UNIVERSITY OF TARTU

Institute of Computer Science

MARKUS3 – Multiplayer Battleship Game

User Manual

Annabell Kuldmaa, Markus Lippus, Robert Roosalu, Heikki Saul

Contents

Introduction 3

System Requirements 3

System Dependencies 3

Set up process 4

Setting up RabbitMQ 4

Setting up the game 4

How to play? 6

Pre-processing 6

Starting the game and logging in 6

Gameplay 7

Leaving and disconnecting 9

Features and known errors 9

Troubleshooting 9

Introduction

This document is a user manual for MARKUS3 – a multiplayer battleship game application. In the following system requirements, setup process and game instructions are described. MARKUS3 does not rely on TCP not UDP, it uses indirect communication paradigm instead, i.e., it uses message queues for communication. Furthermore, in MARKUS3 indirect communication paradigm is combined with RPC.

System Requirements

MARKUS3 requires a (mostly) POSIX compliant operating system to work correctly. Any standard (Linux Standard Base compatible) Linux distribution can be used.

System Dependencies

MARKUS3 requires a standard distribution of Python 2.7 with a number of built-in services to work. Please check the availability of required services if using any non-standard Python distribution. The list following built-in services and their respective dependencies are required:

- uuid
- json
- thread
- sys
- struct
- os
- random

In addition to the aforementioned built-in services, the following external libraries and operating system services are required:

- pika
- numpy
- readline
- termios
- fcntl
- time

MARKUS3 uses the RabbitMQ message broker software. A working RabbitMQ server must be available on the network. Refer to Set up process for a setup guide.

Set up process

Setting up RabbitMQ

To run MARKUS3, a RabbitMQ server is needed to broker messages between clients and servers. Executables and buildable binaries for different operating systems are available at RabbitMQ webpage. Documentation regarding version changes, setup and configuration are also available at the RabbitMQ webpage.

Setting up the game

MARKUS3 has three main components – a login server (`login_server.py`), game servers (`game_server.py`) and clients (`client.py`). The object model of the system is presented as Figure 1 to give more insight into the architecture.

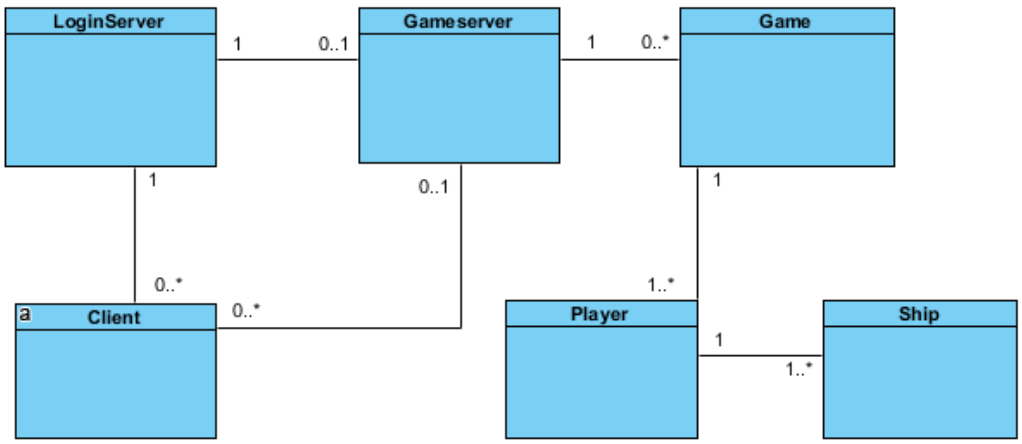

Figure 1. Object model

- The login server handles access to game servers. Only one login server is needed on a network for any number of players.
- The game server acts as a lobby, allowing players to join or create games. The game server can handle multiple games at a time, from any number of players and there can be multiple game servers handled by one login server. Once a player has joined a game server, he/she can choose to join any of the games on that particular server. Switching between game servers is only possible with a new login.
- The client handles all player-side actions, such as logging in, joining or hosting games and the game proper.

- If there is no login server running, a player must first host a login server. The same is true with game servers. If there is no game server on the network, any player must start a game server. Any subsequent players can choose to join that existing server or start their own game servers. If there are both a running login server and a running game server and the player does not want to start a new game server, he/she should start the client to join or create a game. Any player hosting either a login server and/or a game server should also start the client for player-side actions.

How to play?

MARKUS3 uses a command line interface and requires any (mostly) POSIX compliant terminal.

Pre-processing

To host login server run `login_server.py` and to host a game server run `game_server.py`.

Starting the game and logging in

To play MARKUS3, please run `client.py`. Upon starting `client.py`, you are displayed a list of available game servers:

```
NAME        KEY
Server1     GAMESERVER1
Server2     GAMESERVER2
Which server to join?

>Server2
```

Next you must choose a game server to log in and insert a user name. Note that the username must be unique within the game server. If your selected username is already taken, you will be prompted to insert a new one. After choosing which game server to join, you will see the game lobby view, with a list of available games and selection to either join an existing game or start hosting a new game session.

You can join an existing game as follows:

```
Joined Server2

Available games

GAME_1

GAME_2

Do you want to join a session or host a new session? [J]/[H]

>J

Which game?

>GAME_1

Joined GAME_1. Please wait for the host to start the game.
```

You can host a new game (become owner of the game) as follows:

```
Joined Server1

Available games

GAME_1

GAME_2

Do you want to join a session or host a new session? [J]/[H]

>H

Choose the playing field size

>10

Joined to a game. Incoming queue registered to game exchange.

Waiting for players. Enter 'start' to begin game.

Player ROBERT has joined your game.

Player HEIKKI has joined your game.
```

Please note, that if there are no game sessions on the selected game server, it is only possible to host a new game. In order to start the game, the owner must enter

```
>start

Starting game!
```

Gameplay

Gameplay in MARKUS3 is reminiscent of the classic Battleships game. You are shown 2 versions of the playing field – one with your ships and another marking your shots.

You are given two ships – a boat and a very small boat. The server will place the ships on the field for you.

Your mission in the game is to be the last player with ships still on the field.

Your main weapon is a mighty artillery salvo from your warships. This salvo can heavily damage any ship and sink even the mightiest ships in a few hits. To use your artillery salvo, give the command to shoot, followed by the x- and y-coordinates of the target.

The command

```
>shoot 3 0
```

tells your artillery to fire a salvo at the target located at the x-coordinate 3 and the y-coordinate 0.

Your tactical grid-maps shows the locations of your ships and feedback on your artillery salvos. The locations of your ships are denoted by `#` on the tactical grid-map. A section of your ship, that is hit by enemy fire is denoted by `*`. A hit on an enemy ship is denoted by `x` and a missing shot by `o`.

```
Your Ships                    Your Shots

\  0  1  2  3  4  5  6  7  8  9     \  0  1  2  3  4  5  6  7  8  9
0  .  .  #  #  *  #  #  .  .  .     0  .  .  .  x  .  .  .  .  .  .
1  .  #  #  #  #  .  .  .  .  .     1  .  .  .  x  .  .  .  o  .  .
2  .  .  #  .  .  .  .  .  .  .     2  .  .  .  .  .  .  .  o  .  .
3  .  .  #  .  .  .  .  .  .  .     3  .  .  .  .  .  .  .  x  .  .
4  .  .  #  .  .  .  .  .  .  .     4  .  .  .  .  .  .  .  .  .  .
5  .  .  .  .  .  .  .  .  .  #     5  .  .  .  .  .  .  .  .  .  .
6  .  .  .  .  .  .  .  .  .  *     6  .  .  .  .  .  .  .  .  .  .
7  .  .  .  #  .  .  .  .  .  #     7  .  .  .  .  .  .  .  .  .  .
8  .  .  .  #  .  .  .  .  #  .     8  .  .  .  .  .  .  .  .  .  .
9  .  .  .  .  .  .  .  .  .  .     9  .  .  .  .  .  .  .  .  .  .
```

You will get textual feedback if anyone hits your ships

```
One of your ships is under attack by ROBERT
```

```
or when any ship from any player is sunk
```

```
Player ROBERT sunk player HEIKKI ship.
```

If another player succeeded to sunk some other player's ship, you will also receive a notification.

Leaving and disconnecting

You can leave the game at any time with the following command:

```
> leave
```

If you were disconnected during a game, you can rejoin this game if the game is not over yet. The disconnecting handling feature can be tried by Ctrl+C in the client.

Note: If you get disconnected while it is your turn and you are not the host, it may happen that on reconnection you cannot continue playing.

Features and known errors

All the features in the requirements list have been implemented, with the exception of connection timeout handling and as a result, the owners' ability to remove people from the game.

The current architecture limits the ability to handle a complicated state space. Therefore, in some specific states some features might display erroneous behavior.

Troubleshooting

If you run into any errors when running MARKUS3, please check that your system/network fulfills the following:

- Operating system requirements and necessary services
- Python 2.7 and necessary packages
- RabbitMQ server running
- Login server and game server available
- Check if the error is noted in the Features and known errors section