

Εργασία στο μάθημα Προχωρημένα Θέματα Προγραμματισμού Η/Υ (έκδοση 1.0)

Δημιουργία ενός παίκτη του παιχνιδιού Proximity.

Ημερομηνία παράδοσης της εργασίας: **29 – 05 – 2022**
(50% τελικού βαθμού)

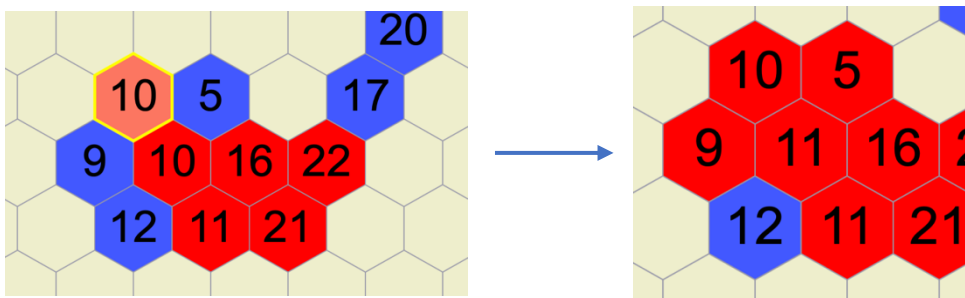
Τι είναι το Proximity?

Παιχνίδι στρατηγικής (<https://www.mathsisfun.com/games/proximity.html>). Υπάρχει ένα ταμπλό από εξάγωνα και δύο παίκτες, ο κόκκινος και ο μπλε. Σε κάθε γύρο ο παίκτης λαμβάνει έναν τυχαίο αριθμό, π.χ. x , από το 1 έως το 20 και τον τοποθετεί σε ένα ελεύθερο κελί του ταμπλό. Εάν το κελί αυτό είναι γείτονας με κελιά:

- (i) του ίδιου χρώματος, τους αυξάνει την τιμή κατά 1.
- (ii) του άλλου χρώματος, τα καταλαμβάνει (δηλαδή τα αλλάζει χρώμα) εάν η τιμή τους είναι μικρότερη από x .

Ένα παράδειγμα φαίνεται στην Εικόνα 1.

Το παιχνίδι τελειώνει όταν δε μείνει κανένα ελεύθερο κελί στο ταμπλό. Κερδίζει ο παίκτης που έχει τους περισσότερους πόντους.

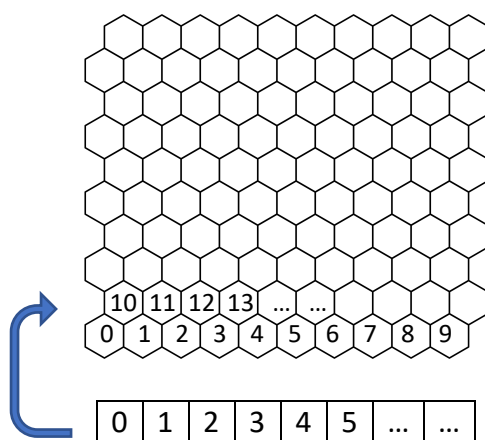


Εικόνα 1. Ο κόκκινος τοποθετεί το 10. Τα μπλε κελιά με τιμές 9 και 5 καταλαμβάνονται ενώ το κελί κόκκινο 10 αυξάνεται κατά 1 και γίνεται 11.

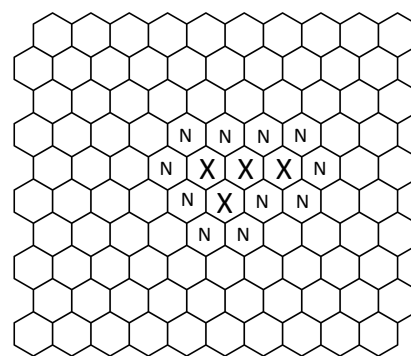
Ζητούμενα της εργασίας

- 1) Δημιουργείστε μια κλάση με όνομα *Cell* η οποία θα περιλαμβάνει τα παρακάτω:
 - a. Instance variable “*value*” # Ακέραιος που δείχνει την τρέχουσα τιμή του κελιού
 - b. Instance variable “*owner*” # Ακέραιος που δείχνει τον ιδιοκτήτη του κελιού. Μπορεί να πάρει τις τιμές 0 (δεν ανήκει σε κανένα παίκτη), 1 (κόκκινο), 2 (πράσινο)
 - c. Methods *getValue()*, *setValue(int)*, *getOwner()*, και *setOwner(int)*. Οι μέθοδοι *set* δίνουν τιμή στα αντίστοιχα instance variables ενώ οι μέθοδοι *get* επιστρέφουν την τιμή των αντίστοιχων instance variables.
- 2) Δημιουργείστε μια κλάση με όνομα *ProximityX* όπου το *X* θα αντικατασταθεί με τον αριθμό της ομάδας σας (π.χ., *Proximity5*, *Proximity12* κλπ.). Η κλάση αυτή θα περιλαμβάνει:
 - a. Instance variable “*pid*”. Ακέραιος που δείχνει τον αριθμό του παίκτη (1 για κόκκινο, 2 για πράσινο)
 - b. Instance variables “*length_X*” and “*length_Y*”. Ακέραιοι που δείχνουν το μέγεθος του ταμπλό.

- c. Methods *setPid(int)* και *setBoardSize(int, int)*. Οι μέθοδοι θέτουν τιμή στα instance variables *pid* και *length_X* και *length_Y*, αντίστοιχα.
- d. Method *getPlayerName()*. Η μέθοδος αυτή επιστρέφει ως string τα ονόματα των φοιτητών της ομάδας (π.χ. "Charisis_Paradopoulos_Ioannidis").
- e. Method *findNeighbours(Cell list)*. Η μέθοδος δέχεται ως είσοδο το ταμπλό, δηλαδή μία λίστα από αντικείμενα τύπου *Cell*, όπου η αρίθμηση ξεκινά από κάτω προς τα πάνω και από αριστερά προς τα δεξιά (Εικόνα 2). Η μέθοδος επιστρέφει μία λίστα από αντικείμενα τύπου *Cell* (ίδιου μεγέθους με αυτή της εισόδου) όπου τα κελιά που είναι γείτονες των κατειλημμένων κελιών (ανεξαρτήτως χρώματος) έχουν *owner* το *pid* του παίκτη και *value* την τιμή μηδέν. Στην Εικόνα 3 με X σημειώνονται τα κατειλημμένα κελιά και με N τα κελιά γείτονες. Εάν δεν υπάρχει κατειλημμένο κελί, τότε η έξοδος είναι ίδια με την είσοδο. Προσοχή στις οριακές συνθήκες.



Εικόνα 2. Αρίθμηση κελιών του ταμπλό



Εικόνα 3. Τα γειτονικά κελιά που πρέπει να εντοπίσει η μέθοδος *findNeighbours()* είναι σημειωμένα με N.

Παράδειγμα: Έστω ότι παίζει ο πράσινος (*owner* = 2) και η κατάσταση του ταμπλό δίνεται από την παρακάτω λίστα (η λίστα αυτή δίνεται ως είσοδος στη συνάρτηση *findNeighbours*)

```
[cell(0,0), cell(0,0), cell(15,1), cell(8,2), cell(0,0), cell(0,0),
cell(0,0), cell(0,0), cell(0,0), cell(0,0), cell(0,0), cell(0,0),
cell(0,0), cell(0,0), cell(0,0), cell(0,0), cell(0,0), ...]
```

Δηλαδή, τα κελιά στις θέσεις 2 και 3 είναι κατειλημμένα. Η συνάρτηση *findNeighbours()* πρέπει να επιστρέψει την παρακάτω λίστα:

```
[cell(0,0), cell(0,2), cell(15,1), cell(8,2), cell(0,2), cell(0,0),
cell(0,0), cell(0,0), cell(0,0), cell(0,0), cell(0,0), cell(0,2),
cell(0,2), cell(0,2), cell(0,0), cell(0,0), cell(0,0), ...]
```

Δηλαδή, τα γειτονικά κελιά είναι αυτά που βρίσκονται στις θέσεις 1, 11, 12, 13 και 4.

- f. Method *placeTile()*. Είσοδοι: (i) ένας ακέραιος αριθμός (ο οποίος πρέπει να τοποθετηθεί σε κάποιο κελί του ταμπλό, (ii) λίστα από cell (δηλαδή το ταμπλό στην τρέχουσα κατάσταση). Έξοδος: (i) η θέση στην οποία θα τοποθετηθεί ο νέος αριθμός. Η μέθοδος *placeTile()* πρέπει να καλεί τη μέθοδο *findNeighbours()*, να αξιολογεί τις πιθανές θέσεις που μπορεί να τοποθετηθεί ο νέος αριθμός και να τον τοποθετεί στην καταλληλότερη θέση. Εάν είναι ο πρώτος αριθμός που τοποθετείται στο ταμπλό τότε πρέπει να τοποθετηθεί στο κεντρικό κελί του ταμπλό. Εάν καμία από τις γειτονικές θέσεις (που βρίσκει η *findNeighbours()*) δεν είναι συμφέρουσα, τότε ο αριθμός μπορεί να τοποθετηθεί σε οποιοδήποτε ελεύθερο κελί του ταμπλό.

- g. Method *findMyNeighbours(int)*. Η μέθοδος αυτή δέχεται ως είσοδο τη θέση ενός κελιού και επιστρέφει όλα τα γειτονικά κελιά, ανεξάρτητα με το εάν είναι κενά ή όχι. Προσοχή στις οριακές συνθήκες (άκρα του ταμπλό).
 - h. Method *applyChanges()*. Είσοδοι: (i) ένας ακέραιος αριθμός (ο οποίος πρέπει να τοποθετηθεί σε κάποιο κελί του ταμπλό, (ii) λίστα από *Cell* (δηλαδή το ταμπλό στην τρέχουσα κατάσταση). Έξοδος: (i) λίστα από *Cell* (δηλαδή η νέα κατάσταση του ταμπλό). Η μέθοδος *applyChanges()* τοποθετεί τον αριθμό στο κατάλληλο κελί (πρέπει να χρησιμοποιηθεί η *placeTile()*) και έπειτα ανανεώνει τις τιμές και το χρώμα των γειτονικών κελιών (πρέπει να χρησιμοποιηθεί η *findMyNeighbours()*) με βάση τους κανόνες του Proximity.
- 3) Δημιουργήστε κυρίως πρόγραμμα το οποίο να αποδεικνύει την καλή λειτουργία των παραπάνω.

Παραδοτέα

1. Ένα αρχείο με όνομα **ProximityX_Player.py**, όπου το X θα αντικατασταθεί από τον αριθμό της ομάδας σας. Το αρχείο αυτό πρέπει να περιλαμβάνει τις κλάσεις *Cell* και *ProximityX* που δημιουργήσατε καθώς και το κυρίως πρόγραμμα που αποδεικνύει την σωστή λειτουργία των κλάσεων.
2. Σύντομη γραπτή αναφορά με όνομα **ProximityX_Report.pdf**, στην οποία πρέπει να περιέχονται τα ακόλουθα:
 - a. Σύντομη περιγραφή του προβλήματος και των αλγορίθμων που υλοποιήσατε.
 - b. Σύντομη περιγραφή της λογικής με την οποία η μέθοδος *placeTile()* αξιολογεί τις διαθέσιμες θέσεις και τοποθετεί ένα αριθμό στο ταμπλό.
 - c. Αποτελέσματα που να δείχνουν την ορθή λειτουργία του κώδικα (π.χ. screenshots) με επεξηγήσεις και σχόλια.
 - d. Πίνακας όπου να φαίνεται με τι ασχολήθηκε κάθε μέλος της ομάδας και σε τι ποσοστό.
 - e. Τον κώδικα (όχι με τη μορφή εικόνας)
3. Ένα .zip αρχείο με όνομα Επώνυμο1_Επώνυμο2_Επώνυμο2.zip (π.χ., Charisis_Papadopoulos_Ioannidis.zip) το οποίο θα περιέχει τα ProximityX_Player.py και ProximityX_Report.pdf. Το zip αρχείο πρέπει να αποσταλεί στο email vaschar@auth.gr έως 29-05-2022.

Προσοχή

- 1) Τα ονόματα των κλάσεων, μεθόδων κλπ. πρέπει να είναι ακριβώς όπως στην εκφώνηση της εργασίας.
- 2) Τα ονόματα των παραδοτέων πρέπει να είναι ακριβώς όπως στην εκφώνηση της εργασίας.
- 3) Ο κώδικας πρέπει να συνοδεύεται από πλήρη τεκμηρίωση (σχόλια). Στην αρχή του αρχείου ProximityX_Player.py πρέπει να υπάρχει επικεφαλίδα σε μορφή σχολίων με τα στοιχεία σας (ονοματεπώνυμο, ΑΕΜ, email). Επίσης, πριν από κάθε μέθοδο (method) πρέπει να υπάρχει επικεφαλίδα με τη μορφή σχολίων όπου θα παρουσιάζεται μια σύντομη περιγραφή της λειτουργικότητας της και θα περιγράφονται οι μεταβλητές τους
- 4) Οι αναφορές και ο κώδικας θα περάσουν από εργαλείο ελέγχου λογοκλοπής. Σε περίπτωση λογοκλοπής η εργασία θα μηδενίζεται.