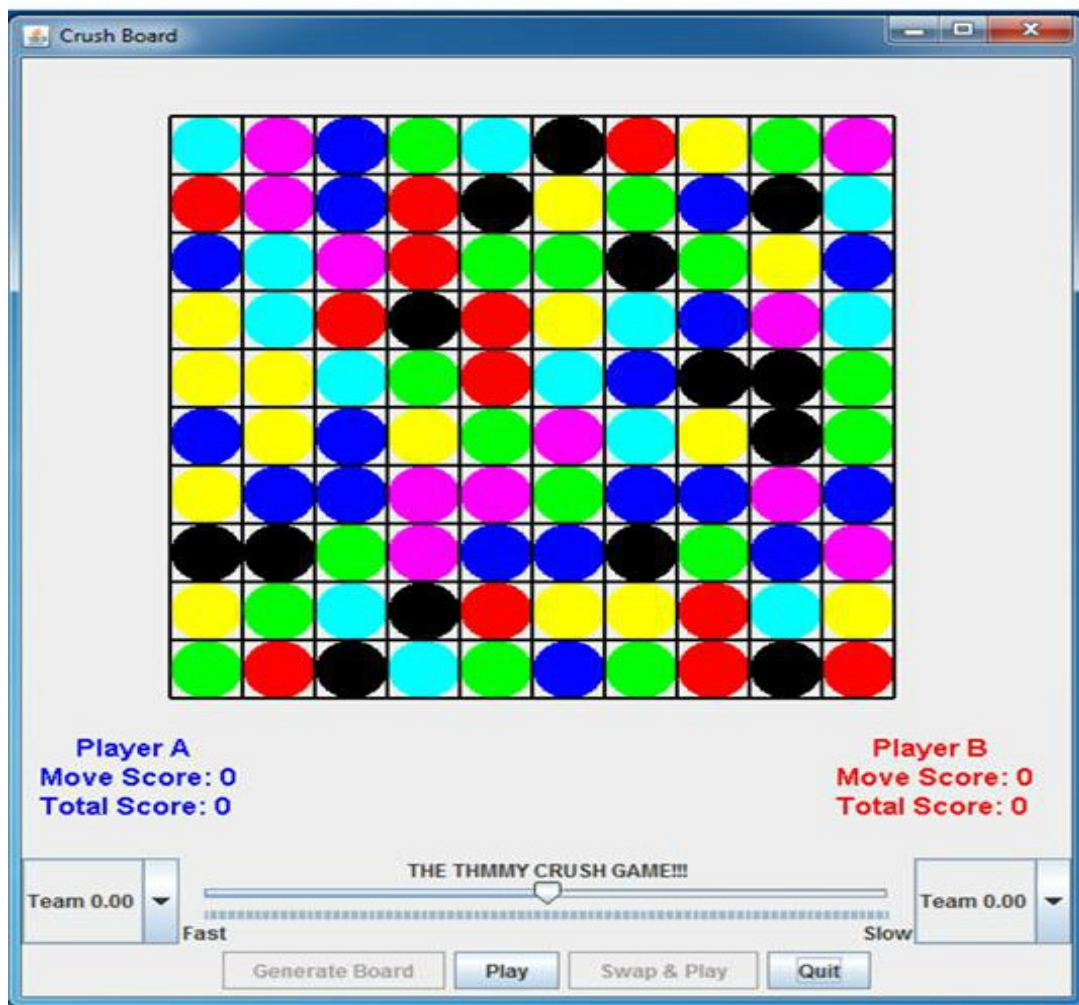


Εργασία Candy Crash

Part A

24/11/2016

- Μπεκιάρης Θεοφάνης ΑΕΜ:8200
- Ευαγγελόπουλος Δημήτρης-Νεκτάριος ΑΕΜ:8561



Περιγραφή προβλήματος

Η εργασία αφορά την ολοκλήρωση της κατασκευής ενός παιχνιδιού, το Candy Crash .Ξεκινώντας, στο πρώτο μέρος της εργασίας , καλούμαστε να συμπληρώσουμε-τροποποιήσουμε δύο κλάσεις: την κλάση Tile, η οποία είναι υπεύθυνη για τα πλακίδια του ταμπλό του παιχνιδιού, και την κλάση RandomPlayer, η οποία είναι υπεύθυνη για την δημιουργία και την ομαλή λειτουργία των παικτών της πλατφόρμας.

Αρχικά, θα πρέπει να κατασκευάσω τους Constructors , με τους οποίους θα αρχικοποιήσω τις μεταβλητές που έχουν μέσα τους οι δύο αυτές κλάσεις.

Έπειτα, καθίσταται αναγκαίο, να δημιουργήσω τους Setters και Getters των μεταβλητών των δύο κλάσεων (Tile και RandomPlayer), οι οποίοι ορίζουν τιμές και τις επιστρέφουν αντίστοιχα.

Επιπλέον για την κλάση RandomPlayer καλούμαστε να υλοποιήσουμε μια συνάρτηση η οποία:

- θα επιλέγει την επόμενη κίνηση του παίχτη με τυχαίο τρόπο και
- θα υπολογίζει και θα επιστρέφει τις θέσεις (συντεταγμένες) των ζαχαρωτών που θα ανταλλάξουν θέση .

Μετά την ολοκλήρωση της πρώτης αυτής εργασίας, θα πρέπει το παιχνίδι να ξεκινάει και οι δύο παίκτες να επιλέγουν τυχαία δύο ζαχαρωτά και να τα αλλάζουν θέση κάνοντας επιτυχημένους συνδυασμούς, μέχρι κάποιος από τους δύο να κερδίσει.

Περιγραφή Αλγορίθμου

- Η κατασκευή των Constructor είναι απλή και την κάνουμε σε όλες τις κλάσεις. Αυτή η διαδικασία δεν μας πήρε καθόλου χρόνο. Οι Constructor (μέθοδος αρχικών συνθηκών) αρχικοποιούν τις μεταβλητές που έχει η Tile. Οι Setters (setId, setX, setY, setColor, setMark) όπως προαναφέραμε ορίζουν τιμές στις μεταβλητές της κλάσης και οι Getters (getId, getX, getY, getColor, getMark) επιστρέφουν τιμές.
- Αντίστοιχα δουλεύουμε και στην RandomPlayer για τη δημιουργία Constructor και συναρτήσεων get και set.

- Στην κλάση RandomPlayer, για την υλοποίηση της συνάρτησης
`public int[] getNextMove(ArrayList<int[]> availableMoves, Board board)`
 δουλεύουμε ως εξής:

Χρειάζεται να επιλέξω τυχαία μια κίνηση από τη λίστα των πιθανών κινήσεων. Για να το κάνω αυτό χρησιμοποιώ την `Math.random` η οποία επιλέγει τυχαία μια τιμή στο διάστημα $[0,1)$. Πολλαπλασιαζόμενη με το 10 δίνει ένα δεκαδικό αριθμό τύπου `float` στο διάστημα $[0,10)$, τον οποίο για να τον μετατρέψουμε σε ακέραιο χρησιμοποιούμε το `próthema(int)`. Αντί για τον αριθμό 10 χρησιμοποιήσαμε το `availableMoves.size()`. Με αυτόν τον τρόπο καταφέραμε να δώσουμε στην μεταβλητή `index` μια τυχαία ακέραια τιμή στο διάστημα $[0, \text{μήκος λίστας}]$.

```
!! index=(int)(Math.random()*availableMoves.size());
```

Την παραπάνω τιμή την χρησιμοποιούμε στην συνάρτηση
`int[] getRandomMove (ArrayList moves , int index)` η οποία θα μας επιστρέψει έναν πίνακα 1×3 της μορφής [στήλη πλακιδίου (x), γραμμή πλακιδίου (y), κατεύθυνση κίνησης (UP,DOWN,LEFT,RIGHT)], της οποίας τα αποτελέσματα τα αποθηκεύω στην `int[] arrayMoves`.

Έπειτα, Αφού πάρουμε τις τιμές x y για την θέση του στοιχείου στο ταμπλό ελέγχουμε την κατεύθυνση κατά την οποία θα κινηθεί και δημιουργούμε τις μεταβλητές x2 y2 που περιέχουν την νέα θέση. Η διαδικασία αυτή γίνεται εύκολα με τη χρήση της `switch` παίρνοντας και τις 4 πιθανές περιπτώσεις για τις πιθανές κινήσεις του πλακιδίου.

Τέλος, για την επιστροφή των στοιχείων δημιουργούμε πίνακα 1×3 που επιστρέφει τις παλιές και τις νέες συντεταγμένες του πλακιδίου.

```
!! int[] returnMatrix=new int[4]
```

Δεδομένου ότι η θέση του στοιχείου που παίρνουμε από την λίστα και η κατεύθυνση κίνησης είναι καλά ορισμένες δεν χρειάζεται έλεγχος για τα επιστρεφόμενα δεδομένα(θα είναι εντός ορίων).