

# **ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ**

## **Candy Crush**



**Εργασία Α:**

## **Random Movement**

**Ράπτης Δημήτριος**

**A.E.M.: 8467**

**e-mail: dmraptis@auth.gr**

**Παπαγεωργίου Θωμάς**

**A.E.M.: 8577**

**e-mail: tompap@live.com**

Σε αυτό το πρώτο μέρος της εργασίας καλούμαστε να συμπληρώσουμε τους Constructors, τους Getters και τους Setters των κλάσεων Tile και RandomPlayer καθώς και την συνάρτηση getNextMove της RandomPlayer.

## Tile:

- Οι μεταβλητές της κλάσης:

```
int id;  
int x;  
int y;  
int color;  
boolean mark;
```

**int id:** είναι μια μεταβλητή που δείχνει το μοναδικό κωδικό ενός πλακιδίου.  
**int x:** η μεταβλητή αυτή δείχνει την θέση του πλακιδίου στον άξονα x'x.  
**int y:** η μεταβλητή αυτή δείχνει την θέση του πλακιδίου στον άξονα y'y.  
**int color:** η μεταβλητή αυτή δείχνει το χρώμα του ζαχαρωτού που περιέχει το πλακίδιο.  
**boolean mark:** η μεταβλητή αυτή συμβολίζει αν το πλακίδιο έχει επιλεγεί ή όχι για να διαγραφεί από το ταμπλό του παιχνιδιού στην επόμενη κίνηση λόγω συμμετοχής του σε τριάδα, τετράδα ή πεντάδα ζαχαρωτών ίδιου χρώματος.

- Ο **Constructor** της κλάσης:

```
public Tile (int id, int x, int y, int color, boolean mark){  
    this.id = id;  
    this.x = x;  
    this.y = y;  
    this.color = color;  
    this.mark = mark;  
}
```

Η μεταβλητή **this.id** συμβολίζει την **id** της κλάσης, ενώ η «σκέτη» **id** την παράμετρο του Constructor. Ομοίως για τις υπόλοιπες μεταβλητές και παραμέτρους.

- **Getters - Setters**

Οι Setters δίνουν τιμή στην μεταβλητή της κλάσης που αναφέρεται στην ονομασία τους. Η τιμή αυτή είναι η παράμετρος που δέχονται.

Οι Getters δεν έχουν ορίσματα αλλά επιστρέφουν την αντίστοιχη μεταβλητή της κλάσης στην οποία αναφέρεται το όνομά τους.

## RandomPlayer:

- Οι μεταβλητές της κλάσης:

```
int id;  
String name;  
int score;
```

**int id:** είναι μια μεταβλητή που δείχνει αν ο παίκτης είναι ο Κόκκινος ή ο Μπλε.

**int name:** η μεταβλητή name δίνει το όνομα που επιθυμούμε στον παίκτη.

**int score:** η μεταβλητή αυτή αποθηκεύει τον αριθμό των ζαχαρωτών που έχει συγκεντρώσει ο παίκτης.

- Ο **Constructor** της κλάσης:

```
public RandomPlayer (Integer pid){  
    this.id = pid.intValue();  
}
```

Η μεταβλητή **this.id** συμβολίζει την **id** της κλάσης, ενώ η **pid** την παράμετρο του Constructor.

Η **pid** είναι της κλάσης **Integer** οπότε για να πάρουμε την ακέραια τιμή πρέπει να καλέσουμε την μέθοδο **intValue()**.

- **Getters - Setters**

Ομοίως με την κλάση **Tile**

- Συνάρτηση **getNextMove**

Δέχεται ως ορίσματα ένα ArrayList τύπου `int[]`, που περιέχει όλες τις επιτρεπτές κινήσεις του παίχτη, και ένα αντικείμενο τύπου `Board`.

Επιστρέφει τις θέσεις (συντεταγμένες) των ζαχαρωτών που θα ανταλλάξουν θέση `[x1, y1, x2, y2]`, όπου `(x1,y1)` είναι η παλιά θέση του ζαχαρωτού που επιλέξατε να μετακινήσετε και `(x2,y2)` η νέα θέση του με ένα πίνακα ακεραίων μεγέθους 4.

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

Επιλέγει τυχαία μια κίνηση από τις διαθέσιμες του χρήστη.

➤ `availableMoves.size()` :

επιστρέφει το μέγεθος του array

➤ `Math.random() * ( availableMoves.size() )` :

τυχαία επιλογή ενός αριθμού μεταξύ 0 και `availableMoves.size()` :

```
nextMove = CrushUtilities.getRandomMove (availableMoves, randomIndex);
```

Επιστρέφει την τυχαία κίνηση που υπολογίσαμε προηγουμένως. Το αρχικό `x` και `y` και την κατεύθυνση που θα ακολουθήσει.

```
int [] moves = new int[4];
```

Ο πίνακας περιέχει κατά σειρά τις εξής θέσεις:

`x1, y1, x2, y2`

Οπότε οι αρχικές συντεταγμένες θα είναι το επιστρεφόμενο `x, y` της συνάρτησης `getRandomMove`.

Για τις τελικές συντεταγμένες εργαζόμαστε ως εξής αναλόγως την μεταβλητή κατεύθυνσης:

➤ UP:

Η συντεταγμένη  $y$  θα αυξηθεί κατά ένα, ενώ η  $x$  θα παραμείνει ίδια.

➤ DOWN:

Η συντεταγμένη  $y$  θα μειωθεί κατά ένα, ενώ η  $x$  θα παραμείνει ίδια.

➤ RIGHT:

Η συντεταγμένη  $y$  θα παραμείνει ίδια, ενώ η συντεταγμένη  $x$  θα αυξηθεί κατά ένα .

➤ LEFT:

Η συντεταγμένη  $y$  θα παραμείνει ίδια, ενώ η συντεταγμένη  $x$  θα μειωθεί κατά ένα .