

ΟΝΤΟΚΕΝΤΡΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ II PROJECT

“Υλοποίηση προσομοίωσης συνόλου πλοίων που κινούνται σε μία
θαλάσσια περιοχή”

Ακαδημαϊκό Έτος: 2015 - 2016

Ονοματεπώνυμο: Δημήτριος Ζαρκάδας || Αριθμός Μητρώου: 4982
Ονοματεπώνυμο: Χαράλαμπος Κανελλόπουλος || Αριθμός Μητρώου: 4994

1. Κώδικες Υλοποίησης & Ιεραρχία Κλάσεων

Η υλοποίηση της εφαρμογής πραγματοποιήθηκε στο παρακάτω περιβάλλον :

- Windows 7 Ultimate, Service Pack 1 (x64)
- Microsoft Visual Studio 2015

Τα σχετικά αρχεία κώδικα C++ που δημιουργήθηκαν και είναι αναγκαία για την υλοποίηση όλης της εφαρμογής (*folder : Codes/Cpp_2015_CEID*) είναι :

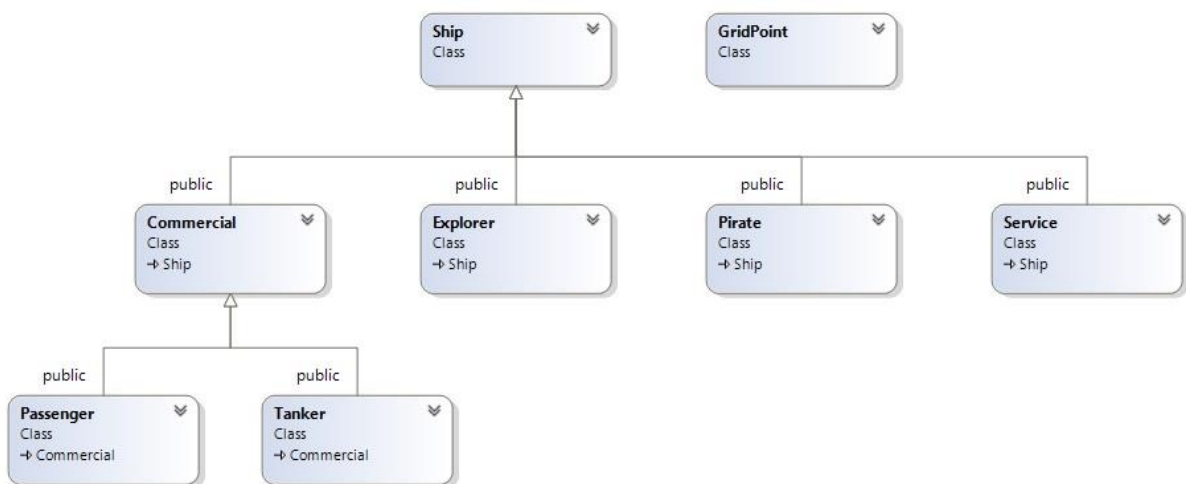
.cpp Files :

- Main.cpp
- GridPoint.cpp
- Ship.cpp
- Commercial.cpp
- Explorer.cpp
- Passenger.cpp
- Pirate.cpp
- Service.cpp
- Tanker.cpp

.h Files :

- GridPoint.h
- Ship.h
- Commercial.h
- Explorer.h
- Passenger.h
- Pirate.h
- Service.h
- Tanker.h

Ο χάρτης της προσομοίωσης δημιουργήθηκε μέσω της κλάσης GridPoint, ενώ όλα τα πλοία που κινούνται μέσα σε αυτόν (υποκλάσεις : Commercial, Explorer, Passenger, Pirate, Service, Tanker) κληρονομούν τις βασικές τους ιδιότητες από την κλάση Ship. Επιπρόσθετα, τα πλοία τύπου Passenger και Tanker κληρονομούν ιδιότητες και από την κλάση Commercial. Παρακάτω, παρατίθεται η σχηματική απεικόνιση της ιεραρχίας των δημιουργηθέντων κλάσεων :



2. Επεξήγηση της Υλοποίησης

Ο **χάρτης** του κόσμου προσομοίωσης υλοποιήθηκε ως ένα δισδιάστατο vector (GridMap) με περιεχόμενα δείκτες προς αντικείμενα τύπου GridPoint. Οι διαστάσεις του χάρτη καθορίζονται από τον χρήστη σε κάθε εκτέλεση του προγράμματος, ο οποίος εισάγει την διάσταση που επιθυμεί (5 έως 20) και παράγεται ένας χάρτης (vector) από (5 x 5) έως και (20 x 20) αντίστοιχα.

Το **σύνολο των πλοίων** που δημιουργούνται κατά την έναρξη, αλλά και κατά τη διάρκεια της προσομοίωσης αποθηκεύονται σε ένα δισδιάστατο vector 7 γραμμών, με περιεχόμενα δείκτες προς αντικείμενα τύπου Ship ως εξής :

Γραμμή :	Τύποι πλοίων που περιέχονται :	Υποκλάσεις που περιέχονται :
#1	Εμπορικά, Επιβατικά, Πετρελαιοφόρα πλοία	Commercial, Passenger, Tanker
#2	Εξερευνητικά πλοία	Explorer
#3	Επιβατικά πλοία	Passenger
#4	Πειρατικά πλοία	Pirate
#5	Επισκευαστικά πλοία	Service
#6	Πετρελαιοφόρα πλοία	Tanker
#7	Όλα τα πλοία	Όλες οι παραπάνω

Το συνολικό πλήθος και το είδος καθενός από τα πλοία που δημιουργούνται κατά την έναρξη της προσομοίωσης υπολογίζεται τυχαία κάθε φορά και κυμαίνεται από 10 έως και $(10 + n^2/2)$, όπου n είναι η διάσταση του χάρτη που έχει εισαγάγει ο χρήστης παραπάνω. Ακόμη, οι διάφορες λειτουργίες των πλοίων καθώς και η διαγραφή/βύθιση κάποιου εξ' αυτών εμφανίζονται στον χρήστη μέσω μηνυμάτων στην κονσόλα, ενώ η μετακίνηση τους σε κάθε γύρο εμφανίζεται στο χάρτη.

Τα **σύμβολα** που χρησιμοποιούνται για την απεικόνιση σημείων και πλοίων στο χάρτη ορίζονται στα αρχεία GridPoint.h και Ship.h αντίστοιχα και συγκεκριμένα είναι :

Σύμβολο :	Σημασία :
---	Απλή θάλασσα
!!!	Λιμάνι
\$\$\$	Τοποθεσία θησαυρού
C	Εμπορικό Πλοίο
E	Εξερευνητικό Πλοίο
A	Επιβατικό Πλοίο
P	Πειρατικό Πλοίο
S	Επισκευαστικό Πλοίο
T	Πετρελαιοφόρο Πλοίο

Επίσης, κάθε σύμβολο πλοίου από τα παραπάνω, συνοδεύεται κατά την απεικόνιση του στο χάρτη από ένα ακέραιο αριθμό (ID) για να ξεχωρίζει από τα ομοειδή του. Ακόμη, ο αριθμός αυτός δείχνει και τη σειρά με την οποία κατασκευάστηκαν τα πλοία ανά είδος (αυξάνεται σειριακά κατά 1 μονάδα, σε κάθε δημιουργία πλοίου του συγκεκριμένου είδους).

Ακόμη, χρησιμοποιήθηκε **υπερφόρτωση των τελεστών** ++, --, <<, < στην κλάση Ship. Οι δύο πρώτοι αφορούν την αύξηση και μείωση αντίστοιχα του θησαυρού του αντίστοιχου πλοίου κατά μία μονάδα, ενώ με τον τελεστή << γίνεται εμφάνιση των στοιχείων συγκεκριμένου πλοίου. Τέλος υπερφορτώσαμε τον τελεστή < έτσι ώστε να συγκρίνονται τα πλοία ως προς την ταχύτητα τους. Οι παραπάνω λειτουργίες μπορούν να κληθούν χρησιμοποιώντας ένα δείκτη προς κάποιο στοιχείο του vector που είναι αποθηκευμένο κάθε πλοίο, ακολουθούμενο από τον σχετικό τελεστή που επιθυμούμε να χρησιμοποιήσουμε.

Όσον αφορά τη **δομή της προσομοίωσης**, αυτή υλοποιήθηκε ως εξής :

- Στην αρχή κάθε γύρου της προσομοίωσης (συνάρτηση `roundStart()`), προτού τα πλοία αρχίσουν να μετακινούνται και να εκτελούν τις λειτουργίες τους, ελέγχεται για καθένα εξ' αυτών εάν στην τρέχουσα θέση του :
 - a. Υπάρχει έντονη κακοκαιρία και σε τέτοια περίπτωση του προκαλείται ζημιά (κλήση συνάρτησης `Ship::WeatherDamage()`).
 - b. Βρίσκεται θησαυρός και σε τέτοια περίπτωση των συλλέγει κι ενημερώνει τον χάρτη πως δεν υπάρχει πλέον θησαυρός στο σημείο (κλήση συνάρτησης `Ship::TreasurePoint()`).
 - c. Βρίσκεται σε γειτονική θέση λιμανιού, και σε τέτοια περίπτωση αν δεν το πλοίο δεν είναι πειρατικό επισκευάζεται κατά ένα μέρος, ενώ αν είναι πειρατικό δέχεται ζημιά (κλήση συνάρτησης `Ship::PortPoint()`).
- Στη συνέχεια του κάθε γύρου (συνάρτηση `roundCont()`), κάθε πλοίο κινείται προς μια τυχαία κατεύθυνση τόσες θέσεις, όσες του επιτρέπει η ταχύτητα του (συνάρτηση `Ship::Move()`) και μετά εκτελεί την προσωπική λειτουργία του, αναλόγως με το είδος πλοίων στο οποίο ανήκει (pure virtual συνάρτηση `Ship::ShipFunc()`). Εδώ να σημειωθεί πως, αφού μετακινηθεί, κάθε πλοίο ελέγχει εάν στη νέα του θέση υπάρχει κακοκαιρία ή θησαυρός με τις συνέπειες που περιγράφηκαν παραπάνω.
- Στο τέλος κάθε γύρου (συνάρτηση `roundEnd()`), ελέγχεται η τιμή της τρέχουσας αντοχής κάθε πλοίου, κι αν αυτή είναι μη θετική τότε το πλοίο θεωρείται «νεκρό» και διαγράφεται από τον χάρτη και το vector των πλοίων (pure virtual συνάρτηση `Ship::destroyShip()`). Επίσης, ανανεώνεται η τιμή της έντασης του καιρού σε κάθε σημείο, αυξανόμενη ή μειωνόμενη τυχαία κατά 1 μονάδα. Κλείνοντας, αν όλα τα πλοία του χάρτη έχουν αφαιρεθεί ήδη ή αν κάποιο πλοίο έχει επιτύχει τον στόχο θησαυρού του τότε η προσομοίωση τελειώνει αυτόματα με το πέρας του γύρου αυτού.

Για να γίνει **παύση της προσομοίωσης**, αρκεί ο χρήστης να πατήσει σε οποιαδήποτε στιγμή το πλήκτρο "p" (σε κεφαλαία ή πεζά γράμματα) του λατινικού αλφαβήτου. Στη συνέχεια μπορεί είτε να συνεχίσει την ροή της προσομοίωσης ξανά πατώντας το πλήκτρο "C" (ή "c"), είτε να εισέλθει στο μενού που του προσφέρει πληθώρα δυνατοτήτων για να διαχειριστεί την προσομοίωση, πατώντας το πλήκτρο "M" (ή "m").

Εάν επιλέξει να εισέλθει στο **μενού**, τότε αφού σβήσει (πλήκτρο "backspace") όλους τους χαρακτήρες που πιθανόν να έχουν μείνει στο τερματικό του απο προηγούμενες ενέργειες, μπορεί να επιλέξει την προς εκτέλεση ενέργεια πληκτρολογώντας το σχετικό αριθμό και πιέζοντας "Enter". Οι διαθέσιμες ενέργειες είναι :

0. Συνέχιση της προσομοίωσης.
1. Προβολή πληροφοριών για κάποιο συγκεκριμένο πλοίο, αφού επιλέξει τον τύπο και το μοναδικό ID του πλοίου που τον ενδιαφέρει.
2. Προβολή πληροφοριών για κάποιο συγκεκριμένο σημείο στο χάρτη, αφού πληκτρολογήσει και τις συντεταγμένες αυτού.
3. Προβολή στατιστικών στοιχείων που αφορούν συνολικά ένα συγκεκριμένο είδος πλοίων, το οποίο θα επιλέξει.

4. Προβολή του συνόλου των πλοίων της προσομοίωσης, διατεταγμένων σε φθίνουσα σειρά με βάση τον θησαυρό ή την τρέχουσα αντοχή ή την ταχύτητα τους.
5. Προσθήκη ή αφαίρεση ενός πλοίου στην προσομοίωση σε τυχαίο ή συγκεκριμένο σημείο του χάρτη.
6. Προσθήκη ή αφαίρεση ενός λιμανιού ή ενός σημείου θησαυρού σε τυχαίο ή συγκεκριμένο σημείο του χάρτη.
7. Τέλος της προσομοίωσης.

Τέλος, κατά την υλοποίηση της εφαρμογής έγιναν οι παρακάτω **παραδοχές** :

- ✓ Έχουμε θεωρήσει ότι τα όρια του χάρτη είναι αυστηρά, δηλαδή το δεξί άκρο δεν ενώνεται με το αριστερό άκρο, αντίστοιχα και το πάνω με το κάτω άκρο. Κατά την δημιουργία του χάρτη αρχικοποιούμε το είδος κάθε σημείου με τυχαιότητα. Συγκεκριμένα κάθε σημείο αρχικοποιείται με πιθανότητα 40% ως απλή θάλασσα, με πιθανότητα 20% ως λιμάνι και με πιθανότητα 40% ως θησαυρός.
- ✓ Έντονες καιρικές συνθήκες θεωρούμε ότι υφίστανται σε κάποιο σημείο όταν η ένταση καιρού του σημείου είναι μεγαλύτερη ή ίση του 7 (max : 10). Από την τιμή αυτή και πάνω τα πλοία παθαίνουν ζημιά (ανάλογα με το μέγεθος της κακοκαιρίας).
- ✓ Οι δύο κλάσεις που ζητείται να δημιουργήσουμε είναι οι Passenger και Tanker οι οποίες είναι αμφοτέρως υποκλάσεις της Commercial. Η διαφορά μεταξύ αυτών των κλάσεων βρίσκεται ποσότητα θησαυρού που συλλέγουν σε κάθε γύρο, όταν εκτελούν τη λειτουργία (ShipFunc) τους.
- ✓ Η κατεύθυνση που ακολουθείται στη μετακίνηση γίνεται με τον εξής τρόπο: Παράγεται ένας αριθμός τυχαία και παίρνουμε το ακέραιο υπόλοιπό του με το 4. Το κάθε πλοίο λοιπόν δοκιμάζει να πάει προς την κατεύθυνση που αντιστοιχεί σε αυτό το νούμερο. Αν δεν τα καταφέρει τότε δοκιμάζει τις υπόλοιπες κατευθύνσεις κυκλικά με τη φορά του ρολογιού. Αν δεν μπορεί να πάει πουθενά με την ταχύτητα που έχει, τότε μένει στη θέση του και δοκιμάζει ξανά στον επόμενο γύρο. Έτσι δεν δίνεται προτεραιότητα πάντα σε μία από τις τέσσερις κατευθύνσεις (0: πάνω, 1: δεξιά, 2: κάτω, 3: αριστερά).
- ✓ Έχουμε θεωρήσει ότι όλα τα πλοία του ίδιου είδους έχουν την ίδια ποσότητα θησαυρού -. Ο ορισμός αυτών των ποσοτήτων βρίσκεται στην αρχή των αντίστοιχων header αρχείων σε εντολές #define.
- ✓ Τα πλοία εξερεύνησης όταν βρεθούν δίπλα σε πειρατικό πλοίο ή σε σημείο με έντονη κακοκαιρία προσπαθούν πρώτα να πάνε στην αντίθετη κατεύθυνση. Αν για κάποιο λόγο δεν τα καταφέρουν δοκιμάζουν τις άλλες 2 κατευθύνσεις με την σειρά που υποδεικνύει η φορά των δεικτών του ρολογιού π.χ. αν ο κίνδυνος είναι πάνω, δοκιμάζει πρώτα κάτω, μετά δεξιά και τέλος αριστερά.
- ✓ Ο θησαυρός στην κλάση GridPoint είναι τύπου bool ενώ στην κλάση Ship είναι τύπου double. Συγκεκριμένα θεωρούμε ότι στον χάρτη κάθε σημείο είτε έχει 1 μονάδα θησαυρού είτε δεν έχει θησαυρό. Όταν ένα πλοίο ανακτά θησαυρό από την θάλασσα τότε αυξάνει τον θησαυρό του κατά 1 και το σημείο αυτό παύει να έχει θησαυρό. Όταν ένα πλοίο για οποιονδήποτε από τους λόγους που αναφέρονται στην εκφώνηση της εργασίας πάρει θησαυρό από κάποιο άλλο πλοίο τότε παίρνει ποσοστό του συνολικού θησαυρού του. Όταν όμως ένα πλοίο βυθιστεί ανεξάρτητα από το πόσο θησαυρό διαθέτει το σημείο στο οποίο βυθίστηκε αποκτά 1 μονάδα θησαυρού.