

Οντοκεντρικός Προγραμματισμός II –C++

Ακαδημαϊκό Έτος 2014-2015

Εργαστηριακή Άσκηση

Στην παρούσα εργασία καλείστε να υλοποιήσετε την προσομοίωση ενός συνόλου από πλοία που κινούνται σε μια θαλάσσια περιοχή. Διαφόρων ειδών πλοία κινούνται μέσα σε έναν προκαθορισμένο χώρο, ο οποίος και θα αποτελεί τον χάρτη του κόσμου προσομοίωσης.

ΠΕΡΙΒΑΛΛΟΝ ΘΑΛΑΣΣΑΣ ΚΑΙ ΕΙΔΗ ΠΛΟΙΩΝ

Κάθε σημείο στον χάρτη αποτελεί αντικείμενο μιας κλάσης με τις παρακάτω μεταβλητές: *Ένταση καιρού* (ακέραια τιμή από 1 έως 10 που δείχνει πόσο έντονες καιρικές συνθήκες επικρατούν στην συγκεκριμένη θέση), *Έχει Θησαυρό* (Boolean τιμή που δείχνει αν υπάρχει θησαυρός στην συγκεκριμένη θέση), *Είναι Λιμάνι* (Boolean που δείχνει αν υπάρχει λιμάνι στην συγκεκριμένη θέση).

Κάθε πλοίο (αντικείμενο αντίστοιχης κλάσης) αναπαριστάται από τις παρακάτω τιμές: Μέγιστη Αντοχή και Τρέχουσα Αντοχή, Ταχύτητα (ακέραια τιμή που δείχνει πόσες θέσεις μπορεί να μετακινηθεί σε κάθε γύρο), Απόθεμα Θησαυρού (ποσότητα θησαυρού που έχει συγκεντρώσει).

Τα πλοία σε κάθε γύρο εκτελούν δυο βασικές λειτουργίες (μέθοδοι της κλάσης Πλοίο):

Μετακίνηση: το πλοίο κινείται με τυχαίο τρόπο όσες θέσεις του επιτρέπει η ταχύτητα του

Λειτουργία: το πλοίο εκτελεί συγκεκριμένες λειτουργίες ανάλογα με το είδος του. Υπόδειξη: Η μέθοδος αυτή μπορεί να οριστεί ως *pure virtual* για την βασική κλάση Πλοίο.

Τα πλοία διαχωρίζονται σε τέσσερις βασικές κατηγορίες (υποκλάσεις):

- Πειρατικό: Έχει μικρότερη ταχύτητα από τα υπόλοιπα πλοία (1 θέση κάθε γύρο) αλλά έχει επιπλέον μια τιμή Δύναμης (πόση ζημιά προκαλεί όταν επιτίθεται σε άλλα πλοία). Στην μέθοδο Λειτουργία το πειρατικό πλοίο επιτίθεται σε ένα πλοίο το οποίο βρίσκεται σε γειτονική θέση και κερδίζει ένα ποσοστό από το θησαυρό του.
- Εμπορικό: Στην μέθοδο Λειτουργία το εμπορικό πλοίο αν υπάρχει λιμάνι σε γειτονική θέση αυξάνει το απόθεμα θησαυρού του κατά μια σταθερή τιμή.
- Επισκευαστικό: Στην μέθοδο Λειτουργία το επισκευαστικό πλοίο αν υπάρχει γειτονικό πλοίο που έχει υποστεί ζημιά, αυξάνει την τρέχουσα αντοχή του με αντάλλαγμα ποσότητα θησαυρού.
- Εξερεύνησης: Στην μέθοδο Λειτουργία το πλοίο εξερεύνησης ελέγχει αν υπάρχει πειρατικό πλοίο ή έντονη κακοκαιρία σε γειτονική θέση και μετακινείται μια θέση προς αντίθετη κατεύθυνση.

ΡΟΗ ΕΚΤΕΛΕΣΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

Στην αρχή της προσομοίωσης δημιουργείται με αυτόματο τρόπο ο χάρτης ορίζοντας με τυχαίο τρόπο συγκεκριμένα σημεία λιμανιών και θησαυρού και δημιουργώντας πλοία όλων των ειδών.

A. Στην αρχή κάθε γύρου γίνεται έλεγχος σε κάθε θέση του χάρτη και:

- Αν σε μια θέση η ένταση του καιρού είναι πολύ υψηλή προκαλείται ζημιά στο πλοίο που βρίσκεται εκεί.

- Αν στην θέση υπάρχει θησαυρός τότε αυξάνεται το απόθεμα θησαυρού του πλοίου που βρίσκεται εκεί.
- Αν σε μία θέση υπάρχει λιμάνι τότε: Αν ένα γειτονικό πλοίο είναι πειρατικό, του προκαλείται ζημιά αλλιώς το πλοίο επισκευάζεται κατά ένα ποσοστό. Υπόδειξη: Αν διατηρείτε όλα τα πλοία σε έναν πίνακα τύπου *Πλοίο* τότε μπορείτε να χρησιμοποιήσετε την συνάρτηση *dynamic_cast* για να ελέγξετε αν ένα πλοίο είναι συγκεκριμένου τύπου. Εναλλακτικά μπορείτε να χρησιμοποιήσετε πολυμορφισμό ορίζοντας μια μέθοδο στην κλάση πλοίο και υπερφορτώνοντας την στις υποκλάσεις.

Β. Για κάθε πλοίο καλείται η μέθοδος μετακίνησης. Στην συνέχεια για κάθε πλοίο καλείται η μέθοδος λειτουργίας.

Γ. Στο τέλος κάθε γύρου:

- Γίνεται έλεγχος και αφαιρείται όποιο πλοίο έχει αντοχή κάτω από 0. Η θέση στην οποία βρισκόταν μετατρέπεται σε θησαυρό.
- Ανανεώνονται οι τιμές έντασης του καιρού σε κάθε θέση κατά μια μονάδα με τυχαίο τρόπο προς τα πάνω ή προς τα κάτω.
- Συνθήκες Τερματισμού Προσομοίωσης:
 - ο Αν δεν υπάρχει κανένα πλοίο στον χάρτη, η προσομοίωση τερματίζει με σχετικό μήνυμα.
 - ο Αν ένα από τα πλοία έχει συγκεντρώσει μια προκαθορισμένη ποσότητα/στόχος θησαυρού, η προσομοίωση τερματίζει με σχετικό μήνυμα.

ΥΠΟΔΕΙΞΕΙΣ

Δημιουργήστε 2 δικές σας κλάσεις πλοίων ως υποκλάσεις κάποιων από τις τέσσερις που περιγράφηκαν παραπάνω. Για κάθε μία μπορείτε να προσθέσετε δικά σας μέλη και να κάνετε υπερφόρτωση της μεθόδου λειτουργίας ή και της μεθόδου μετακίνησης. Υπόδειξη: Μπορείτε να χρησιμοποιήσετε την **super** ώστε να αποφύγετε την επανάληψη κώδικα που έχετε υλοποιήσει στην κλάση από την οποία κληρονομείτε.

Δώστε προσοχή στα δικαιώματα μεταβλητών και μεθόδων. Υλοποιήστε set και get μεθόδους με τους απαραίτητους ελέγχους και αποφύγετε άμεση πρόσβαση στις μεταβλητές των κλάσεων.

Χρησιμοποιήστε πολυμορφισμό για την συνάρτηση λειτουργίας και αξιοποιήστε το στον κώδικα σας (μπορείτε να έχετε αποθηκευμένα όλα τα πλοία σε έναν πίνακα ή vector της κλάσης *Πλοίο* και να καλείτε για το καθένα την συνάρτηση λειτουργίας).

Χρησιμοποιήστε static μεταβλητές και μεθόδους για να μπορείτε να διατηρείτε και να εμφανίζετε στατιστικές πληροφορίες σχετικά με την προσομοίωση: Για παράδειγμα συνολικός αριθμός πλοίων και αριθμός πλοίων κάθε τύπου, ποσότητα θησαυρού που έχει ανακτηθεί, ποσότητα ζημιάς που έχει υποστεί κάθε είδος πλοίου. Μπορείτε να χρησιμοποιήσετε static μεταβλητές και για την κλάση που αναπαριστά ένα σημείο στον χάρτη και να διατηρείτε σε αυτήν πληροφορίες που αφορούν συνολικά όλες τις θέσεις του χάρτη.

Κάντε υπερφόρτωση τουλάχιστον 4 τελεστών για την απλοποίηση του κώδικα. Για παράδειγμα χρήση τελεστών +/- για αύξηση μείωση του θησαυρού ενός πλοίου, χρήση τελεστή << για εκτύπωση πληροφορίας σχετικής με ένα πλοίο.

Προσπαθήστε να συγκεντρώσετε όλες τις παραμέτρους (ποσότητα θησαυρού στόχος, αρχικές τιμές αντοχής κάθε είδους πλοίου, αριθμός λιμανιών και θησαυρού κατά την έναρξη της προσομοίωσης κτλ) της προσομοίωσης σε συγκεκριμένο σημείο ή αρχείο ως σταθερές (χρήση της #define).

Διεπαφή: Κατά την διάρκεια προσομοίωσης ο χρήστης μπορεί να κάνει παύση και μέσω μενού να ζητήσει πληροφορίες ή και να επέμβει. Συγκεκριμένα μπορεί να ζητήσει πληροφορίες για (1) συγκεκριμένο πλοίο, (2) συγκεκριμένη θέση στον χάρτη, (3) συγκεκριμένο τύπο πλοίου ή (4) γενικές πληροφορίες για την προσομοίωση (πχ κατάταξη πλοίων με βάση τον θησαυρό που έχουν ή τη ζημιά που έχουν προκαλέσει). Επίσης μπορεί να επέμβει προσθέτοντας ή αφαιρώντας πλοία στον χάρτη ή τροποποιώντας μια θέση στον χάρτη ώστε να προσθέσει/αφαιρέσει λιμάνι ή θησαυρό.

ΚΑΝΟΝΕΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

1. Σε κάθε επανάληψη της προσομοίωσης εμφανίζονται σύντομα μηνύματα που περιγράφουν τι συμβαίνει πχ (το πλοίο Α μετακινήθηκε, το πλοίο Β έπαθε ζημιά λόγω κακοκαιρίας, το πλοίο εξερεύνησης Γ μετακινήθηκε για να αποφύγει πειρατικό πλοίο κτλ.)
2. Για κάθε πλοίο καταγράφεται το πόσες θέσεις έχει μετακινηθεί από την αρχή της εξομοίωσης και πόση ζημιά έχει υποστεί. Για ένα πειρατικό πλοίο καταγράφεται επιπλέον πόση ζημιά έχει προκαλέσει. Για τα επισκευαστικά πλοία καταγράφεται η ποσότητα θησαυρού που κέρδισαν από επιδιορθώσεις και για τα εμπορικά η ποσότητα που κέρδισαν από επισκέψεις σε λιμάνια.
3. Θεωρούμε ότι όλα τα οχήματα κινούνται συνεχώς μέσα στον κόσμο.

Σημείωση: Για οτιδήποτε δεν είναι ξεκάθαρο από την εκφώνηση μπορείτε να κάνετε δικές σας παραδοχές και υποθέσεις. Ειδικά για την ποσότητα με την οποία μειώνετε και αυξάνετε κάθε φορά την αντοχή και το θησαυρό κάθε πλοίου, κάντε δικές σας παραδοχές προσπαθώντας να κάνετε ρεαλιστική την προσομοίωση.

Διεπαφή με τον χρήστη

Η προσομοίωση θα υλοποιηθεί με διεπαφή κονσόλας. Ο χρήστης θα έχει την δυνατότητα να κάνει παύση της προσομοίωσης για να δει την συνολική κατάσταση του κόσμου ή/και πληροφορίες για κάποιο όχημα ή θέση του κόσμου, να εισάγει εξωτερικούς παράγοντες και να τερματίσει την προσομοίωση.

Γραφική Διεπαφή με χρήση της βιβλιοθήκης GLUT (Προαιρετικό!!!)

Η υλοποίηση γραφικής παραθυρικής διεπαφή για την οπτικοποίηση της εξέλιξης της προσομοίωσης είναι προαιρετική. Για την γραφική διεπαφή μπορεί να χρησιμοποιηθεί η βιβλιοθήκη GLUT (<http://www.lighthouse3d.com/opengl/glut/>). Για την οπτικοποίηση μπορείτε να κάνετε όποιες συμβάσεις επιθυμείτε: π.χ. να χρησιμοποιήσετε διαφορετικά χρώματα και σχήματα για κάθε οργανισμό ή να χρησιμοποιήσετε εικόνες bitmap.

ΟΔΗΓΙΕΣ

1. Βαθμολόγηση:
 - a. Για να υπάρξει προσμέτρηση της προγραμματιστικής εργασίας στον τελικό βαθμό, ο βαθμός της γραπτής εξέτασης θα πρέπει να είναι μεγαλύτερος ή ίσος του 5.
 - b. Η προγραμματιστική εργασία είναι υποχρεωτική και πρέπει να υποβληθεί για να προσμετρηθεί στην εξεταστικής του Ιανουαρίου. Αν η εργασία υποβληθεί την περίοδο του Ιουνίου-Σεπτεμβρίου συμμετέχετε μόνο στην αντίστοιχη γραπτή εξέταση. Αν δεν υποβάλλετε εργασία παίρνετε 0 στο αντίστοιχο ποσοστό.

- c. Η προγραμματιστική εργασία συμμετέχει με βάρος 30% στην τελική βαθμολόγηση του μαθήματος. Σε περίπτωση που υλοποιηθεί το προαιρετικό ερώτημα της γραφικής διεπαφής συμμετέχει με βάρος 40%. Σε περίπτωση που η εργασία υποβληθεί τον Σεπτέμβριο (ή Ιούνιο) η εργασία συμμετέχει με βάρος 15% (χωρίς γραφικά) και 20% (με γραφικά).
 - d. Η βαθμολογία στην προγραμματιστική άσκηση δεν κατοχυρώνεται για επόμενη ακαδημαϊκή χρονιά.
2. Το πρόγραμμά σας θα πρέπει να υλοποιεί όλο το σύνολο των κλάσεων και διεπαφών για την υποστήριξη των παραπάνω λειτουργικών απαιτήσεων
 3. Το πρόγραμμα πρέπει να δημιουργηθεί χρησιμοποιώντας αντικειμενοστραφή σχεδιασμό (ιεραρχία κλάσεων, κληρονομικότητα, πολυμορφισμός κ.λ.π.).
 4. Η εργασία γίνεται σε ομάδες των 1-3 ατόμων
 5. Απορίες υποβάλλονται στο φόρουμ του μαθήματος
 6. Η παράδοση της εργασίας γίνεται μέσω του **eclass.upatras.gr**
 7. Ημερομηνία Παράδοσης **10/01/2015**
 8. **Παραδοτέα:** Η άσκηση θα παραδίδεται υπό την μορφή ενός ZIP αρχείου, με όνομα <AM1>_<AM2>_<AM3>.zip (πχ. 1821_1822_1822.zip) όπου AMx οι αριθμοί μητρώου των μελών της ομάδας, και το οποίο θα περιέχει τα ακόλουθα:
 1. Πηγαίο κώδικα των κλάσεων που υλοποιούν την άσκηση (όχι τα εκτελέσιμα αρχεία!!)
 2. Σύντομη επεξήγηση του προγράμματος σας (.pdf).
 3. Διάγραμμα κλάσεων. Ένα σχεδιάγραμμα που δείχνει τη δομή και τις σχέσεις μεταξύ των κλάσεων του σχεδιασμού σας
 4. Ένα αρχείο txt με τις παραμέτρους εκτέλεσης του προγράμματος.
 5. Ένα αρχείο περιγραφής της μεταγλώττισης και σύνδεσης του προγράμματος (makefile ή DevC++ project file ή Visual Studio Solution File ή code::blocks project file)