Παραδοχές

Για την υλοποίηση του προγράμματος, έχουν γίνει οι εξής παραδοχές:

1. Παραδοχές υλοποίησης κόσμου.
    - Ο χάρτης είναι μεγέθους 10x10
    - Με την εκκίνηση του προγράμματος αυτόματα δημιουργούνται:
        - Τυχαίος αριθμός λιμανιών, με κάθε θέση να έχει πιθανότητα 10% να είναι λιμάνι.
        - Τυχαίος αριθμός θησαυρών, με κάθε θέση να έχει πιθανότητα $(1/8)$ να είναι λιμάνι.
        - 12 πλοία (δυο από το κάθε είδος), τα οποία τοποθετούνται στον χάρτη με τυχαία σειρά και σε τυχαίες ελεύθερες θέσεις.
        - Καιρικές συνθήκες σε κάθε θέση μεταξύ των τιμών [1,10]. Όποια θέση έχει τιμή > 7 θεωρείται ότι υπάρχει κακοκαιρία.
    - Η καταστροφή που προκαλείται σε ένα πλοίο λόγο κακοκαιρίας είναι 30.
    - Όταν ένα πλοίο (εκτός από τα Πειρατικά) είναι δίπλα σε λιμάνι, η αντοχή που λαμβάνει είναι το 10% της τρέχουσας αντοχής του. Ενώ, άμα είναι Πειρατικό δέχεται χτύπημα 30.

2. Παραδοχές υλοποίησης πλοίων.

    - Δημιουργία 12 πλοίων, αρχικοποιημένα με βάση τον παρακάτω πίνακα.

Τύπος Πλοίου	**Μέγιστη Αντοχή**	**Ταχύτητα**	**Θησαυρός**	**Δύναμη**
Πειρατικό	500	1	50	20
Εμπορικό	150	3	100	-
Εξερεύνησης	300	5	80	-
Επισκευαστικό	250	2	50	-
Υποβρύχιο	300	1	50	-
Λιμενικό	300	1	150	-

    *Τα πλοία που ανήκουν στις κατηγορίες Υποβρύχιο και Λιμενικό είναι αυτά που προστέθηκαν από εμάς και οι λειτουργείς τους είναι:
        - Το Υποβρύχιο αν εντοπίσει σε γειτονική θέση κάποιο πλοίο, τότε το καταστρέφει και παίρνει τον θησαυρό του. Όμως στην $3^η$ επαφή του με κάποιο πλοίο, το Υποβρύχιο καταστρέφεται και το $3^ο$ πλοίο παίρνει τον θησαυρό του.
        - Το Λιμενικό αν εντοπίσει Πειρατικό, τότε του προκαλεί ζημία ίση με την τρέχουσα αντοχή του Λιμενικού. Στην περίπτωσή, όμως, που το Λιμενικό έχει λιγότερη αντοχή από το Πειρατικό, το Λιμενικό καταστρέφεται και προκαλεί στο Πειρατικό ζημία ίση με την αντοχή που του είχε πριν καταστραφεί.
    - Τα Πειρατικά πλοία, παράλληλα με την επίθεση που κάνουν, αποσπούν και ένα 20% από τον θησαυρό που έχει το πλοίο θύμα.

- Τα Εμπορικά πλοία, όταν βρεθούν σε γειτονική θέση με κάποιο λιμάνι αυξάνουν τον θησαυρό τους κατά 50.
- Τα Επισκευαστικά πλοία, μόλις επισκευάσουν κάποιο πλοίο σαν αντάλλαγμα παίρνουν θησαυρό μεγέθους ίσο με την ζημία που επισκεύασαν στο πλοίο.
- Κάθε πλοίο σε κάθε γύρο αλληλοεπιδρά μέσω της μεθόδου λειτουργίας του μόνο με ένα πλοίο. Δηλαδή , για παράδειγμα, αν ένα πειρατικό έχει 4 γειτονικά πλοία , θα επιτεθεί μόνο στο ένα.

3. Παραδοχές, όσο αφορά την λειτουργία των τελεστών που υπερφορτώθηκαν.
   - `Operator + (double)`
     Με αυτή την υπερφόρτωση επιτυγχάνεται η επισκευή ενός πλοίου (εκτός από τα Πειρατικά) όταν βρεθεί δίπλα σε λιμάνι.
   - `Operator −− (int)`
     Με αυτή την υπερφόρτωση επιτυγχάνεται η μείωση της αντοχής κατά 30 των πλοίων που βρίσκονται σε θέση με κακοκαιρία.
   - `Operator > (Ship*)`
     Με αυτή την υπερφόρτωση επιτυγχάνεται ο έλεγχος για την ταξινόμηση bubblesort με βάση τον θησαυρό.
   - `Operator ≪ (ostream &, Ship*)`
     Με αυτή την υπερφόρτωση επιτυγχάνεται εμφάνιση πληροφοριών για ένα πλοίο.

Ιεραρχία κλάσεων

Στο πρόγραμμα δημιουργήθηκε η εξής ιεραρχία κλάσεων. Αρχικά, όσον αφορά τον κόσμο, κατασκευάστηκε η κλάση **Location.** Η κλάση αυτή αποτελεί τον πυρήνα της προσομοίωσης καθώς αναπαριστά κάθε σημείο στο χάρτη. Αρχικοποιεί τον καιρό και τα λιμάνια αλλά και έχει μια μέθοδο weatherChange() η οποία στο τέλος κάθε γύρου αλλάζει τον καιρό κατά μία μονάδα με τυχαίο τρόπο. Παράλληλα υλοποιεί την μέθοδο IsEmpty() η οποία ελέγχει αν σε ένα σημείο υπάρχει λιμάνι ή πλοίο.

Στην παραπάνω κλάση βασίζεται η κλάση **Map.** Ουσιαστικά η κλάση αυτή δημιουργεί ένα πίνακα 10x10 στον οποίο τοποθετούνται δείκτες σε σημεία τύπου Location. Η κλάση αυτή μετράει και τον αριθμό των λιμανιών που υπάρχει στον χάρτη αλλά και τα σημεία που έχουν καιρό > 7.

Η κλάση **Ship** είναι μια ξεχωριστή κλάση η οποία αποτελεί τη βάση για όλα τα πλοία. Περιέχει τους default constructors και destructors κάθε πλοίου αλλά και μεταβλητές οι οποίες αποθηκεύουν την αντοχή, το θησαυρό, την ταχύτητα, το όνομα, τη θέση, τον τύπο, τις κινήσεις και τη συνολική ζημιά που έχει υποστεί κάθε πλοίο. Παράλληλα αρχικοποιεί static μεταβλητές οι οποίες καταγράφουν τον αριθμό των πλοίων που υπάρχουν στην προσομοίωση όσο και την ποσότητα θησαυρού που έχει ανακτηθεί. Η κλάση Ship, επίσης, έχει μια μέθοδο setLocation(Map&) η οποία δέχεται σαν όρισμα το χάρτη της προσομοίωσης, και αφού παραχθεί ένα ζευγάρι τυχαίων συντεταγμένων προσπαθεί να τοποθετήσει ένα πλοίο σε αυτές (κριτήριο για

αυτή την τοποθέτηση είναι αν η εν λόγω θέση έχει ήδη πλοίο ή λιμάνι). Αν αποτύχει παράγει άλλο ζευγάρι συντεταγμένων κοκ. Ύστερα, υπάρχουν οι μέθοδοι printLocation() και printType() οι οποίες εμφανίζουν πληροφορίες σχετικά με την τοποθεσία και τον τύπο του πλοίου (εμπορικό, πειρατικό κα.) αλλά και η μέθοδος printInfo() η οποία καλεί τις δυο παραπάνω και εμφανίζει τη συνολική κατάσταση του πλοίου. Όσον αφορά την μέθοδο movement(int, int, int, Map*), αυτή ουσιαστικά υλοποιεί την κίνηση του πλοίου με την εξής λογική: επιλέγεται μια τυχαία πορεία για το πλοίο και αν δεν μπορεί να μετακινηθεί προς εκείνη, η πορεία μεταβάλλεται και προσπαθεί να κινηθεί προς μια άλλη κατεύθυνση. Αν μετά τον έλεγχο και των 4 κατευθύνσεων το πλοίο έχει αποτύχει στην κίνηση του , τότε μένει στην αρχική του θέση. Η μέθοδος operation(int, int, Map*, vector<Ship*>) είναι η μέθοδος η οποία περιγράφει την λειτουργία του πλοίου και επειδή γίνεται overwrite σε υποκλάσεις έχει δηλωθεί ως virtual. Τέλος, η findShip(int, int, vector<Ship*>), δέχεται σαν όρισμα ένα ζευγάρι συντεταγμένων και ένα vector που περιέχει πλοία και επιστρέφει ένα δείκτη προς το αντικείμενο που βρίσκεται στις παραπάνω συντεταγμένες.

Η κλάση **PirateShip** είναι μια υποκλάση της **Ship**. Έχει δυο παραπάνω ιδιότητες: attackingDamage, η οποία δείχνει την ζημιά που υφίσταται το πλοίο στο οποίο επιτίθεται, και causedDamage, η οποία καταμετρά τη συνολική ζημιά την οποία έχει υποστεί ένα πειρατικό. Όσον αφορά τη μέθοδο λειτουργίας του, το πειρατικό ελέγχει αν γύρω του υπάρχουν γειτονικά πλοία και αν βρει, εκτελεί τη λειτουργία που αναφέρεται παραπάνω.

Η κλάση **MerchantShip** είναι μια υποκλάση της **Ship.** Έχει επιπλέον την ιδιότητα portTreasure, η οποία αναπαριστά τον θησαυρό που κερδίζει από επισκέψεις σε λιμάνια μέσω της λειτουργίας της. Τα εμπορικά σε αντίθεση με τα υπόλοιπα πλοία θα πάρουν όσο περισσότερο θησαυρό μπορούν από λιμάνια, δηλαδή σε κάθε γύρο μπορούν να επισκεφτούν παραπάνω από ένα λιμάνια.

Η κλάση **RepairShip** είναι μια υποκλάση της **Ship.** Έχει μια παραπάνω ιδιότητα, την totalHeal, η οποία καταμετρά τη συνολική επιδιόρθωση που έχει καταφέρει σε άλλα πλοία. Όπως αναφέρεται και παραπάνω, το πλοίο επιδιόρθωσης ελέγχει αν γύρω του υπάρχουν λαβωμένα πλοία, και αν βρει διορθώνει ένα από αυτά.

Η κλάση **ExplorationShip** είναι και αυτή υποκλάση της κλάσης **Ship**. Η λειτουργία της κλασης αυτής είναι ότι ελέγχει αν σε μια τοποθεσία έχει γειτονικά κακό καιρό ή πειρατικό πλοίο και κινείται μια θέση προς την αντίθετη κατεύθυνση. Σε κάθε γύρο μπορεί να κάνει μέχρι δυο κινήσεις, μια λόγω πειρατικού και μια λόγω καιρού. Παράλληλα η κλαση **ExplorationShip** έχει μια μέθοδο movement(int direction, Map* myMap) η οποία υλοποιεί την παραπάνω λειτουργία. Ουσιαστικά δέχεται σαν όρισμα μια κατεύθυνση και αν η κίνηση είναι νόμιμη τότε την εκτελεί, αλλιώς το πλοίο μένει στην αρχική του θέση.

Η κλάση **NavalShip** κληρονομεί την κλάση **ExplorationShip.** Σκοπός των πλοίων αυτού του τύπου είναι να εντοπίζουν πειρατικά πλοία και να τους καταστούν όσο το δυνατόν μεγαλύτερη ζημιά γίνεται. Η ακριβής λειτουργία τους περιγράφεται παραπάνω.

Τέλος, η κλάση **Submarine** κληρονομεί την κλάση **PirateShip**. Έχει μια ιδιότητα kills, η οποία αναπαριστά τα πλοία τα οποία έχει καταστρέψει. Η ακριβής λειτουργία τους περιγράφεται παραπάνω και όπως αναφέρεται και στις παραδοχές σε κάθε γύρο μπορούν να επιδράσουν μόνο σε ένα πλοίο.

Ροή εκτέλεσης

Με την εκκίνηση του προγράμματος, δημιουργείται ένας πίνακας 10x10 ως χάρτης. Παράλληλα τοποθετούνται τα λιμάνια και οι θησαυροί και ορίζεται καιρός. Όλα με τυχαίο τρόπο. Επίσης, δημιουργούνται τα πλοία, τα οποία αποθηκεύονται σε ένα Vector και τοποθετούνται στον χάρτη με τυχαία σειρά στις θέσεις που είναι ελεύθερες. Ακόμα, γίνεται μια τυχαία ανακατάταξη στο Vector, ώστε να μην είναι προκαθορισμένες οι θέσεις στις οποίες αντιστοιχεί κάθε τύπος πλοίου και στην συνέχεια το κάθε πλοίο λαμβάνει ως όνομα μέχρι τη καταστροφή του ή το τέλος της προσομοίωσης το Ship <θέση στο Vector> .Τέλος, εμφανίζεται ένα Menu επιλογών για την διαχείριση του προγράμματος. Το Menu, έχει τις εξής επιλογές:

1. Για να συνεχιστεί η προσομοίωση.
2. Για να εμφανιστούν πληροφορίες για ένα πλοίο δίνοντας το όνομά του ή για περισσότερα πλοία δίνοντας την τιμή -1.
3. Για να εμφανιστούν πληροφορίες σχετικά με μια θέση στον χάρτη, δίνοντας τις συντεταγμένες.
4. Για να εμφανιστούν πληροφορίες σχετικά με έναν από τους τύπους των πλοίων, δίνοντας τον τύπο που θέλουμε. Συγκεκριμένα,
    1. Πειρατικό
    2. Εμπορικό
    3. Επισκευαστικό
    4. Εξερεύνησης
    5. Λιμενικό
    6. Υποβρύχιο
5. Για να εμφανιστούν γενικές πληροφορίες σχετικά με την προσομοίωση. Συγκεκριμένα,
    1. Πλήθος θέσεων με κακοκαιρία.
    2. Πλήθος θέσεων με λιμάνι.
    3. Συνολικό ποσό μαζεμένου θησαυρού.
6. Για να μπορεί να προσθαφαιρεί ένα πλοίο στο Vector. Συγκεκριμένα,
    1. Για προσθήκη πλοίου με χρήση του τύπου του. Συγκεκριμένα,
        1. Πειρατικό
        2. Εμπορικό
        3. Επισκευαστικό
        4. Εξερεύνησης
        5. Λιμενικό
        6. Υποβρύχιο
    2. Για αφαίρεση πλοίου με χρήση του ονόματός του.
    3. Για να αφαιρεθούν όλα τα πλοία.
7. Για να τροποποιήσουμε μια συγκεκριμένη θέσης τον χάρτη, δίνοντας συντεταγμένες. Συγκεκριμένα,
    1. Για να προσθέσουμε λιμάνι.

    2. Για να αφαιρέσουμε λιμάνι.
    3. Για να προσθέσουμε θησαυρό εισάγοντας το ποσό.
    4. Για να αφαιρέσουμε θησαυρό εισάγοντας το ποσό.
8. Για να εμφανιστεί μια αναπαράσταση του χάρτη. Συγκεκριμένα,
    1. Για αναπαράσταση των πλοίων του χάρτη.
    2. Για αναπαράσταση των λιμανιών του χάρτη.
    3. Για αναπαράσταση του καιρού σε όλα τα σημεία του χάρτη.
    4. Για αναπαράσταση του θησαυρού στον χάρτη.
9. Για να ταξινομηθούν τα πλοία μέσα στο Vector σε φθίνουσα σειρά. Συγκεκριμένα,
    1. Με βάση το θησαυρό που έχουν.
    2. Με βάση την τρέχουσα αντοχή.
    3. Με βάση τον τύπο τους.
10. Για να τερματιστεί η προσομοίωση. Συγκεκριμένα,
    y. Για επανεκκίνηση προσομοίωσης.
    n. Για έξοδο.