

# Project Abalone Report

By Viktor Tonchev and Dimitar Popov

## Functional Requirements of the Application

Functional Requirements for the Server:

1. In the run method of the AbaloneServer.java class the user is asked for port and then the application tries to create a socket with the given port and connect the user to it.
2. When the socket can't be created at the port an exception is thrown and the user is notified to try with a new port.
3. The servers implements the Runnable interface. Whenever a new user connects the server starts a thread for its handler in the run method.
4. This is implemented by the AbaloneServerTUI.java class.
5. The classes ServerProtocol.java and ProtocolMessages.java are used throughout the program to ensure this.

Functional requirements for the Client:

1. This is implemented by the AbaloneClientTUI.java class.
2. The client has a SimpleGame in it that can handle up to 4 players. It has a board with basic functionality. It also has a HashMap that stores the name of a player and their color to be able to distinguish between them. There is also a class AbaloneClientAI.java that supports an AI with basic functionality and can be connected to a 2-player game. The AI is barely functional(cannot actually win) but we didn't have time to think of an actual AI so we had to go with it.
3. When connecting as a bot in the client it should be specified the difficulty of the bot(AiDumb or AiSmart). This is handled in the start method of the AbaloneServerTUI.java class.
4. The client can type hint when they are in a game and it is their turn and the program will provide them with one. The ClientTUI receives it and uses the handGiveHint method of the Client class to print it.
5. When a game ends, the client is notified and the user is given the option to enter a new command in the same manner as if they had just connected to the server. They can join a game, create a game, exit, etc. When the game ends normally(win or draw) the server sends a GOVER message to the client through the handler where the handleGameOver method is called and the users are notified and the local games on their clients are reset. The server also calls the quitActiveGame method to remove the games from the handlers of the users.
6. Whenever a game is finished a GOVER message is sent to all players in it. In the case of someone quitting, disconnecting or their client crashing this message informs the other clients(handleGameOver), removes them from the game(quitActiveGame), and deletes it from the server(doQuit). All users can then choose what to do next.

7. This is handled by the ClientHandler in the shutdown method which is called when there is a ServerUnavailableException.
8. The ClientProtocol.java and ProtocolMessages.java classes are used throughout the program to ensure this.

Global Requirements:

1. The client has a local SimpleGame class, while the server uses the NetworkGame class in the ClientHandler for every client. They are always synchronized since they are both updated whenever a move is made.
2. Throughout the program there are various checks for user input. The Client, Server, ServerTUI, ClientTUI and the ClientHandler classes all have such checks and appropriate error messages when the input is invalid.
3. Our implementation is based on the hotel implementation from week seven which is based on the MVC model.
4. There is a package with exceptions where all exceptions are defined by us and then used throughout the program.

Model-View-Controller pattern:

**Model:** In the ClientHandler class there is a NetworkGame that plays the role of the model in our implementation.

**View:** The ServerTUI and ClientTUI play the role of the view in our implementation.

**Controller:** The ClientHandler class plays the role of the controller in our implementation.

The rest of the classes help the MVC pattern work. They are connected in the following way: The ClientTUI accepts user input. It then classifies it and sends it over to the Client. The Client then calls the corresponding handle method which does what needs to be done and redirects to the Server through the ClientHandler. The Server then executes the command and sends a message to the Client through the ClientHandler where the Client sends it to the ClientTUI that updates the view. Of course, for the different commands and cases things work differently but this is the general workflow of the system.

Unit Testing:

ASCIIArtUtils.java: Coverage - 95.5% Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package.

Ball.java: Coverage - 45.0%- Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package. Nothing to really test here.

Board.java: Coverage - 94.5% - Covered in the Junit GameTest.java class with all other classes in the abalone package. Test results and expected results match.

ClientBoard.java: Coverage - 99.4% - Covered in the Junit GameTest.java class with all other classes in the abalone package. Test results and expected results match.

Color.java: Coverage 40.7% - Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package.

Direction.java: Coverage 84.3& - Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package.

Field.java: Coverage 100.0% - Covered in the Junit GameTest.java class with all other classes in the abalone package. Test results and expected results match.

Game.java: Coverage 22.3% - Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package.

GameInfo.java: Coverage 0.0% - Not covered. Nothing to really test here.

HumanPlayer.java: Coverage 100.0% - Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package. Nothing to really test here.

Move.java: Coverage 0.0% - Not covered. Nothing to really test here.

NetworkGame.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

Player.java: Coverage 71.4% - Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package.

PlayerQuitException: Coverage 0.0% - Not covered. Nothing to really test here.

SelectionOrientation.java: Coverage 100.0% - Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package.

SimpleBoard.java: Coverage 94.8% - Covered by proxy in the Junit GameTest.java class with all other classes in the abalone package.

SimpleGame.java: Coverage 0.0% - Not covered. Nothing to really test here.

AbaloneClient.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

AbaloneClientAI.java - Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

AbaloneClientTUI.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

AbaloneClientView.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

ExitProgram.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

InvalidMoveException.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

InvalidSelectionException.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

InvalidPasswordException.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

JoinGameException.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

PlyerQuitException.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

ProtocolException.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

ServerUnavailableException.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

ClientProtocol.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

ProtocolMessages.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

ServerProtocol.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

AbaloneClientHandler.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

AbaloneServer.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

AbaloneServerTUI.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

AbaloneServerView.java: Coverage 0.0% - Coverage is like that because it isn't covered in a Junit test. It is covered by our manual testing of the game, server, client handler and client.

For classes in the abalone package just run the GameTest.java class. The networking part of the application is covered in the tests below.

### For starting a server:

```
Welcome to the Abalone Server! Starting...
Please enter the server port.
8888888888

*** Error in input: Integer input outside of legal range, 888888888888.
*** Expecting: Integer in the range -2147483648 to 2147483647
*** Discarding Input: (end-of-line)

Please re-enter:

Welcome to the Abalone Server! Starting...
Please enter the server port.
123456
Attempting to open a socket at 192.168.178.21 on port 123456...
Exception in thread "Thread-0" java.lang.IllegalArgumentException: Port value out of range: 123456
    at java.base/java.net.ServerSocket.<init>(ServerSocket.java:238)
    at ss.project.abalone.server.AbaloneServer.setup(AbaloneServer.java:143)
    at ss.project.abalone.server.AbaloneServer.run(AbaloneServer.java:86)
    at java.base/java.lang.Thread.run(Thread.java:834)

Welcome to the Abalone Server! Starting...
Please enter the server port.
8888
Attempting to open a socket at 192.168.178.21 on port 8888...
Server started at port 8888
```

### For starting a client:

---

```
Input the IP Adress of the server you want to connect to :
192.168.100.100
Input the port on which you want to connect:
8888
Input the mode in which you want to run the client("Human" for normal human player,"AiDumb" for a Dumb Ai or "AiSmart" for a Smart Ai ):
Human
Attempting to connect to /192.168.100.100:8888...
ERROR: could not create a socket on 192.168.100.100 and port 8888.
Do you want to open a new socket?
yes
Attempting to connect to /192.168.100.100:8888...
ERROR: could not create a socket on 192.168.100.100 and port 8888.
Do you want to open a new socket?
no
Okay. Goodbye.
```

Or if a wrong is inputted:

```
Input the IP Adress of the server you want to connect to :
192.168.178.21
Input the port on which you want to connect:
1111
Input the mode in which you want to run the client("Human" for normal human player,"AiDumb" for a Dumb Ai or "AiSmart" for a Smart Ai ):
Human
Attempting to connect to /192.168.178.21:1111...
ERROR: could not create a socket on 192.168.178.21 and port 1111.
Do you want to open a new socket?
```

Or:

```
AbaloneClient (2) [Java Application] C:\Program Files\Java\jdk-11.0.5\bin\javaw.exe (Feb 1, 2020, 7:40:10 PM)
AbaloneClient (2) [Java Application] C:\Program Files\Java\jdk-11.0.5\bin\javaw.exe (Feb 1, 2020, 7:40:10 PM)
Input the port on which you want to connect:
8888
Input the mode in which you want to run the client("Human" for normal human player,"AiDumb" for a Dumb Ai or "AiSmart" for a Smart Ai ):
Human
Attempting to connect to /192.168.178.21:8888...
Connected
Wellcome to the Abalone client!

Below is the list of supported commands!
Connect to server - c:<username>
Disconnect from server - exit
Create game - create:<game_name>:<password>:<capacity>
Join game - join:<game_name>:<pass>
List game - list
Make a move - move:<m1>:<m2>:<m3>:<direction>
Quit game - quit
Request a hint for a move - hint
Enter command:
```

```
AbaloneServer (2) [Java Application] C:\Program Files\Java\jdk-11.0.5\bin\javaw.exe (Feb 1, 2020, 7:33:53 PM)
Welcome to the Abalone Server! Starting...
Please enter the server port.
8888
Attempting to open a socket at 192.168.178.21 on port 8888...
Server started at port 8888
New client [Client 01] connected!
```

```
c:dimitar
User dimitar connected to the server!
Enter command:
```

```
AbaloneServer (2) [Java Application] C:\Program Files\Java\jdk-11.0.5\bin\javaw.exe (Feb 1, 2020, 7:44:27 PM)
Welcome to the Abalone Server! Starting...
Please enter the server port.
8888
Attempting to open a socket at 192.168.178.21 on port 8888...
Server started at port 8888
New client [Client 01] connected!
> [Client 01] Incoming: c:dimitar
user [dimitar] connected!
```

```
c:dimitar
User dimitar connected to the server!
Enter command:
join:<game>:<pass>
Please send 'list' command to the server first to obtain information for the active games!
Enter command:
list
Invalid game info received from the server - There are no games on this server. You may join another one...
Enter command:
move:<23>:<24>:<DR>
You are not in a game!
Enter command:
quit
You are not in a game!
Enter command:
hint
You are not in a game.
Enter command:
```

```
Enter command:  
create:dimitargame:pass1:2  
You have created the game dimitargame on the server!
```

Current game situation:

```
 01  02  03  04  05  
 06  07  08  09  10  11  
 12  13  14  15  16  17  18  
 19  20  21  22  23  24  25  26  
 27  28  29  30  31  32  33  34  35  
 36  37  38  39  40  41  42  43  
 44  45  46  47  48  49  50  
 51  52  53  54  55  56  
 57  58  59  60  61
```

Please, send ready command when ready!

```
Enter command:  
move:14:15:R  
Declare you're ready so the game can start/ you can quit!  
Enter command:  
quit  
Declare you're ready so the game can start/ you can quit!  
Enter command:  
list  
You cannot look at other games while you are in one!  
Enter command:
```

On the server's console, this is printed out:

```
Welcome to the Abalone Server! Starting...  
Please enter the server port.  
8888  
Attempting to open a socket at 192.168.178.21 on port 8888...  
Server started at port 8888  
New client [Client 01] connected!  
> [Client 01] Incoming: c:dimitar  
user [dimitar] connected!  
> [Client 01] Incoming: list  
> [Client 01] Incoming: create:dimitargame:pass1:2  
Game [dimitargame] created!  
dimitar joined the game [dimitargame]!
```

Now, user dimitar can input "ready"

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, send ready command when ready!  
Enter command:  
**move:14:15:R**  
Declare you're ready so the game can start/ you can quit!  
Enter command:  
**quit**  
Declare you're ready so the game can start/ you can quit!  
Enter command:  
**list**  
You cannot look at other games while you are in one!  
Enter command:  
**join:whateverothergame**  
Please send 'list' command to the server first to obtain information for the active games!  
Enter command:  
**ready**  
Please wait the server to give the turn to you...

Now, after another client Viktor connects, he can join the game and start the game can start:

```

Enter command:
c:viktor
User viktor connected to the server!
Enter command:
wrongCommand
Invalid command entered wrongCommand
Enter command:
list
Game:dimitargame ,connected users:1,max users:2 has_pass
Enter command:
join:dimitargame:pass1
You have joined the game dimitargame on the server!

Current game situation:

```

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

```

Please, send ready command when ready!
Enter command:
ready
Please wait the server to give the turn to you...
Player viktor, the game has started, please wait for your turn!

```

The server says that :

```

New client [Client 02] connected!
> [Client 02] Incoming: c:viktor
user [viktor] connected!
> [Client 02] Incoming: list
> [Client 02] Incoming: join:dimitargame:pass1
viktor joined the game [dimitargame]!
Sending message to user [viktor]!
Message sent to user [viktor]!
> [Client 02] Incoming: ready
Ready message received from user [viktor]!
Sending start to user [dimitar]!
Start sent to user [dimitar]!
Sending start to user [viktor]!
Start sent to user [viktor]!
Giving turn to user [dimitar]!
Turn given to user [dimitar]!

```

Then, dimitar can input a move:

```

c:michael
You are already connected as dimitar to this server!
Enter command:
list
You cannot look at other games while you are in one!
Enter command:
hint
Suggested move: Move the field: 10 in direction: DR
Enter command:
move:10:DR
Your move was accepted by the server!
The situation in the dimitargame after the move is:

Current game situation:

```

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

## Server:

```
Giving turn to user [dimitar]!
Turn given to user [dimitar]!
> [Client 01] Incoming: move:10:DR
The game board before the move is:
 01 02 03 04 05
 06 07 08 09 10 11
 12 13 14 15 16 17 18
 19 20 21 22 23 24 25 26
 27 28 29 30 31 32 33 34 35
 36 37 38 39 40 41 42 43
 44 45 46 47 48 49 50
 51 52 53 54 55 56
 57 58 59 60 61

The game board after the move is:
 01 02 03 04 05
 06 07 08 09 10 11
 12 13 14 15 16 17 18
 19 20 21 22 23 24 25 26
 27 28 29 30 31 32 33 34 35
 36 37 38 39 40 41 42 43
 44 45 46 47 48 49 50
 51 52 53 54 55 56
 57 58 59 60 61
```

## Player viktor:

```
Please, wait the server to give the turn to you...
Player viktor, please enter your turn:
Enter command:
move:ifWeEnterAnInvalidInputOnWhatever:Place:OfThisCommand
Direction for the move is not valid:OfThisCommand move:ifWeEnterAnInvalidInputOnWhatever:Place:OfThisCommand
Enter command:

Invalid command entered
Enter command:
move:22:23:L
Invalid move: Field at 22 does not have a ball on it. Please make a valid selection!
Enter command:
move:14:15:16:R
Invalid move: Player viktor(BLUE), please select only balls from your color!
Enter command:
move:47:54:DR
Invalid move: Player viktor(BLUE) you can't push your own balls with a move (60)
Enter command:
move:55:56:R
Invalid move: Player viktor(BLUE), you can't push your balls outside of the board (56)
Enter command:
move:56:See?AllInvalidUserInputForMoveIsHadnled
Direction for the move is not valid:See?AllInvalidUserInputForMoveIsHadnled move:56:See?AllInvalidUserInputForMoveIsHadnled
Enter command:
move:46:47:48:UL
Your move was accepted by the server!
The situation in the dimitargame after the move is:

Current game situation:
```

Current game situation:

01	02	03	04	05
06	07	08	09	10
11				
12	13	14	15	16
17				18
19	20	21	22	23
24	25	26		
27	28	29	30	31
32	33	34	35	
36	37	38	39	40
41	42	43		
44	45	46	47	48
49	50			
51	52	53	54	55
56				
57	58	59	60	61

Please, wait the server to give the turn to you...

## Inline moves:

## Moving three balls:

Current game situation:

01	02	03	04	05
06	07	08	09	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50
51	52	53	54	55
56	57	58	59	60
61				

Please, wait the server to give the turn to you...  
Player dimitar, please enter your turn:

Enter command:

Enter command:

MOVE:03:9:16:DR

move:03:9:16:DR

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Current game situation:

01	02	03	04	05
06	07	08	09	10
11	12	13	14	15
16	17	18		
19	20	21	22	23
24	25	26		
27	28	29	30	31
32	33	34	35	
36	37	38	39	40
41	42	43		
44	45	46	47	48
49	50			
51	52	53	54	55
56				
57	58	59	60	61

### Moving two balls:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

move:60:55:UR

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

### Moving one ball:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

move:55:60:DR

The situation in the dimitargame after the move is:

Current game situation:

### Sidestep move:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...  
The situation in the dimitargame after the move is:

Current game situation:

move:11:18:L

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

### Sumito two/three players:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

Player viktor, please enter your turn:

Enter command:

move:38:39:40:R

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Please, wait the server to give the turn to you...  
Player viktor, please enter your turn:

Enter command:

move:38:39:40:R

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

Please, wait the server to give the turn to you...

Enter command:

move:38:39:40:R

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

Please, wait the server to give the turn to you...

Player dimitar, please enter your turn:

Enter command:

move:42:43:L

Invalid move: Player dimitar(YELLOW), you can't push two balls with your selection

Enter command:

Please, wait the server to give the turn to you...

Player viktor, please enter your turn:

Enter command:

move:39:40:41:R

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

Please, wait the server to give the turn to you...

Player viktor, please enter your turn:

Enter command:

move:40:41:42:R

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Please, wait the server to give the turn to you...

Player viktor, please enter your turn:

Enter command:

move:40:41:42:R

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

## SUMITO/MOVE FOUR PLAYER GAME:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

Player 3, please enter your turn:

Enter command:

move:31:23:UR

Your move was accepted by the server!

The situation in the 1 after the move is:

Player 3, please enter your turn:

Enter command:

move:31:23:UR

Your move was accepted by the server!

The situation in the 1 after the move is:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

**End of game**(in this case in a two player game but the same applies for all types of games):

Player viktor makes the final move:

Current game situation:

01	02	03	04	05
06	07	08	09	10
12	13	14	15	16
17	18			
19	20	21	22	23
24	25	26		
27	28	29	30	31
32	33	34	35	
36	37	38	39	40
41	42	43		
44	45	46	47	48
49	50			
51	52	53	54	55
56				
57	58	59	60	61

Please, wait the server to give the turn to you...

Player viktor, please enter your turn:

Enter command:

move:32:33:34:R

Your move was accepted by the server!

The situation in the dimitargame after the move is:

Player viktor sees:

move: 32:33:34:R

Your move was accepted by the server!

Your move was accepted by the server!  
The situation in the dimitargame after the move is:

Current game situation:

Please, wait the server to give the turn to you...

Game over. Player viktor has won!

Enter command:

He has won and the game is over, so he can do everything he could have before the game started. The other players (in this case the game is for two, so player dimitar) see:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

Game over. Player viktor has won!

Enter command:

Overall, all players see the same thing and can do everything they could have before the game was over. The game name is deleted.

#### Move for teams:

Please, wait the server to give the turn to you...

Player 1, please enter your turn:

Enter command:

move:8:15:23:DL

Your move was accepted by the server!

The situation in the 1 after the move is:

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Current game situation:

01	02	03	04	05				
06	07	08	09	10	11			
12	13	14	15	16	17	18		
19	20	21	22	23	24	25	26	
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	
44	45	46	47	48	49	50		
51	52	53	54	55	56			
57	58	59	60	61				

Please, wait the server to give the turn to you...

Player 1, please enter your turn:

Enter command:

move:8:15:23:DL

Your move was accepted by the server!

The situation in the 1 after the move is:

Please, wait the server to give the turn to you...

**If a player quits a game, the following is printed out:**

```
| Enter command:  
| quit  
| You have quit from the game game  
| Enter command:
```

On all the other players'(in the particular game) consoles the following is printed out:

```
| Please, wait the server to give the turn to you...  
| Someone quit the game.  
| Enter command:
```

**If a player exits a game, the following is printed out:**

```
| exit  
| Closing the connection...|
```

On all the other players'(in the particular game) consoles the following is printed out:

```
| Please, wait the server to give the turn to you...  
| Someone quit the game.  
| Enter command:
```

**The AI is started in the following way:** start a normal client and when asked input AiSmart for smart mode or AiDumb for dumb mode. Then the client should be connected to a game in the following way:

```
Input the IP Adress of the server you want to connect to :  
192.168.178.21  
Input the port on which you want to connect:  
8888  
Input the mode in which you want to run the client("Human" for normal human player,"AiDumb" for a Dumb Ai or "AiSmart" for a Smart Ai ):  
AiSmart  
Attempting to connect to /192.168.178.21:8888...  
Connected  
Wellcome to the Abalone client!  
  
Below is the list of supported commands!  
Connect to server - c:<username>  
Disconnect from server - exit  
Create game - create:<game_name>:<password>:<capacity>  
Join game - join:<game_name>:<pass>  
List game - list  
Make a move - move:<m1>:<m2>:<m3>:<direction>  
Quit game - quit  
Request a hint for a move - hint  
Enter command:  
c:aBot  
User aBot connected to the server!  
Enter command:  
create:IWantToCreateAGameButIAmABot:pass:2  
Invalid command: A bot cannot create a game  
Enter command:  
list  
Game:dimitargame ,connected users:2,max users:2 has_pass  
Game:aGameWithABot ,connected users:1,max users:2 has_pass  
Enter command:  
join:aGameWithABot:pass  
You have joined the game aGameWithABot on the server!  
  
Current game situation:  
    01 02 03 04 05  
    06 07 08 09 00 11  
12 13 14 15 16 17 18
```

Then, when the actual human player does a move, the AI(Client) will make the move on its own:

move:14:15:16:DR  
Your move was accepted by the server!  
The situation in the aGameWithABot after the move is:

Current game situation:

01	02	03	04	05
06	07	08	09	10
12	13	14	15	16
17	18			
19	20	21	22	23
24	25	26		
27	28	29	30	31
32	33	34	35	
36	37	38	39	40
41	42	43		
44	45	46	47	48
49	50			
51	52	53	54	55
56				
57	58	59	60	61

Please, wait the server to give the turn to you...  
The situation in the aGameWithABot after the move is:

Current game situation:

01	02	03	04	05
06	07	08	09	10
11				
12	13	14	15	16
17	18			
19	20	21	22	23
			24	25
26				
27	28	29	30	31
32	33	34	35	
36	37	38	39	40
41	42	43		
44	45	46	47	48
49	50			
51	52	53	54	55
56				
57	58	59	60	61

Please, wait the server to give the turn to you...  
Player vldai, please enter your turn:  
Enter command:

These are just some examples of the extensive tests that we did. Our tests included going through a full game(both draw and win) for 2,3 and 4 players, testing a full game with a bot(both win and draw), testing all user commands and their variants (for example quitting, disconnecting, crashing in the middle of a game; not being able to join/create a game while in one; not being able to do anything before connecting; etc.), testing various possible moves(with and without sumitos, valid and invalid), validation of all user input, etc. Some errors we found while testing were that the move method of the ClientBoard class method, various errors with user input, “end cases” and other various small bugs across the project. We found these with both the Junit tests and visual testing of the UI.

## Metrics report:

Below are the metrics for the most important classes:

### Board.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	549	32.294	38.839	149	/MetricsAndStuff/src/ss/project/abalone/Board.java	validateAndExpand...
> McCabe Cyclomatic Complexity (avg/max per method)		10.941	16.878	56	/MetricsAndStuff/src/ss/project/abalone/Board.java	validateAndExpand...
> Weighted methods per Class (avg/max per type)	186	186	0	186	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Lack of Cohesion of Methods (avg/max per type)		0.286	0	0.286	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Number of Parameters (avg/max per method)		1.647	1.135	4	/MetricsAndStuff/src/ss/project/abalone/Board.java	initForFour
> Nested Block Depth (avg/max per method)		2.471	2.145	10	/MetricsAndStuff/src/ss/project/abalone/Board.java	validateAndExpand...
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Number of Overridden Methods (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Number of Attributes (avg/max per type)	2	2	0	2	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Number of Static Attributes (avg/max per type)	3	3	0	3	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Number of Methods (avg/max per type)	17	17	0	17	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Specialization Index (avg/max per type)		0.059	0	0.059	/MetricsAndStuff/src/ss/project/abalone/Board.java	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	598					

### ClientBoard.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	96	24	20.857	57	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	move
> McCabe Cyclomatic Complexity (avg/max per method)		10	8.689	24	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	move
> Weighted methods per Class (avg/max per type)	40	40	0	40	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Lack of Cohesion of Methods (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Parameters (avg/max per method)		1	0.707	2	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	expandForSummito
> Nested Block Depth (avg/max per method)		2.75	1.479	5	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	move
> Depth of Inheritance Tree (avg/max per type)		2	0	2	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Overridden Methods (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Methods (avg/max per type)	4	4	0	4	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Specialization Index (avg/max per type)		0.5	0	0.5	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	109					

### NetworkGame.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	98	4.9	6.848	21	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	join
> McCabe Cyclomatic Complexity (avg/max per method)		1.9	1.841	8	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	join
> Weighted methods per Class (avg/max per type)	38	38	0	38	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Lack of Cohesion of Methods (avg/max per type)		0.828	0	0.828	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Number of Parameters (avg/max per method)		0.55	0.805	3	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	NetworkGame
> Nested Block Depth (avg/max per method)		1.4	0.86	4	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	isOver
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Number of Attributes (avg/max per type)	11	11	0	11	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Number of Methods (avg/max per type)	20	20	0	20	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/NetworkGame...	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	159					

### SimpleBoard.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	320	20	22.655	72	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	toString
> McCabe Cyclomatic Complexity (avg/max per method)		6.25	11.611	48	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	calculateNeigbourl...
> Weighted methods per Class (avg/max per type)	100	100	0	100	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Lack of Cohesion of Methods (avg/max per type)		0.286	0	0.286	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Number of Parameters (avg/max per method)		1.438	1.059	4	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	initForFour
> Nested Block Depth (avg/max per method)		1.75	0.75	4	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	calculateNeigbourl...
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Number of Children (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Number of Overridden Methods (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Number of Attributes (avg/max per type)	2	2	0	2	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Number of Static Attributes (avg/max per type)	3	3	0	3	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Number of Methods (avg/max per type)	16	16	0	16	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
> Specialization Index (avg/max per type)		0.062	0	0.062	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	363					

### SimpleGame.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	34	3.778	4.613	15	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	reset
> McCabe Cyclomatic Complexity (avg/max per method)		1.667	1.054	4	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	reset
> Weighted methods per Class (avg/max per type)	15	15	0	15	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Lack of Cohesion of Methods (avg/max per type)		0.75	0	0.75	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Number of Parameters (avg/max per method)		0.667	0.816	2	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	SimpleGame
> Nested Block Depth (avg/max per method)		1.333	0.667	3	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	setPlayerName
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Number of Overridden Methods (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Number of Attributes (avg/max per type)	5	5	0	5	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Number of Methods (avg/max per type)	9	9	0	9	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
> Specialization Index (avg/max per type)		0.111	0	0.111	/MetricsAndStuff/src/ss/project/abalone/SimpleGame.ja...	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	60					

## The whole abalone package:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	1301	10.842	20.864	149	/MetricsAndStuff/src/ss/project/abalone/Board.java	validateAndExpand...
> McCabe Cyclomatic Complexity (avg/max per method)		4	8.772	56	/MetricsAndStuff/src/ss/project/abalone/Board.java	validateAndExpand...
> Weighted methods per Class (avg/max per type)	480	28.235	46.108	186	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Lack of Cohesion of Methods (avg/max per type)		0.323	0.338	0.844	/MetricsAndStuff/src/ss/project/abalone/Field.java	
Afferent Coupling	7					
Efferent Coupling	2					
> Number of Parameters (avg/max per method)		0.958	1.044	4	/MetricsAndStuff/src/ss/project/abalone/SimpleBoard.j...	initForFour
> Nested Block Depth (avg/max per method)		1.508	1.147	10	/MetricsAndStuff/src/ss/project/abalone/Board.java	validateAndExpand...
Instability	0.222					
Abstractness	0					
Normalized Distance	0.778					
> Depth of Inheritance Tree (avg/max per type)		1.412	0.6	3	/MetricsAndStuff/src/ss/project/abalone/PlayerQuitExce...	
> Number of Children (avg/max per type)	2	0.118	0.322	1	/MetricsAndStuff/src/ss/project/abalone/Player.java	
> Number of Overridden Methods (avg/max per type)	5	0.294	0.456	1	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Attributes (avg/max per type)	58	3.412	3.107	11	/MetricsAndStuff/src/ss/project/abalone/NetworkGame....	
> Number of Static Attributes (avg/max per type)	65	3.824	13.329	57	/MetricsAndStuff/src/ss/project/abalone/ASCIIArtUtil.ja...	
> Number of Methods (avg/max per type)	117	6.882	6.028	20	/MetricsAndStuff/src/ss/project/abalone/NetworkGame....	
> Number of Static Methods (avg/max per type)	3	0.176	0.381	1	/MetricsAndStuff/src/ss/project/abalone/Direction.java	
> Specialization Index (avg/max per type)		0.048	0.118	0.5	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Classes	17					
> Number of Interfaces	0					
Total Lines of Code	1738					

## AbaloneClient.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	357	9.395	12.03	61	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleGiveHint
> McCabe Cyclomatic Complexity (avg/max per method)		2.763	3.716	23	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleGiveHint
> Weighted methods per Class (avg/max per type)	105	105	0	105	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Lack of Cohesion of Methods (avg/max per type)		0.886	0	0.886	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Parameters (avg/max per method)		0.526	0.716	3	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleCreateGame
> Nested Block Depth (avg/max per method)		1.921	1.222	5	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleGiveHint
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Attributes (avg/max per type)	15	15	0	15	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Methods (avg/max per type)	37	37	0	37	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Static Methods (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	489					

### AbaloneClientAI.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	162	32.4	23.838	56	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	makeSmartMove
> McCabe Cyclomatic Complexity (avg/max per method)		10.2	7.44	18	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	makeDumbMove
> Weighted methods per Class (avg/max per type)	51	51	0	51	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Lack of Cohesion of Methods (avg/max per type)		0.5	0	0.5	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Parameters (avg/max per method)		0.4	0.8	2	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	AbaloneClientAI
> Nested Block Depth (avg/max per method)		2.6	1.2	4	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	makeDumbMove
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Attributes (avg/max per type)	2	2	0	2	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Methods (avg/max per type)	5	5	0	5	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	182					

### AbaloneClientTUI.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	312	20.8	43.815	170	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleUserInput
> McCabe Cyclomatic Complexity (avg/max per method)		5.867	12.479	50	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleUserInput
> Weighted methods per Class (avg/max per type)	88	88	0	88	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Lack of Cohesion of Methods (avg/max per type)		0.822	0	0.822	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Parameters (avg/max per method)		0.4	0.49	1	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	AbaloneClientTUI
> Nested Block Depth (avg/max per method)		1.667	1.535	6	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleUserInput
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Attributes (avg/max per type)	5	5	0	5	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Methods (avg/max per type)	15	15	0	15	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	369					

### The whole abalone.client package:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	832	12.606	25.149	170	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleUserInput
> McCabe Cyclomatic Complexity (avg/max per method)		3.818	7.292	50	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleUserInput
> Weighted methods per Class (avg/max per type)	252	63	37.276	105	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Lack of Cohesion of Methods (avg/max per type)		0.552	0.351	0.886	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
Afferent Coupling	0					
Efferent Coupling	4					
> Number of Parameters (avg/max per method)		0.5	0.657	3	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleCreateGame
> Nested Block Depth (avg/max per method)		1.682	1.394	6	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	handleUserInput
Instability	1					
Abstractness	0.25					
Normalized Distance	0.25					
> Depth of Inheritance Tree (avg/max per type)		0.75	0.433	1	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Children (avg/max per type)	1	0.25	0.433	1	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Attributes (avg/max per type)	22	5.5	5.766	15	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Methods (avg/max per type)	65	16.25	12.517	37	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Static Methods (avg/max per type)	1	0.25	0.433	1	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/client/Abalone...	
> Number of Classes	4					
> Number of Interfaces	1					
> Total Lines of Code	1056					

### AbaloneClientHandler.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	155	14.091	28.906	104	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	handleCommand
> McCabe Cyclomatic Complexity (avg/max per method)		3.545	6.514	24	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	handleCommand
> Weighted methods per Class (avg/max per type)	39	39	0	39	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Lack of Cohesion of Methods (avg/max per type)		0.75	0	0.75	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Number of Parameters (avg/max per method)		0.636	0.881	3	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	AbaloneClientHand...
> Nested Block Depth (avg/max per method)		1.909	1.73	7	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	handleCommand
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Number of Attributes (avg/max per type)	8	8	0	8	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Number of Methods (avg/max per type)	11	11	0	11	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/Abalone...	
Number of Classes	1					
Number of Interfaces	0					
> Total Lines of Code	196					

### AbaloneServer.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	301	17.706	19.201	82	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:17.706	doMove
> McCabe Cyclomatic Complexity (avg/max per method)		3.765	3.718	17	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:3.765	doMove
> Weighted methods per Class (avg/max per type)	64	64	0	64	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:64.0	
> Lack of Cohesion of Methods (avg/max per type)		0.731	0	0.731	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.731	
> Number of Parameters (avg/max per method)		1.529	1.377	5	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:1.529	doCreateGame
> Nested Block Depth (avg/max per method)		2.588	1.375	5	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:2.588	doMove
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:1.0	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
> Number of Attributes (avg/max per type)	6	6	0	6	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:6.0	
> Number of Static Attributes (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:1.0	
> Number of Methods (avg/max per type)	16	16	0	16	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:16.0	
> Number of Static Methods (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:1.0	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	356					

### AbaloneServerTUI.java:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	8	2	1	3	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:2.0	getInt
> McCabe Cyclomatic Complexity (avg/max per method)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:1.0	AbaloneServerTUI
> Weighted methods per Class (avg/max per type)	4	4	0	4	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:4.0	
> Lack of Cohesion of Methods (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
> Number of Parameters (avg/max per method)		0.75	0.433	1	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.75	showMessage
> Nested Block Depth (avg/max per method)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:1.0	AbaloneServerTUI
> Depth of Inheritance Tree (avg/max per type)		1	0	1	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:1.0	
> Number of Children (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
> Number of Attributes (avg/max per type)	1	1	0	1	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:1.0	
> Number of Static Attributes (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
> Number of Methods (avg/max per type)	4	4	0	4	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:4.0	
> Number of Static Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/AbaloneServerTUI.java:0.0	
Number of Classes	1					
Number of Interfaces	0					
Total Lines of Code	25					

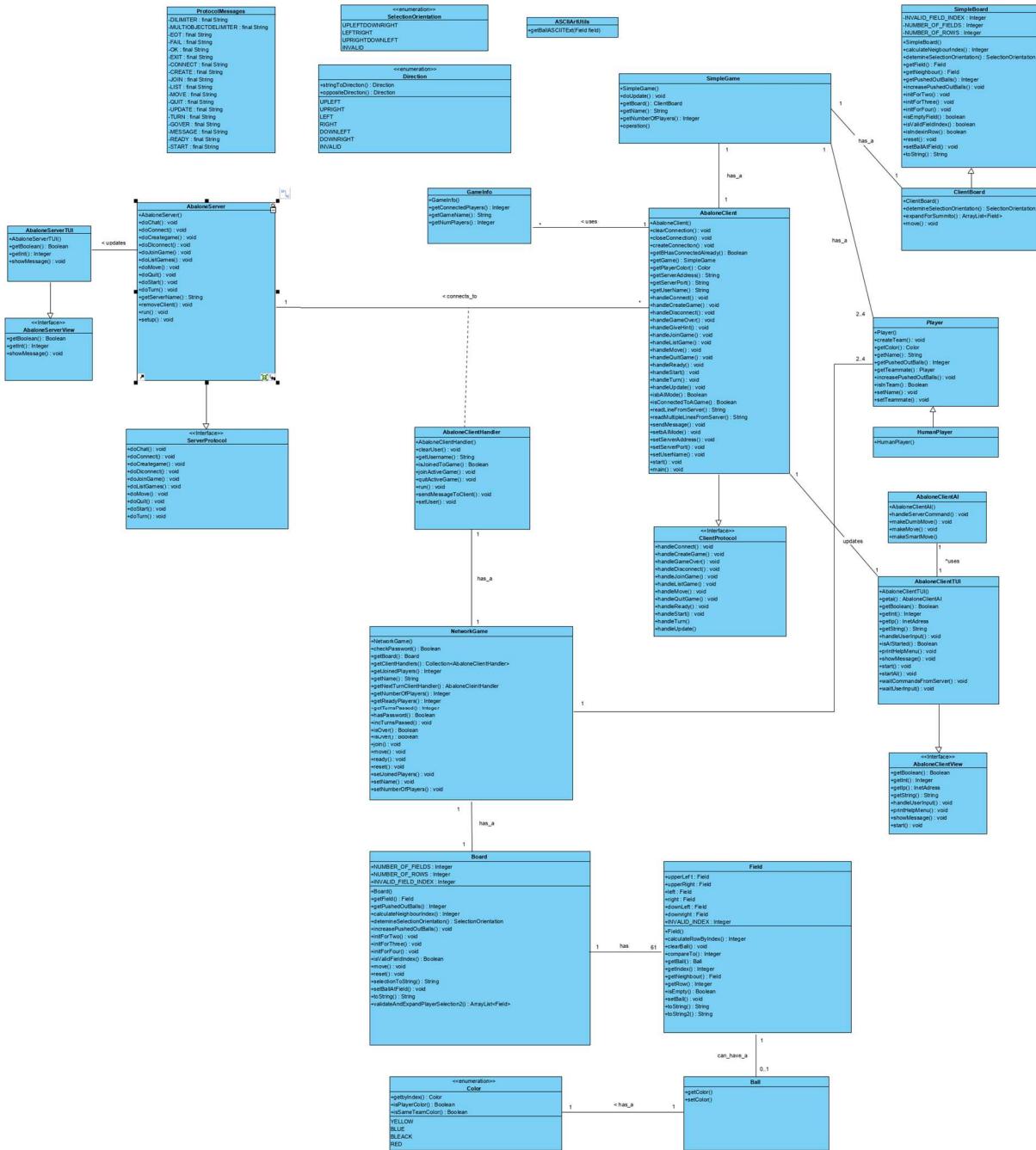
### The whole abalone.server package:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	464	13.257	21.935	104	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	handleCommand
> McCabe Cyclomatic Complexity (avg/max per method)		3.143	4.605	24	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	handleCommand
> Weighted methods per Class (avg/max per type)	110	27.5	25.578	64	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Lack of Cohesion of Methods (avg/max per type)		0.37	0.37	0.75	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
Afferent Coupling	2					
Efferent Coupling	3					
> Number of Parameters (avg/max per method)		1.114	1.165	5	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	doCreateGame
> Nested Block Depth (avg/max per method)		1.971	1.576	7	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	handleCommand
Instability	0.6					
Abstractness	0.25					
Normalized Distance	0.15					
> Depth of Inheritance Tree (avg/max per type)		0.75	0.433	1	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Number of Children (avg/max per type)	1	0.25	0.433	1	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Number of Overridden Methods (avg/max per type)	0	0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Number of Attributes (avg/max per type)	15	3.75	3.345	8	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Number of Static Attributes (avg/max per type)	1	0.25	0.433	1	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Number of Methods (avg/max per type)	34	8.5	5.315	16	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Number of Static Methods (avg/max per type)	1	0.25	0.433	1	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Specialization Index (avg/max per type)		0	0	0	/MetricsAndStuff/src/ss/project/abalone/server/Abalon...	
> Number of Classes	4					
> Number of Interfaces	1					
> Total Lines of Code	583					

## The whole project:

Metric	Total	Mean	Std. Dev.	Maxi...	Resource causing Maximum	Method
> Method Lines of Code (avg/max per method)	4297	13.428	28.948	229	/MetricsAndStuff/src/ss/project/abalone/test/GameTest.java	testValidateDownLeftSumToMoves
> McCabe Cyclomatic Complexity (avg/max per method)		3.506	6.744	56	/MetricsAndStuff/src/ss/project/abalone/Board.java	validateAndExpandPlayerSelection2
> Weighted methods per Class (avg/max per type)	1122	29.526	44.64	186	/MetricsAndStuff/src/ss/project/abalone/Board.java	
> Lack of Cohesion of Methods (avg/max per type)		0.249	0.334	0.886	/MetricsAndStuff/src/ss/project/abalone/client/AbaloneClient.java	
> Afferent Coupling (avg/max per packageFragment)		3.571	3.245	9	/MetricsAndStuff/src/ss/project/abalone/exception	
> Efferent Coupling (avg/max per packageFragment)		1.714	1.385	4	/MetricsAndStuff/src/ss/project/abalone/client	
> Number of Parameters (avg/max per method)		0.787	0.999	5	/MetricsAndStuff/src/ss/project/abalone/protocol/ServerProtocol.java	doCreateGame
> Nested Block Depth (avg/max per method)		1.525	1.24	10	/MetricsAndStuff/src/ss/project/abalone/Board.java	validateAndExpandPlayerSelection2
> Instability (avg/max per packageFragment)		0.444	0.398	1	/MetricsAndStuff/src/ss/project/abalone/client	
> Abstractness (avg/max per packageFragment)		0.167	0.231	0.667	/MetricsAndStuff/src/ss/project/abalone/protocol	
> Normalized Distance (avg/max per packageFragment)		0.461	0.415	1	/MetricsAndStuff/src/ss/project/abalone/exception	
> Depth of Inheritance Tree (avg/max per type)		1.5	0.967	3	/MetricsAndStuff/src/ss/project/abalone/PlayerQuitException.java	
> Number of Children (avg/max per type)	6	0.158	0.365	1	/MetricsAndStuff/src/ss/project/abalone/Player.java	
> Number of Overridden Methods (avg/max per type)	5	0.132	0.338	1	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Attributes (avg/max per type)	100	2.632	3.55	15	/MetricsAndStuff/src/ss/project/abalone/client/AbaloneClient.java	
> Number of Static Attributes (avg/max per type)	113	2.974	9.775	57	/MetricsAndStuff/src/ss/project/abalone/ASCIIArtUtil.java	
> Number of Methods (avg/max per type)	262	6.895	7.437	37	/MetricsAndStuff/src/ss/project/abalone/client/AbaloneClient.java	
> Number of Static Methods (avg/max per type)	58	1.526	8.469	53	/MetricsAndStuff/src/ss/utils/TextIO.java	
> Specialization Index (avg/max per type)		0.021	0.083	0.5	/MetricsAndStuff/src/ss/project/abalone/ClientBoard.java	
> Number of Classes (avg/max per packageFragment)	38	5.429	5.206	17	/MetricsAndStuff/src/ss/project/abalone	
> Number of Interfaces (avg/max per packageFragment)	4	0.571	0.728	2	/MetricsAndStuff/src/ss/project/abalone/protocol	
> Number of Packages	7					
> Total Lines of Code	5385					

Some of these metrics aren't ideal, however we didn't really have the time to deal with that so we just left them as is. For example, some methods have high cyclomatic complexity, making them hard to test and fix in case something goes wrong. The other problematic one is weighted methods per class which is high for a lot of the classes, making them hard to maintain.



When it comes to the protocol, we followed it but there were some minor deviations. For example, our update function doesn't include the color of the player, but the username and our move function message is conveyed by `move:<index>;<direction>` instead of `move:<index>::<direction>`. We did this because the people in our protocol group made changes to the protocol every other hour up until 29<sup>th</sup> and we couldn't keep up with all the changes. Our protocol messages only deviate from the final protocol slightly where they were the ones that were originally accepted, but were randomly changed by someone who wanted to make the protocol fit their implementation at a point when it was too late for us to change it without this fundamentally impacting our project. Another reason was that people

who had no place changing it did so without the explicit consent of the rest of the group. Instead of the protocol being edited by the protocol manager other people had access to it. This resulted in a lot of changes that were discussed and decided on by 3 or 4 people. At that point we had already implemented a big part of the project based on the already existing protocol and had to change a lot of our code. At one point we couldn't keep up anymore and had to prioritize the functionality of our server over implementing the various protocol changes.

#### Reflection:

1. Since we had already done the time planning on the design project, we had a much better understanding on how to plan our time for this project. Thus, this time the time-management plan was much better than last time and as a result we had a much more structured workflow.
2. Since the plan was better, our planning corresponded to the actual project to a great extent. There were only a few things that took more time than expected: the time we spent on handling invalid input and synchronizing the commands (i.e. that you cannot list the current games on the server if you are already connected to a game, that you cannot join a game if you are not connected to the server, etc.), and when we had to make changes to our project because of sudden changes of the protocol (as we explained above).
3. In the last few days of working on the project we had to crunch and work all day to make sure our project was functional and met all the requirements that were specified in the module guide. This didn't really impact the quality of our project since we just used a lot of extra time to make sure we didn't get anything wrong.
4. We learned that you shouldn't leave a lot of work to be done in the last stages of a project since if new things come up you won't be able to deal with them. For the next project it would be better to try and be done ahead of time so we can deal with anything that comes along if it comes to it.
5. Two do's would be to make sure you actually have a plan instead of just going along with the flow, since it really helps, and working together instead of just doing things separately and then trying to fit it all together when neither of you knew what the other was doing since that ends up taking more time than just doing it as a team. Two don'ts would be to not plan a lot of your work for the end of the project and to make sure your group understands that the protocol needs to be decided on by everybody and on time.