Pentago Game Specification Document

CT30A6000 Communications Software, Protocols and Architectures

Group 03

Joonas Blomberg, 0342152

Arttu Hanska, 0342181

Table of Contents

1. System Overview

The application is an implementation of the game named Pentago. It's a two player game that is played on a 6x6-tile board that is divided into four 3x3-tile sub-boards. The players take turns placing marbles on the tiles and then rotating one sub-board. The rotation is 90° in any direction and is done after player's own turn. The winner is the one that first has a vertical, horizontal or diagonal line of five marbles.

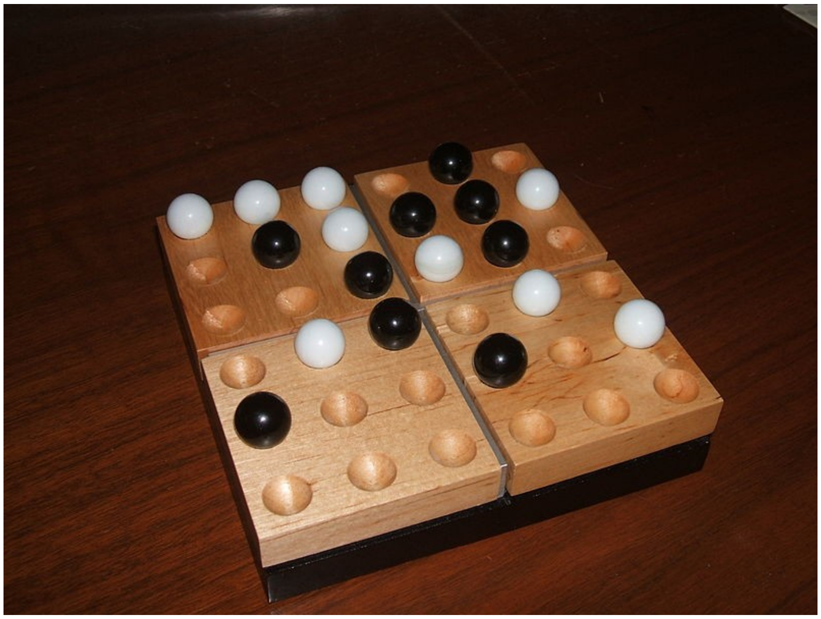


Illustration 1: Pentago board on winning position for white [1]

The system has two clients that are connected to one server. There can be many games on the server at any one time. The server handles the game logic and a client handles the user interface. The connection is run by TCP [2].

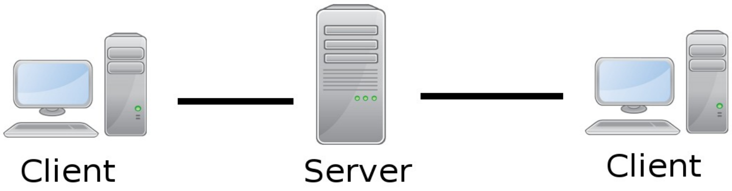


Illustration 2: System overview

2. Use Case Diagram

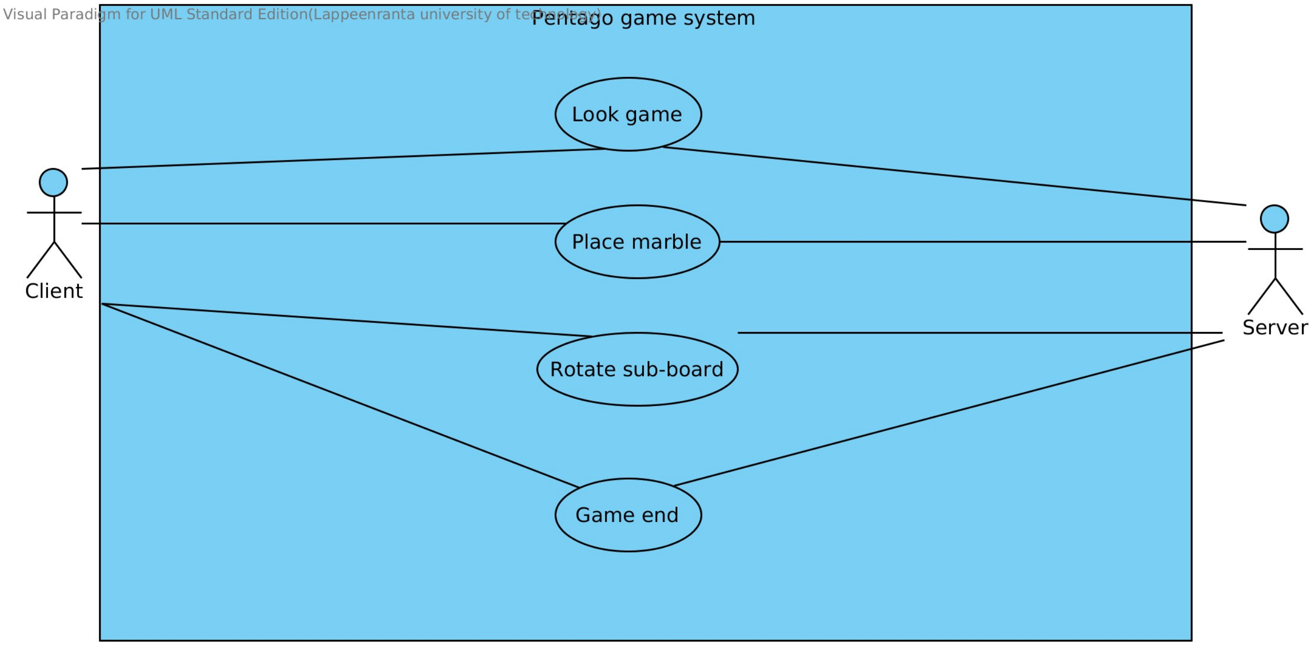

Visual Paradigm for UML Standard Edition(Lappeenranta university of technology)
Pentago game system
Look game
Place marble
Rotate sub-board
Game end
Client
Server

Illustration 3: Pentago game system use cases

3. Sequence Diagrams

3.1 Create game sequence diagram

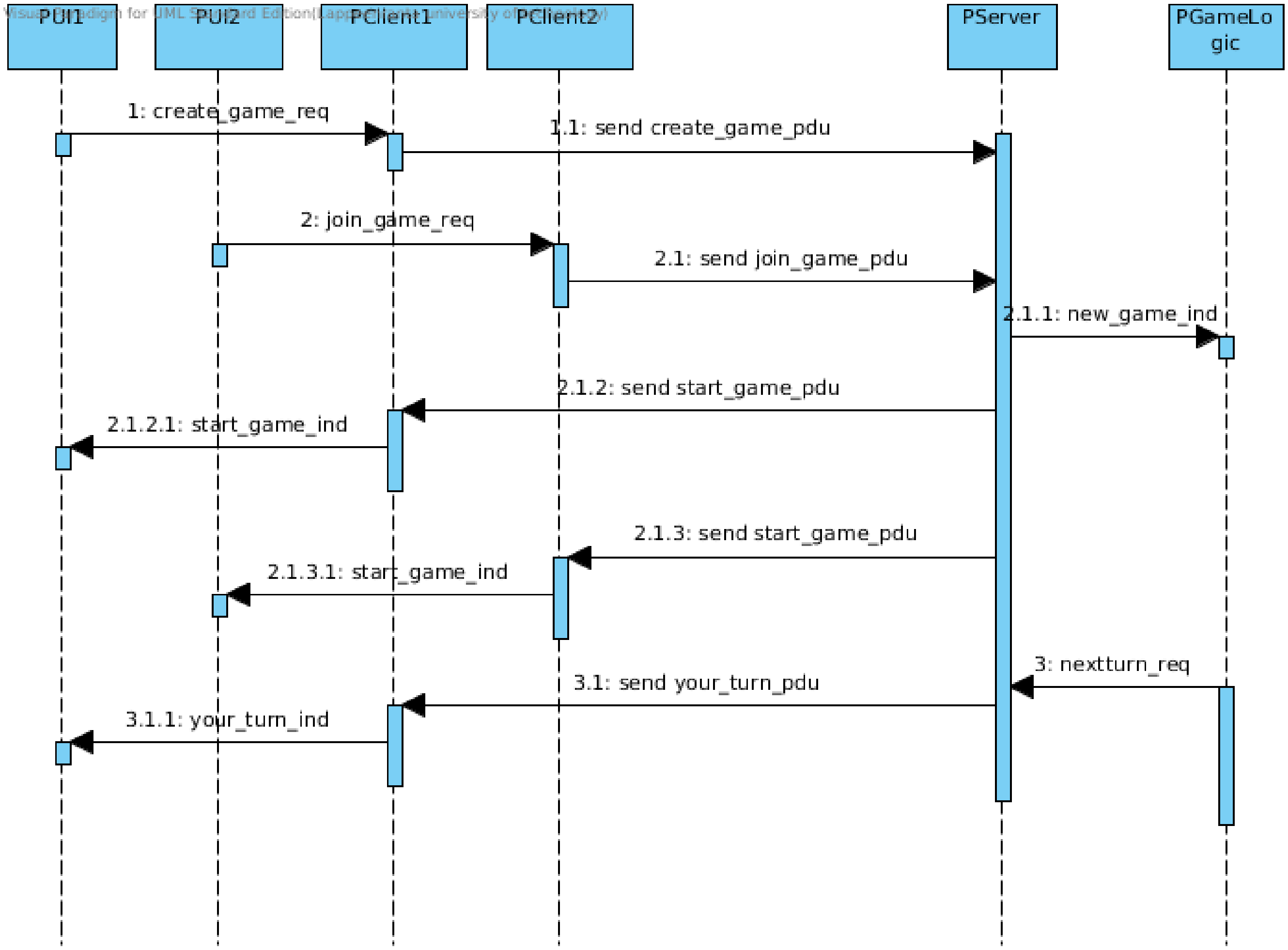


Illustration 4: Create game sequence diagram

3.2 Place marble sequence diagram

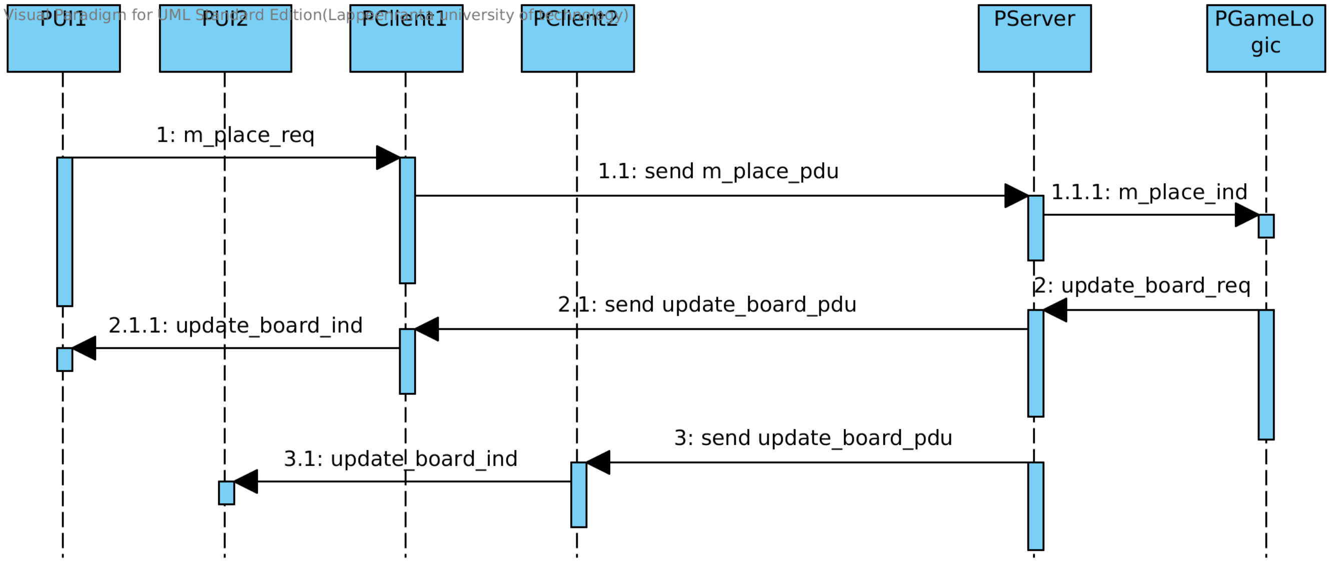


Illustration 5: Place marble sequence diagram

3.3 Game error sequence diagram

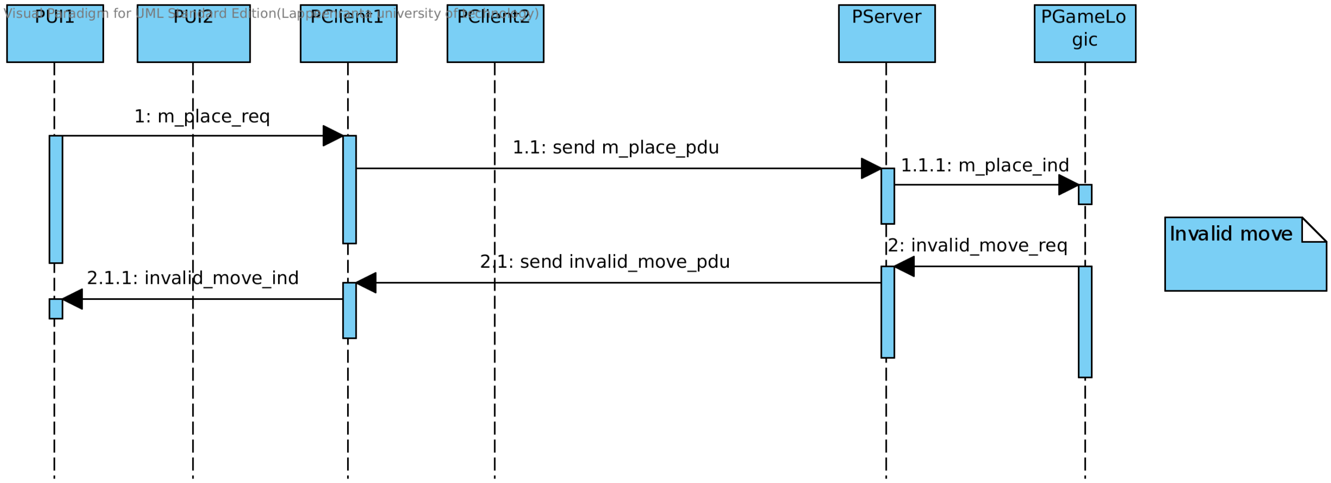


Illustration 6: Place marble error sequence diagram

3.4 Rotate sub-board sequence diagram

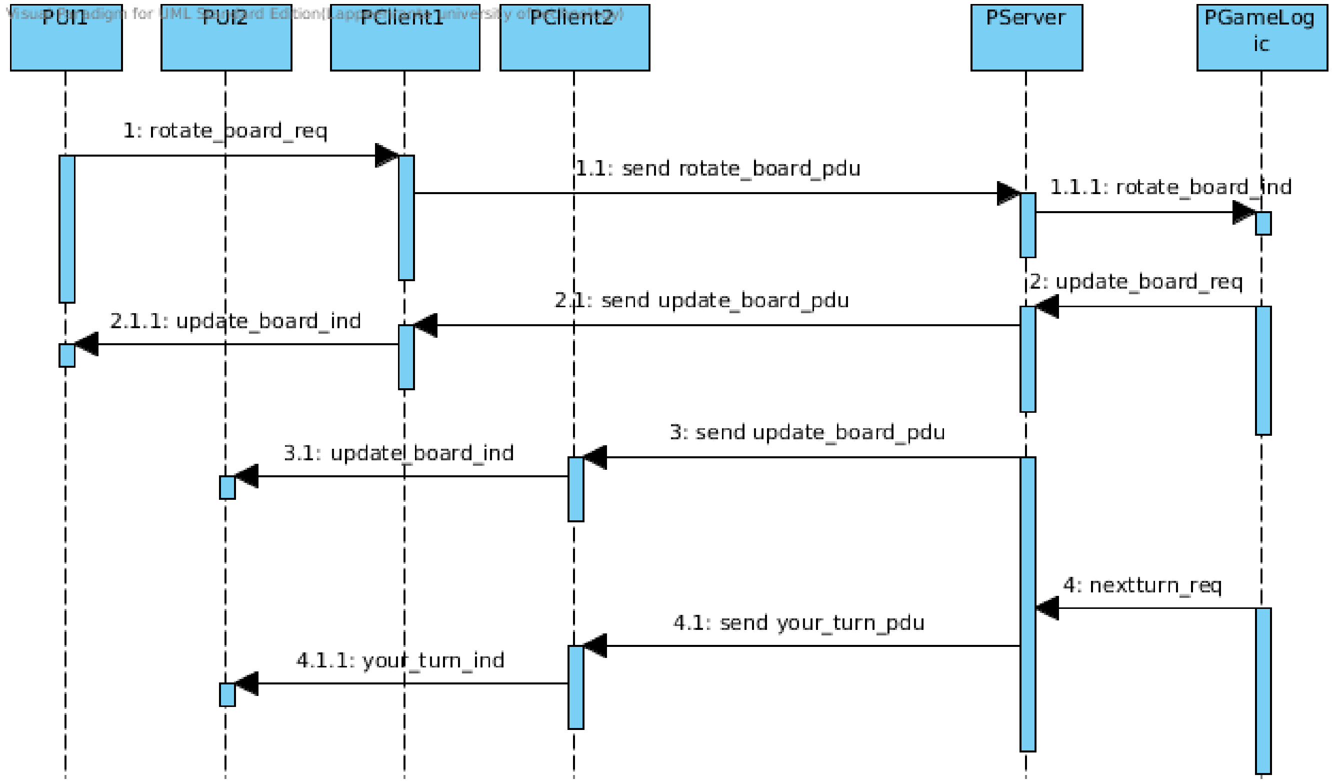


Illustration 7: Rotate sub-board sequence diagram

3.5 Game end sequence diagram

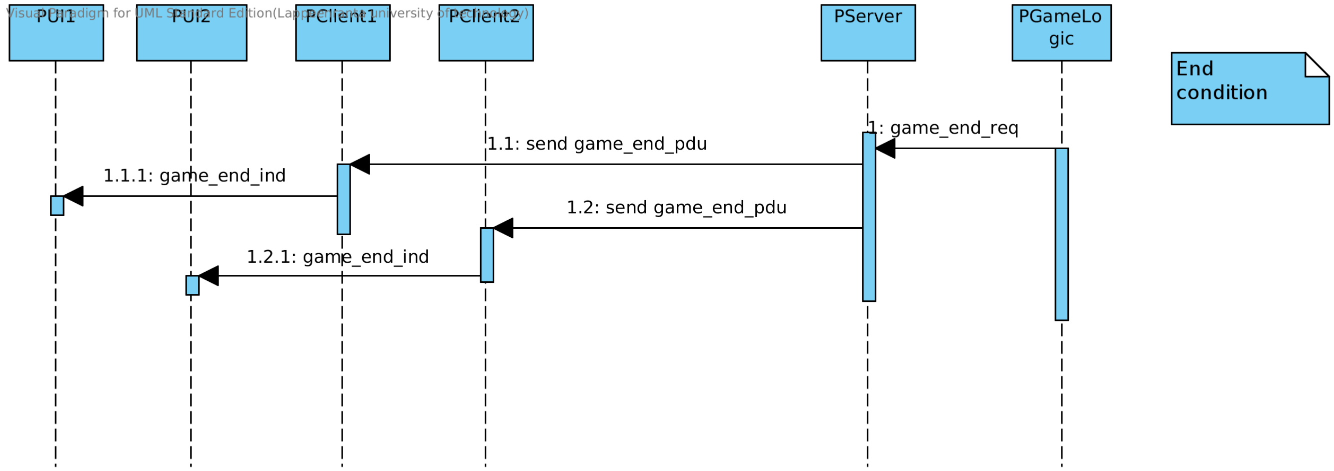


Illustration 8: Game end sequence diagram

4. Game service Layer

4.1. Functional layer requirements

ID	Description
F01	Pentago game logic is handled by server
F02	Provides game error handling managed by server
F03	Server can run multiple parallel sessions
F04	Provides game piece placement for a Pentago game
F05	Provides sub-board rotation for a Pentago game
F06	Provides end game contition for a Pentago game
F07	Provides turn indication for a Pentago game
F08	Provides logic to join Pentago game against other player

4.2. Layer constraints

ID	Description
C01	Transmission is handled by TCP

5. Layer model

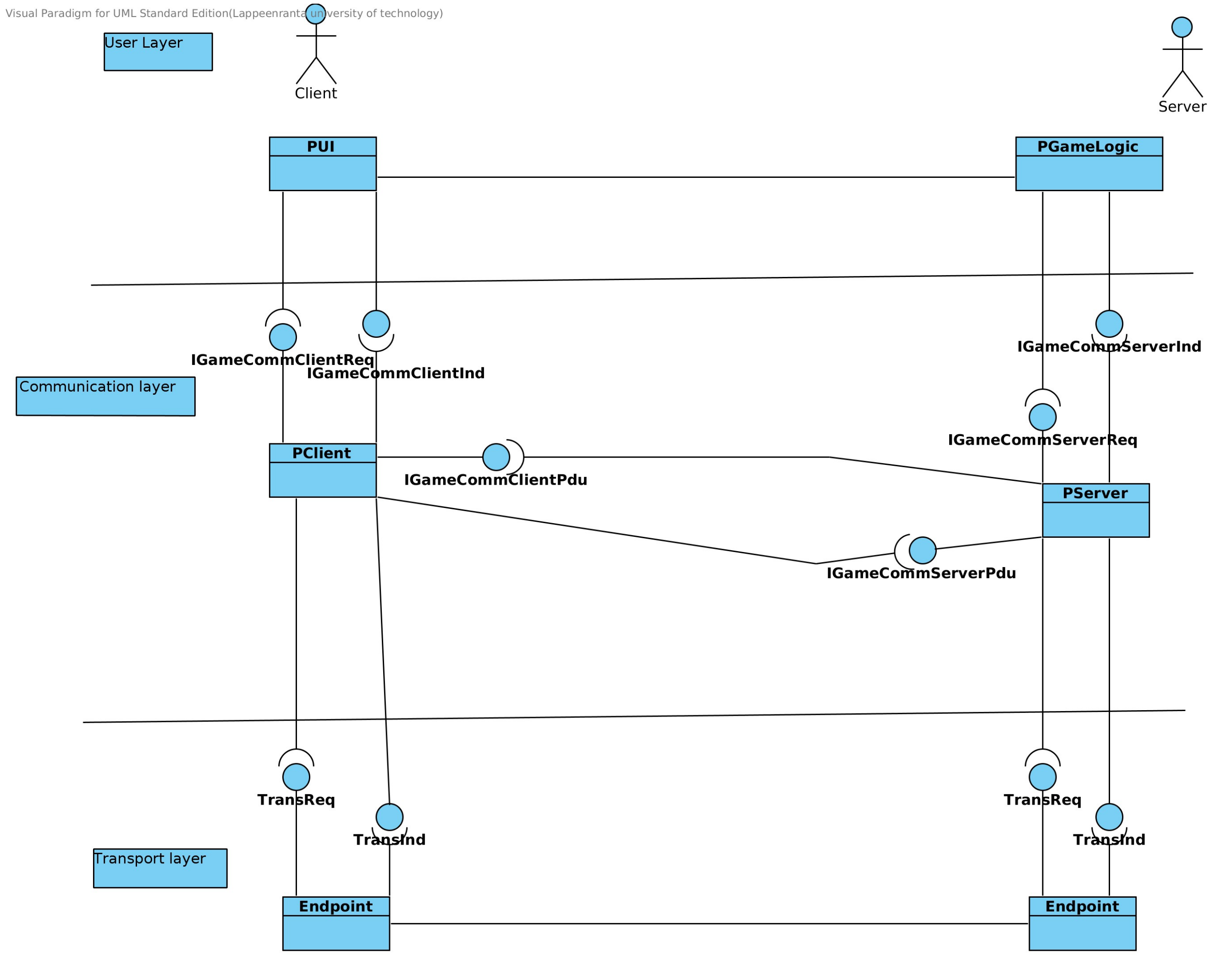


Illustration 9: Layer model entities

6. Interfaces

6.1 Client interfaces

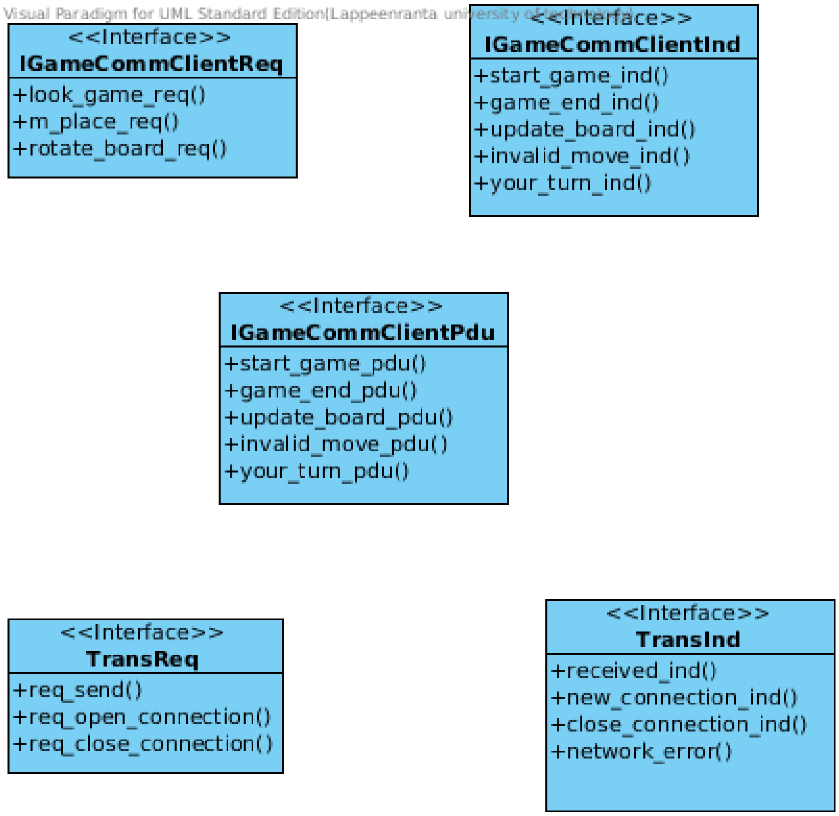


Illustration 10: Client interfaces

6.2 Server interfaces

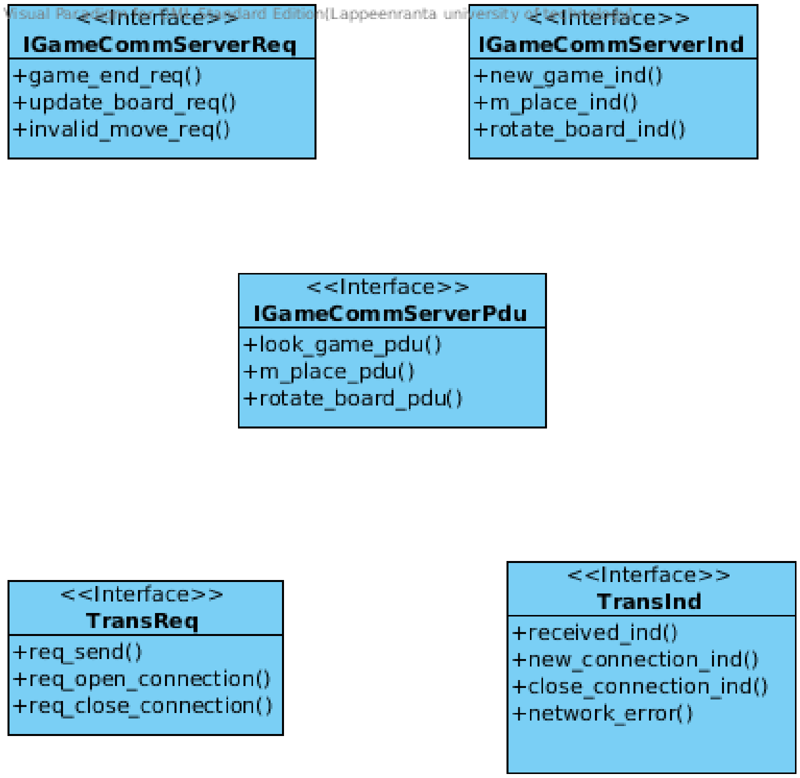


Illustration 11: Server interfaces

7. State Machine Diagrams

7.1. Client State Machine Diagram

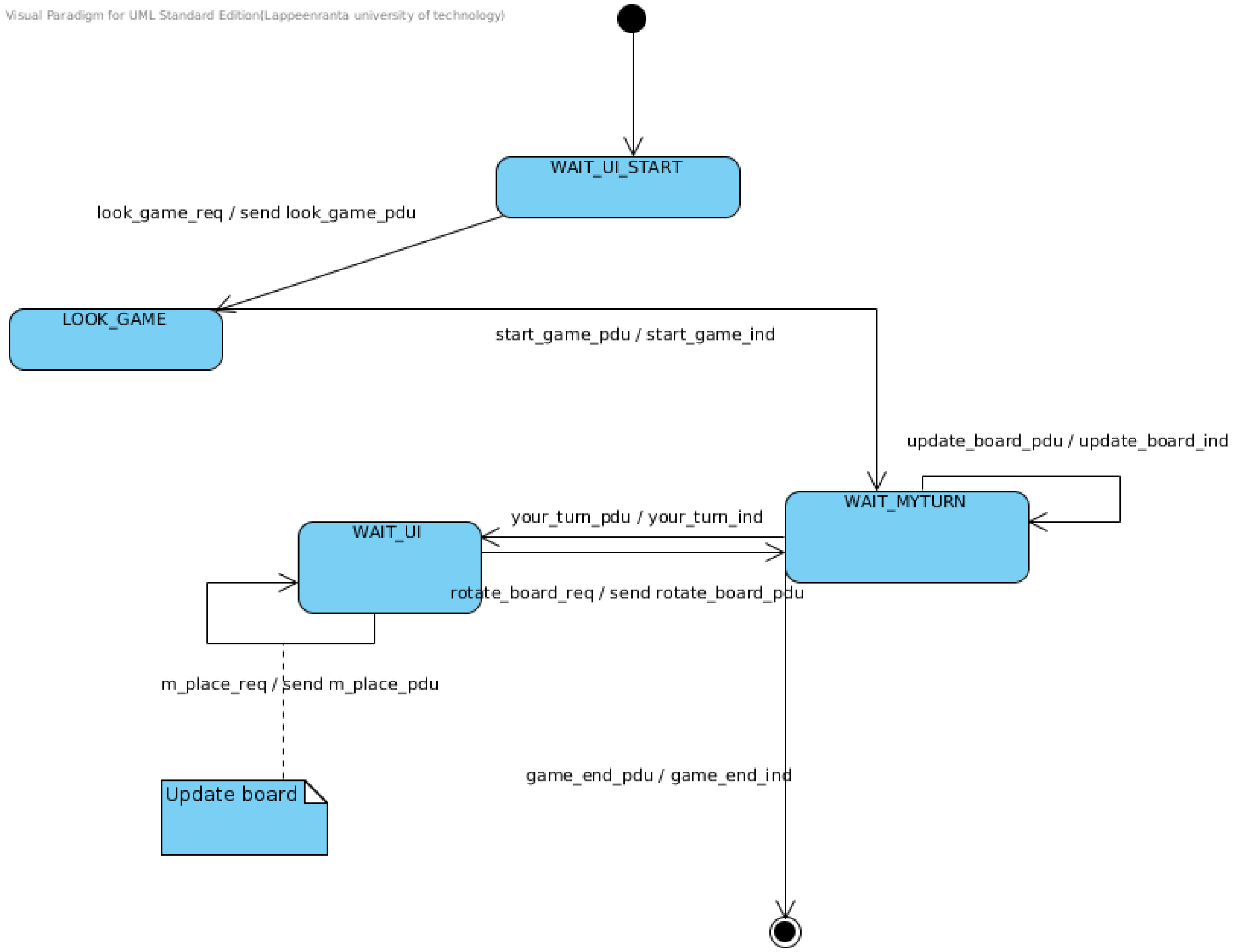
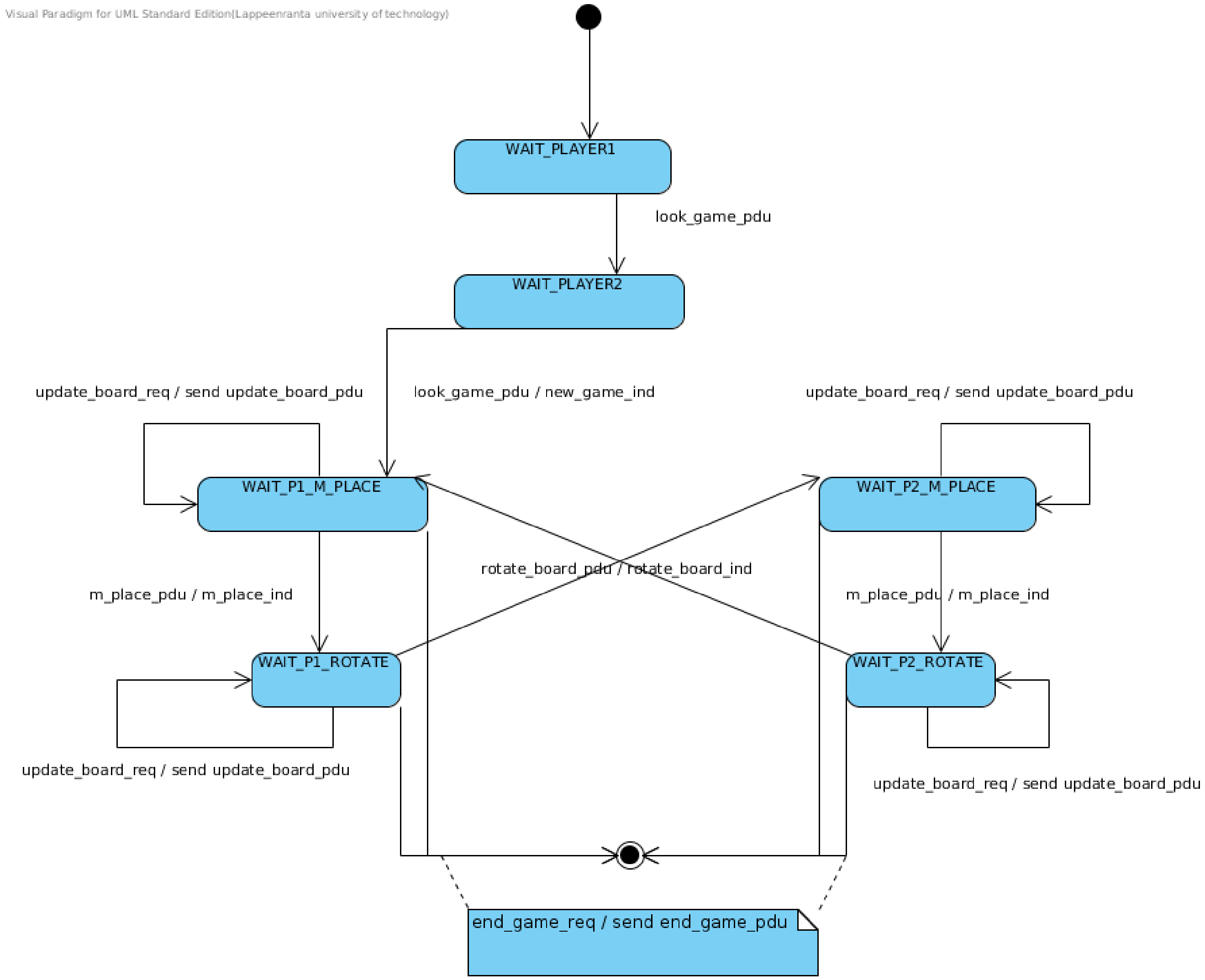

Visual Paradigm for UML Standard Edition(Lappeenranta university of technology)
WAIT_UI_START
look_game_req / send look_game_pdu
LOOK_GAME
start_game_pdu / start_game_ind
update_board_pdu / update_board_ind
WAIT_MYTURN
WAIT_UI
your_turn_pdu / your_turn_ind
rotate_board_req / send rotate_board_pdu
m_place_req / send m_place_pdu
Update board
game_end_pdu / game_end_ind

Illustration 12: Client state machine diagram

7.2. Server State Machine Diagram



Visual Paradigm for UML Standard Edition(Lappeenranta university of technology)
WAIT_PLAYER1
look_game_pdu
WAIT_PLAYER2
update_board_req / send update_board_pdu
look_game_pdu / new_game_ind
update_board_req / send update_board_pdu
WAIT_P1_M_PLACE
WAIT_P2_M_PLACE
rotate_board_pdu / rotate_board_ind
m_place_pdu / m_place_ind
m_place_pdu / m_place_ind
WAIT_P1_ROTATE
WAIT_P2_ROTATE
update_board_req / send update_board_pdu
update_board_req / send update_board_pdu
end_game_req / send end_game_pdu

Illustration 13: Server state machine diagram

8. Abstract Message Definitions

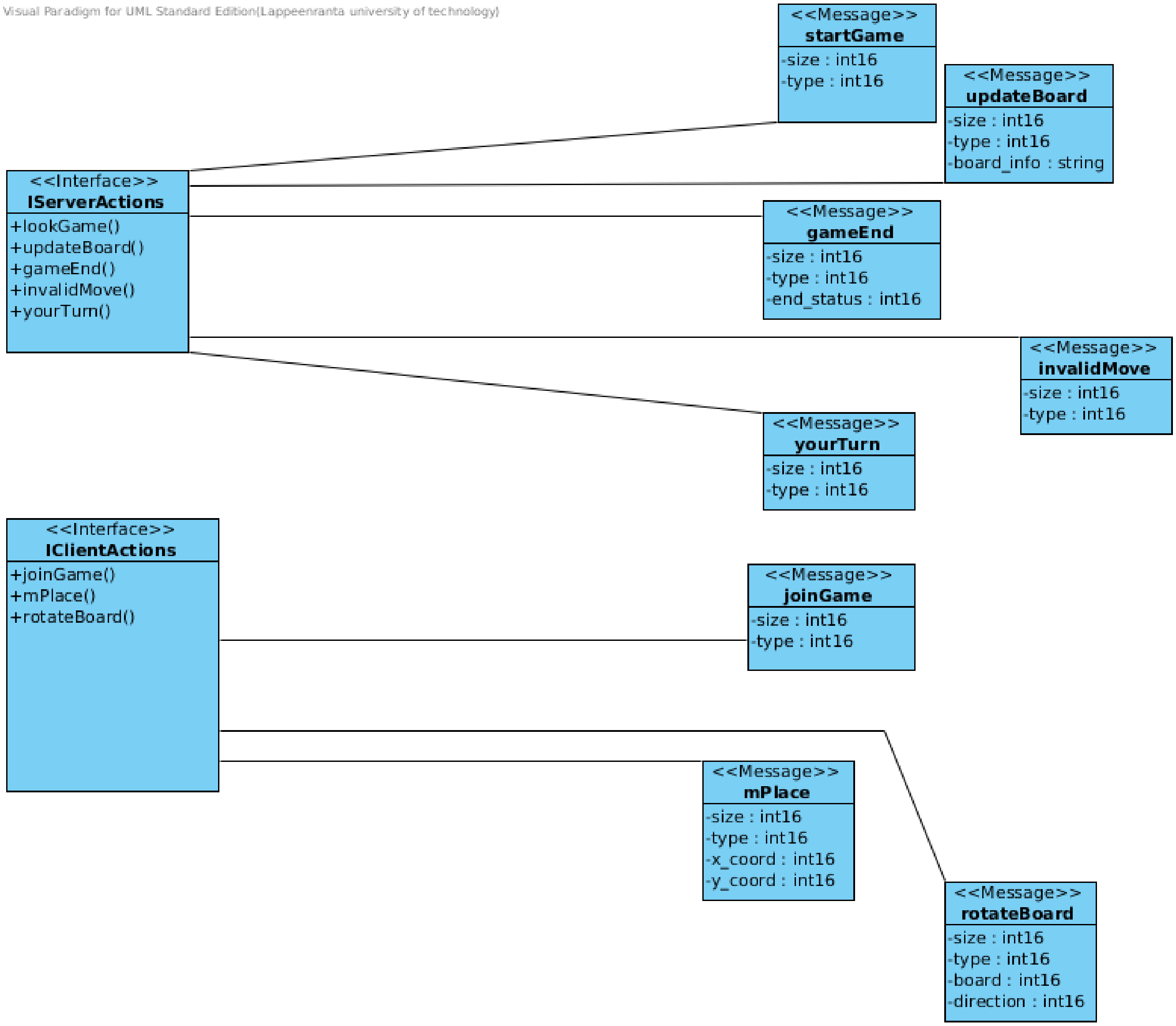


Illustration 14: Abstract messages

9. Concrete message definitions

look_game_pdu				
	0...7	8...15	16...23	24...31
0	size		type	

update_board_pdu				
	0...7	8...15	16...23	24...31
0	size		type	
1	strlen		board_info	board_info
…	board_info			
N	board_info		PADDING	

game_end_pdu				
	0...7	8...15	16...23	24...31
0	size		type	
1	end_status		PADDING	

invalid_move_pdu				
	0...7	8...15	16...23	24...31
0	size		type	

your_turn_pdu				
	0...7	8...15	16...23	24...31
0	size		type	

join_game_pdu				
	0...7	8...15	16...23	24...31
0	size		type	

m_place_pdu				
	0...7	8...15	16...23	24...31
0	size		type	
1	x_coord		y_coord	

rotate_board_pdu				
	0...7	8...15	16...23	24...31
0	size		type	
1	board		direction	

7. Type catalogue

type – 16-bit integer

- 0 – look_game_pdu
- 1 – update_board_pdu
- 2 – game_end_pdu
- 3 – invalid_move_pdu
- 4 – your_turn_pdu
- 5 – join_game_pdu
- 6 – m_place_pdu
- 7 – rotate_board_pdu

size – 16-bit integer

- PDU message size

string – iso8859_15 8-bit

strlen – 16-bit integer

- string length

board_info – string

- contains number for each board tile
    - 0 – empty
    - 1 – player 1's piece
    - 2 – player 2's piece
- 32 characters
- first number in left upper corner (origo)
    - table updated from left to right and from up to down

end_status – 16-bit integer

- 0 – draw
- 1 – player 1 wins
- 2 – player 2 wins

x_coord – 16-bit integer

- number between 0 … 5
- origo at left upper corner

y_coord – 16-bit integer

- number between 0 … 5
- origo at left upper corner

board – 16-bit integer

- number between 0 … 3

direction – 16-bit integer

- boolean for rotation direction

References

1. Pentago, internet reference, URL: http://en.wikipedia.org/wiki/Pentago, accessed 4.12.2012
2. Transmissio control protocol, internet reference, URL: http://tools.ietf.org/html/rfc793, accessed 4.12.2012