

Образовательный центр Motorola в СПбГУ**Высокоуровневое описание
проекта
«BattleShips»****Версия D01.00.00.**

Документ подготовлен:			
Имя:	А. Егоров И. Кривченко	Адрес электронной почты:	andreyegorov@pisem.net marsiana_k@mail.ru
Дата:		14 July 2003	
Документ распечатан из файла:		C:\hld_d01_00_00.doc	

История разработки.

Версия	Описание	Дата	Автор
D00.01.00	Начальная версия документа.	04 июля 2003 г.	А. Егоров
D00.02.00	Добавлено описание интерфейсов для стратегии.	04 июля 2003 г.	И. Кривченко
D00.03.00	Добавлено описание интерфейсов для симулятора.	05 июля 2003 г.	А. Егоров
D01.00.00	Первая версия (черновая) документа.	07 июля 2003 г.	А. Егоров И. Кривченко

Содержание.

1. Введение.....	7
1.1 Цель.....	7
1.2 Сокращения.....	7
1.3 Соглашения.....	7
1.4 Ссылки.....	7
2. Описание архитектуры проекта.....	8
2.1 Общая структура проекта.....	8
2.2 Диаграмма блоков.....	9
2.2.1 Блок симулятора (Simulator).....	9
2.2.2 Блок стратегий (Strategy).....	10
2.3 Описание интерфейсов.....	10
2.3.1 Поведение симулятора.....	10
2.3.2 Поведение стратегии.....	19
2.4 Основные алгоритмы.....	22
2.4.1 Центр масс кораблей.....	22
2.4.2 Движение кораблей при отсутствии противника.....	22
2.4.3 Область целевой точки.....	22
2.4.4 Координаты точек обстрела.....	22
2.4.5 Угол между векторами.....	23
2.4.6 Положение корабля при изменении скорости.....	23
2.4.7 Изменение положение снаряда.....	23

1. Введение.

1.1 Цель.

Этот документ описывает архитектуру программного проекта «BattleShips». Данный документ предназначен непосредственно для разработчиков программного продукта «BattleShips».

Документ содержит все уровни описания проекта: описание на уровне системы, описание на уровне блоков и описание на уровне интерфейсов между блоками.

1.2 Сокращения.

GUI.....	Graphical User Interface
MSC	Message Sequence Chart (ITU-T z.120)
SDL	Specification and Description Language (ITU-T z.100)

1.3 Соглашения.

- «Стратегия» - модуль проекта, управляющий поведением ему принадлежащих кораблей.
- «Симулятор» - модуль, отвечающий за прием и отправку сигналов стратегиям, синхронизацию информации о кораблях каждого флота, управление синхронизацией процесса отображения текущего положения кораблей и выстрелов на графическом интерфейсе.
- «Клиент» - рабочая станция или терминал, на котором запускается GUI.
- «Сервер» - рабочая станция или терминал, на котором запускается непосредственно симулятор со стратегиями.
- Термины «сообщения» и «сигнал» обозначают одно и то же.
- «Флот» - группа кораблей, принадлежащих одной стратегии.
- «Мель» - глубина, меньшая посадки корабля, но не суша.
- «Так проверки» - такт симуляции, на котором проверяется возможность окончания процесса симуляции.
- «Поле видимости» - участок карты, который попадает в зону обзора, по крайней мере, одного корабля данного флота.
- «Коллизия» - возможность столкновения кораблей на следующем такте симуляции или посадка корабля на мель через 2 и менее тактов симуляции.

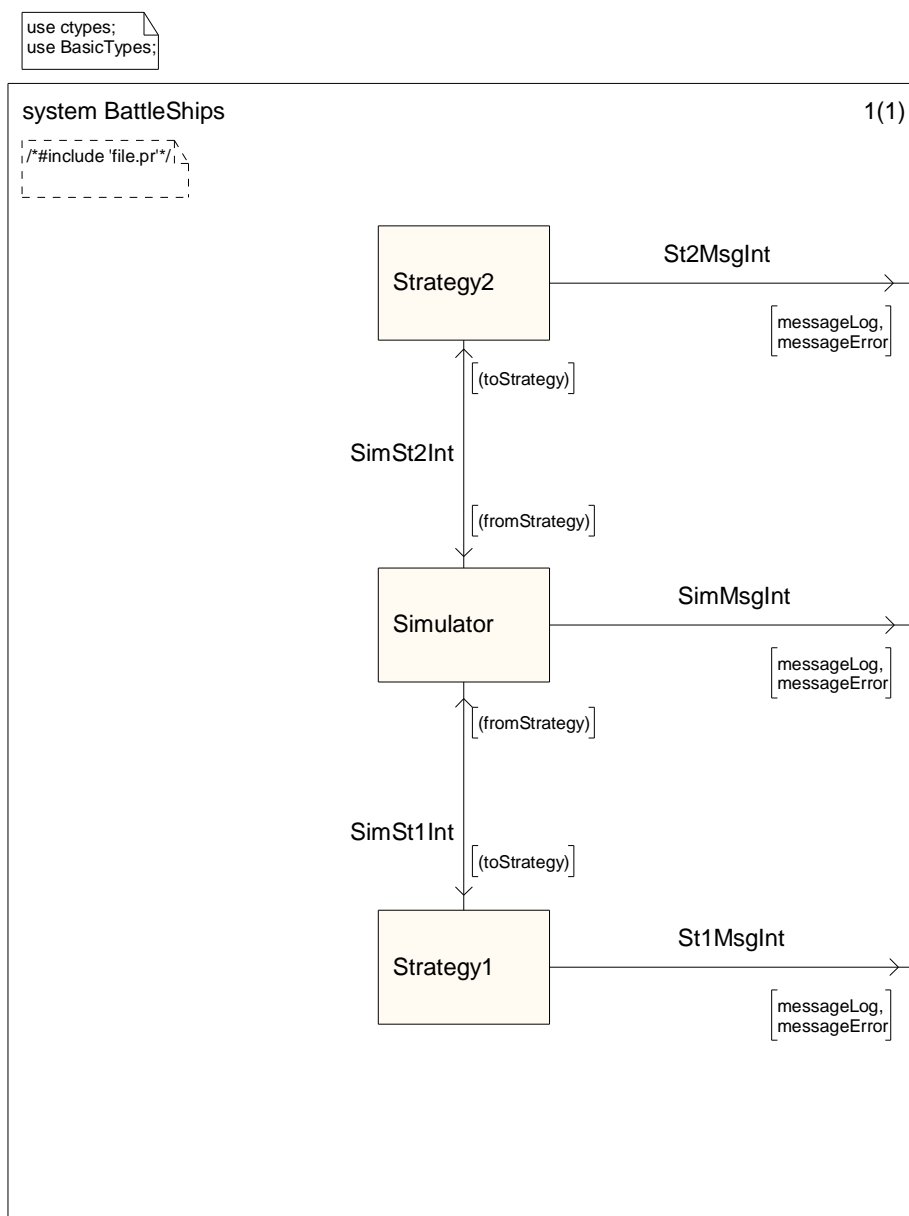
1.4 Ссылки.

- [1] Функциональная спецификация проекта «BattleShips».

2. Описание архитектуры проекта.

2.1 Общая структура проекта.

В проекте BattleShips можно выделить три основные части: блок симулятора (Simulator), блоки стратегий (Strategy1 и Strategy2) и графический интерфейс (GUI). Блоки симулятора и стратегий необходимо реализовать на языке спецификаций SDL, а графический интерфейс необходимо реализовать на объектно-ориентированном языке программирования высокого уровня JAVA. Общая структура проекта выглядит следующим образом:



Взаимодействие между блоком симулятора и блоками стратегий происходит с помощью списков сигналов, описанных в п. В.8 документа [1], по каналам SimSt1Int и SimSt2Int соответственно с первой и второй стратегиями. Кроме того, эти блоки имеют возможность отправлять сообщения messageLog и messageError (см. п. В.7 документа [1]) в «окружающую среду» по каналам SimMsgInt, St1MsgInt и St2MsgInt соответственно для симулятора, стратегии 1 и стратегии 2 – это необходимо для возможности ведения журнала событий и вывода любой отладочной информации.

Взаимодействие с графическим интерфейсом блок симулятора осуществляет с помощью передачи информации по сокетах.

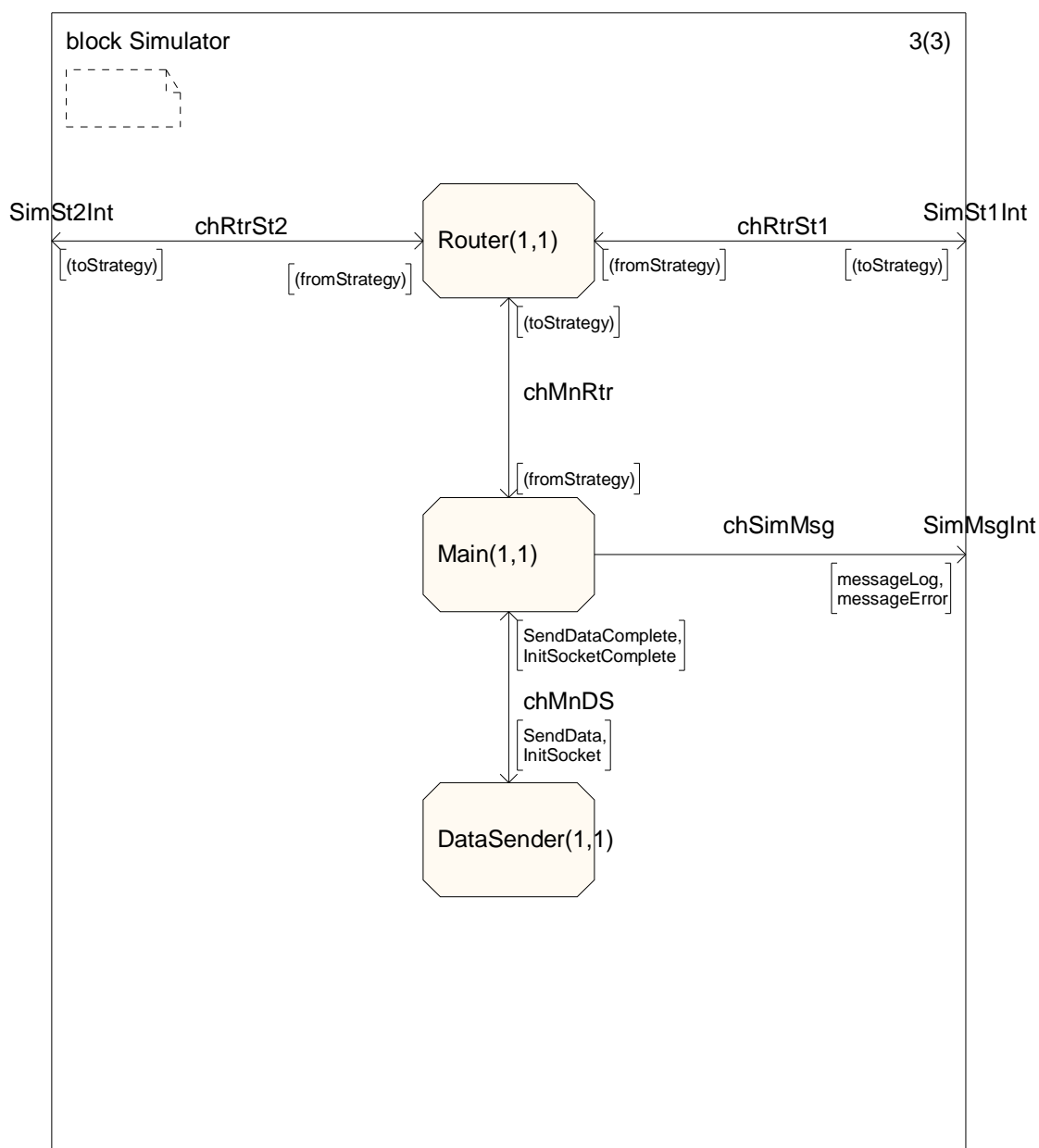
2.2 Диаграмма блоков.

2.2.1 Блок симулятора (Simulator).

Блок симулятора состоит из трех модулей:

- главный модуль – процесс Main; этот модуль отвечает за хранение и обработку данных обо всех флотах и выстрелах, прием сигналов от стратегий и отправку им ответов, управление процессом симуляции морского боя.
- коммуникационный модуль – процесс DataSender; этот модуль отвечает за процесс взаимодействия с графическим интерфейсом: установление соединения, отправка данных о карте, кораблях и выстрелах.
- коммуникационный модуль – процесс Router; этот модуль отвечает за прием сигналов от симулятора и стратегий и перенаправление их стратегиям и симулятору соответственно.

Структура блока симулятора приведена на рисунке ниже:

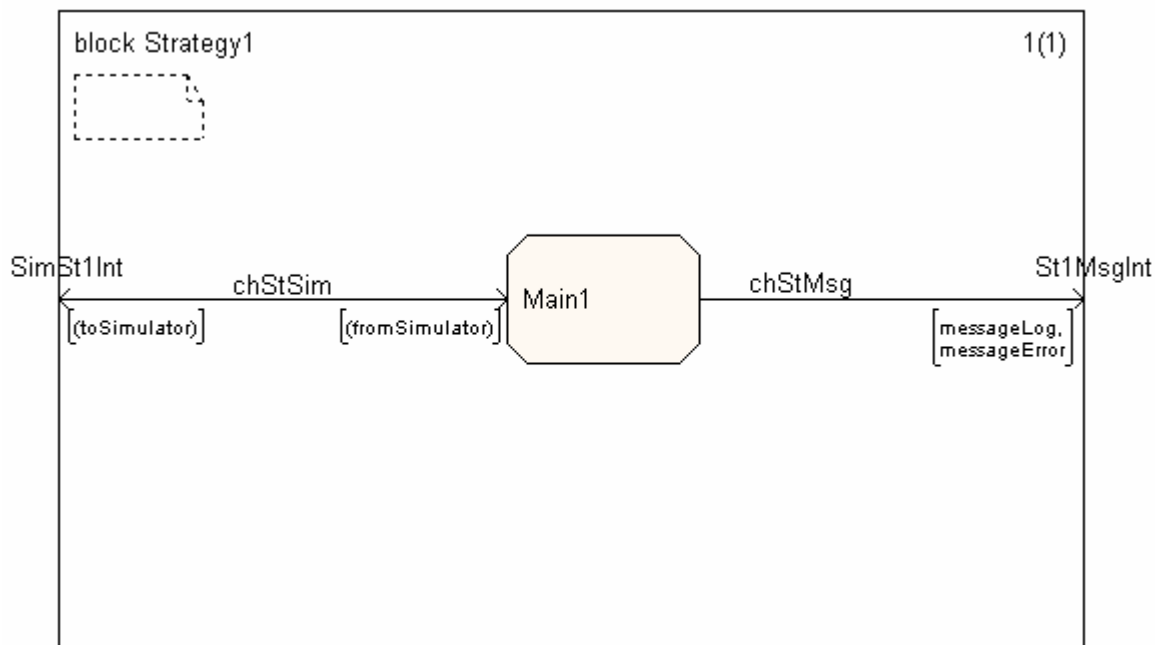


Из рисунка видно, что сигналы, поступающие от стратегий из каналов SimSt1Int и SimSt2Int, разводятся соответственно по путям chRtrSt1 и chRtrSt2 в процесс Router. Процесс Main посылает и принимает сигналы процессам Router и DataSender соответственно по путям chMnRtr и chMnDS.

Кроме того, процесс Main может посылать сигналы messageLog и messageError по пути chSimMsg в канал SimMsgInt.

2.2.2 Блок стратегий (Strategy).

Каждый из блоков стратегий состоит только из одного процесса, отвечающего за поведение стратегии во время симуляции морского боя, и его структура приведена ниже (рассмотрим на примере блока Strategy1 – для блока Strategy2 архитектура точно такая же, разница лишь в индексе: вместо «1» везде стоит индекс «2»):



Процесс Main1 получает сигналы от симулятора по пути chStSim через канал SimSt1Int и посылает сигналы messageLog и messageError в «окружающую среду» по пути chStMsg в канал St1MsgInt.

2.3 Описание интерфейсов.

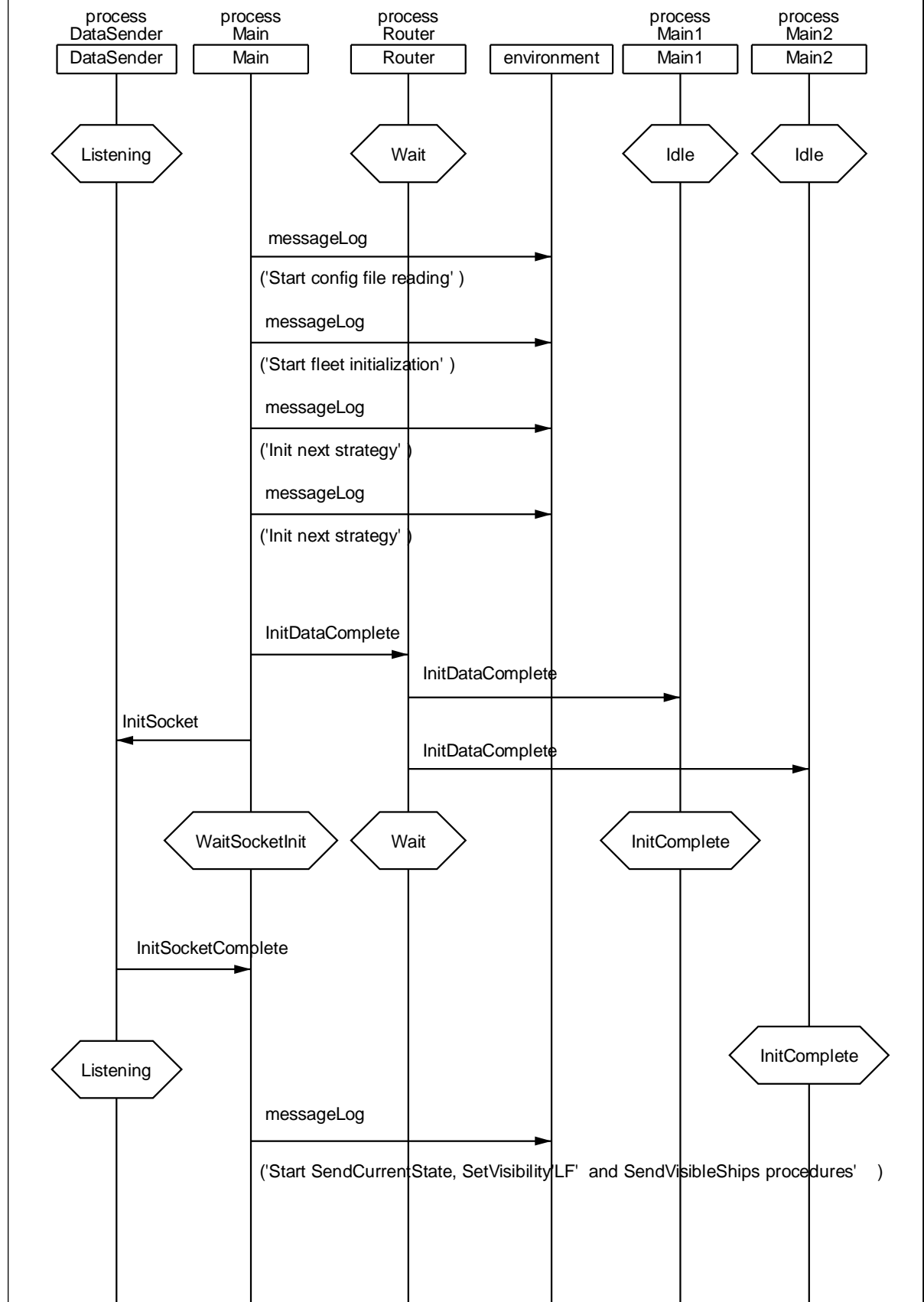
Эта секция описывает интерфейс взаимодействия стратегии с симулятором, а также интерфейс взаимодействия симулятора со стратегией и интерфейс между модулями внутри самого симулятора. Кроме того, описан протокол взаимодействия с GUI.

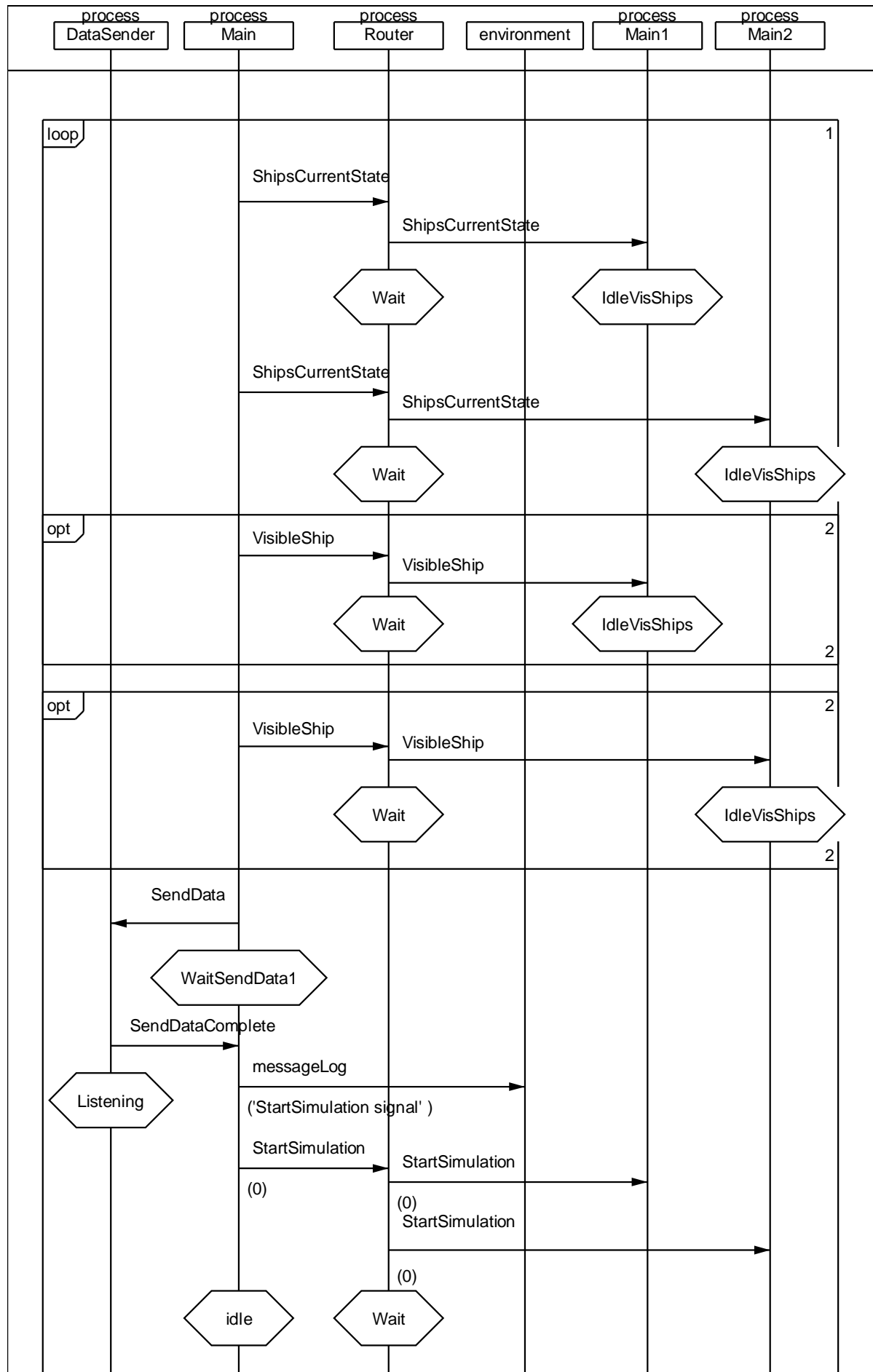
2.3.1 Поведение симулятора.

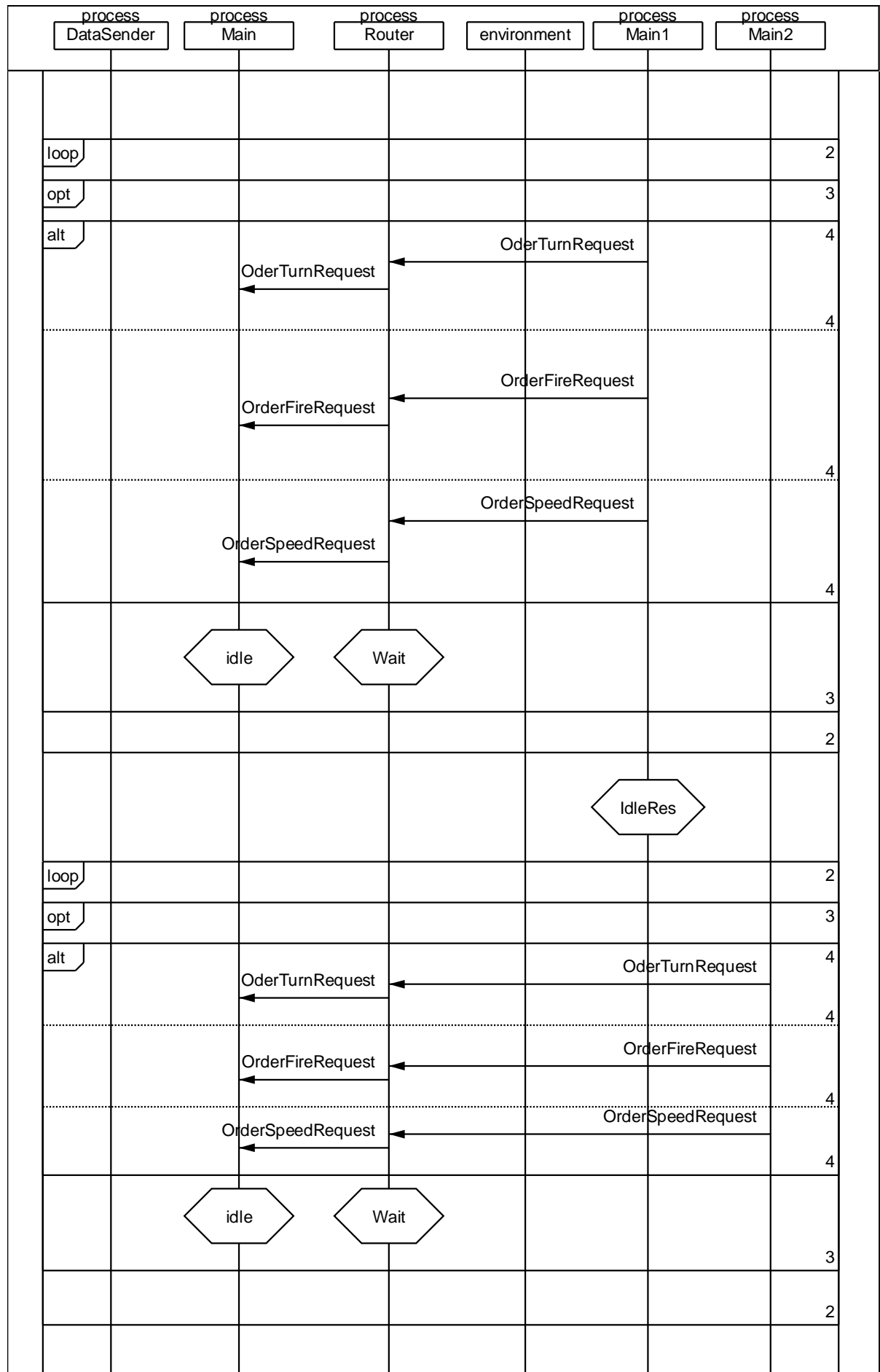
2.3.1.1 Взаимодействие симулятора со стратегиями.

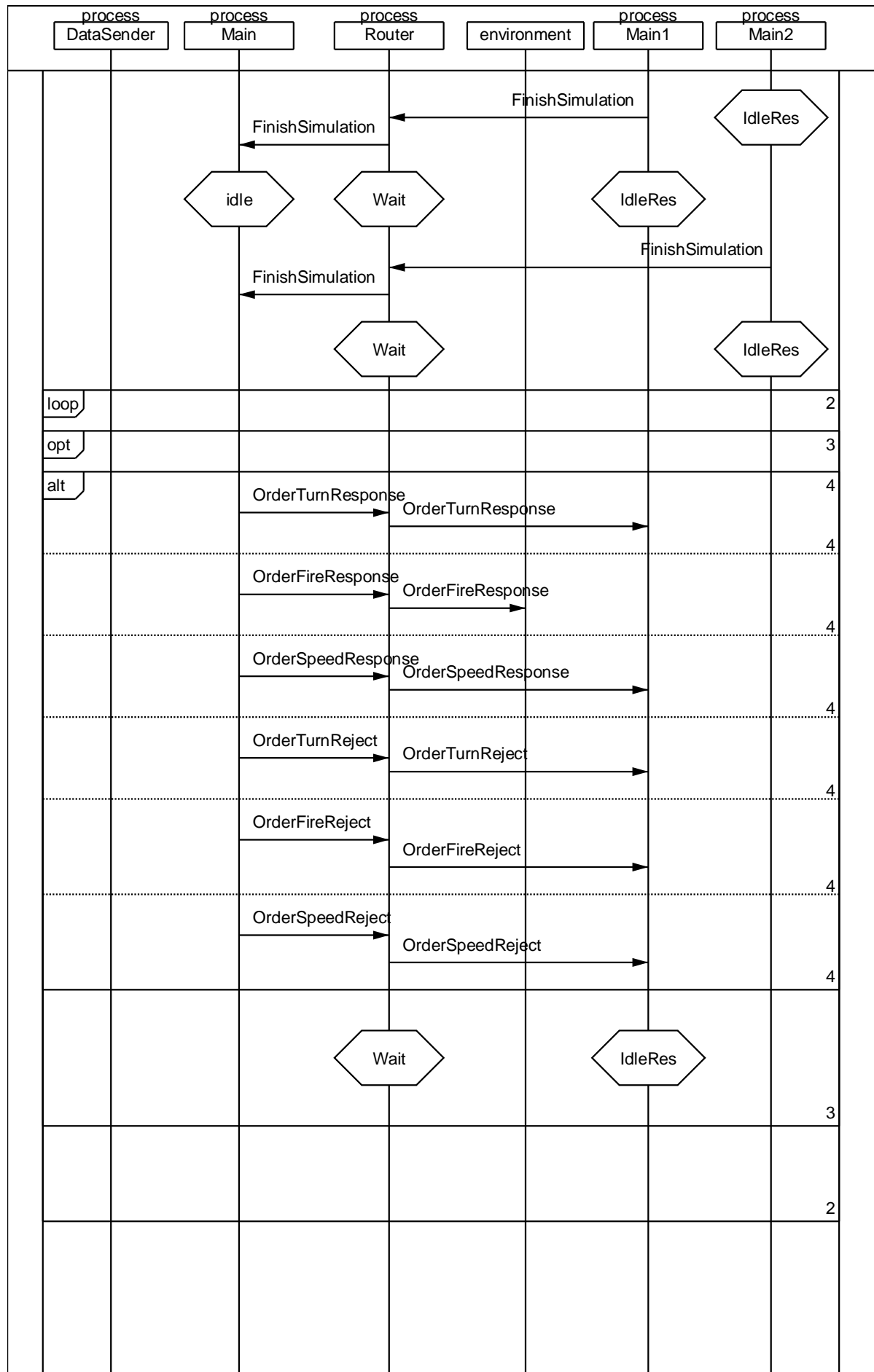
- 2.3.1.1.1 После успешного окончания инициализации данных по флотам и выстрелам (об этом могут свидетельствовать сообщения messageLog) симулятор должен отправить сигнал один InitDataComplete, который из процесса Router отправляется непосредственно уже каждой стратегии.
- 2.3.1.1.2 После того, как был отправлен стратегиям сигнал InitDataComplete, необходимо отправить сигнал InitSocket процессу DataSender. По окончании выполнения инициализации соединения с GUI, процесс DataSender должен отправить сообщение InitSocketComplete.
- 2.3.1.1.3 После этого симулятор должен отправить каждой из стратегий сигнал ShipsCurrentState.
- 2.3.1.1.4 Если есть вражеские корабли в области видимости каждой из стратегий, то этим стратегиям отправляется сигнал VisibleShip.
- 2.3.1.1.5 Затем процесс Main отправляет сигнал SendData процессу DataSender для возможности отправить данные GUI. По окончании все операций процесс DataSender высылает сигнал SendDataComplete.
- 2.3.1.1.6 Такт симуляции начинается с отправки одного сообщения StartSimulation, который отправляется уже каждой стратегии из процесса Router.

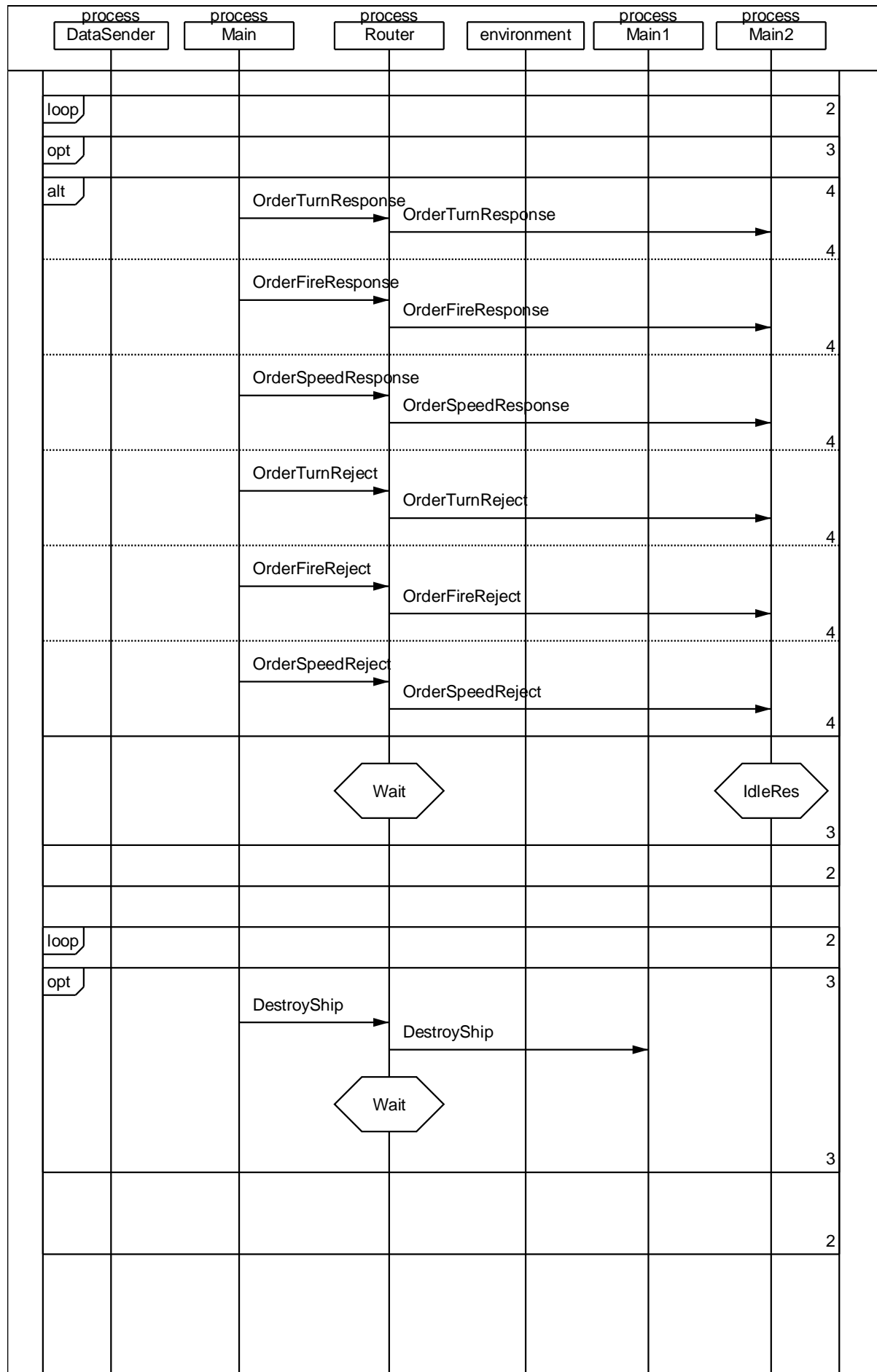
MSC Diagram

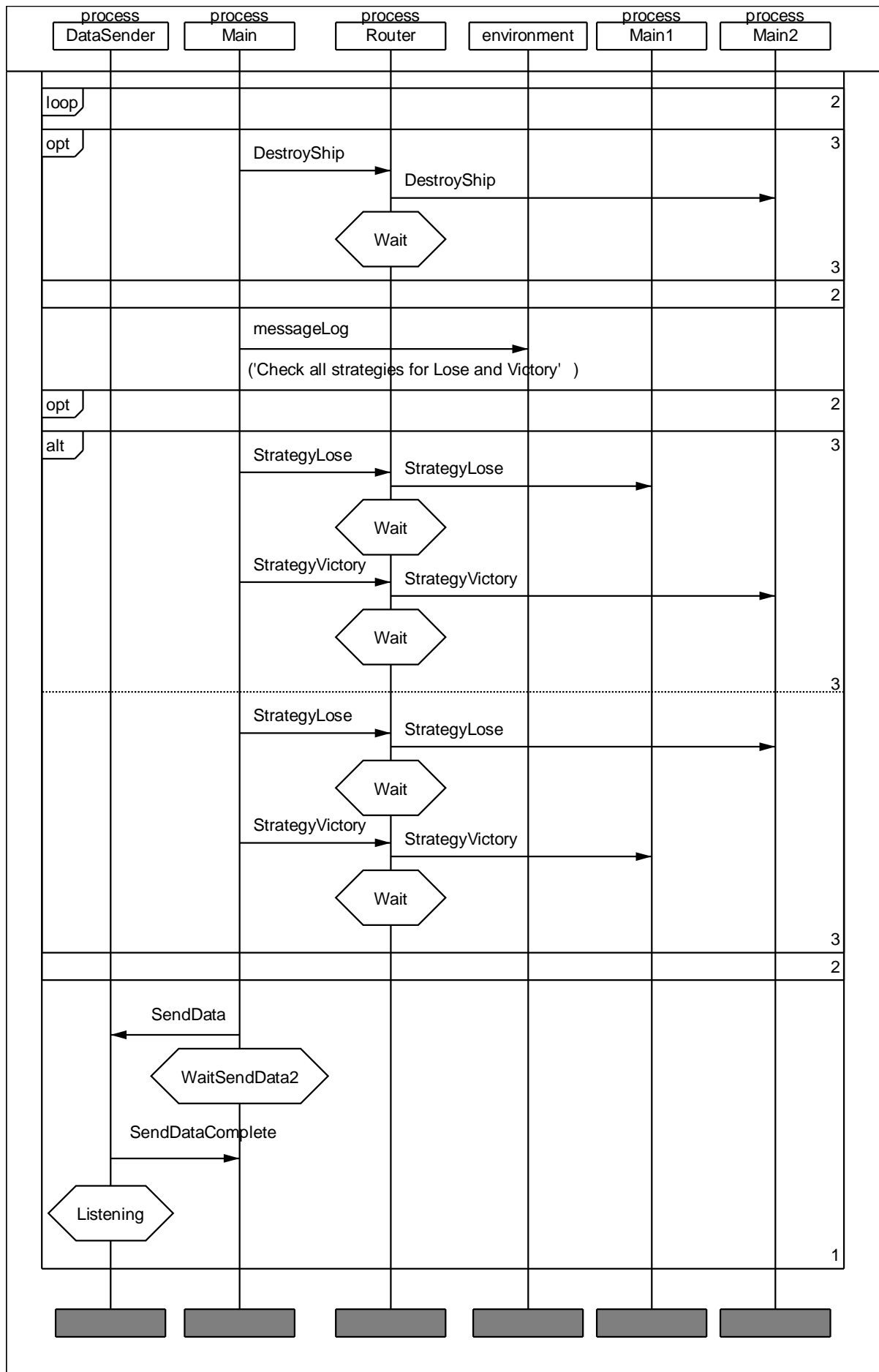






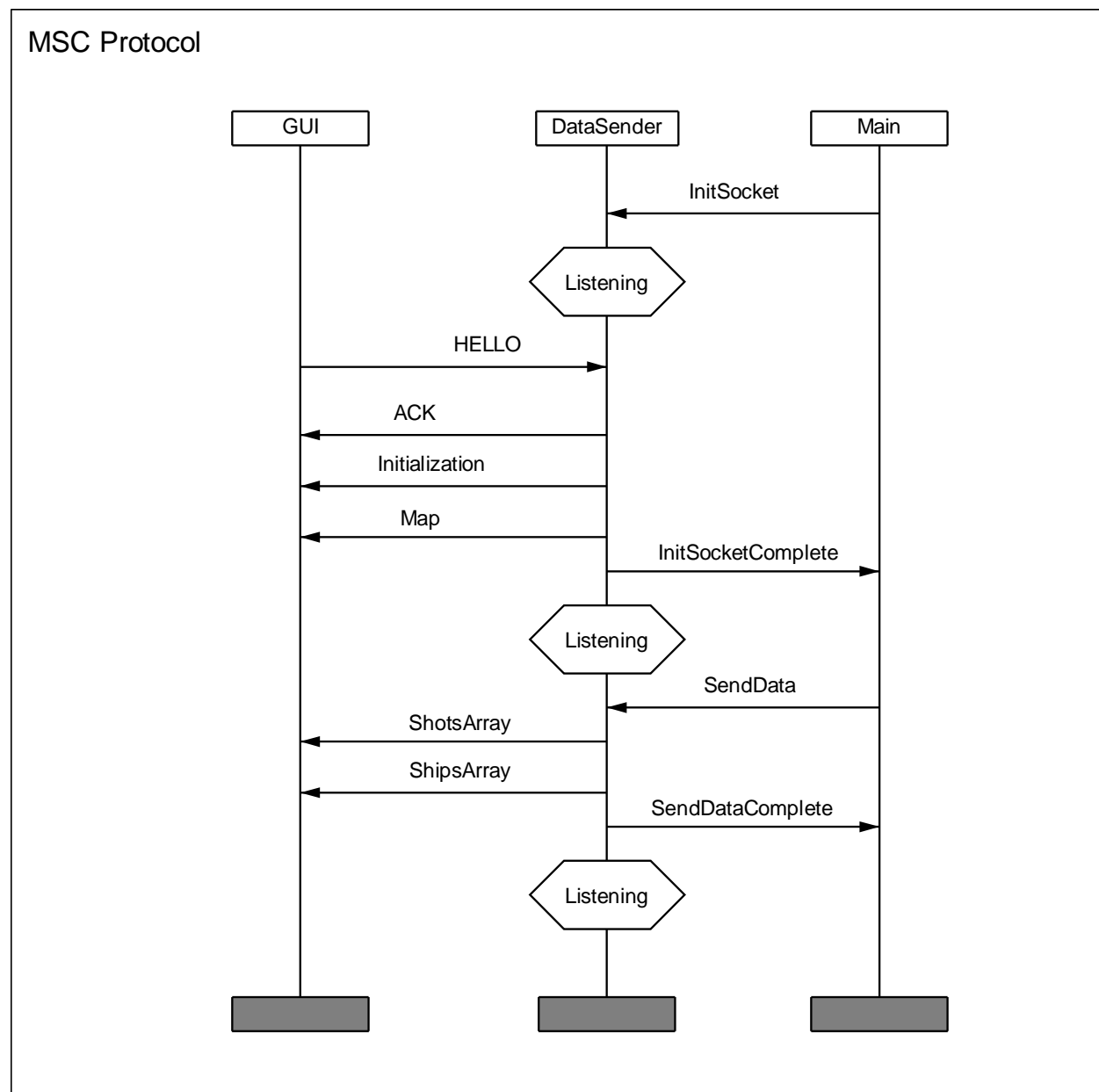






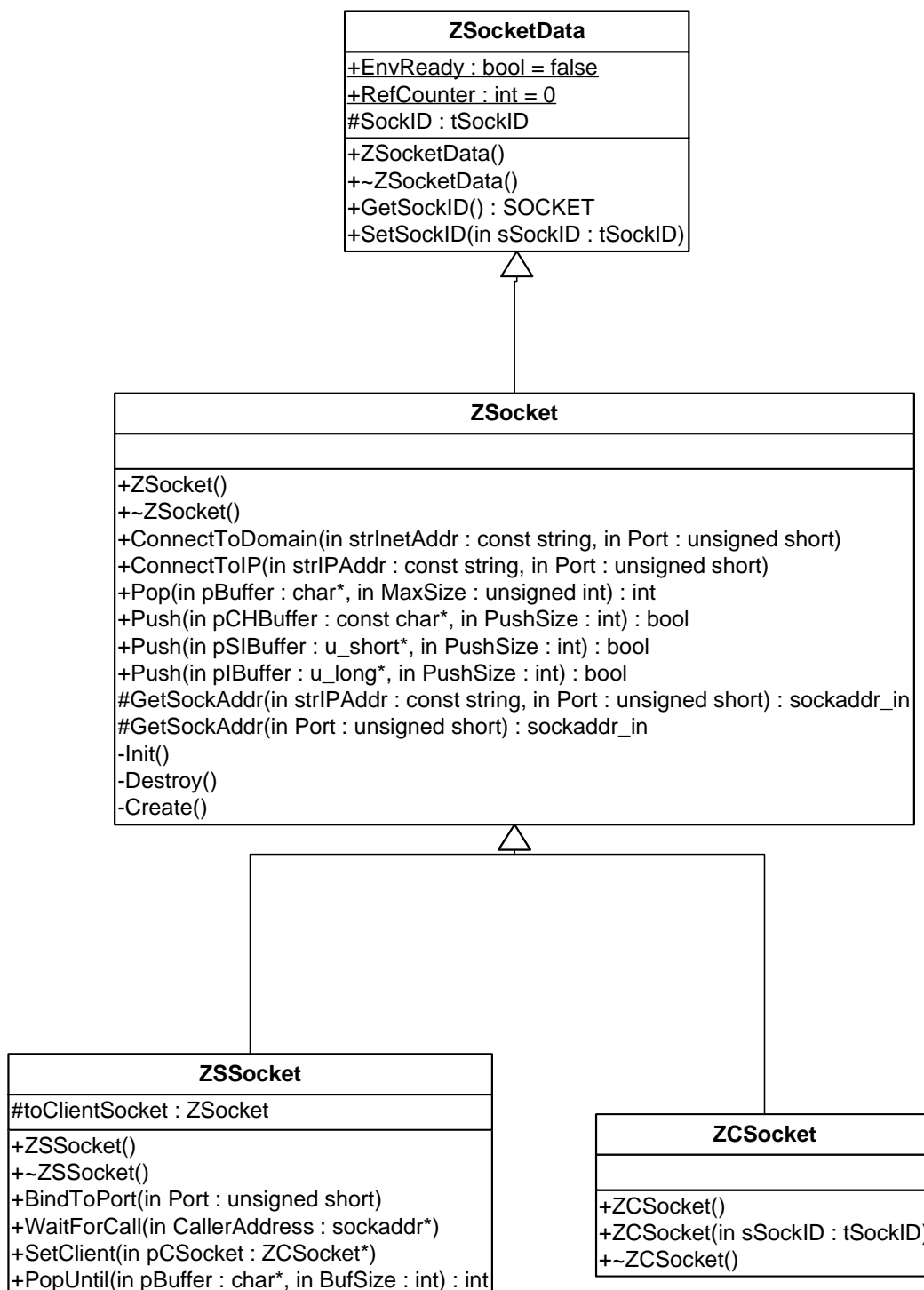
- 2.3.1.1.7 Затем симулятор может от каждой из стратегий принимать сигналы с приказами для кораблей своего флота: OrderSpeedRequest, OrderFireRequest, OrderTurnRequest. По окончании симулятору должен прийти сигнал FinishSimulation от каждой стратегии.
- 2.3.1.1.8 После этого каждой из стратегий отправляются ответы на приказы: OrderSpeedResponse, OrderFireResponse, OrderTurnResponse, если приказ был принят к выполнению, и OrderSpeedReject, OrderFireReject, OrderTurnReject, если приказ не был принят к исполнению.
- 2.3.1.1.9 Затем симулятор может отправить сигналы DestroyShip тем стратегиям, корабли которых были уничтожены.
- 2.3.1.1.10 Далее, если определилась стратегия-победитель, то ей отправляется сигнал StrategyVictory, а другой стратегии StrategyLose.
- 2.3.1.1.11 После чего выполняются действия, аналогичные п. 2.3.1.1.5.
- 2.3.1.1.12 Если на этом такте симуляции не определилась стратегия победитель, то далее выполняются все действия, начиная с п. 2.3.1.1.3.

2.3.1.2 Протокол взаимодействия с GUI.



- 2.3.1.2.1 После того, как от процесса Main пришел сигнал InitSocket процессу DataSender, в нем выполняются действия по инициализации соединения с GUI.
- 2.3.1.2.2 После того, как GUI установит соединения с симулятором, от него приходит сообщение HELLO, на которое симулятор отправляет сообщение подтверждения ACK. Затем симулятор отправляет данные GUI в сообщениях Initialization и Map. По окончании отправляется сигнал InitSocketComplete процессу Main.
- 2.3.1.2.3 После того, как от процесса Main приходит сигнал SendData процесс DataSender отправляет сообщения ShotsArray и ShipsArray GUI, а затем уже возвращает управление процессу Main, отправив сигнал SendDataComplete.

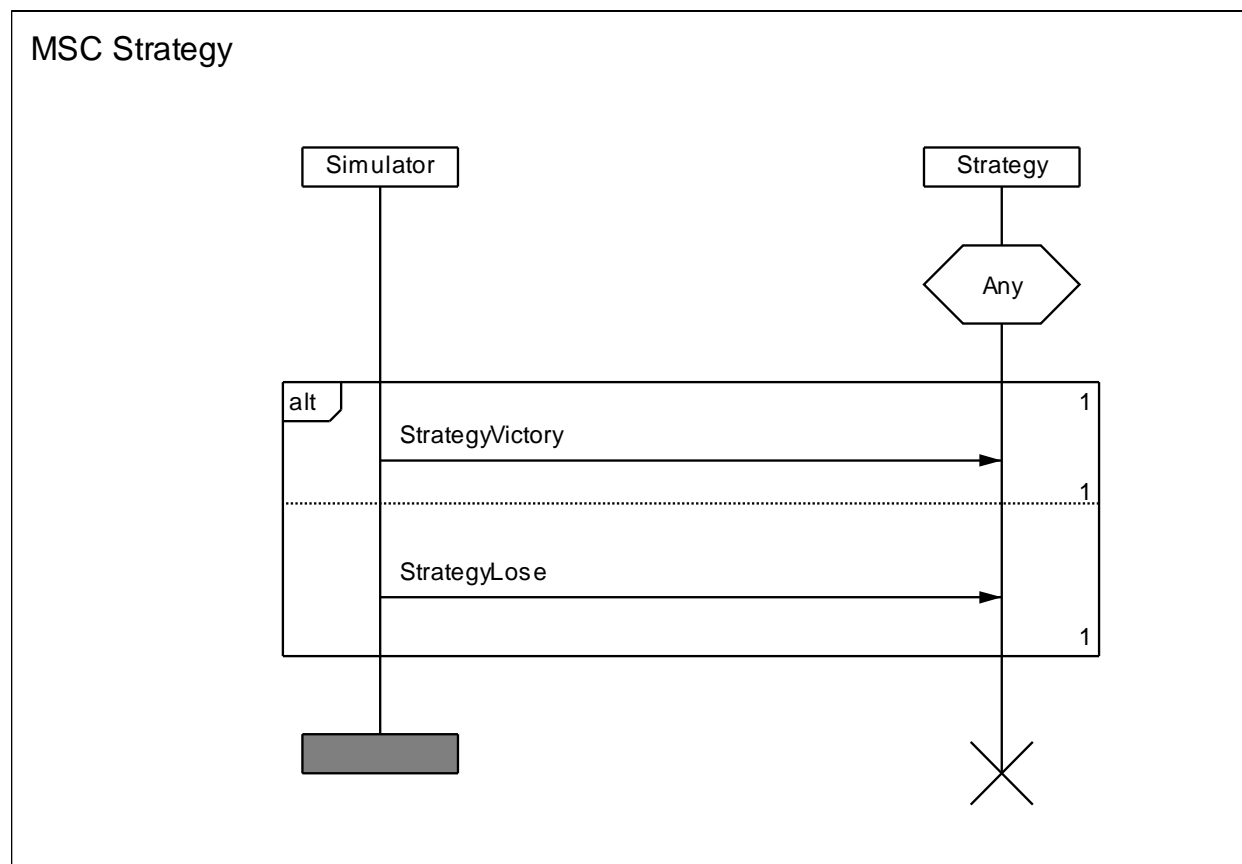
Взаимодействие процесса DataSender с GUI осуществляется через вызовы экспортируемых функций. При этом соединение с помощью сокетов должно быть реализована в виде набора классов, диаграмма взаимодействий которых представлена на рисунке:



2.3.2 Поведение стратегии.

2.3.2.1 Окончание работы стратегии.

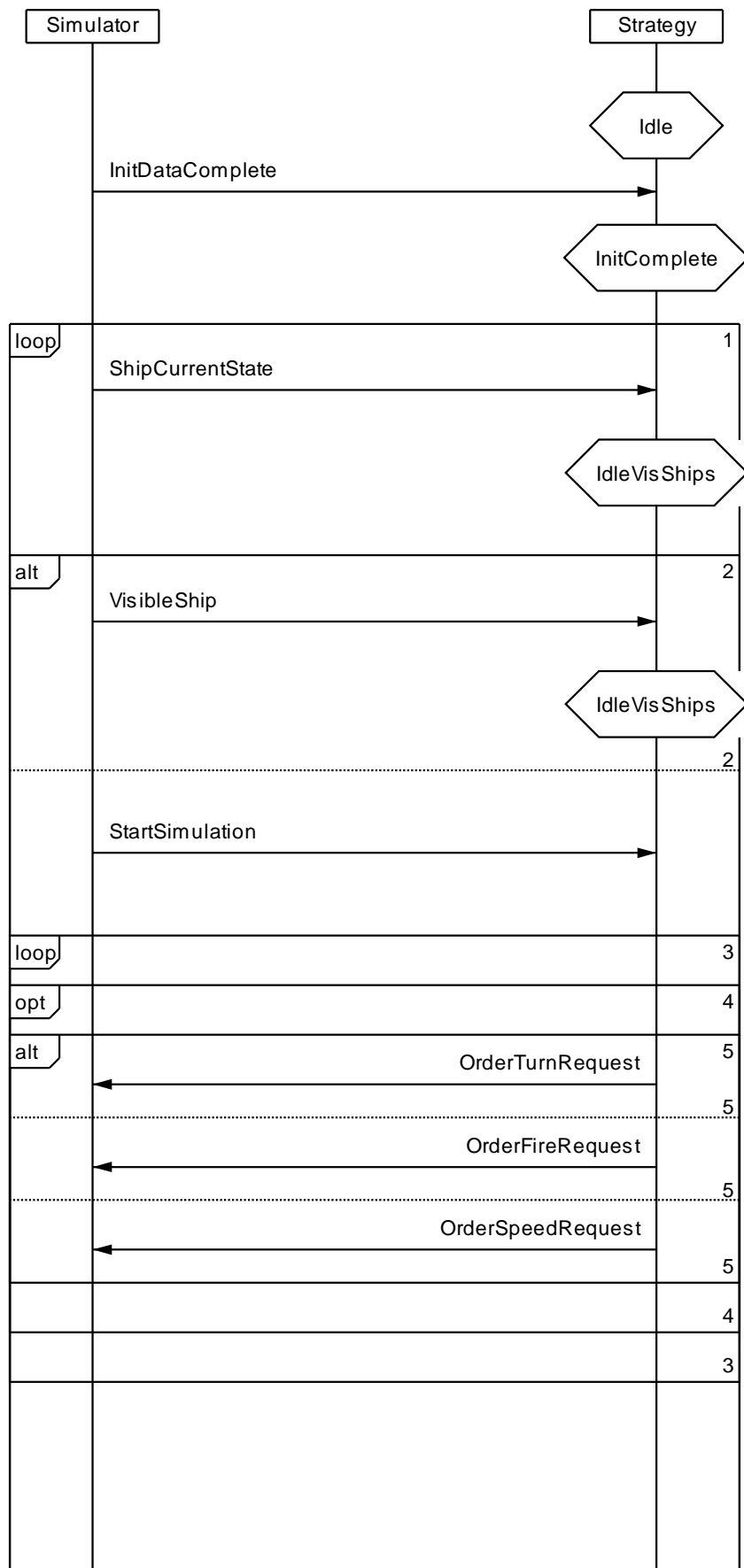
- 2.3.2.1.1 Если в любом состоянии стратегии приходят сигналы StrategyVictory или StrategyLose, стратегия завершает свою работу.

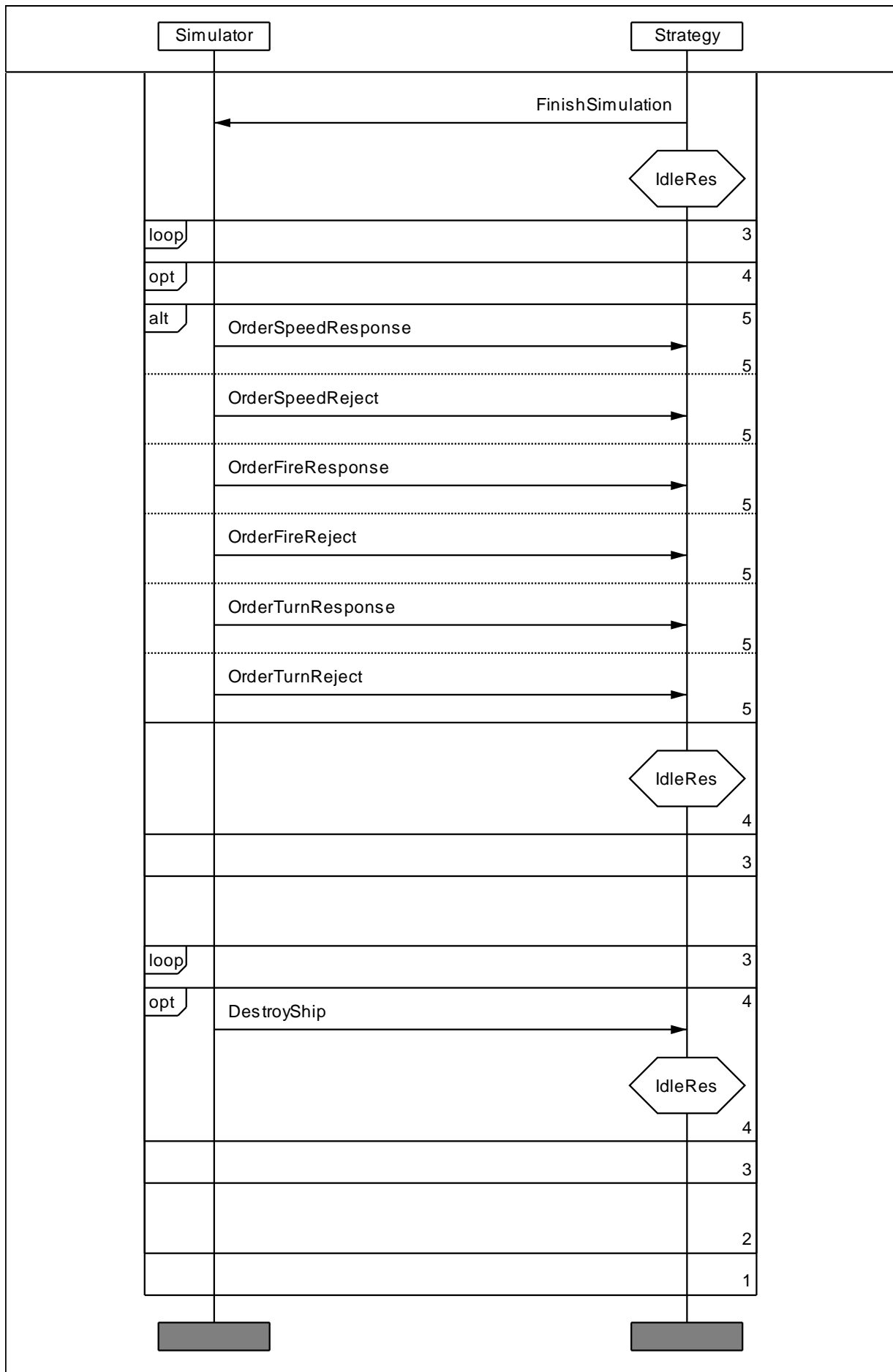


2.3.2.2 Взаимодействие стратегии с симулятором.

- 2.3.2.2.1 В начальный момент стратегия находится в состоянии Idle ожидания прихода сигнала InitDataComplete.
- 2.3.2.2.2 После прихода сигнала InitDataComplete, показывающего, что инициализация данных проведена симулятором успешно, стратегия переходит в состояние InitComplete и ожидает сигнала ShipsCurrentState с информацией о кораблях флота.
- 2.3.2.2.3 После прихода сигнала ShipsCurrentState стратегия переходит в состояние IdleVisShips, где она ожидает прихода сигналов либо VisibleShip, содержащего список вражеских кораблей с их характеристиками, либо StartSimulation, говорящего о том, что стратегия может начинать такт симуляции и посылать сообщения с приказами для своих кораблей.
- 2.3.2.2.4 Если в состоянии IdleVisShips в стратегию поступил сигнал VisibleShip, вычисляются центр масс вражеских кораблей (см. п. 2.4.1) и общее число вражеских кораблей, а затем снова осуществляется переход в состояние IdleVisShips.

MSC Strategy





- 2.3.2.2.5 Если в состоянии IdleVisShips в стратегию поступил сигнал StartSimulation, то стратегия перебирает все корабли флота и посылает необходимые сигналы с приказами этим кораблям (см. п. 5.2 документа [1]): OrderSpeedRequest, OrderTurnRequest или OrderFireRequest. После отсылки всех приказов стратегия отправляет симулятору сигнал FinishSimulation, говорящий о том, что все приказы кораблям были отосланы, и переходит в состояние IdleRes, в котором ожидает результатов приказов.
- 2.3.2.2.6 В ответ на каждый сигнал с приказом для корабля стратегия получает либо подтверждение выполнения приказа – OrderFireResponse, OrderTurnResponse, OrderSpeedResponse, либо сигналы о невозможности выполнить приказ – OrderFireReject, OrderTurnReject или OrderSpeedReject. В ответ на приход любого из этих сигналов стратегия остается в состоянии IdleRes.
- 2.3.2.2.7 После того, как все сигналы о результатах выполнения приказов были высланы, стратегия может получить сигналы DestroyShip, сообщающие об уничтоженных кораблях. После прихода каждого из сигналов DestroyShip стратегия остается в состоянии IdleRes.
- 2.3.2.2.8 Находясь в состоянии IdleRes, стратегия может получить сигнал ShipsCurrentState с текущей информацией о флоте, по получении которого выполняются все действия, начиная с п. 2.3.2.2.3.

2.4 Основные алгоритмы.

2.4.1 Центр масс кораблей.

Центр масс кораблей флота вычисляется по следующей формуле:

$$x_0 = \sum_{i=1}^N \frac{x_i}{N}, y_0 = \sum_{i=1}^N \frac{y_i}{N}$$

где N – количество кораблей во флоте, x_i и y_i – координаты i -го корабля, x_0 и y_0 – координаты центра масс флота.

2.4.2 Движение кораблей при отсутствии противника.

Движение кораблей при отсутствии противника осуществляется по прямоугольнику, углы которого являются центрами четвертей карты (целевыми точками):

$$X_1 = \frac{3}{4} MaxXCoordinate, Y_1 = \frac{3}{4} MaxYCoordinate$$

$$X_2 = \frac{3}{4} MaxXCoordinate, Y_2 = \frac{1}{4} MaxYCoordinate$$

$$X_3 = \frac{1}{4} MaxXCoordinate, Y_3 = \frac{1}{4} MaxYCoordinate$$

$$X_4 = \frac{1}{4} MaxXCoordinate, Y_4 = \frac{3}{4} MaxYCoordinate$$

где пары (X_i, Y_i) – целевые точки, $MaxXCoordinate$ – размер карты по оси абсцисс, $MaxYCoordinate$ – размер карты по оси ординат.

2.4.3 Область целевой точки.

Область целевой точки задается множеством следующих точек (x, y) :

$$x \in [X_i - 5, X_i + 5]$$

$$y \in [Y_i - 5, Y_i + 5]$$

где пары (X_i, Y_i) – целевые точки.

2.4.4 Координаты точек обстрела.

Для ведения огня по кораблям координаты точки обстрела определяются из расчета положения корабля и его скорости:

$$x_{i+1} = x_i + v_x$$

$$y_{i+1} = y_i + v_y$$

где x_i и y_i – координаты корабля на текущем такте симуляции, x_{i+1} и y_{i+1} – координаты корабля на следующем такте симуляции (точки обстрела), v_x и v_y – составляющие скорости корабля по осям абсцисс и ординат соответственно.

2.4.5 Угол между векторами.

Для корректировки направления движения кораблей к цели используется формула для вычисления угла между векторами:

$$\cos(\alpha) = \frac{a_x b_x + a_y b_y}{|a||b|}$$

где вектор a с координатами a_x и a_y – вектор, направленный от точки положения корабля до цели, вектор b с координатами b_x и b_y – вектор, направленный от точки положения корабля до точки, в которой окажется корабль при соответствующем повороте (налево, направо или при движении вперед). Для определения непосредственно направления поворота используется следующая формула:

$$\cos(\alpha)_{\min} = \min\{\cos(\alpha)_L, \cos(\alpha)_R, \cos(\alpha)_S\}$$

где $\cos(\alpha)_L$ – косинус угла при повороте налево, $\cos(\alpha)_R$ – косинус угла при повороте направо, $\cos(\alpha)_S$ – косинус угла при движении вперед. Соответственно в качестве направления поворота выбирается то значение, которое соответствует минимальному значению косинуса.

2.4.6 Положение корабля при изменении скорости.

Для определения положения корабля на следующем такте симуляции в случае изменения его скорости используется формула:

$$Speed = OldSpeed + MaxAcceleration * FactorAccel$$

где $Speed$ – новая скорость корабля, $OldSpeed$ – текущая скорость корабля, $MaxAcceleration$ – максимальное ускорение для данного типа корабля, $FactorAccel$ – множитель в приказе об ускорении.

2.4.7 Изменение положение снаряда.

Если снаряд не может достичь цели за один такт симуляции, то вычисление новых координат снаряда производится по следующим формулам.

$$x = cx + \frac{speed \cdot (fx - cx)}{\sqrt{(fx - cx)^2 + (fy - cy)^2}}$$

$$y = cy + \frac{speed \cdot (fy - cy)}{\sqrt{(fx - cx)^2 + (fy - cy)^2}}$$

где x и y – новые значения координат снаряда, cx и cy – координаты снаряда на данном такте симуляции, fx и fy – координаты цели, $speed$ – скорость снаряда данного типа (количество клеток, преодолеваемое за один такт симуляции).