Istanbul University
Computer Engineering
Lesson: Ağ Programlama
Project: Catch The Figure

Emre Erinç - 1306150016
May 2019

Description of this document

This document specifies how it works on server and client for lesson project. Socket operations doing with Java Netty Framework, GUI design and operations doing with Java Swing API.

Table of Contents

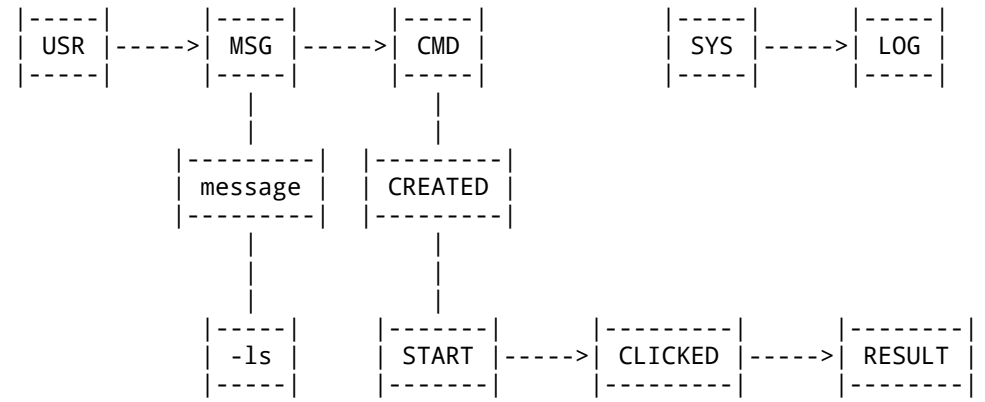
1. Introduction....1
2. States....1
3. States Descriptions....2
3.1. USR....2
3.2. MSG....3
3.3. CMD....4
3.4. SYS....5
3.5. LOG....5

## 1. Introduction

Created a protocol for a lesson project. Protocol based on Java Netty Framework for socket messages.

## 2. States

USR
MSG
-ls
message
CMD
CREATED
CLICKED
START
RESULT
SYS



## 3. State Descriptions

### 3.1. USR

Accept any user want to connect to server with nick name. If user the first member to connect to server, server assign him/her as a HOST. Other members assign as a SUBSCRIBER.

Participant Message : Member who want to join lobby, should enter their nick
"[USR] – *nick*"

Server Message : When any member join the lobby, see this welcome message
"[SYS] - " + "WELCOME TO LOBBY!\r\n" +
"[SYS] - " + "YOUR INFO : *ip:port*\r\n" +
"[SYS] - Server Time : *serverTime*\r\n" +
"[SYS] - Client Role : <<HOST>> or <<SUBSCRIBER>> \r\n"

Server Message : If any member join the lobby, other members views this message
"[SYS] - A new user has joined\n"

Server Message : If any member disconnect from server, other member views this message.
"[SYS] - <<*nick*>> has left\n"

### 3.2. MSG

Communication messages between members.

Participant Message : Want to send message
"[MSG] *message*\r\n"

Server Message : Lobby members take messages and shows to client GUI
"[MSG] - <<*sender*>> *message*\n"

If any member want learn how many member at lobby, message need to contain "-ls" parameter.

Server Message : Only message sender views result *userList*
"Connected User List : *userList*\n"

### 3.3. CMD

CREATED : Give figure creation info from host user and send this information to all participants.

Host Message : Host client app generate a figure and send to server

```
"[CMD] - CREATED " +
"| Type : <<figureType>> " +
"| Color : <<figureColor>> " +
"| Bounds : <<"figureBounds>> " +
"| FigureCreation : <<figureCreatedAt>> \r\n"
```

Server Message : Client app generates created figure except for host

```
"[SYS] - <<host>> CREATED " +
"| Type : <<figureType>> " +
"| Color : <<figureColor>> " +
"| Bounds : <<figureBounds>> " +
"| FigureCreation : <<figureCreatedAt>> \r\n"
```

START : Take game start command from host user and send to all participants.

Host Message : Host sends a start game request

```
"[CMD] - START " +
"| Interval : <<figureCreationInterval>> " +
"| X : <<windowHeigth>> " +
"| Y : <<windowWidth>> " +
"| FigureCount : <<figureCount>> " +
"| Points : <<figurePoints>>\r\n"
```

Server Message : Client app starts game with given parameters

```
"[SYS] - <<host>> STARTED " +
"| Interval : <<figureCreationInterval>> " +
"| X : <<windowHeigth>> " +
"| Y : <<windowWidth>> " +
"| FigureCount : <<figureCount>> " +
"| Points : <<figurePoints>>\r\n"
```

CLICKED : Give figure click info from any user and send this information to all participants.

User Message : Send any figure click to server

```
"[CMD] - CLICKED " +
"| Time : <<clickTime>> " +
"| FigureCreation : <<figureCreatedAt>> \r\n"
```

Participant Message From Server : Clicked figure removes from all members

```
"[SYS] - <<sender>> CLICKED " +
"| Time : <<clickTime>> " +
"| FigureCreation : <<figureCreatedAt>> \r\n"
```

Participant Message From Server (Reject – to sender) :

```
"[SYS] - ALREADY_CLICKED\n"
```

RESULT : Calculate clicked figure count by server and control it. If every figures were CLICKED status send to all participants COLLECT message and participants sends there points. Server collect them and send as a RESULT.

Server Message :

```
"[SYS] – COLLECT"
```

Participant Message :

```
"[CMD] - RESULT | Points : <<points>> \r\n"
```

Server Message :

```
"[SYS] - RESULT - resultList \r\n"
```

### 3.4. SYS

This states includes all server based messages.

### 3.5. LOG

This is not exactly state. Just for logging any transaction for serverside.

Server Log :

```
"[LOG] - <<incomingMessageAddress>> " +
" all other below messages \r\n"
```