**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

**Отчет по курсовой работе**

**по курсу "Операционные системы"**

Студент: Сливин М.В.

группа: М8О-208Б-17

№ по списку: 20

Работа выполнена \_\_.\_\_.\_\_\_\_

Преподаватель: Миронов Е.С.

Отчет сдан \_\_.\_\_.\_\_\_\_

Оценка

Москва 2019

**Цель:**

* Приобретение практических навыков в использовании знаний, полученных в течение курса
* Проведение исследования в выбранной предметной области

**Задание:**

Необходимо спроектировать и реализовать программный прототип в соответствии с выбранным вариантом. Произвести анализ и сделать вывод на основании данных, полученных при работе программного прототипа.

*Вариант задания*:

Создание клиента для передачи мгновенных личных сообщений

На основе любой из выбранных технологий:

* Pipes
* Sockets
* Сервера очередей
* И другие

Необходимо реализовать консоль-серверную игру крестики нолики . Должны быть поддержка нескольких игр сразу.

Игры проводятся на поле 5 на 5 клеток. Для победы необходимо получить три элемента, следующих друг за другом, либо по горизонтали, либо по вертикали или же по любой из диагоналей.

Одновременно в игре участвуют 3 человека (крестики, нолики, квадратики). То есть если на сервер зашло 7 человек, то 6 игроков начали игру, а один игрок ждет еще двух.

При выходе одного игрока сервер уведомляет об этом двух оставшихся, и последующая игра будет проводиться между ними.

Для программы выбран способ передачи сообщений *ZeroMQ* .

**Введение:**

Реализация консольно-серверной игры происходит в несколько этапов :

* Разбиение задачи на подзадачи , то есть разработать алгоритмы для всех программ , составляющих полноценную игру.
* Анализ сложности алгоритма .
* Написать скелет будущей программы и затем добавлять в него больше кода .
* Тестирование программы для поиска и исправления ошибок .
* Краш тесты (например проверить устойчивость программы на вход миллиона игроков).

Так как программа игры должна быть устойчива при любой сложившейся ситуации, потенциально, сложность ее реализации значительно возрастает. Именно поэтому я решил сфокусироваться на отказоустойчивости программы в ущерб ее производительности. Поэтому, несмотря на то что моя программа реализует полный функционал, она не является идеальной.

Для написания кода я выбрал стандартный паттерн общения между клиентами и серверами — *Request — Response*, суть которого состоит в том, чтоклиент отправляет данные на сервер и получает оттуда сообщение о успехе или неуспехе действия . Хоть схема и простая , но при написании кода не раз возникала путаница и приходилось долго исследовать код на возникновение логических ошибок, а также тратить значительные усилия на их исправления. Именно поэтому следует изначально разработать архитектуру программы в подробностях, а только потом переводить концепцию на программный код.

**Решение:**

Так как по условиям моей задачи необходима поддержка нескольких игр сразу, я решил размещать базу данных игр в обычном массиве. Несмотря на то, что это решение не является эффективным, оно позволило мне сразу перейти к разработке игры без предварительных трудностей при создании хорошего хранилища. Стоит заметить, что мою программу можно легко переделать на линейном списке для большей эффективности, если это необходимо.

Далее я написал заголовочный файл, который хранит описание структуры сообщения и стандартного игровой зоны. Понятие игровой зоны состоит из самого игрового поля 5х5 элементов, а также параметров, отвечающих за состояние той или иной игры, а именно:

* *game\_end -* параметр, фиксирующий то или иное завершение игры для отправки информации об этом другим игрокам).
* *current\_figure -* параметр, отвечающий за последовательность игровых действий игроков.
* *status -* по смыслу является схожим с *game\_end*, но данный параметр фиксирует синхронизацию начала игры, а также опрос игроками сервера в предыгровых ситуациях.
* *who\_left —* параметр, опосредованно передающий игрокам информацию о том, что один из игроков покинул игру.
* *count\_left -* параметр, регулирующий, что два игрока одновременно покинули игру на предыдущем шаге.

Реализация клиента следующая :

При запуске игры происходит идентификация пользователя в системе, где ему присваивается порядковый номер. Уже на этапе регистрации игрок получает заданную фигуру, соответствующую его номеру. Реализуется данный алгоритм на стороне сервера при обращении к нему клиентов. Для упрощения работы с игровой зоной, на клиенте на каждом шаге происходит полное копирование текущей зоны в локальную область.

После непосредственного подключения игрок попадает в очередь ожидания, где опрашивает сервер с заданным периодом времени на предмет наличия еще двух игроков. Когда определенное количество игроков набирается, сервер «активирует» игровую зону (playzone.status = 1). Сразу после этого игрок из очереди ожидания попадает в игру, где либо он начинает ход, либо же попадает уже в другую очередь, отвечающую за последовательный ввод игроков.

Когда параметр игровой зоны current\_figure поменяется на фигуру, присвоенную клиенту, ему будет предложен ввод координат новой фигуры с предварительной отрисовкой поля. Ввод сначала проверяется на корректность, а сразу после — на проверку условия выигрыша игрока и абсолютной заполненности поля.

Согласно дополнительному условию моей задачи на этапе ожидания хода происходит отлавливание сигнала прерывания процесса клиента и передача информации об этом с помощью уже упомянутых параметров игровой зоны.

**Сценарий выполнения работы:**

**Три игрока. Один игрок покидает игру, не дождавшись своего хода.**

***server***:

spknnk@ubuntu:~/Рабочий стол/OS/xo\_new$ ./server

Server is starting...

Status: tcp://\*:5000

0 1 2 3 4

0 | \_ | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: -1

mdF: -1

dF: 4

pF: 0

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 1

!!!!MSG1

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: -1

mdF: -1

dF: 4

pF: 0

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 2

!!!!MSG1

>Players in queue: 2

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 2

!!!!MSG3

>Playzone status: 0

>ID: 2

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 2

!!!!MSG3

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: -1

mdF: -1

dF: 4

pF: 0

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 3

Game 0 is ready!

0 1 2 3 4

0 | \_ | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

!!!!MSG4

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 0

!!!!MSG3

>Playzone status: 0

>ID: 2

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 0

!!!!MSG3

>Playzone status: 1

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

!!!!MSG2

>Playzone status: 1

>ID: 2

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

!!!!MSG2

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 0

>Coords sended: 1

PG ID: 0

mdF: 1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

~~1~~

>>>>>>>>>>>>>>>>dF: 2

>>>>>>>>>>>>>>>>pF: 2

Client ID: 3

Coordinates: 0 0

~~~ Null counter: 25

0 1 2 3 4

0 | x | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

COORDS SENDED CHANGE

>Playzone status: 1

>ID: 2

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 3

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt3 pz3

>-----------------------

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 1

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz3

>-----------------------

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 1

>Coords sended: 1

PG ID: 0

mdF: 2

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz3

>-----------------------

~~2~~

>>>>>>>>>>>>>>>>dF: 1

>>>>>>>>>>>>>>>>pF: 1

Client ID: 1

Coordinates: 0 1

~~~ Null counter: 24

0 1 2 3 4

0 | x | o | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt3 pz3

>-----------------------

COORDS SENDED CHANGE

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz3

>-----------------------

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 2

>Coords sended: 1

PG ID: 0

mdF: 1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt3 pz3

>-----------------------

~~1~~

>>>>>>>>>>>>>>>>dF: 2

>>>>>>>>>>>>>>>>pF: 2

Client ID: 3

Coordinates: 0 2

~~~ Null counter: 23

0 1 2 3 4

0 | x | o | x | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt3 pz3

>-----------------------

COORDS SENDED CHANGE

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 1

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt3 pz3

>-----------------------

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 3

>Coords sended: 1

PG ID: 0

mdF: 2

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt3 pz3

>-----------------------

~~2~~

>>>>>>>>>>>>>>>>dF: 1

>>>>>>>>>>>>>>>>pF: 1

Client ID: 1

Coordinates: 0 3

~~~ Null counter: 22

0 1 2 3 4

0 | x | o | x | o | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt3 pz3

>-----------------------

COORDS SENDED CHANGE

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 1

pF: 1

Game\_end\_data: 4

Game\_end\_playg: 4

WHO\_LEFT: dt3 pz3

>-----------------------

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 4

>Coords sended: 1

PG ID: 0

mdF: 1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 4

WHO\_LEFT: dt3 pz3

>-----------------------

Game finished!

~~1~~

>>>>>>>>>>>>>>>>dF: 2

>>>>>>>>>>>>>>>>pF: 2

Client ID: 3

Coordinates: 0 4

~~~ Null counter: 21

0 1 2 3 4

0 | x | o | x | o | x

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

***player1***:

spknnk@ubuntu:~/Рабочий стол/OS/xo\_new$ ./client

Connected to tcp://localhost:5000

0.---------------------

ID: 0

Game ready status: 0

Game ready notific: 0

Playzone status: 0

Figure: -1

dF: 4

PG ID: -1

Game end: 0

-----------------------

IM IN ELSE

Wait a moment...

!!!MESSAGE SENDED2!!!

Sorry, you wasn't registred. Now you're allowed to play!

Your ID:1

INFO STATUS: 0

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Your game is ready!(1)

1.---------------------

ID: 1

Game ready status: 1

Game ready notific: 1

Playzone status: 1

Figure: -1

dF: 1

PG ID: 0

Game end: 0

-----------------------

MYFIGURE: 2

mdF: 2

df: 1

!!!MESSAGE SENDED1.5!!!

Your turn now!(0)

0 1 2 3 4

0 | x | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

~~~ Null counter: 24

~Please, enter coordinates of your figure: 0 1

0 1 2 3 4

0 | x | o | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

!!!MESSAGE SENDED1!!!

>>>WARNING:PLAYER LEFT THE GAME

2.---------------------

ID: 1

Game ready status: 1

Game ready notific: 1

Playzone status: 1

Figure: 2

dF: 1

PG ID: 0

Game end: 0

-----------------------

MYFIGURE: 2

mdF: 2

df: 1

!!!MESSAGE SENDED1.5!!!

Waiting for enemy turn!

MYFIGURE: 2

mdF: 2

df: 1

!!!MESSAGE SENDED1.5!!!

Your turn now!(0)

0 1 2 3 4

0 | x | o | x | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

~~~ Null counter: 22

~Please, enter coordinates of your figure: 0 3

0 1 2 3 4

0 | x | o | x | o | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

!!!MESSAGE SENDED1!!!

>>>WARNING:PLAYER LEFT THE GAME

3.---------------------

ID: 1

Game ready status: 1

Game ready notific: 1

Playzone status: 1

Figure: 2

dF: 1

PG ID: 0

Game end: 0

-----------------------

MYFIGURE: 2

mdF: 2

df: 1

^C:(

Waiting for enemy turn!

Disconnected(1.0)!

***player2(который покидает игру)***:

spknnk@ubuntu:~/Рабочий стол/OS/xo\_new$ ./client

Connected to tcp://localhost:5000

0.---------------------

ID: 0

Game ready status: 0

Game ready notific: 0

Playzone status: 0

Figure: -1

dF: 4

PG ID: -1

Game end: 0

-----------------------

IM IN ELSE

Wait a moment...

!!!MESSAGE SENDED2!!!

Sorry, you wasn't registred. Now you're allowed to play!

Your ID:2

INFO STATUS: 0

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Waiting for other players...

!!!MESSAGE SENDED3!!!

Your game is ready!(1)

1.---------------------

ID: 2

Game ready status: 1

Game ready notific: 1

Playzone status: 1

Figure: -1

dF: 1

PG ID: 0

Game end: 0

-----------------------

MYFIGURE: 3

mdF: 3

df: 1

^C:(

Waiting for enemy turn!

Disconnected(1.1)!

***player3***:

spknnk@ubuntu:~/Рабочий стол/OS/xo\_new$ ./client

Connected to tcp://localhost:5000

0.---------------------

ID: 0

Game ready status: 0

Game ready notific: 0

Playzone status: 0

Figure: -1

dF: 4

PG ID: -1

Game end: 0

-----------------------

IM IN ELSE

Wait a moment...

!!!MESSAGE SENDED2!!!

Your game is ready!(2)

INFO STATUS: 0

1.---------------------

ID: 3

Game ready status: 1

Game ready notific: 1

Playzone status: 1

Figure: -1

dF: 1

PG ID: 0

Game end: 0

-----------------------

MYFIGURE: 1

0 1 2 3 4

0 | \_ | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

~~~ Null counter: 25

~Please, enter coordinates of your figure: 0 0

0 1 2 3 4

0 | x | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

!!!MESSAGE SENDED1!!!

Nice turn!

2.---------------------

ID: 3

Game ready status: 1

Game ready notific: 1

Playzone status: 1

Figure: 1

dF: 2

PG ID: 0

Game end: 0

-----------------------

MYFIGURE: 1

mdF: 1

df: 2

!!!MESSAGE SENDED1.5!!!

Waiting for enemy turn!

MYFIGURE: 1

mdF: 1

df: 2

!!!MESSAGE SENDED1.5!!!

Your turn now!(0)

0 1 2 3 4

0 | x | o | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

~~~ Null counter: 23

~Please, enter coordinates of your figure: 0 2

0 1 2 3 4

0 | x | o | x | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

!!!MESSAGE SENDED1!!!

>>>WARNING:PLAYER LEFT THE GAME

3.---------------------

ID: 3

Game ready status: 1

Game ready notific: 1

Playzone status: 1

Figure: 1

dF: 2

PG ID: 0

Game end: 0

-----------------------

MYFIGURE: 1

mdF: 1

df: 2

!!!MESSAGE SENDED1.5!!!

Waiting for enemy turn!

MYFIGURE: 1

mdF: 1

df: 2

!!!MESSAGE SENDED1.5!!!

Your turn now!(0)

0 1 2 3 4

0 | x | o | x | o | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

~~~ Null counter: 21

~Please, enter coordinates of your figure: 0 4

0 1 2 3 4

0 | x | o | x | o | x

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

!!!MESSAGE SENDED1!!!

OTHER PLAYERS HAS LEFT THE GAME

4.---------------------

ID: 3

Game ready status: 1

Game ready notific: 1

Playzone status: 1

Figure: 1

dF: 1

PG ID: 0

Game end: 4

-----------------------

GAME END

**Две параллельные игры** (в виду большого объема листинга будут показаны логи только на стороне сервера, так как они полностью дают представление о том, что происходит на локальных клиентах):

***server***:

spknnk@ubuntu:~/Рабочий стол/OS/xo\_new$ ./server

Server is starting...

Status: tcp://\*:5000

0 1 2 3 4

0 | \_ | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 0

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 1

!!!!MSG1

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 0

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 0

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 2

!!!!MSG1

>Players in queue: 2

!!!!MSG3

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 4

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 3

Game 0 is ready!

0 1 2 3 4

0 | \_ | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

!!!!MSG4

>Playzone status: 0

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 0

!!!!MSG3

>Playzone status: 0

>ID: 2

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 0

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 0

!!!!MSG3

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: -1

mdF: -1

dF: 4

pF: 0

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 4

!!!!MSG1

>Players in queue: 1

!!!!MSG3

>Playzone status: 1

>ID: 1

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

!!!!MSG2

>Playzone status: 1

>ID: 2

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

!!!!MSG2

>Playzone status: 0

>ID: 4

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 1

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 1

!!!!MSG3

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 0

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 5

!!!!MSG1

>Players in queue: 2

!!!!MSG3

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 2

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 3

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 0

>ID: 4

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 1

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 2

!!!!MSG3

>Playzone status: 0

>ID: 5

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 1

mdF: -1

dF: 0

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 2

!!!!MSG3

>Playzone status: 0

>ID: 0

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 0

mdF: -1

dF: 0

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

New user!

>Number of players: 6

Game 1 is ready!

0 1 2 3 4

0 | \_ | \_ | \_ | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

!!!!MSG4

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 2

>Coords sended: 1

PG ID: 0

mdF: 1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

~~1~~

>>>>>>>>>>>>>>>>dF: 2

>>>>>>>>>>>>>>>>pF: 2

Client ID: 3

Coordinates: 0 2

~~~ Null counter: 25

0 1 2 3 4

0 | \_ | \_ | x | \_ | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

COORDS SENDED CHANGE

>Playzone status: 1

>ID: 2

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 3

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 0

>ID: 4

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 1

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 0

!!!!MSG3

>Playzone status: 0

>ID: 5

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 1

mdF: -1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Players in queue: 0

!!!!MSG3

>Playzone status: 1

>ID: 6

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 4

>Coords sended: 1

PG ID: 1

mdF: 1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

~~1~~

>>>>>>>>>>>>>>>>dF: 2

>>>>>>>>>>>>>>>>pF: 2

Client ID: 6

Coordinates: 0 4

~~~ Null counter: 25

0 1 2 3 4

0 | \_ | \_ | \_ | \_ | x

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 1

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 3

>Coords sended: 1

PG ID: 0

mdF: 2

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

~~2~~

>>>>>>>>>>>>>>>>dF: 3

>>>>>>>>>>>>>>>>pF: 3

Client ID: 1

Coordinates: 0 3

~~~ Null counter: 24

0 1 2 3 4

0 | \_ | \_ | x | o | \_

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 2

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 3

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

COORDS SENDED CHANGE

>Playzone status: 1

>ID: 4

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 1

mdF: -1

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

!!!!MSG2

>Playzone status: 1

>ID: 5

>FIG: -1

>Game ready status: 0

>Game ready notific: 0

>Coords: -1 -1

>Coords sended: 0

PG ID: 1

mdF: -1

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

!!!!MSG2

>Playzone status: 1

>ID: 6

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 1

mdF: 1

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 1

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 5

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 1

mdF: 3

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 4

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: 0 2

>Coords sended: 1

PG ID: 1

mdF: 2

dF: 2

pF: 2

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

~~2~~

>>>>>>>>>>>>>>>>dF: 3

>>>>>>>>>>>>>>>>pF: 3

Client ID: 4

Coordinates: 0 2

~~~ Null counter: 24

0 1 2 3 4

0 | \_ | \_ | o | \_ | x

1 | \_ | \_ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 6

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 1

mdF: 1

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 2

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: 1 1

>Coords sended: 1

PG ID: 0

mdF: 3

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

~~0~~

>>>>>>>>>>>>>>>>dF: 1

>>>>>>>>>>>>>>>>pF: 1

Client ID: 2

Coordinates: 1 1

~~~ Null counter: 23

0 1 2 3 4

0 | \_ | \_ | x | o | \_

1 | \_ | □ | \_ | \_ | \_

2 | \_ | \_ | \_ | \_ | \_

3 | \_ | \_ | \_ | \_ | \_

4 | \_ | \_ | \_ | \_ | \_

1MY PG STATUS IS: 1

2MY PG STATUS IS: 1

!!!!MSG0

>Playzone status: 1

>ID: 3

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 1

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

COORDS SENDED CHANGE

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 5

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 1

mdF: 3

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

COORDS SENDED CHANGE

>Playzone status: 1

>ID: 4

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 1

mdF: 2

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 6

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 1

mdF: 1

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 2

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 3

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 1

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 2

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 4

>FIG: 2

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 1

mdF: 2

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 6

>FIG: 1

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 1

mdF: 1

dF: 3

pF: 3

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

>Playzone status: 1

>ID: 2

>FIG: 3

>Game ready status: 1

>Game ready notific: 1

>Coords: -1 -1

>Coords sended: 2

PG ID: 0

mdF: 3

dF: 1

pF: 1

Game\_end\_data: 0

Game\_end\_playg: 0

WHO\_LEFT: dt0 pz0

>-----------------------

**Листинг программы** :

**CLIENT.C**

#include <inttypes.h>

#include <pthread.h>

#include <stdio.h>

#include <signal.h>

#include <stdlib.h>

#include <string.h>

#include <zmq.h>

#include "arg.h"

volatile sig\_atomic\_t flag = 0;

int im\_left = 0;

void sig\_check(int sig){

switch (sig){

case SIGINT: {

if (flag == 0){

flag = sig;

im\_left++;

puts(":(");

} else if (flag == sig){

flag = 0;

puts(" Server unblocked!");

}

break;

}

case SIGQUIT:

flag = sig;

}

}

void\* SendMessage(void\* arg, Message\_t\* p){

Message\_t\* data = (Message\_t\*)arg;

zmq\_msg\_t message;

zmq\_msg\_init\_size(&message, sizeof(Message\_t));

memcpy(zmq\_msg\_data(&message), data, sizeof(Message\_t));

zmq\_msg\_send(&message, data->requester, 0);

zmq\_msg\_close(&message);

zmq\_msg\_init(&message);

zmq\_msg\_recv(&message, data->requester, 0);

data = (Message\_t\*)zmq\_msg\_data(&message);

printf("%s\n", data->message);

p->id = data->id;

p->playground\_id = data->playground\_id;

p->game\_is\_ready = data->game\_is\_ready;

//p->playzone = data->playzone;

p->playzone.status = data->playzone.status;

p->playzone.current\_figure = data->playzone.current\_figure;

p->playzone.game\_end = data->playzone.game\_end;

for(int i = 0; i < DIM; ++i) {

for(int j = 0; j < DIM; ++j) {

p->playzone.coord[i][j] = data->playzone.coord[i][j];

}

}

p->ready\_notification = data->ready\_notification;

p->coords\_sended = data->coords\_sended;

p->playzone.who\_left = data->playzone.who\_left;

p->playzone.count\_left = data->playzone.count\_left;

zmq\_msg\_close(&message);

return 0;

}

int main() {

void\* context = zmq\_ctx\_new();

void\* server = zmq\_socket(context, ZMQ\_REQ);

//save id,sckt in file

// IS READY TO PLAY

int sckt = 5000;

// FILE \*f = fopen("id.txt", "r");

// fscanf(f, "%d %d", &id, &sckt);

char adress[25];

sprintf(adress, "%s%d", "tcp://localhost:", sckt);

zmq\_connect(server, adress);

printf("\nConnected to %s\n\n", adress);

/\* ZMQ \*/

Message\_t data;

data.id = 0; // not registred on server

data.game\_is\_ready = 0;

data.ready\_notification = 0;

data.requester = server;

data.coords\_sended = 0;

data.x\_coord = -1;

data.y\_coord = -1;

data.figure = -1;

int game\_is\_ready\_flag = 0;

int info\_status = 0;

data.playzone.status = 0; // 4 - winner found

data.playzone.who\_left = 0;

data.playzone.game\_end = 0;

data.playzone.count\_left = 0;

while(1) {

//Debugging

printf("%d.---------------------\n", info\_status);

printf("ID: %d\n", data.id);

printf("Game ready status: %d\n", data.game\_is\_ready);

printf("Game ready notific: %d\n", data.ready\_notification);

printf("Playzone status: %d\n", data.playzone.status);

printf("Figure: %d\n", data.figure);

printf("dF: %d\n", data.playzone.current\_figure);

printf("PG ID: %d\n", data.playground\_id);

printf("Game end: %d\n", data.playzone.game\_end);

printf("-----------------------\n");

info\_status++;

signal(SIGINT, sig\_check);

signal(SIGQUIT, sig\_check);

if((im\_left == 1) && (data.playzone.who\_left != 0)){

data.playzone.game\_end = 4;

data.playzone.count\_left++;

SendMessage(&data, &data);

printf("Disconnected(0.0)!\n");

return 0;

}

if((im\_left == 1) && (data.playzone.who\_left == 0)){

data.playzone.who\_left = data.figure;

data.playzone.count\_left++;

SendMessage(&data, &data);

printf("Disconnected(0.1)!\n");

return 0;

}

if(data.playzone.game\_end == 4){

printf("GAME END\n");

return 0;

}

if((data.game\_is\_ready == 1) && (data.id != 0)) {

data.x\_coord = -1;

data.y\_coord = -1;

if(data.figure == -1){

if(data.id % 3 == 0) data.figure = x;

if(data.id % 3 == 1) data.figure = o;

if(data.id % 3 == 2) data.figure = n;

}

int wrong\_coords = 0;

coords:

while((data.x\_coord < 0) || (data.y\_coord < 0) || (data.x\_coord > 4) || (data.y\_coord > 4)) {

printf("MYFIGURE: %d\n", data.figure);

if(wrong\_coords == 1){

printf("Enter correct coordinates!\n");

wrong\_coords = 0;

}

if(data.figure != data.playzone.current\_figure){

//printf("Waiting for enemy turn(1)...\n");

//while(data.figure != data.playzone.current\_figure){

printf("mdF: %d\n", data.figure);

printf("df: %d\n", data.playzone.current\_figure);

sleep(3);

data.x\_coord = -1;

data.y\_coord = -1;

data.coords\_sended = 2;

signal(SIGINT, sig\_check);

signal(SIGQUIT, sig\_check);

if((im\_left == 1) && (data.playzone.who\_left != 0)){

data.playzone.game\_end = 4;

data.playzone.count\_left++;

SendMessage(&data, &data);

printf("Disconnected(1.0)!\n");

return 0;

}

if((im\_left == 1) && (data.playzone.who\_left == 0)){

data.playzone.who\_left = data.figure;

data.playzone.count\_left++;

SendMessage(&data, &data);

printf("Disconnected(1.1)!\n");

return 0;

}

if((data.playzone.game\_end == 4) || (data.playzone.count\_left == 2)){

printf("GAME END\n");

return 0;

}

printf("!!!MESSAGE SENDED1.5!!!\n");

SendMessage(&data, &data);

}

if(data.figure == data.playzone.current\_figure){

PlayzonePrint(&data.playzone);

if((PlayzoneIsFull(&data.playzone) == 1) || (data.playzone.game\_end == 4)){

printf("\n\nPlayzone is full. Game finished!\n");

data.playzone.game\_end = 4;

data.coords\_sended = 1;

printf("!!!MESSAGE SENDED FULL!!!\n");

SendMessage(&data, &data);

return 0;

}

if(data.playzone.status == 4){

printf("BETTER LUCK NEXT TIME!\n");

data.coords\_sended = 1;

printf("!!!MESSAGE SENDED LOSTGAME!!!\n");

SendMessage(&data, &data);

return 0;

}

if(data.playzone.game\_end == 4){

printf("Other players left the game. Game finished!\n");

return 0;

}

printf("~Please, enter coordinates of your figure: ");

wrong\_coords = 1;

scanf("%d %d", &data.x\_coord, &data.y\_coord);

if((data.playzone.coord[data.x\_coord][data.y\_coord] != 0) || (data.x\_coord > 4) || (data.y\_coord > 4)){

data.x\_coord = -1;

data.y\_coord = -1;

goto coords;

}

data.playzone.coord[data.x\_coord][data.y\_coord] = data.figure;

PlayzonePrint(&data.playzone);

//CHECK FOR WINNER

if(PlayzoneCheckForWinner(&data.playzone, data.x\_coord, data.y\_coord) == 1){

//TODO

printf("IM SENDING INFO THAT OTHER PLAYERS LOSE\n");

data.playzone.status = 4;

data.coords\_sended = 1;

printf("!!!MESSAGE SENDED WINNER!!!\n");

SendMessage(&data, &data);

return 0;

}

data.coords\_sended = 1;

printf("!!!MESSAGE SENDED1!!!\n");

SendMessage(&data, &data);

}

}

} else {

printf("IM IN ELSE\n");

if((data.game\_is\_ready == 1) && (game\_is\_ready\_flag == 0)) { // Start of the game

game\_is\_ready\_flag++;

PlayzonePrint(&data.playzone);

}

if(data.id == 0) {

printf("Wait a moment...\n");

printf("!!!MESSAGE SENDED2!!!\n");

SendMessage(&data, &data);

}

printf("INFO STATUS: %d\n", info\_status-1);

while((data.game\_is\_ready == 0) && (data.id != 0)){

//sleep(5);

data.x\_coord = -1;

data.y\_coord = -1;

signal(SIGINT, sig\_check);

signal(SIGQUIT, sig\_check);

if((im\_left == 1) && (data.playzone.who\_left != 0)){

data.playzone.game\_end = 4;

data.playzone.count\_left++;

SendMessage(&data, &data);

printf("Disconnected(2.0)!\n");

return 0;

}

if((im\_left == 1) && (data.playzone.who\_left == 0)){

data.playzone.who\_left = data.figure;

data.playzone.count\_left++;

SendMessage(&data, &data);

printf("Disconnected(2.1)!\n");

return 0;

}

printf("!!!MESSAGE SENDED3!!!\n");

SendMessage(&data, &data);

}

}

//sleep(1);

}

/////////

zmq\_close(server);

zmq\_ctx\_destroy(context);

return 0;

}

**SERVER.C**

#include <signal.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <stropts.h>

#include <sys/ioctl.h>

#include <time.h>

#include <unistd.h>

#include <zmq.h>

#include "arg.h"

volatile sig\_atomic\_t flag = 0;

#define Send\_msg(DATA, MESSAGE, INFO, SERVER) \

memcpy(DATA->message, INFO, strlen(INFO) + 1); \

zmq\_msg\_init\_size(&MESSAGE, sizeof(Message\_t)); \

memcpy(zmq\_msg\_data(&MESSAGE), DATA, sizeof(Message\_t)); \

zmq\_msg\_send(&MESSAGE, SERVER, 0); \

zmq\_msg\_close(&MESSAGE);

void PlayzoneInit(Playzone\* p){

for(int i = 0; i < DIM; i++){

for(int j = 0; j < DIM; j++){

p->coord[i][j] = nan;

}

}

}

void TCPString(int\* currPort, char\* sckt){

char\* tcp = "tcp://localhost:";

char s\_currPort[5];

sprintf(s\_currPort, "%d", \*currPort);

snprintf(sckt, 22, "%s%s", tcp, s\_currPort);

printf("Status: %s\n", sckt);

}

void sig\_check(int sig){

switch (sig){

case SIGINT: {

if (flag == 0){

flag = sig;

puts(" Server blocked!");

} else if (flag == sig){

flag = 0;

puts(" Server unblocked!");

}

break;

}

case SIGQUIT:

flag = sig;

}

}

int main(int argc, char \*\*argv) {

printf("Server is starting...\n");

int currPort = 5000;

char adress[13];

sprintf(adress, "%s%d", "tcp://\*:", currPort);

printf("Status: %s\n", adress);

size\_t playersInQueue = 0;

size\_t player\_id = 0;

Playzone p;

PlayzoneInit(&p);

PlayzonePrint(&p);

void \*context = zmq\_ctx\_new();

void \*server = zmq\_socket(context, ZMQ\_REP);

if (zmq\_bind(server, adress)) {

puts("ERROR: Bind. Server is invalid\n");

}

Playzone playground[50];

char info[90]; // info to client

for (int i = 0; i < 50; ++i) {

playground[i].status = 0;

playground[i].current\_figure = 1;

playground[i].game\_end = 0;

playground[i].who\_left = 0;

}

int playgroud\_iterator = 0;

while (1) {

signal(SIGINT, sig\_check);

signal(SIGQUIT, sig\_check);

if (flag == SIGQUIT) {

zmq\_close(server);

zmq\_ctx\_destroy(context);

printf("Server is down\n");

return 0;

}

if (!flag) {

zmq\_msg\_t message;

zmq\_msg\_init(&message);

zmq\_msg\_recv(&message, server, 0);

Message\_t \*data = (Message\_t \*) zmq\_msg\_data(&message);

zmq\_msg\_close(&message);

if((playground[data->playground\_id].current\_figure != 0) && (data->playzone.current\_figure != 0)){ // F5

data->playzone.current\_figure = playground[data->playground\_id].current\_figure;

//data->playzone.game\_end = playground[data->playground\_id].game\_end;

}

for(int i = 0; i < DIM; ++i) {

for(int j = 0; j < DIM; ++j) {

data->playzone.coord[i][j] = playground[data->playground\_id].coord[i][j];

}

}

if(data->playzone.who\_left != 0){ // F5

playground[data->playground\_id].who\_left = data->playzone.who\_left;

data->playzone.who\_left = playground[data->playground\_id].who\_left; // fix later

}

if(data->playzone.game\_end != 0)

playground[data->playground\_id].game\_end = data->playzone.game\_end;

if(data->playzone.count\_left != 0){

playground[data->playground\_id].count\_left = data->playzone.count\_left;

}

printf(">Playzone status: %d\n", data->playzone.status);

printf(">ID: %d\n", data->id);

printf(">FIG: %d\n", data->figure);

printf(">Game ready status: %d\n", data->game\_is\_ready);

printf(">Game ready notific: %d\n", data->ready\_notification);

printf(">Coords: %d %d\n", data->x\_coord, data->y\_coord);

printf(">Coords sended: %d\n", data->coords\_sended);

printf("PG ID: %d\n", data->playground\_id);

printf("mdF: %d\n", data->figure);

printf("dF: %d\n", data->playzone.current\_figure);

printf("pF: %d\n", playground[data->playground\_id].current\_figure);

printf("Game\_end\_data: %d\n", data->playzone.game\_end);

printf("Game\_end\_playg: %d\n", playground[data->playground\_id].game\_end);

printf("\n\nWHO\_LEFT: dt%d pz%d\n\n", data->playzone.who\_left, playground[data->playground\_id].who\_left);

printf(">-----------------------\n\n");

if (data->coords\_sended == 2) {

if (data->figure == data->playzone.current\_figure) {

printf("COORDS SENDED CHANGE\n");

data->playzone.status = playground[data->playground\_id].status;

Send\_msg(data, message, "Your turn now!(0)\n", server);

} else {

data->playzone.who\_left = playground[data->playground\_id].who\_left;

data->playzone.status = playground[data->playground\_id].status;

Send\_msg(data, message, "Waiting for enemy turn!\n", server);

}

}

if (data->coords\_sended == 1) {

if(playground[data->playground\_id].game\_end == 4){

data->playzone.game\_end = 4;

Send\_msg(data, message, "OTHER PLAYERS HAS LEFT THE GAME", server);

printf("Game finished!\n");

}

if (data->figure == playground[data->playground\_id].current\_figure) {

// if (playground[data->playground\_id].current\_figure < 3) {

// ++playground[data->playground\_id].current\_figure;

// ++data->playzone.current\_figure;

// } else {

// playground[data->playground\_id].current\_figure = 1;

// data->playzone.current\_figure = 1;

if (playground[data->playground\_id].current\_figure == 1) {

playground[data->playground\_id].current\_figure = 2;

data->playzone.current\_figure = 2;

if(playground[data->playground\_id].who\_left == 2){

playground[data->playground\_id].current\_figure = 3;

data->playzone.current\_figure = 3;

}

} else {

if (playground[data->playground\_id].current\_figure == 2) {

playground[data->playground\_id].current\_figure = 3;

data->playzone.current\_figure = 3;

if(playground[data->playground\_id].who\_left == 3){

playground[data->playground\_id].current\_figure = 1;

data->playzone.current\_figure = 1;

}

} else {

if (playground[data->playground\_id].current\_figure == 3) {

playground[data->playground\_id].current\_figure = 1;

data->playzone.current\_figure = 1;

if(playground[data->playground\_id].who\_left == 1){

playground[data->playground\_id].current\_figure = 2;

data->playzone.current\_figure = 2;

}

}

}

}

if(playground[data->playground\_id].who\_left != 0){

Send\_msg(data, message, ">>>WARNING:PLAYER LEFT THE GAME", server);

}

printf("\n\n\n>>>>>>>>>>>>>>>>dF: %d\n", data->playzone.current\_figure);

printf(">>>>>>>>>>>>>>>>pF: %d\n", playground[data->playground\_id].current\_figure);

printf("Client ID: %d\n", data->id);

printf("Coordinates: %d %d\n", data->x\_coord, data->y\_coord);

if ((data->x\_coord >= 0) && (data->y\_coord >= 0)) {

data->playzone.who\_left = playground[data->playground\_id].who\_left;

data->playzone.count\_left = playground[data->playground\_id].count\_left;

if(PlayzoneIsFull(&playground[data->playground\_id]) == 0){

playground[data->playground\_id].coord[data->x\_coord][data->y\_coord] = data->figure;

data->playzone.coord[data->x\_coord][data->y\_coord] = data->figure;

if(PlayzoneCheckForWinner(&playground[data->playground\_id], data->x\_coord, data->y\_coord) == 1){

data->playzone.status = 4;

playground[data->playground\_id].status = 4;

Send\_msg(data, message, "", server);

}

} else {

//TODO

printf("Game finished!\n");

//return 0;

}

}

for (int i = 0; i < DIM; ++i) {

for (int j = 0; j < DIM; ++j) {

data->playzone.coord[i][j] = playground[data->playground\_id].coord[i][j];

}

}

PlayzonePrint(&playground[data->playground\_id]);

printf("\n\n1MY PG STATUS IS: %d\n", playground[data->playground\_id].status);

printf("2MY PG STATUS IS: %d\n\n", data->playzone.status);

printf("!!!!MSG0\n");

Send\_msg(data, message, "Nice turn!\n", server);

} else {

printf("~~~~~~~~~~~~~~~~~~\n");

Send\_msg(data, message, "It's not your turn now!\n", server);

}

} else {

if (data->id == 0) {

printf("New user!\n");

player\_id++;

playersInQueue++;

data->id = player\_id;

data->playground\_id = playgroud\_iterator;

printf(">Number of players: %ld\n", player\_id);

if ((playersInQueue != 3) && (data->game\_is\_ready == 0)) {

sprintf(info, "%s%d%c", "Sorry, you wasn't registred. Now you're allowed to play!\nYour ID:",

data->id, '\0');

printf("!!!!MSG1\n");

Send\_msg(data, message, info, server);

}

}

// Checking if game is ready (for all players)

if ((data->playzone.status == 1) && (data->ready\_notification == 0)) {

data->game\_is\_ready = 1;

data->ready\_notification = 1;

printf("!!!!MSG2\n");

Send\_msg(data, message, "Your game is ready!(1)\0", server);

}

if ((playersInQueue != 3) && (data->game\_is\_ready == 0)) {

data->playzone = playground[playgroud\_iterator];

data->playzone.status = playground[data->playground\_id].status;

printf(">Players in queue: %ld\n", playersInQueue);

printf("!!!!MSG3\n");

Send\_msg(data, message, "Waiting for other players...\0", server);

}

if (playersInQueue == 3) {

if ((data->coords\_sended == 3) || (playground[data->playground\_id].status == 5)){

playground[data->playground\_id].status = 5;

data->playzone.status = playground[data->playground\_id].status;

Send\_msg(data, message, "Somebody left the game(2)!\n", server);

}

printf("Game %d is ready!\n", playgroud\_iterator);

PlayzoneInit(&playground[playgroud\_iterator]);

PlayzonePrint(&playground[playgroud\_iterator]);

playground[playgroud\_iterator].status = 1;

data->playzone = playground[playgroud\_iterator];

data->game\_is\_ready = 1;

playgroud\_iterator++;

playersInQueue = 0;

data->ready\_notification = 1;

printf("!!!!MSG4\n");

Send\_msg(data, message, "Your game is ready!(2)\0", server);

}

}

data->coords\_sended = 0;

}

sleep(1);

}

zmq\_close(server);

zmq\_ctx\_destroy(context);

printf("Server is down!\n");

return 0;

}

**ARG.H**

#ifndef ARG\_H

#define ARG\_H

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <time.h>

#include "zmq.h"

#define SIZE\_LOG 100

#define MSG\_SIZE 2048

#define DIM 5 //dimension of playgroud

#define nan 0

#define x 1

#define o 2

#define n 3

#define error(msg) \

do { perror(msg); exit(EXIT\_FAILURE); } while (0)

typedef struct{

int game\_end; // 0 by deflault

// 1,2,3 - players

// 4 - playzone is full

int current\_figure;

int status;

int coord[DIM][DIM];

int who\_left;

int count\_left;

}Playzone;

typedef struct \_msg{

int id;

Playzone playzone;

int playground\_id;

int game\_is\_ready;

int ready\_notification;

int coords\_sended;

int x\_coord;

int y\_coord;

int figure;

void\* requester;

char message[50];

int trash[100];

} Message\_t;

void PlayzonePrint(Playzone\* p){

printf(" 0 1 2 3 4\n");

for (int i = 0; i < DIM; ++i) {

printf("%d ", i);

for (int j = 0; j < DIM; ++j) {

if(p->coord[i][j] == 0) printf("| \_ ");

if(p->coord[i][j] == 1) printf("| x ");

if(p->coord[i][j] == 2) printf("| o ");

if(p->coord[i][j] == 3) printf("| □ ");

}

printf("\n");

}

}

int PlayzoneIsFull(Playzone\* p){

int null\_counter = 0;

for(int i = 0; i < DIM; i++){

for(int j = 0; j < DIM; j++){

if(p->coord[i][j] == nan) null\_counter++;

}

}

if(null\_counter > 0){

printf("~~~ Null counter: %d\n", null\_counter);

return 0;

}

printf("~~~Zone is full\n");

return 1;

}

int PlayzoneCheckForWinner(Playzone\* p, int x\_coord, int y\_coord){

int curr\_fig = p->coord[x\_coord][y\_coord];

int bingo = 1;

// Right diag

if( ((x\_coord-1) >= 0) && ((y\_coord+1) < DIM) ){

if(p->coord[x\_coord-1][y\_coord+1] == curr\_fig){

bingo++;

if( ((x\_coord-2) < DIM) && ((y\_coord+2) >= 0) ){

if(p->coord[x\_coord-2][y\_coord+2] == curr\_fig){

bingo++;

}

}

}

}

if( ((x\_coord+1) < DIM) && ((y\_coord-1) >= 0) ){

if(p->coord[x\_coord+1][y\_coord-1] == curr\_fig){

bingo++;

if( ((x\_coord+2) < DIM) && ((y\_coord-2) >= 0) ){

if(p->coord[x\_coord+2][y\_coord-2] == curr\_fig){

bingo++;

}

}

}

}

if(bingo >= 3){

printf("WINNER WINNER CHICKEN DINNER!!!\n");

return 1;

} else bingo = 1;

// Left diag

if( ((x\_coord-1) >= 0) && ((y\_coord-1) >= 0) ){

if(p->coord[x\_coord-1][y\_coord-1] == curr\_fig){

bingo++;

if( ((x\_coord-2) >= 0) && ((y\_coord-2) >= 0) ){

if(p->coord[x\_coord-2][y\_coord-2] == curr\_fig){

bingo++;

}

}

}

}

if( ((x\_coord+1) < DIM) && ((y\_coord+1) < DIM) ){

if(p->coord[x\_coord+1][y\_coord+1] == curr\_fig){

bingo++;

if( ((x\_coord+2) >= 0) && ((y\_coord+2) >= 0) ){

if(p->coord[x\_coord+2][y\_coord+2] == curr\_fig){

bingo++;

}

}

}

}

if(bingo >= 3){

printf("WINNER WINNER CHICKEN DINNER!!!\n");

return 1;

} else bingo = 1;

// Gor

if((x\_coord-1) >= 0){

if(p->coord[x\_coord-1][y\_coord] == curr\_fig){

bingo++;

if((x\_coord-2) >= 0){

if(p->coord[x\_coord-2][y\_coord] == curr\_fig){

bingo++;

}

}

}

}

if((x\_coord+1) < DIM){

if(p->coord[x\_coord+1][y\_coord] == curr\_fig){

bingo++;

if((x\_coord+2) < DIM){

if(p->coord[x\_coord-2][y\_coord] == curr\_fig){

bingo++;

}

}

}

}

if(bingo >= 3){

printf("WINNER WINNER CHICKEN DINNER!!!\n");

return 1;

} else bingo = 1;

// Vert

if((y\_coord-1) >= 0){

if(p->coord[x\_coord][y\_coord-1] == curr\_fig){

bingo++;

if((y\_coord-2) >= 0){

if(p->coord[x\_coord][y\_coord-2] == curr\_fig){

bingo++;

}

}

}

}

if((y\_coord+1) < DIM){

if(p->coord[x\_coord][y\_coord+1] == curr\_fig){

bingo++;

if((y\_coord+2) < DIM){

if(p->coord[x\_coord][y\_coord+2] == curr\_fig){

bingo++;

}

}

}

}

if(bingo >= 3){

printf("WINNER WINNER CHICKEN DINNER!!!\n");

return 1;

} else bingo = 1;

return 0;

}

#endif

**Вывод**:

Данная программа реализует заданную игру в полном объеме. При отладке программы было принято много спорных решений по вопросам эффективности для обеспечения корректной и стабильной работы компонентов. Большинство из недостатков программы можно заменить путем значительного усложнения алгоритмической модели.

Возможно, стоило бы заменить и паттерн очереди сообщений на более быстрый аналог, но я считаю, что *Request-Response* вполне неплохо справляется с поставленной задачей, а также, самое главное, интуитивно прост в использовании.

В течение работы над данным проектом я смог не раз потренироваться в отладке программ, а также получил опыт разработки более-менее среднего по объему приложения. Данная работа была весьма полезна для оттачивания навыка программирования. Если же рассматривать саму суть работы, то я считаю, что она не охватывает множество тем, рассказанных в течение курса, а развивает именно одну – тему очередей сообщений(одну из самых полезных, как мне кажется). Конечно, можно было бы искусственно «прикрутить» и другие разделы, но тогда бы пострадала логическая целостность получившегося продукта.