Χειμερινό Εξάμηνο 2018-2019

**ΟΜΑΔΑ:**

Σφακιωτάκης Σακελλάρης τπ4514

Αλεξανδρής Κωσταντίνος τπ4479

**ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ**

**1**

**Περιεχόμενα**

1. Δομή κώδικα παιχνιδιού…………………………………..3
2. Περιγραφή κώδικα……………………………………………4
3. Design Patterns……………………………………………….10
4. UML Diagrams…………………………………………………11

**2**

**Δομή κώδικα παιχνιδιού**

Πακέτα:

* GameGUI

1. playingBoardFrame.java

* PlayerInfoGUI

1. InfoFrame.java

* ShipPlacementGUI

1. BoardPanel.java
2. BottomPanel.java
3. ShipPanel.java
4. ShipPlacementFrame.java
5. TopPanel.java

* Ships

1. Aircraft.java
2. Battle.java
3. Destroyer.java
4. Patrol.java
5. Ship.java(interface)
6. ShipFactory.java
7. Submarine.java

* battleShipgame

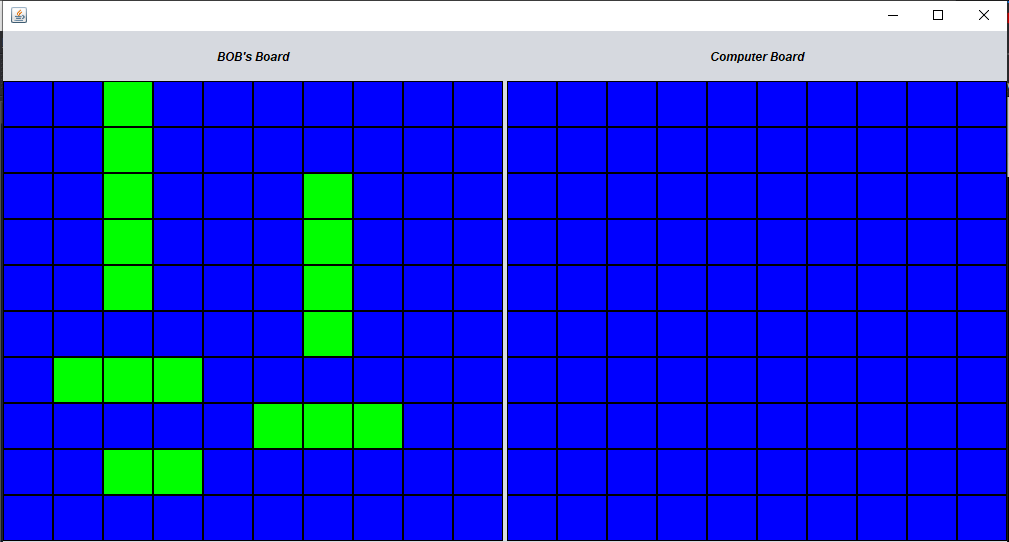
1. BattleshipGame.java(main)

**3**

**Περιγραφή Αρχείων**

* GameGUI

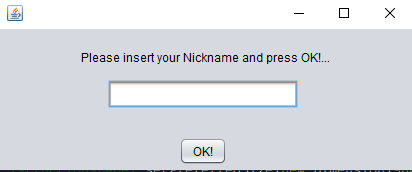
-Το αρχείο κώδικα playingBoardFrame.java είναι υπεύθυνο για το τελευταίο κομμάτι του παιχνιδιού το οποίο παίρνει σαν ορίσματα τους δύο πίνακες με την πληροφορία από το πεδίο που έχουν τοποθετηθεί τα πλοία του χρήστη αλλά και του υπολογιστή. Στην συνέχεια κατασκευάζει το frame και ξεκινάει η μάχη του χρήστη με τον υπολογιστή. Επίσης το αρχείο αυτό διαθέτη την λειτουργία AI για τον υπολογιστή ώστε να μην δουλεύει τυχαία τέλος αναδεικνύει τον νικητή της ναυμαχίας.



* PlayerInfoGUI

-Το αρχείο InfoFrame.java αποτελεί το frame το οποίο εμφανίζεται στην αρχή του παιχνιδιού και ζητάει από τον χρήστη να του δώσει ένα όνομα.

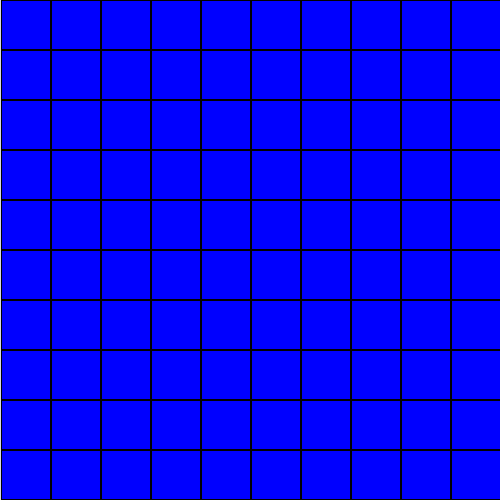
**4**



* ShipPlacementGUI

Σε αυτό το πακέτο γίνεται η κατασκευή του frame της τοποθέτησης των πλοίων.

-Το αρχείο BoardPanel.java είναι αυτό που δημιούργει το πεδίο τοποθέτησης του κάθε πλοίου. Επίσης δίνει την δυνατότητα στον χρήστη να τοποθετήσει τα πλοία του όπου θέλει μέσα στο πεδίο εφόσον μπορεί να τοποθετηθεί στο σημείο που επιθυμεί πράγμα που του το κάνει εμφανές. Δίνει την δυνατότητα αλλαγής θέσης μετά την τοποθέτηση του πλοίου αν αλλάξει γνώμη ο χρήστης. Τέλος ακόμα δημιουργεί τον πίνακα με τα πλοία του υπολογιστή.



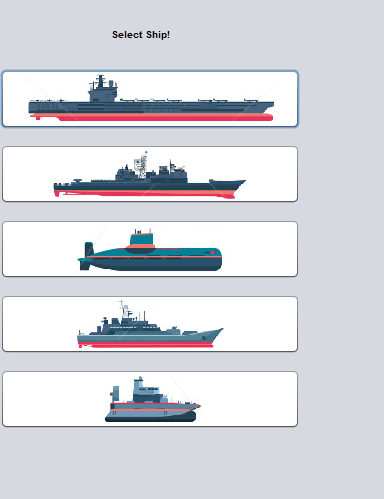
**5**

-Το αρχείο BottomPanel.java είναι το κάτω κομμάτι του frame που γίνεται η τοποθέτηση των πλοίων . Διαθέτει δύο κουμπιά το ένα για την αλλαγή κατεύθυνσης του πλοίου και ένα για να πάει το παιχνίδι στο τελικό frame δηλαδή αρχίσει η ναυμαχία. Το κουμπί είναι αρχικά ανενεργό και στην συνέχεια αν ο χρήστης τοποθετήσει τότε όλα τα πλοία τότε ενεργοποιείτε.



-Το αρχείο ShipPanel.java αποτελεί το μέρος όπου ο χρήστης επιλέγει ποιο πλοίο θέλει να τοποθετήσει. Ουσιαστικά σε αυτή το αρχείο δημιουργούνται τα πλοία και με την επιλογή ανάλογου κουμπιού δίνει την δυνατότητα επιλογής πλοίου από τον χρήστη ώστε να το τοποθετήσει στο πεδίο που τοποθετούνται στο διπλανό πάνελ .Επίσης δίνει την δυνατότητα όταν ο χρήστης τοποθετήσει ένα πλοίο να απενεργοποιηθεί η δυνατότητα επιλογής του έτσι ώστε να μην μπορεί να γίνει πολλαπλή τοποθέτηση του.

**6**

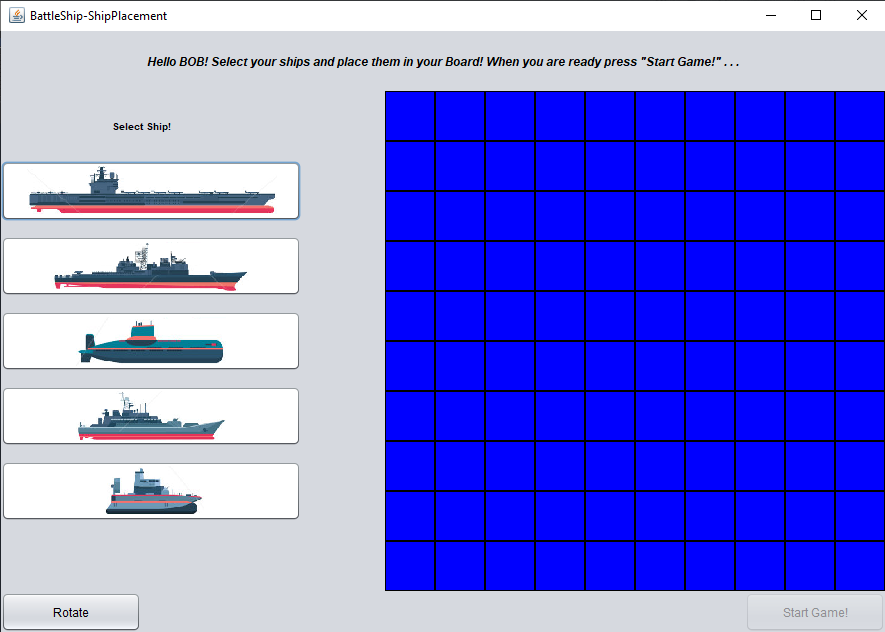


-Το αρχείο ΤopPanel.java είναι απλά ένα Panel το οποίο βρίσκετε στην κορυφή του frame και είναι καλωσορίζει τον χρήστη σε αυτό το frame με το όνομα του.



-Το αρχείο ShipPlacementFrame.java αποτελεί το αρχείο όπου δημιουργείται το frame στο οποίο ο χρήστης επιλέγει και τοποθετεί τα πλοία πριν αρχίσει την ναυμαχία. Εδώ δημιουργούνται όλα τα στιγμιότυπα τον panel που του ανήκουν.

**7**



* Ships

-Το αρχείο Ships.java αποτελεί το συμβόλαιο στο οποίο θα βασιστούν τα άλλα πλοία και έτσι όλα τα πλοία που το κάνουν implement να υλοποιήσουν κάποια συγκεκριμένα χαρακτηριστικά που πρέπει να έχουν όλες την κάνουν Implement.

-Τα αρχεία Aircraft.java, Battle.java, Destroyer.java, Patrol.java και Submarine.java αποτελούν τα αρχεία τα οποία η κλάση του κάνει implement το interface Ship και υλοποιούν ότι τους υποχρεώνει και τοποθετούν μία εικόνα στα κουμπιά επιλογής του πλοίου.

-Το αρχείο ShipFactory.java είναι αυτό το οποίο με βάση το όρισμα που παίρνει επιστρέφει το αντίστοιχό αντικείμενο.

**8**

* Battleshipgame

-Το αρχείο BattleshipGame.java αποτελεί το αρχείο το οποίο έχει την θέση της main. Η οποία η λειτουργία είναι να δημιουργήσει του παιχνιδιού και να το κάνει ορατό.

**9**

**Design Patterns**

Σε αυτή την εργασία χρησιμοποιούνται δυο ειδών Design Patterns τα οποία είναι τα εξής:

**Singleton:**

Στην κλάση InfoFrame.java το χρησιμοποιούμε λόγο της απλότητας. Το συγκεκριμένο frame δημιουργείται μόνο για να παίρνει δεδομένα από τον χρήστη και να τα δίνει στο πρόγραμμα και μετά καταστρέφεται. Οπότε αυτό το απλό Design Pattern μας εξυπηρετεί λόγο της λιγοστών λειτουργειών του.

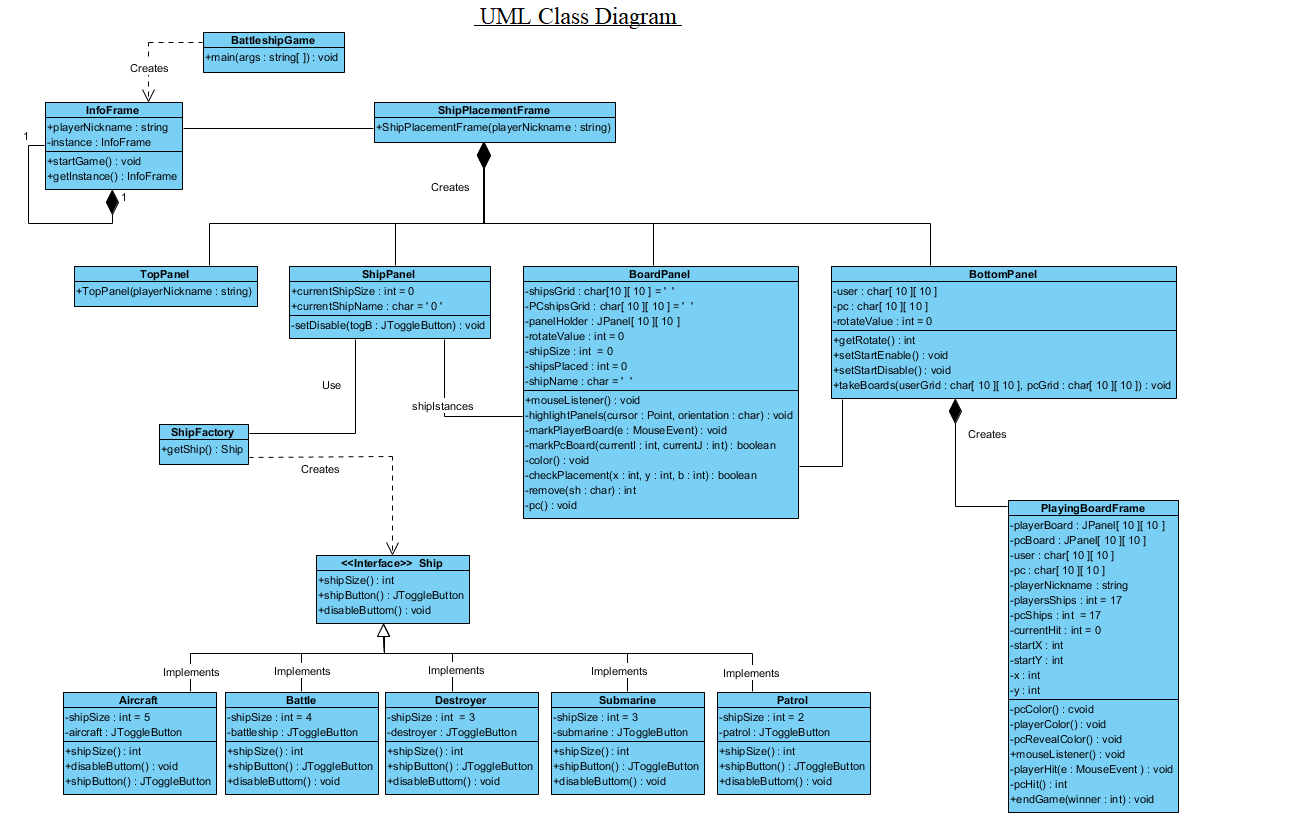
**Factory:**

Στο πακέτο Ships χρησιμοποιούμε το factory pattern λόγο των πολλαπλών κλάσεων τύπου ship. Πράγμα που μας δίνει την δυνατότητα να δημιουργούμε ποιο εύκολα τα αντικείμενα και επίσης μας δίνει την δυνατότητα να τα επεξεργαστούμε και γίνει η αλλαγή όπου και να χρησιμοποιείται. Τέλος μας δίνει την ευελιξία να μην ασχολείται η κλάση που τα χρησιμοποιεί με την δημιουργεί αλλά απλά να ζητάει και να δέχεται το αντικείμενο.

**10**

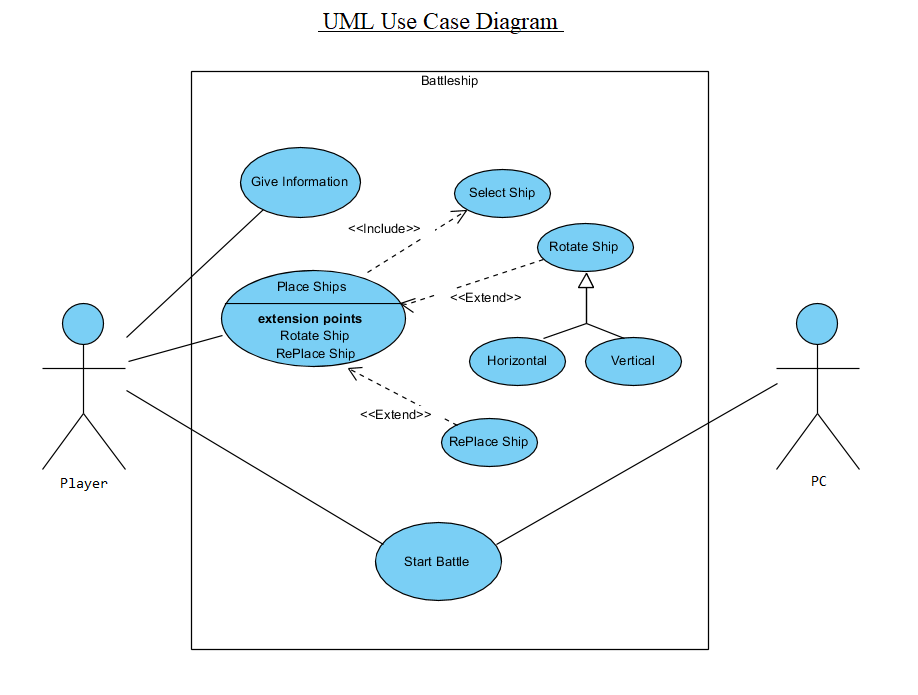
**UML Diagrams**

UML Class Diagram:



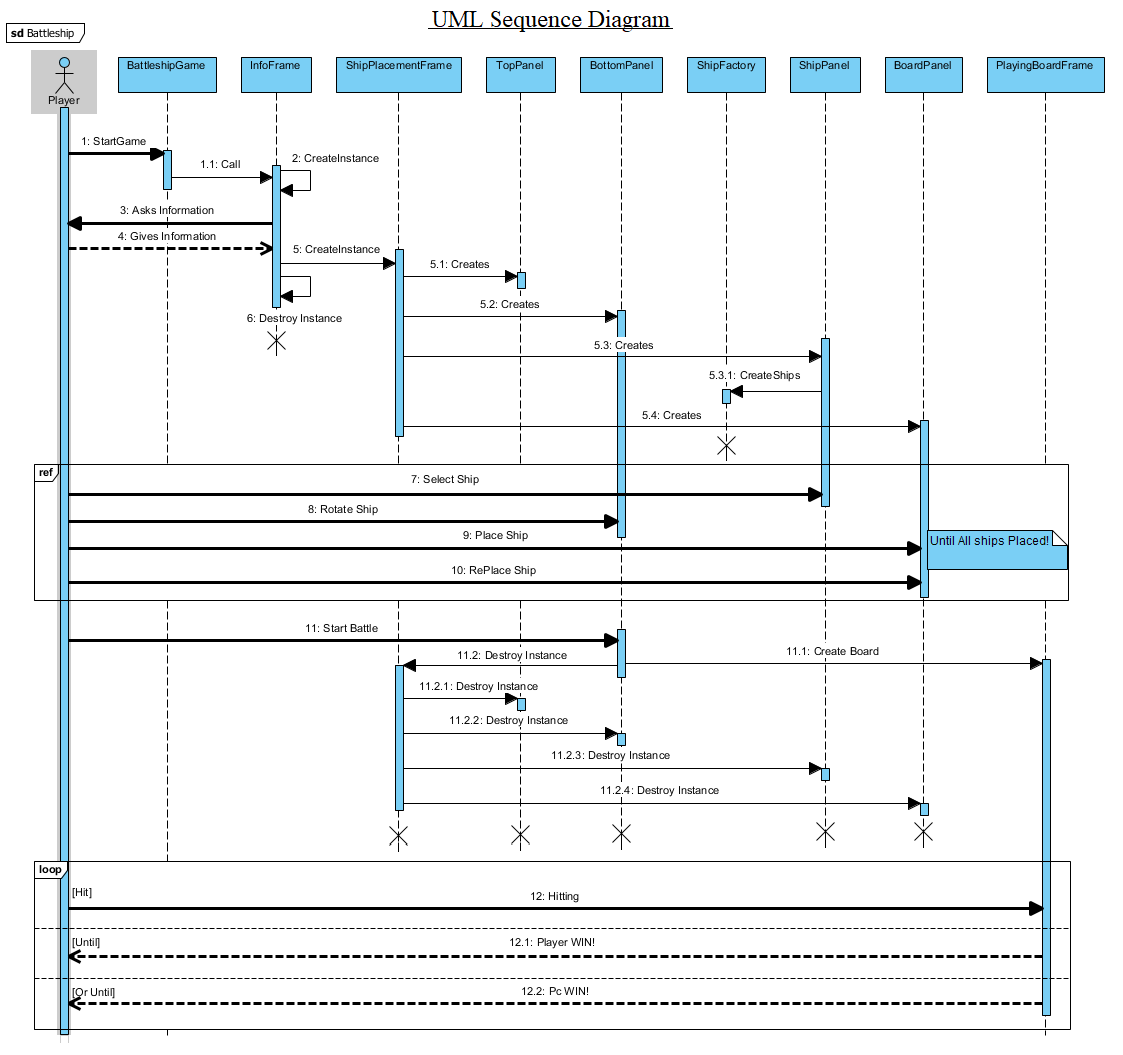
**11**

UML Case Diagram:



**12**

UML Sequence Diagram:



**13**