStatus[i])

4.5 Ανάθεση τιμής στην *newdir*

Με βάση την παραπάνω συνθήκη ελέγχου δημιουργούμε μια δομή *if-else* , έτσι ώστε όταν η συνθήκη είναι αληθής να εκχωρείται στην *newdir* η τιμή της *tempdir*, ενώ όταν είναι ψευδής, η *newdir* να παραμένει μη έγκυρη (επανεκτελείται λοιπόν, ο βρόχος ελέγχου *while*) και η *tempdir* παίρνει μια καινούρια τυχαία τιμή, όπως στο βήμα 1:

Code Implementation:

Εκχώρηση τιμής στην `newdir`:

```
newGhostDirection[i] = tempGhostDirection[i];
```

4.6 Δημιουργία Βρόχου Ελέγχων

Όμοια με το 3ο βήμα:

Τα βήματα 4 και 5 επαναλαμβάνονται για κάθε φάντασμα. Επομένως υλοποιούνται εντός ενός βρόχου επανάληψης με:

- αρχική συνθήκη: `μετρητής = 0`
- συνθήκη ελέγχου: `μετρητής < PacmanUtilities.numberOfGhosts`
- βήμα αύξησης : `+1`

ο οποίος θα περιέχει το βρόχο `while` και τη δομή `if-else`, εμφωλευμένη στο βρόχο `while`.

Code Implementation:

```
for (int i = 0; i < PacmanUtilities.numberOfGhosts; i++)
```

Ο `μετρητής` συμβολίζεται με τη μεταβλητή `int i`.

4.7 Επιστροφή Πίνακα Τιμών

Τέλος η συνάρτηση πρέπει να επιστρέφει έναν πίνακα ακεραίων που να καθορίζει την κίνηση των φαντασμάτων. Αυτός είναι ο `int[] newGhostDirection`:

Code Implementation:

```
return newGhostDirection;
```