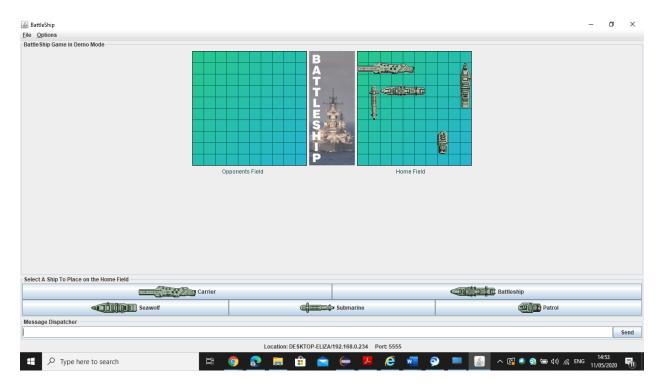# Battleship Game

Name: Ioana-Eliza Georgescu
Student Number: ST20131753
BSc (Hons) Computer Science
Second Assignment for Advanced Programming

This is an interactive game created in Java, called Battleship. This game is a two-player game, using client server.

Running (BattleshipGameJava is the right file with the right executable game.) – The Complex Project is another Java Battleship Game, but couldn't manage to make it work, but it has the "Play with PC" mode.

To run the game, the user should execute the ant file, otherwise the file could be run as a Java Application in Java's console.
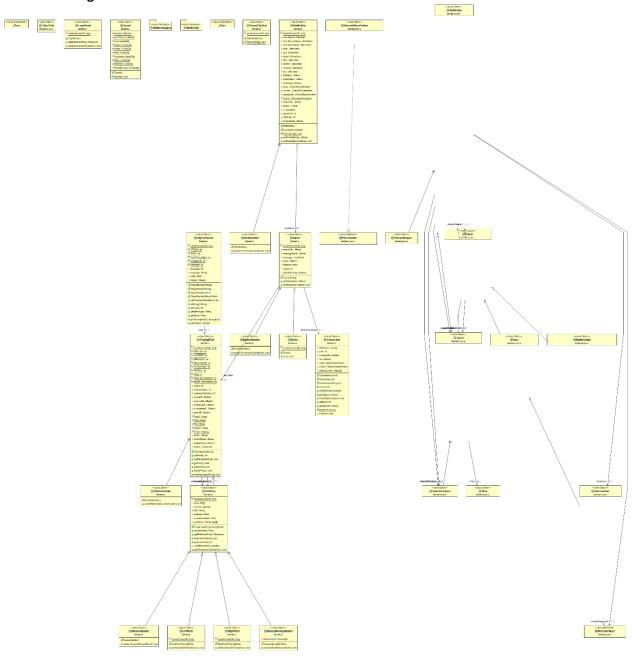

Additional Information

The game presents:
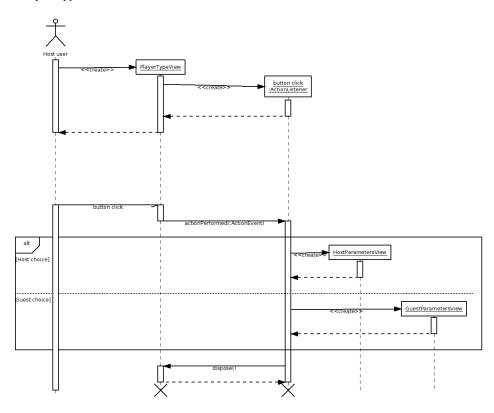- Graphics;
- Sound;
- Easy Access and friendly interface;

All the UML Diagrams can be very well observed by zooming them.

UML Case Diagram

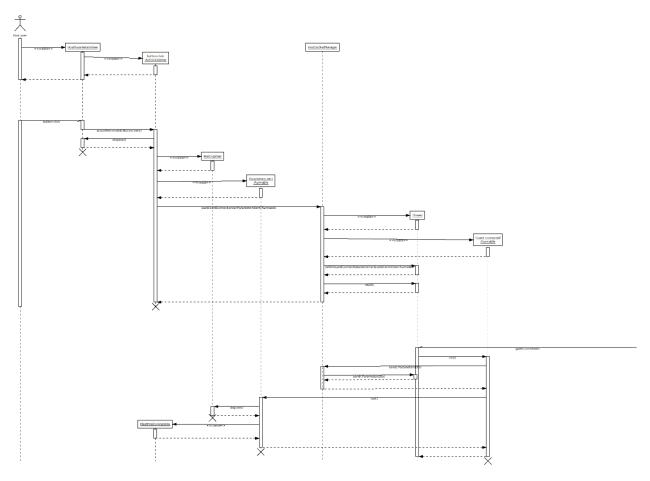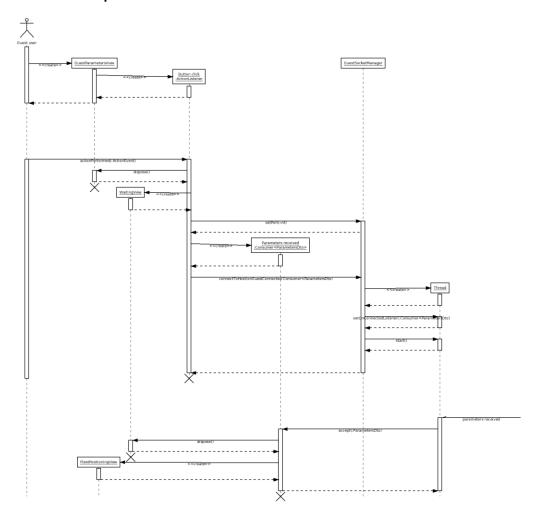# UML Sequence Diagrams for Battleship Game in Java

Player type

Parameters

Host parameters

- ▪ *Write the server's parameters;*

- ▪ *Create and run the server;*

- ▪ *Waiting guest;*

- ▪ *Sending game parameters;*

- ▪ *View Display;*

Guest parameters

- ✓ *Writing server parameters;*
- ✓ *Reception of game parameters;*
- ✓ *View Display;*
- ✓ *Fleet positioning*

Display

Host user

FleetPositioningView

Positions received
:Consumer<PositionsDto>

SocketManager

Thread

<<create>>

setOnPositionsReceivedListener(:Consumer<PositionsDto>)

setOnPositionsReceivedListener(:Consumer<PositionsDto>)

all boats placed

send(:PositionsDto)

send(:PositionsDto)

alt

[positions received]

<<create>>

WaitingView

<<create>>

GameView

parameters received

accept(:PositionsDto)

alt

[waiting view displayed]

dispose()

GameView

<<create>>