Project Report For Collecto Game Application

By Yordan Tsintsov(s2331993)
and Yuchen Sun(s2207818)

Justification of minimal requirements

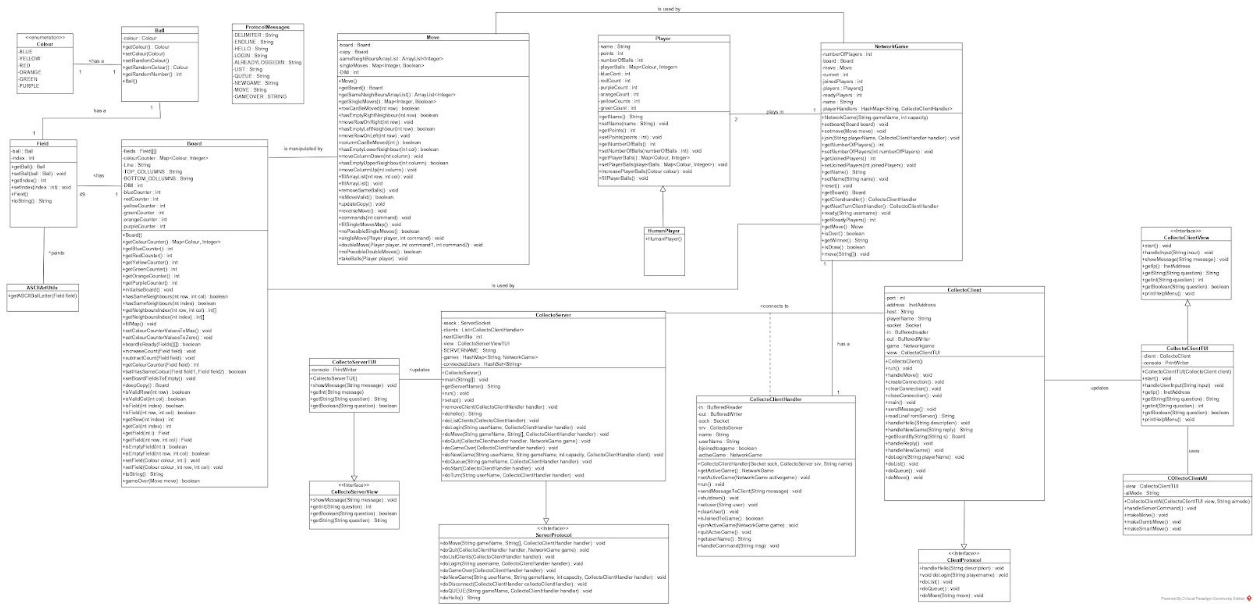
Functional Requirements for the Server:

1. In the run method of the CollectoServer class, the user is asked for a port and then the application tries to create a socket with the given port and to connect the user to it.
2. If the socket can't be created at the port, an exception is thrown and the user is asked to try to connect with another port.
3. The server uses the Runnable interface in order to deal with multiple clients. When a new user is connected to the server, a new thread is started in the run() method for its client handler.
4. In order for this to work, classes ServerProtocol and ProtocolMessages are used.

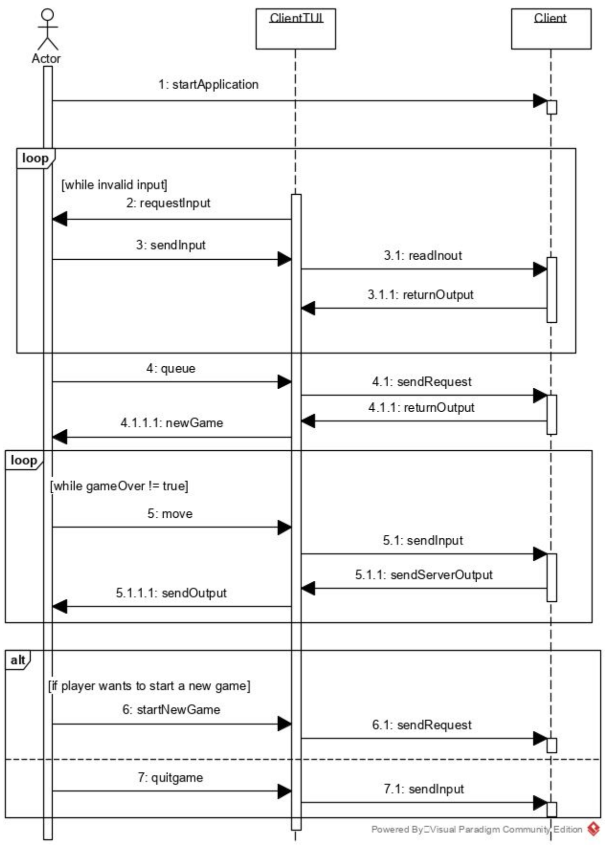
Functional requirements for the Client:

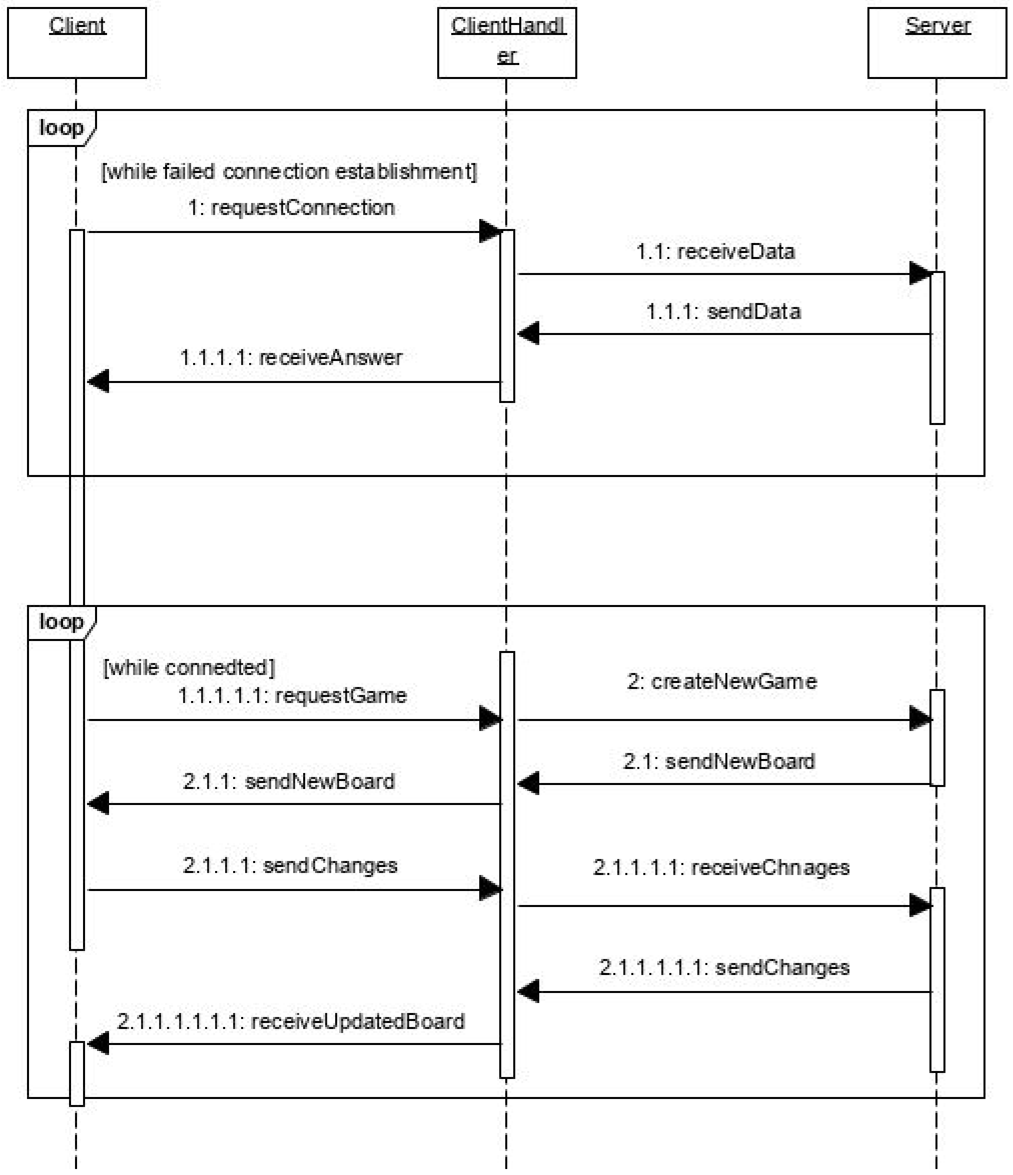
1. When a user inputs some messages in the local console the client needs to send this message to the server.
2. When the server sends a message back the client should receive it.
3. The client should convert the message from the server to a correct format. For example, if the server sends a "MOVE~3" back the client needs to split this message to MOVE and 3. Then the client will do this movement in it's own board and print this board on the console.

Realised Design



For our project, we followed the MVC pattern. In ClientHandler we have a NetworkGame that plays the role of a model, the Server and Client TUIs are the views and the ClientHandler is the controller. Our project follows a precise structure. We have a package with all the necessary implementation and classes to create a game board, control what is happening on the board, create players and create games. If we look at the class diagram we can see that all the classes from package gamedesign are connected to each other. We also have a separate package for the server-side of the project and a package for the client-side of the project. From the diagram, we can see that the client and server are connected to each other through the CollectoServer, the CollectoClient and the CollectoClientHandler. The networking part of the project is connected to the board, movement and game classes in order to function properly.
Our project also contains a package exceptions that contains all the exceptions that can be thrown while running the application. We have a package test that contains all the unit and integration testing and lastly, we have a package protocols with all the necessary protocols for the server and client to communicate.
By looking at the diagram we can see that one board can have 49 fields and 1 field has always 1 ball with 1 colour. One board is controlled by one or more move classes and one game is created with the board. We also have one server and many clients and client handlers that deal with the multiple clients. The Player class can have 2 instances created in the NetworkGame class.
An important matter for us is that we decided on making every field have a constant index when printed out. We were required to make every colour type represented as a number but we decided that it would be more suitable to make the fields to be numbered with their own index and just to make the fields themselves coloured in their respectful colour. We think that this would make board orientation easier for players.


Actor
ClientTUI
Client
1: startApplication
loop
[while invalid input]
2: requestInput
3: sendInput
3.1: readInout
3.1.1: returnOutput
4: queue
4.1: sendRequest
4.1.1: returnOutput
4.1.1.1: newGame
loop
[while gameOver != true]
5: move
5.1: sendInput
5.1.1: sendServerOutput
5.1.1.1: sendOutput
alt
[if player wants to start a new game]
6: startNewGame
6.1: sendRequest
7: quitgame
7.1: sendInput
Powered By Visual Paradigm Community Edition



Concurrency Mechanism

The reason we need to have concurrency design is this game needs to satisfy the requirement that two clients battle on the server. So we need to realise a function that we can at least run two clients at the same time. That’s the reason why we use thread when we program our client class.

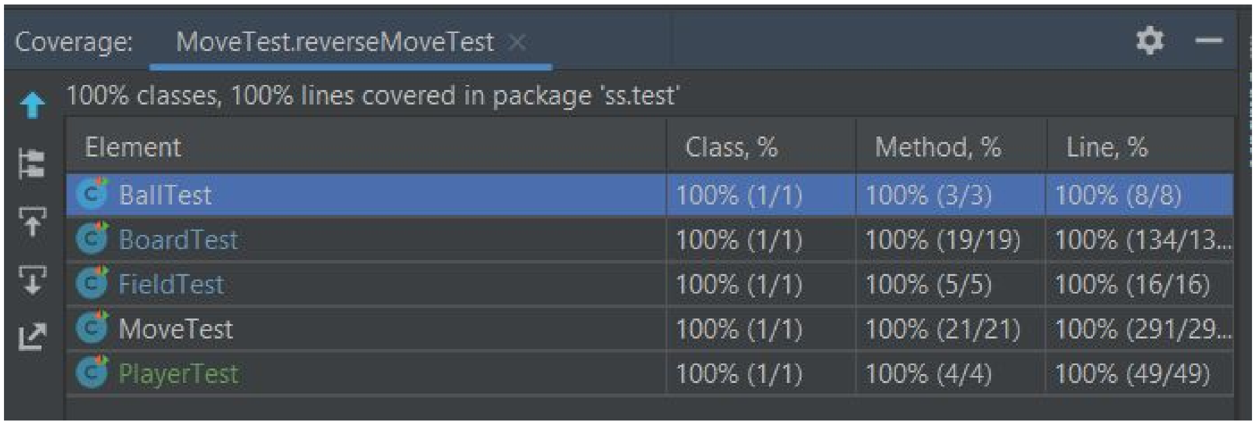
The first thing to ensure safety is by not sharing variables between threads as much as possible. We need to make sure that the variable can only be used in its own thread. Second, we tried to set the type of every variable to private. So the variable won’t be changed by other threads. We also use synchronization mechanisms. It is used to ensure that whenever the server receives the message from the client and makes replies both clients will get the same reply and show it on the console. It's really important in our game because we need to update the board after the player makes every movement.

Design Reflection

At the beginning of the project, we did not design this project in accordance with our final needs. We just divide the whole project into several parts to realize them one by one. But when we merged these completed parts together, we found that there were a lot of problems. Because of the existence of thread, it's really difficult to run the program after combining them. We need to use only one user to test whether the server and the client are smooth, and then implement the interaction between the two players.
At the beginning of the design, we should implement the server and client functions first. We should implement all the received messages that the server and client will handle according to the sample game given by the professor. After that, we will implement the specific board and movement of the game according to the written client. If we did a simple analysis at the very beginning and did what I just said, we would save a lot of time to modify the code.
We started to make a board and realise the move of this board. So at that time we only write the method we need to use but we ignore the method we will use in client and server. So that's the disadvantage of starting from realising the board and move.
We all learn from this time's design that we need to spend some time on making a plan that which part we should do first will be better. This will greatly reduce our time to modify the code in the end. And we also learn that we need to imagine that we are the player who will play our game, so we will find more problems and implement it in time.

Testing

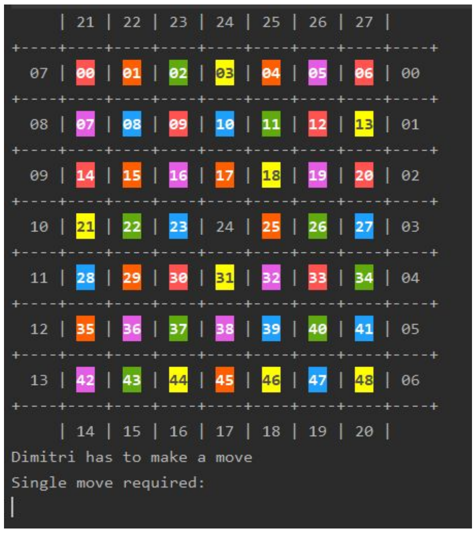
Classes Ball, Board, Field, Player and Move have all their methods covered by JUnit tests in the test package of the project with a 100% coverage. Some of the methods in those classes could not be tested so they were not included in the testing.



100% classes, 100% lines covered in package 'ss.test'

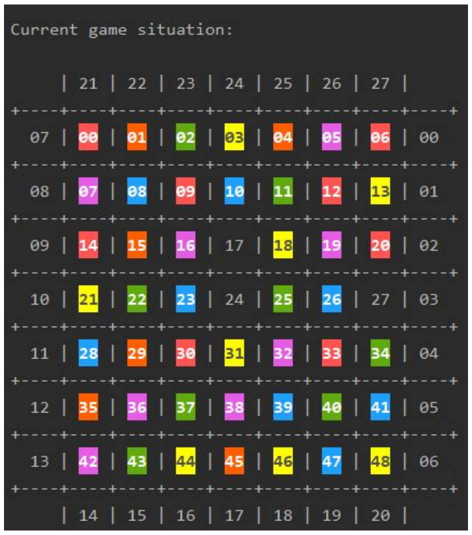
Element	Class, %	Method, %	Line, %
BallTest	100% (1/1)	100% (3/3)	100% (8/8)
BoardTest	100% (1/1)	100% (19/19)	100% (134/13...
FieldTest	100% (1/1)	100% (5/5)	100% (16/16)
MoveTest	100% (1/1)	100% (21/21)	100% (291/29...
PlayerTest	100% (1/1)	100% (4/4)	100% (49/49)

Testing for the Player, Board and Move classes was provided also by creating a prototype of the game in the Game class. Then an instance of Game is created in the class Collecto with a public static void main method. There we extensively tested if the move commands manipulate the board of the game correctly and if the board itself is created correctly. Because the commands require user input they could not be tested using JUnit tests. There we tried various cases and commands to see every possible outcome of the board manipulation. In the Game method, there is no functionality that checks if a player made an invalid double move in the command range of 0 and 27. All other functionality required to play a normal game is provided in the Game class. It also displays at the end of the game the points and balls a player has. Because we cannot provide coverage pictures and JUnit tests we tested the system by running the Collecto class and showing you the following results:



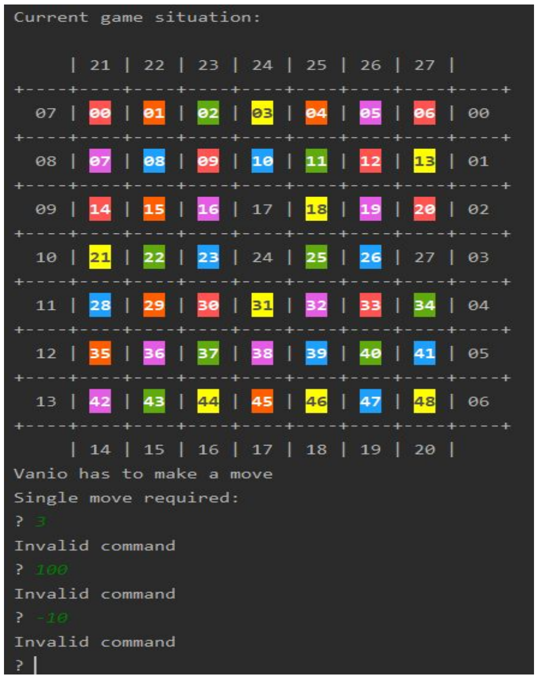
```
     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
Dimitri has to make a move
Single move required:
|
```

This is when we run the main method. If we enter 3 as input we will move the third row to the left and get rid of 2 orange balls and receive the following output:



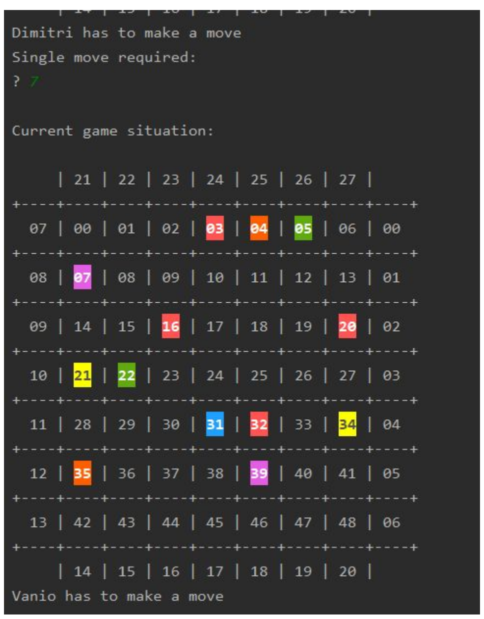
```
Current game situation:

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
```

Afterwards, we entered as input the command 3 again. According to the game rules, this move should be invalid and the player is required to enter a new valid move. Furthermore, if we enter a number above 27 or less than 0, we will still receive an “Invalid command” message because the number is not in the commands range. The game will continue showing warnings until we enter a valid single. An example of these situations can be seen in the screenshot below:



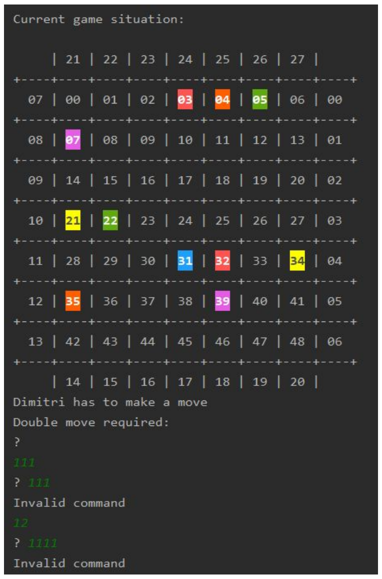
```
Current game situation:

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
Vanio has to make a move
Single move required:
? 3
Invalid command
? 100
Invalid command
? -10
Invalid command
? |
```

In the next screenshot, we can see that the players get their turns as they should normally:

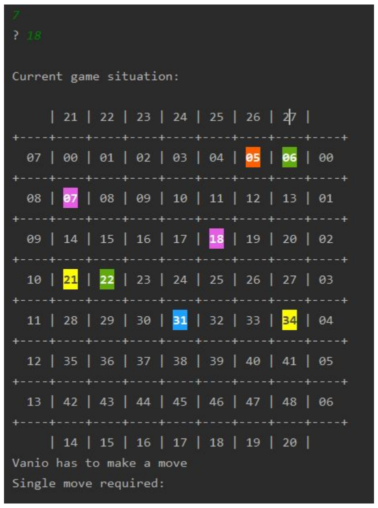


```
Dimitri has to make a move
Single move required:
? 7

Current game situation:

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
Vanio has to make a move
```

Next, we can see that once there are no possible single moves a double move is required. Furthermore, moves that are out of the range of the allowed commands are not accepted. Also, it can be seen that if after a double move a single move is possible then it is required to be done instead of accepting a double move:

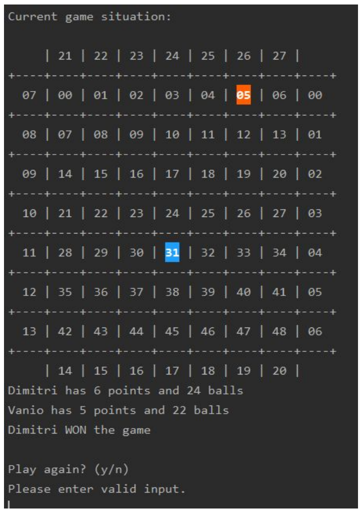


```
Current game situation:

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
Dimitri has to make a move
Double move required:
?
111
? 111
Invalid command
12
? 1111
Invalid command
```



```
7
? 18

Current game situation:

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
Vanio has to make a move
Single move required:
```

Once no more possible single or double moves can be performed the game is over. We receive the balls and points both players have and the winner of the game:

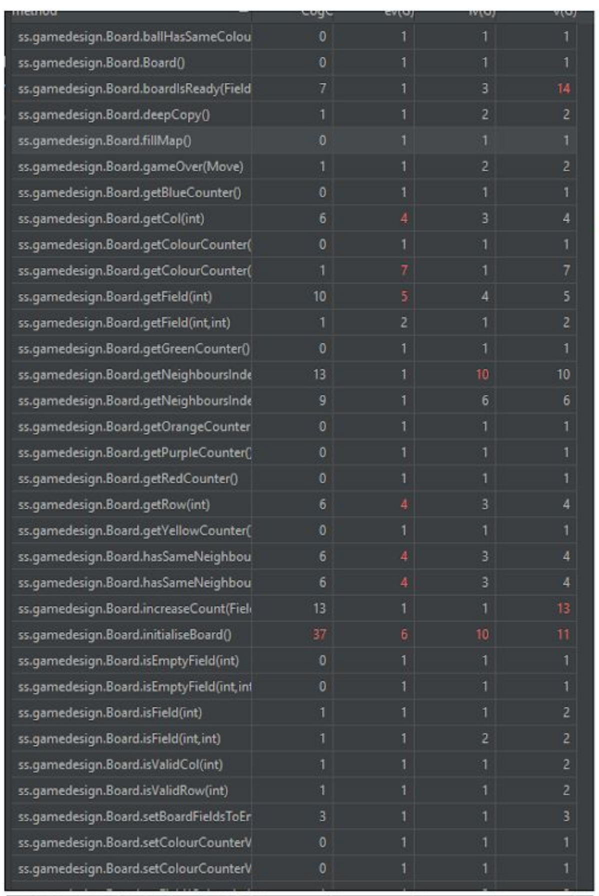
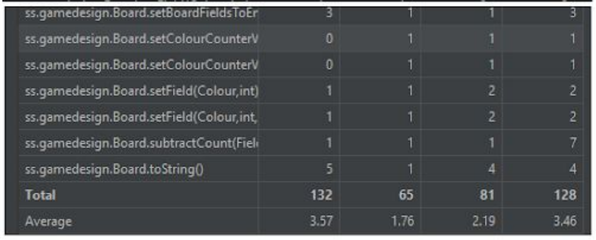


```
Current game situation:

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
Dimitri has 6 points and 24 balls
Vanio has 5 points and 22 balls
Dimitri WON the game

Play again? (y/n)
Please enter valid input.
```

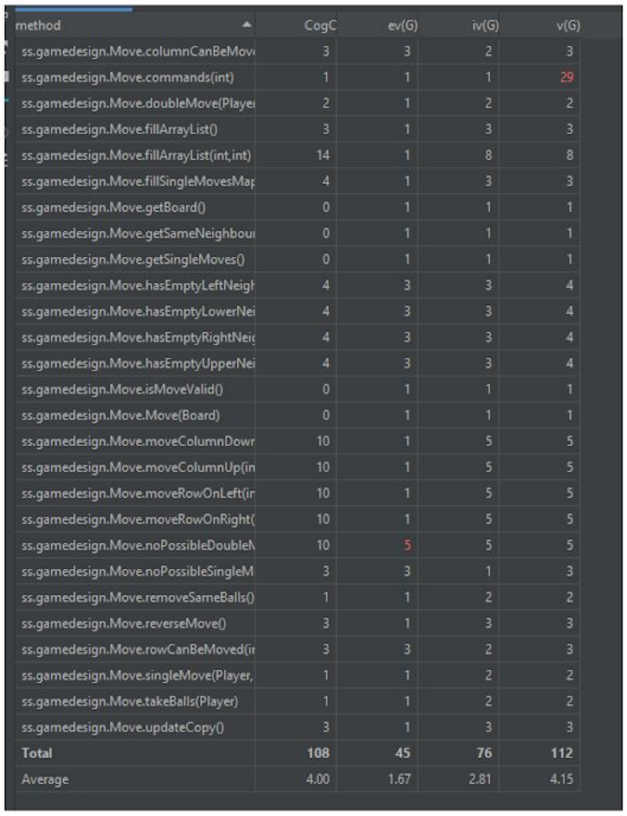
For classes NetworkGame, Colour, ASCIIArtUtils we cannot provide JUnit tests because there are no methods to be tested on their own.

Metrics:

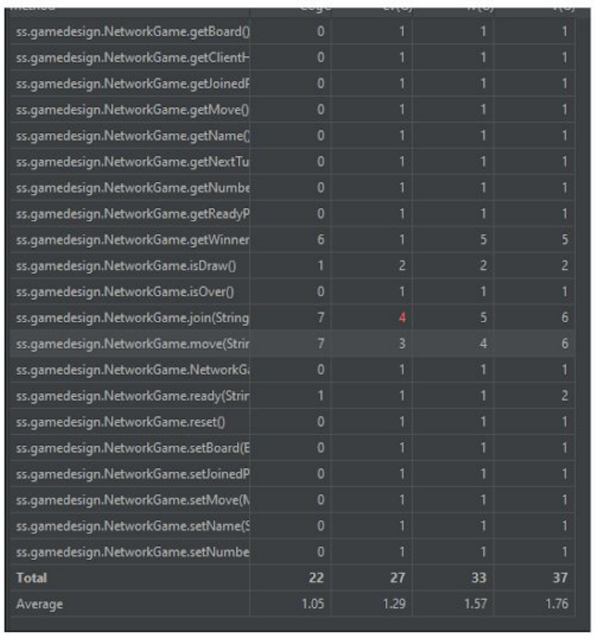
gameDesign classes metrics:

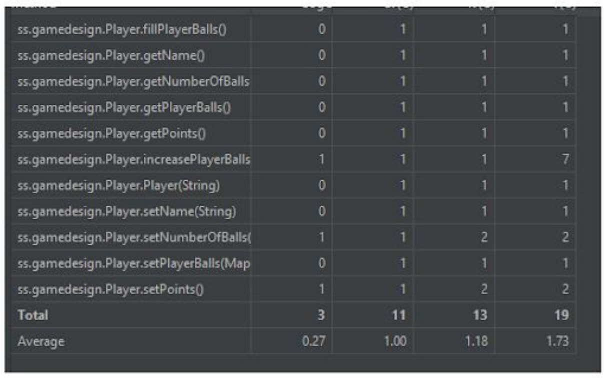
ss.gamedesign.Board.ballHasSameColou	0	1	1	1
ss.gamedesign.Board.Board()	0	1	1	1
ss.gamedesign.Board.boardIsReady(Field	7	1	3	14
ss.gamedesign.Board.deepCopy()	1	1	2	2
ss.gamedesign.Board.fillMap()	0	1	1	1
ss.gamedesign.Board.gameOver(Move)	1	1	2	2
ss.gamedesign.Board.getBlueCounter()	0	1	1	1
ss.gamedesign.Board.getCol(int)	6	4	3	4
ss.gamedesign.Board.getColourCounter(	0	1	1	1
ss.gamedesign.Board.getColourCounter(	1	7	1	7
ss.gamedesign.Board.getField(int)	10	5	4	5
ss.gamedesign.Board.getField(int,int)	1	2	1	2
ss.gamedesign.Board.getGreenCounter()	0	1	1	1
ss.gamedesign.Board.getNeighboursInde	13	1	10	10
ss.gamedesign.Board.getNeighboursInde	9	1	6	6
ss.gamedesign.Board.getOrangeCounter	0	1	1	1
ss.gamedesign.Board.getPurpleCounter()	0	1	1	1
ss.gamedesign.Board.getRedCounter()	0	1	1	1
ss.gamedesign.Board.getRow(int)	6	4	3	4
ss.gamedesign.Board.getYellowCounter(	0	1	1	1
ss.gamedesign.Board.hasSameNeighbou	6	4	3	4
ss.gamedesign.Board.hasSameNeighbou	6	4	3	4
ss.gamedesign.Board.increaseCount(Fiel	13	1	1	13
ss.gamedesign.Board.initialiseBoard()	37	6	10	11
ss.gamedesign.Board.isEmptyField(int)	0	1	1	1
ss.gamedesign.Board.isEmptyField(int,int	0	1	1	1
ss.gamedesign.Board.isField(int)	1	1	1	2
ss.gamedesign.Board.isField(int,int)	1	1	2	2
ss.gamedesign.Board.isValidCol(int)	1	1	1	2
ss.gamedesign.Board.isValidRow(int)	1	1	1	2
ss.gamedesign.Board.setBoardFieldsToEr	3	1	1	3
ss.gamedesign.Board.setColourCounterV	0	1	1	1
ss.gamedesign.Board.setColourCounterV	0	1	1	1
ss.gamedesign.Board.setField(Colour,int)	1	1	2	2
ss.gamedesign.Board.setField(Colour,int,	1	1	2	2
ss.gamedesign.Board.subtractCount(Fiel	1	1	1	7
ss.gamedesign.Board.toString()	5	1	4	4
Total	**132**	**65**	**81**	**128**
Average	3.57	1.76	2.19	3.46



method	CogC	ev(G)	iv(G)	v(G)
ss.gamedesign.Move.columnCanBeMov	3	3	2	3
ss.gamedesign.Move.commands(int)	1	1	1	29
ss.gamedesign.Move.doubleMove(Player	2	1	2	2
ss.gamedesign.Move.fillArrayList()	3	1	3	3
ss.gamedesign.Move.fillArrayList(int,int)	14	1	8	8
ss.gamedesign.Move.fillSingleMovesMap	4	1	3	3
ss.gamedesign.Move.getBoard()	0	1	1	1
ss.gamedesign.Move.getSameNeighbou	0	1	1	1
ss.gamedesign.Move.getSingleMoves()	0	1	1	1
ss.gamedesign.Move.hasEmptyLeftNeigh	4	3	3	4
ss.gamedesign.Move.hasEmptyLowerNei	4	3	3	4
ss.gamedesign.Move.hasEmptyRightNeig	4	3	3	4
ss.gamedesign.Move.hasEmptyUpperNei	4	3	3	4
ss.gamedesign.Move.isMoveValid()	0	1	1	1
ss.gamedesign.Move.Move(Board)	0	1	1	1
ss.gamedesign.Move.moveColumnDowr	10	1	5	5
ss.gamedesign.Move.moveColumnUp(in	10	1	5	5
ss.gamedesign.Move.moveRowOnLeft(ir	10	1	5	5
ss.gamedesign.Move.moveRowOnRight(	10	1	5	5
ss.gamedesign.Move.noPossibleDoubleN	10	5	5	5
ss.gamedesign.Move.noPossibleSingleM	3	3	1	3
ss.gamedesign.Move.removeSameBalls()	1	1	2	2
ss.gamedesign.Move.reverseMove()	3	1	3	3
ss.gamedesign.Move.rowCanBeMoved(ir	3	3	2	3
ss.gamedesign.Move.singleMove(Player,	1	1	2	2
ss.gamedesign.Move.takeBalls(Player)	1	1	2	2
ss.gamedesign.Move.updateCopy()	3	1	3	3
Total	**108**	**45**	**76**	**112**
Average	4.00	1.67	2.81	4.15

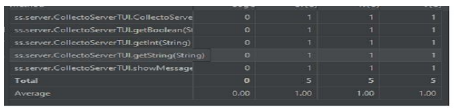


ss.gamedesign.NetworkGame.getBoard()	0	1	1	1
ss.gamedesign.NetworkGame.getClientH	0	1	1	1
ss.gamedesign.NetworkGame.getJoinedP	0	1	1	1
ss.gamedesign.NetworkGame.getMove()	0	1	1	1
ss.gamedesign.NetworkGame.getName()	0	1	1	1
ss.gamedesign.NetworkGame.getNextTu	0	1	1	1
ss.gamedesign.NetworkGame.getNumbe	0	1	1	1
ss.gamedesign.NetworkGame.getReadyP	0	1	1	1
ss.gamedesign.NetworkGame.getWinner	6	1	5	5
ss.gamedesign.NetworkGame.isDraw()	1	2	2	2
ss.gamedesign.NetworkGame.isOver()	0	1	1	1
ss.gamedesign.NetworkGame.join(String	7	4	5	6
ss.gamedesign.NetworkGame.move(Strir	7	3	4	6
ss.gamedesign.NetworkGame.NetworkG	0	1	1	1
ss.gamedesign.NetworkGame.ready(Strir	1	1	1	2
ss.gamedesign.NetworkGame.reset()	0	1	1	1
ss.gamedesign.NetworkGame.setBoard(E	0	1	1	1
ss.gamedesign.NetworkGame.setJoinedP	0	1	1	1
ss.gamedesign.NetworkGame.setMove(N	0	1	1	1
ss.gamedesign.NetworkGame.setName(S	0	1	1	1
ss.gamedesign.NetworkGame.setNumbe	0	1	1	1
Total	**22**	**27**	**33**	**37**
Average	1.05	1.29	1.57	1.76

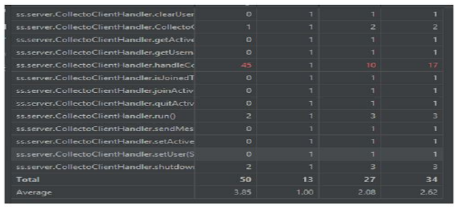


ss.gamedesign.Player.fillPlayerBalls()	0	1	1	1
ss.gamedesign.Player.getName()	0	1	1	1
ss.gamedesign.Player.getNumberOfBalls	0	1	1	1
ss.gamedesign.Player.getPlayerBalls()	0	1	1	1
ss.gamedesign.Player.getPoints()	0	1	1	1
ss.gamedesign.Player.increasePlayerBalls	1	1	1	7
ss.gamedesign.Player.Player(String)	0	1	1	1
ss.gamedesign.Player.setName(String)	0	1	1	1
ss.gamedesign.Player.setNumberOfBalls(	1	1	2	2
ss.gamedesign.Player.setPlayerBalls(Map	0	1	1	1
ss.gamedesign.Player.setPoints()	1	1	2	2
Total	**3**	**11**	**13**	**19**
Average	0.27	1.00	1.18	1.73

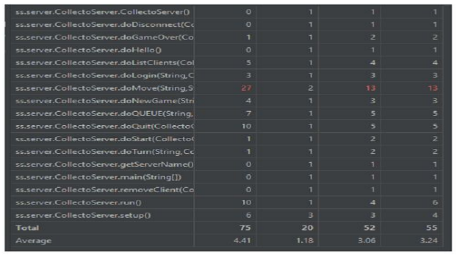
Server classes metrics:



ss.server.CollectoServerTUI.CollectoServe	0	1	1	1
ss.server.CollectoServerTUI.getBoolean(St	0	1	1	1
ss.server.CollectoServerTUI.getInt(String)	0	1	1	1
ss.server.CollectoServerTUI.getString(String)	0	1	1	1
ss.server.CollectoServerTUI.showMessage	0	1	1	1
Total	**0**	**5**	**5**	**5**
Average	0.00	1.00	1.00	1.00

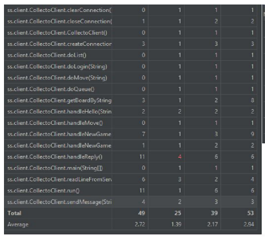


ss.server.CollectoClientHandler.clearUser	0	1	1	1
ss.server.CollectoClientHandler.Collecto(	1	1	2	2
ss.server.CollectoClientHandler.getActive	0	1	1	1
ss.server.CollectoClientHandler.getUsern	0	1	1	1
ss.server.CollectoClientHandler.handleCc	45	1	10	17
ss.server.CollectoClientHandler.isJoinedT	0	1	1	1
ss.server.CollectoClientHandler.joinActiv	0	1	1	1
ss.server.CollectoClientHandler.quitActiv	0	1	1	1
ss.server.CollectoClientHandler.run()	2	1	3	3
ss.server.CollectoClientHandler.sendMes	0	1	1	1
ss.server.CollectoClientHandler.setActive	0	1	1	1
ss.server.CollectoClientHandler.setUser(S	0	1	1	1
ss.server.CollectoClientHandler.shutdow	2	1	3	3
Total	**50**	**13**	**27**	**34**
Average	3.85	1.00	2.08	2.62

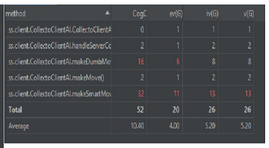


ss.server.CollectoServer.CollectoServer()	0	1	1	1
ss.server.CollectoServer.doDisconnect(Cc	0	1	1	1
ss.server.CollectoServer.doGameOver(Co	1	1	2	2
ss.server.CollectoServer.doHello()	0	1	1	1
ss.server.CollectoServer.doListClients(Col	5	1	4	4
ss.server.CollectoServer.doLogin(String,C	3	1	3	3
ss.server.CollectoServer.doMove(String,S	27	2	13	13
ss.server.CollectoServer.doNewGame(Stri	4	1	3	3
ss.server.CollectoServer.doQUEUE(String,	7	1	5	5
ss.server.CollectoServer.doQuit(Collecto(	10	1	5	5
ss.server.CollectoServer.doStart(Collecto(	1	1	2	2
ss.server.CollectoServer.doTurn(String,Cc	1	1	2	2
ss.server.CollectoServer.getServerName()	0	1	1	1
ss.server.CollectoServer.main(String[])	0	1	1	1
ss.server.CollectoServer.removeClient(Co	0	1	1	1
ss.server.CollectoServer.run()	10	1	4	6
ss.server.CollectoServer.setup()	6	3	3	4
Total	**75**	**20**	**52**	**55**
Average	4.41	1.18	3.06	3.24

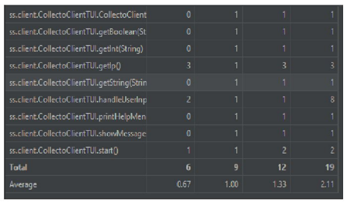
Client classes metrics



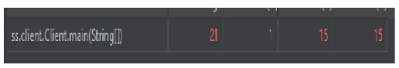
ss.client.CollectoClient.clearConnection(	0	1	1	1
ss.client.CollectoClient.closeConnection(	1	1	2	2
ss.client.CollectoClient.CollectoClient()	0	1	1	1
ss.client.CollectoClient.createConnectior	3	1	3	3
ss.client.CollectoClient.doList()	0	1	1	1
ss.client.CollectoClient.doLogin(String)	0	1	1	1
ss.client.CollectoClient.doMove(String)	0	1	1	1
ss.client.CollectoClient.doQueue()	0	1	1	1
ss.client.CollectoClient.getBoardByString	3	1	2	8
ss.client.CollectoClient.handleHello(Strin	2	2	2	2
ss.client.CollectoClient.handleMove()	0	1	1	1
ss.client.CollectoClient.handleNewGame	7	1	3	9
ss.client.CollectoClient.handleNewGame	1	1	2	2
ss.client.CollectoClient.handleReply()	11	4	6	6
ss.client.CollectoClient.main(String[])	0	1	1	1
ss.client.CollectoClient.readLineFromSen	6	3	2	4
ss.client.CollectoClient.run()	11	1	6	6
ss.client.CollectoClient.sendMessage(Stri	4	2	3	3
Total	**49**	**25**	**39**	**53**
Average	2.72	1.39	2.17	2.94



method ▲	CogC	ev(G)	iv(G)	v(G)
ss.client.CollectoClientAI.CollectoClientA	0	1	1	1
ss.client.CollectoClientAI.handleServerCc	2	1	2	2
ss.client.CollectoClientAI.makeDumbMo	16	6	8	8
ss.client.CollectoClientAI.makeMove()	2	1	2	2
ss.client.CollectoClientAI.makeSmartMov	32	11	13	13
Total	**52**	**20**	**26**	**26**
Average	10.40	4.00	5.20	5.20

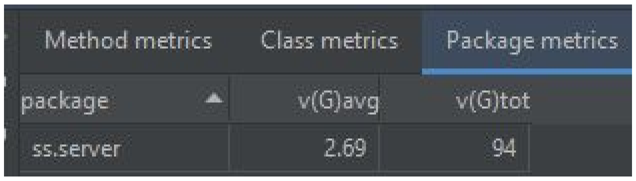


ss.client.CollectoClientTUI.CollectoClient	0	1	1	1
ss.client.CollectoClientTUI.getBoolean(St	0	1	1	1
ss.client.CollectoClientTUI.getInt(String)	0	1	1	1
ss.client.CollectoClientTUI.getIp()	3	1	3	3
ss.client.CollectoClientTUI.getString(Strin	0	1	1	1
ss.client.CollectoClientTUI.handleUserInp	2	1	1	8
ss.client.CollectoClientTUI.printHelpMen	0	1	1	1
ss.client.CollectoClientTUI.showMessage	0	1	1	1
ss.client.CollectoClientTUI.start()	1	1	2	2
Total	**6**	**9**	**12**	**19**
Average	0.67	1.00	1.33	2.11

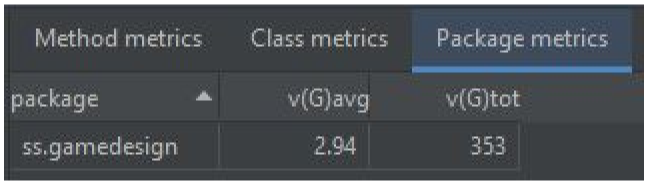


ss.client.Client.main(String[])	21	1	15	15

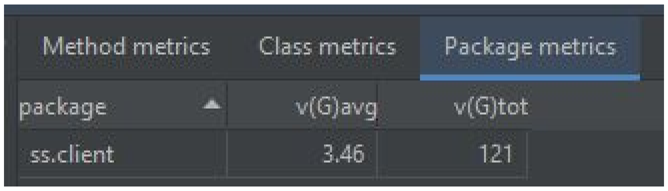
Most important packages metrics and the whole project metrics:


Method metrics | Class metrics | Package metrics

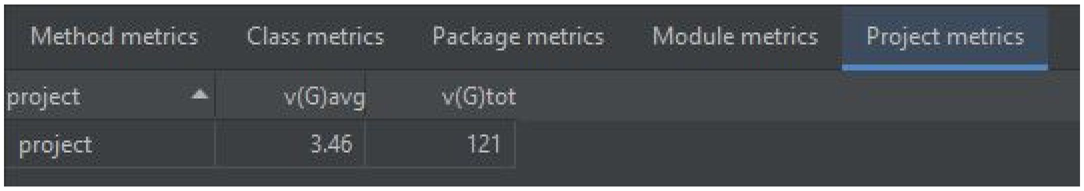
package	v(G)avg	v(G)tot
ss.server	2.69	94


Method metrics | Class metrics | Package metrics

package	v(G)avg	v(G)tot
ss.gamedesign	2.94	353


Method metrics | Class metrics | Package metrics

package	v(G)avg	v(G)tot
ss.client	3.46	121

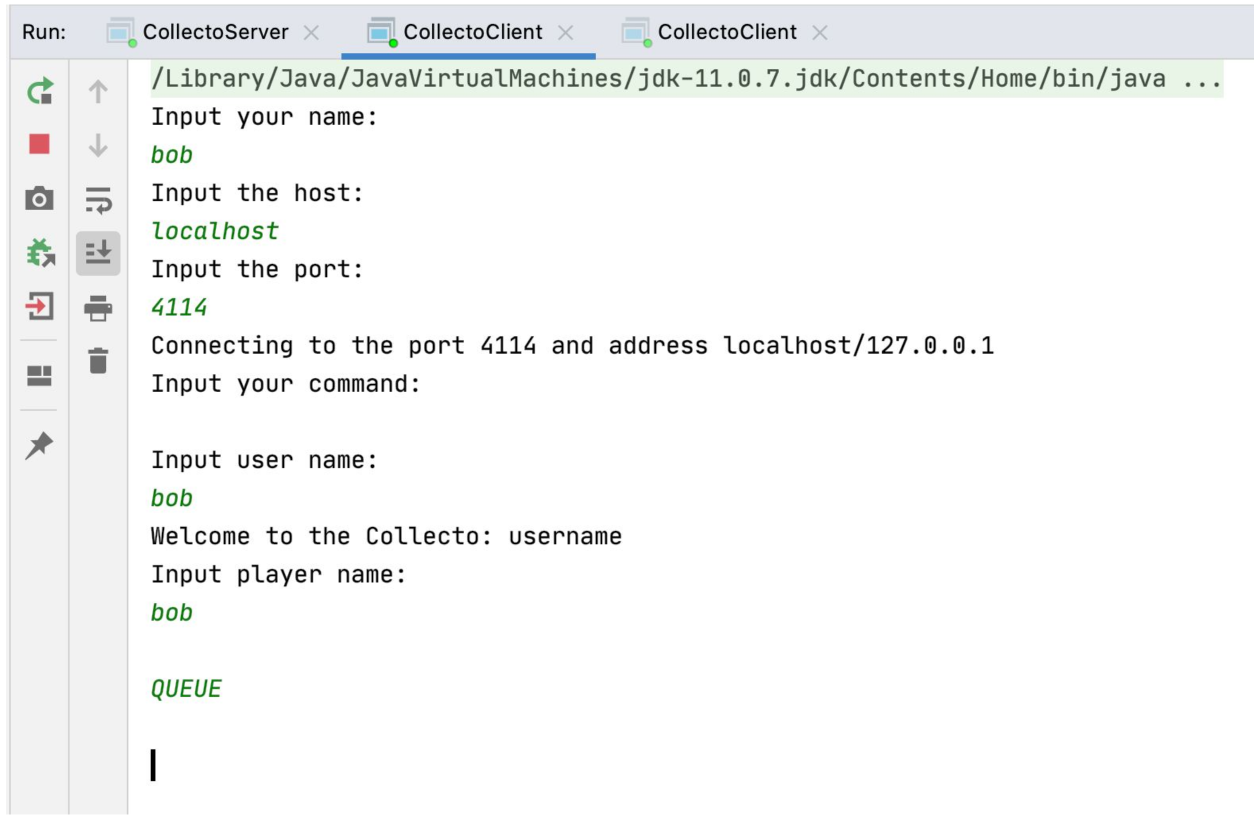

Method metrics | Class metrics | Package metrics | Module metrics | Project metrics

project	v(G)avg	v(G)tot
project	3.46	121

Although we tried to make our methods and classes as simple and clean as possible, we were hard-pressed by the deadline, so we were unable to take care of every single method. However, we do think that we have a relatively low metrics complexity. Our code is easy to maintain as long as it is documented. A problem that could arise could be the size of our bigger classes - like board and move. But even they are relatively ordered. We tried to keep them as small as possible but we do believe that when making a project of this scope big classes are unavoidable. Our unit tests cover the two most complex classes - board and move. They have the most methods and algorithms and as it can be seen from the screenshots they have the most complex methods in terms of metrics. That is why we focused on testing every single possible method of these classes that requires testing. We also tested the possible different scenarios for them so that we could see if they have any bugs that break the game. Our tests prove that all the methods in gameDesign are working correctly without bugs.

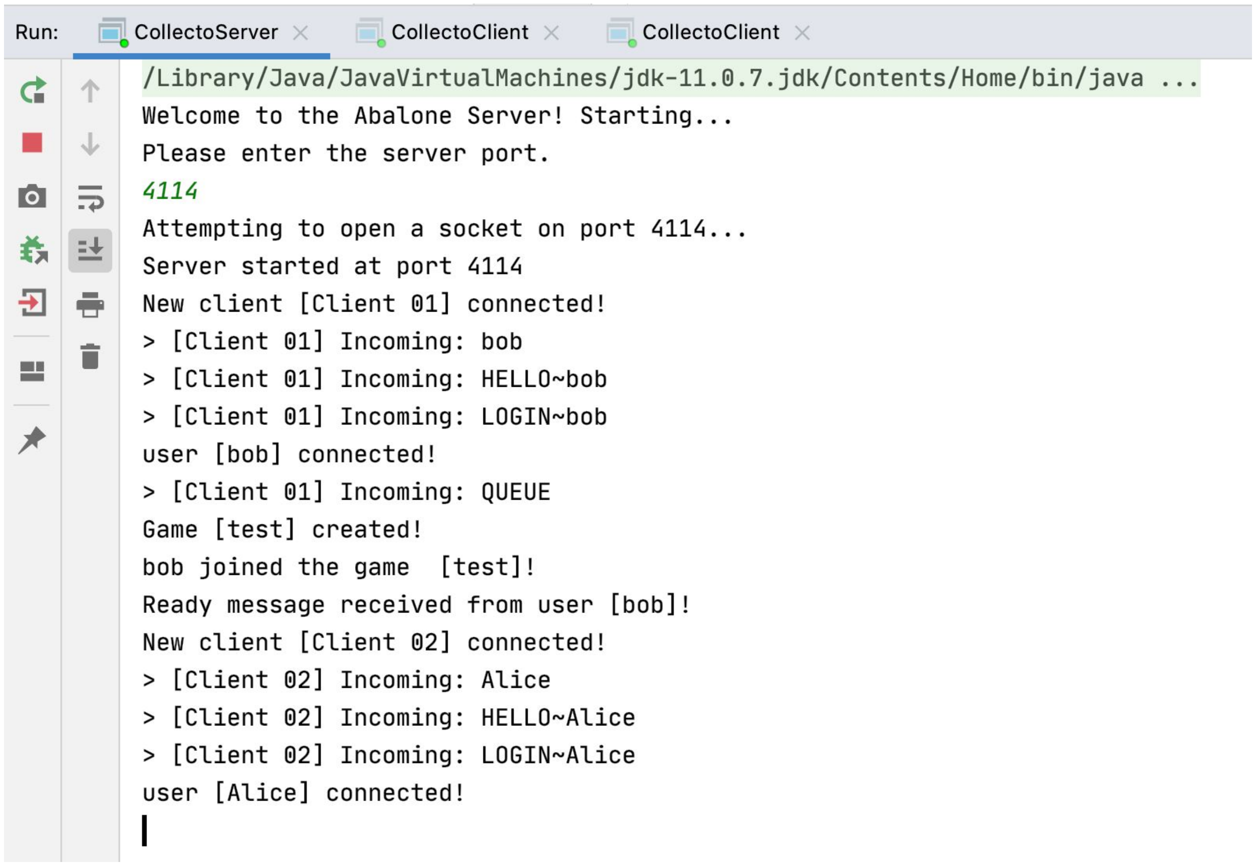
A problem that occurred while doing the project was that we focused a great deal of our time on having a perfectly working board and game logic. Because we focused on them too much, we ended up not having enough time to extensively test the whole system. We did test our end product but we do believe it was not enough. Although we did not fulfil our ambition to fully test every aspect of the game, we do believe that our testing strategy was a good choice. By extensively testing everything from the beginning, we were assured that while working on the later stages of the project there were no bugs in the basic part of our implementation. Also when faced with a bug in the networking part of the project we would know that the problem was not something from the gamedesign package.

Because there is no actual way to write unit tests for the networking part of the project we did the following tests:
For connecting a client to the server:



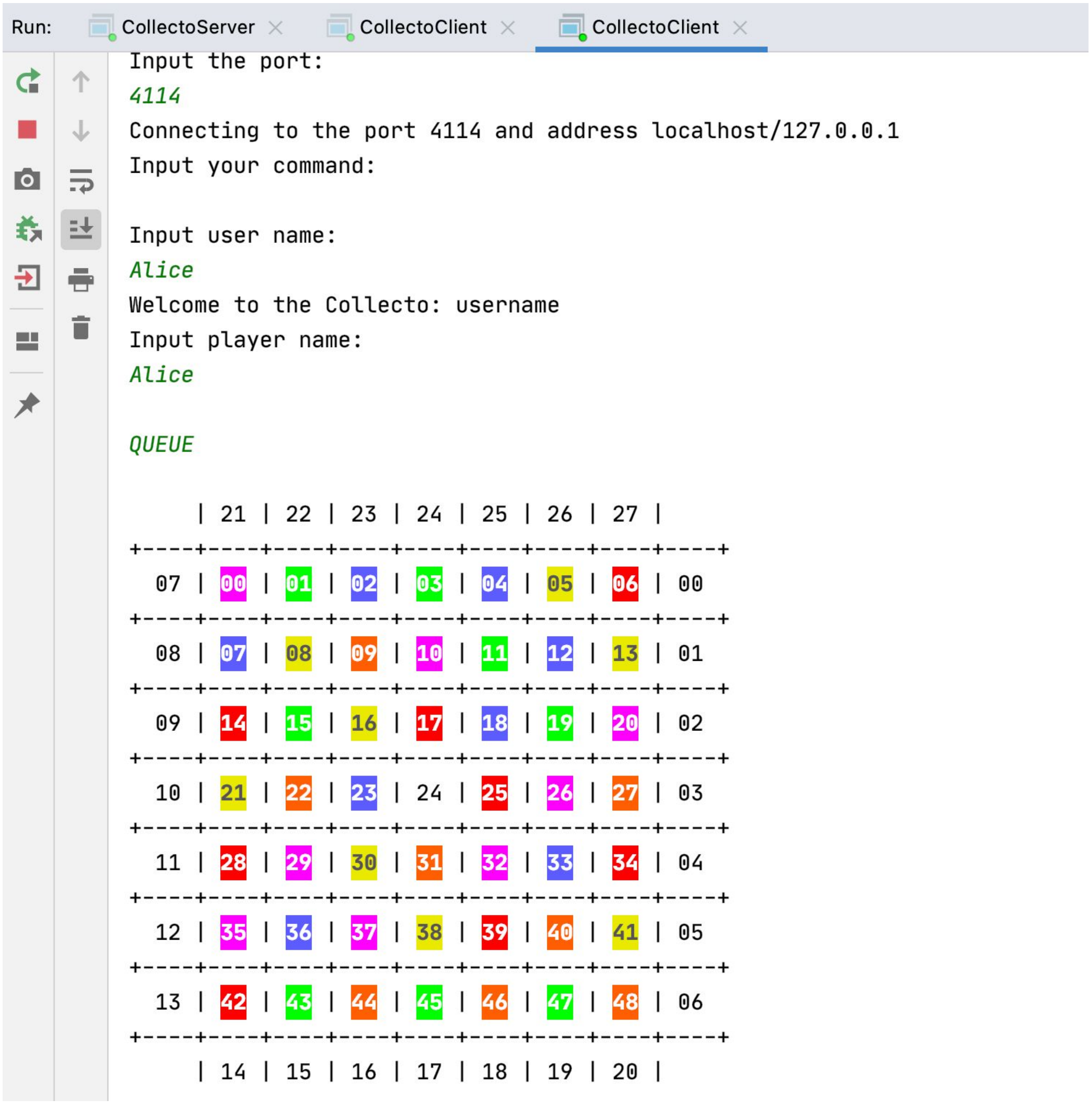
```
/Library/Java/JavaVirtualMachines/jdk-11.0.7.jdk/Contents/Home/bin/java ...
Input your name:
bob
Input the host:
localhost
Input the port:
4114
Connecting to the port 4114 and address localhost/127.0.0.1
Input your command:

Input user name:
bob
Welcome to the Collecto: username
Input player name:
bob

QUEUE
```

For starting a server:

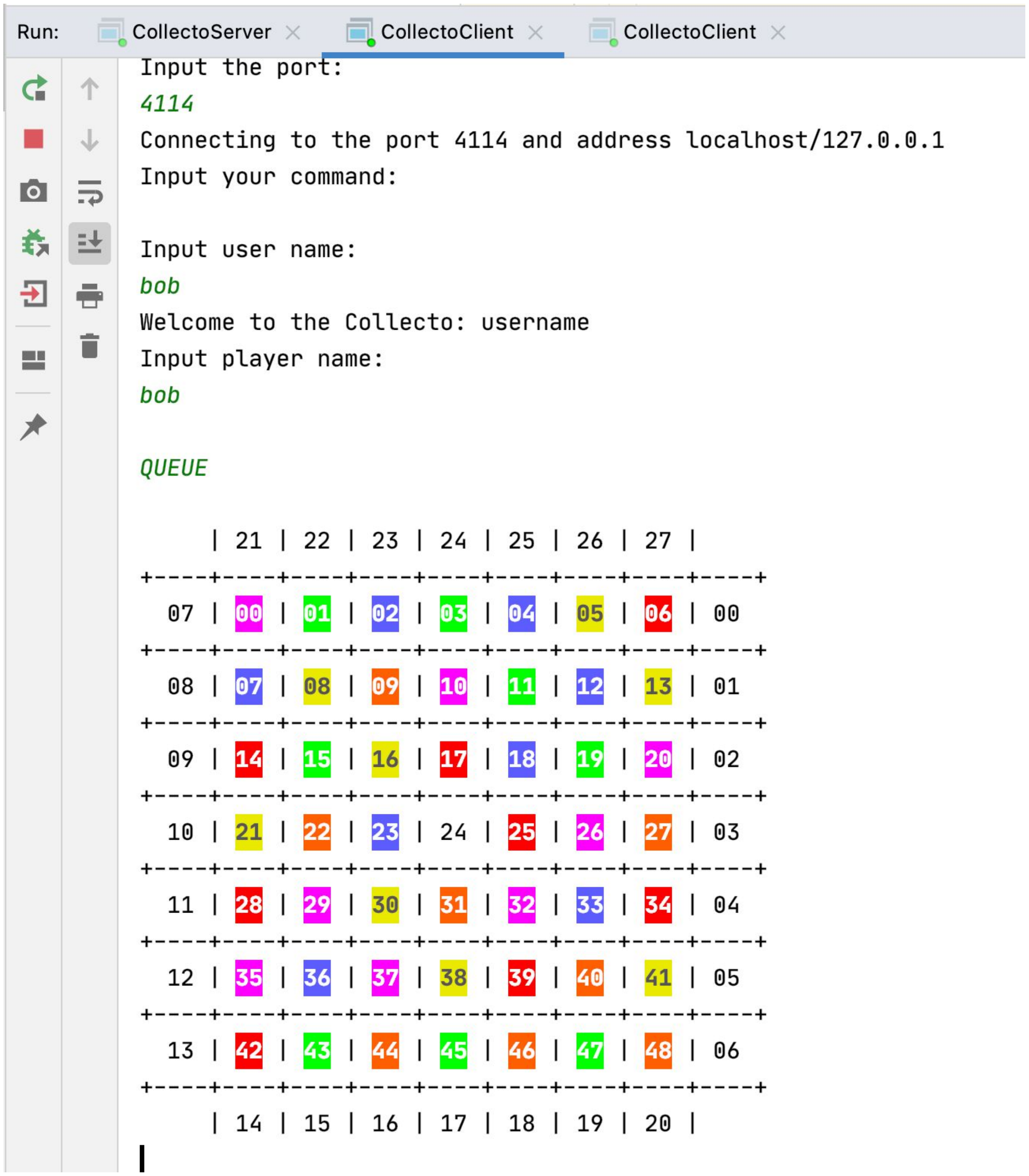


```
/Library/Java/JavaVirtualMachines/jdk-11.0.7.jdk/Contents/Home/bin/java ...
Welcome to the Abalone Server! Starting...
Please enter the server port.
4114
Attempting to open a socket on port 4114...
Server started at port 4114
New client [Client 01] connected!
> [Client 01] Incoming: bob
> [Client 01] Incoming: HELLO~bob
> [Client 01] Incoming: LOGIN~bob
user [bob] connected!
> [Client 01] Incoming: QUEUE
Game [test] created!
bob joined the game  [test]!
Ready message received from user [bob]!
New client [Client 02] connected!
> [Client 02] Incoming: Alice
> [Client 02] Incoming: HELLO~Alice
> [Client 02] Incoming: LOGIN~Alice
user [Alice] connected!
```

The new board after 2 players are queued for a new game:

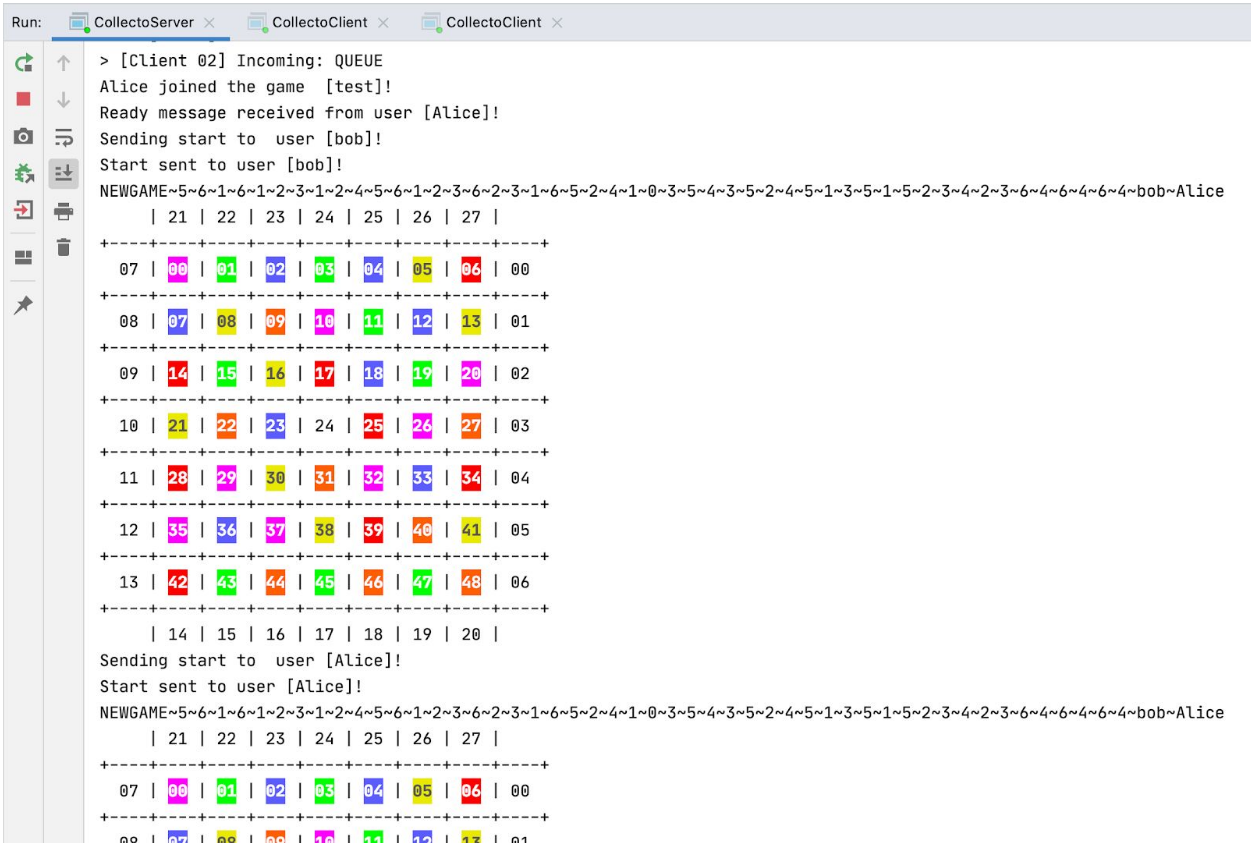


```
Input the port:
4114
Connecting to the port 4114 and address localhost/127.0.0.1
Input your command:

Input user name:
Alice
Welcome to the Collecto: username
Input player name:
Alice

QUEUE

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
```



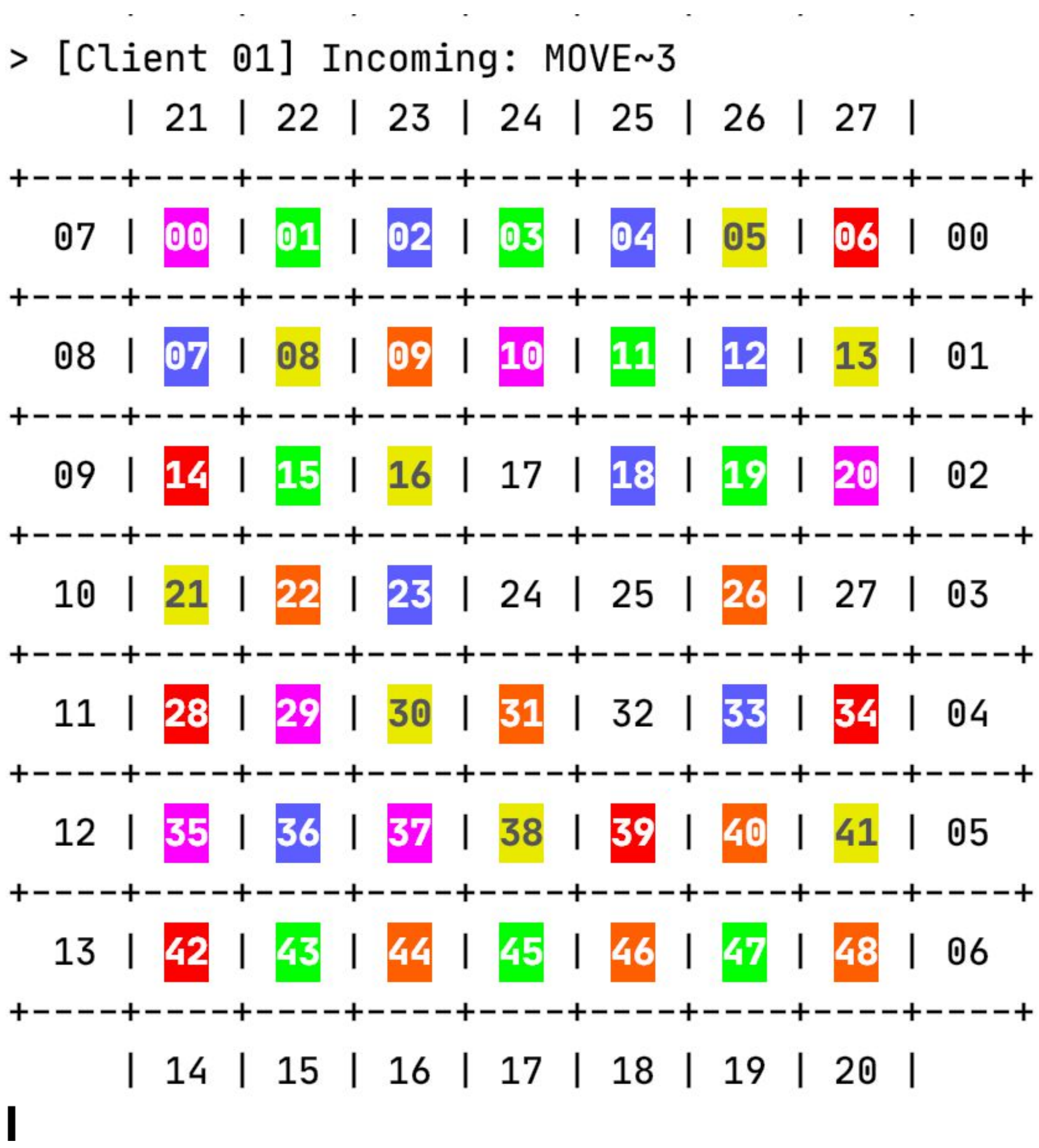
```
Input the port:
4114
Connecting to the port 4114 and address localhost/127.0.0.1
Input your command:

Input user name:
bob
Welcome to the Collecto: username
Input player name:
bob

QUEUE

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
```

The server output when players are queued for a game and start then begin a new game:

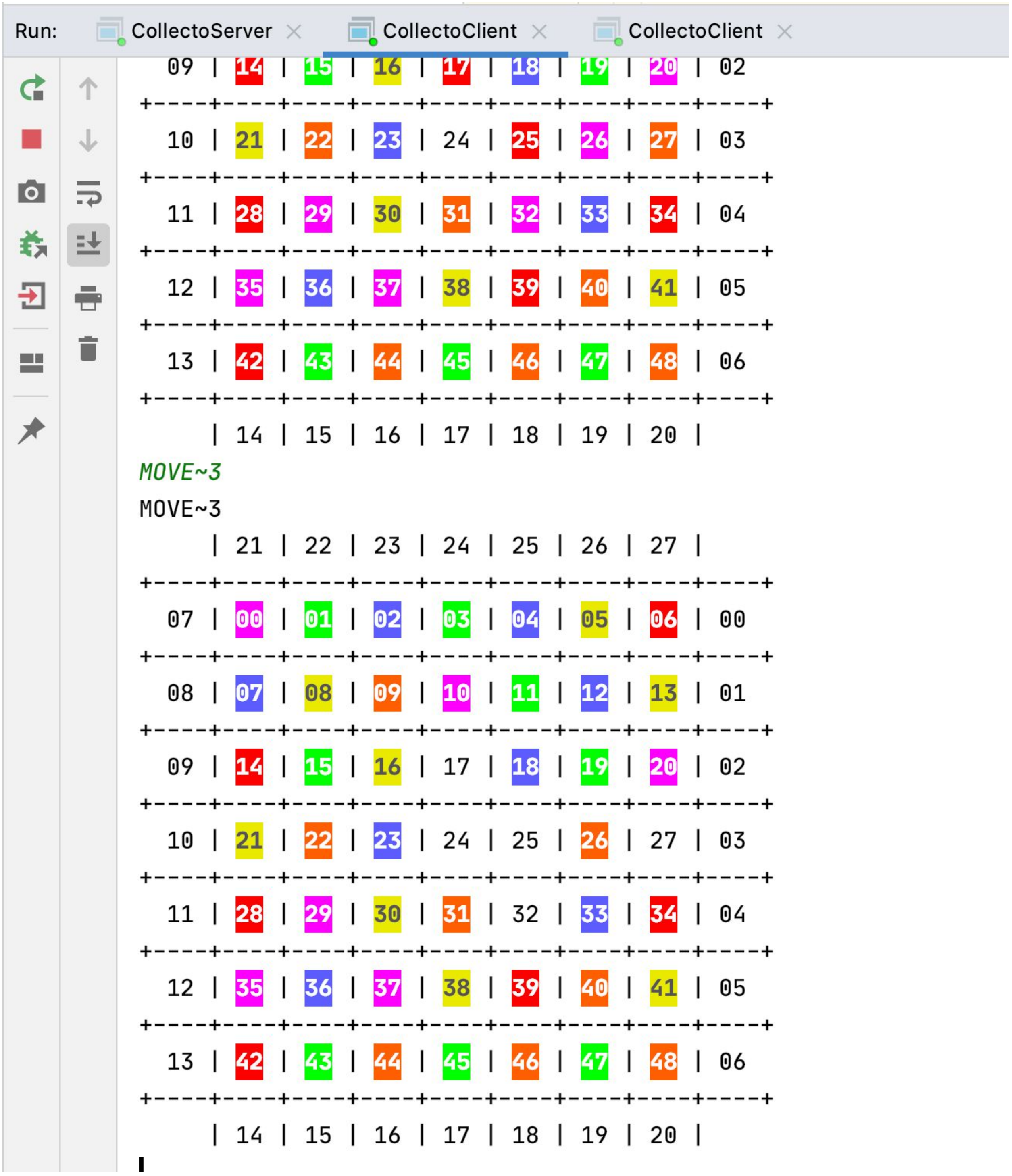


```
> [Client 02] Incoming: QUEUE
Alice joined the game  [test]!
Ready message received from user [Alice]!
Sending start to  user [bob]!
Start sent to user [bob]!
NEWGAME~5~6~1~6~1~2~3~1~2~4~5~6~1~2~3~6~2~3~1~6~5~2~4~1~0~3~5~4~3~5~2~4~5~1~3~5~1~5~2~3~4~2~3~6~4~6~4~6~4~bob~Alice
     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
Sending start to  user [Alice]!
Start sent to user [Alice]!
NEWGAME~5~6~1~6~1~2~3~1~2~4~5~6~1~2~3~6~2~3~1~6~5~2~4~1~0~3~5~4~3~5~2~4~5~1~3~5~1~5~2~3~4~2~3~6~4~6~4~6~4~bob~Alice
     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
```

For using the move command and checking if the board is updatet apropriately:

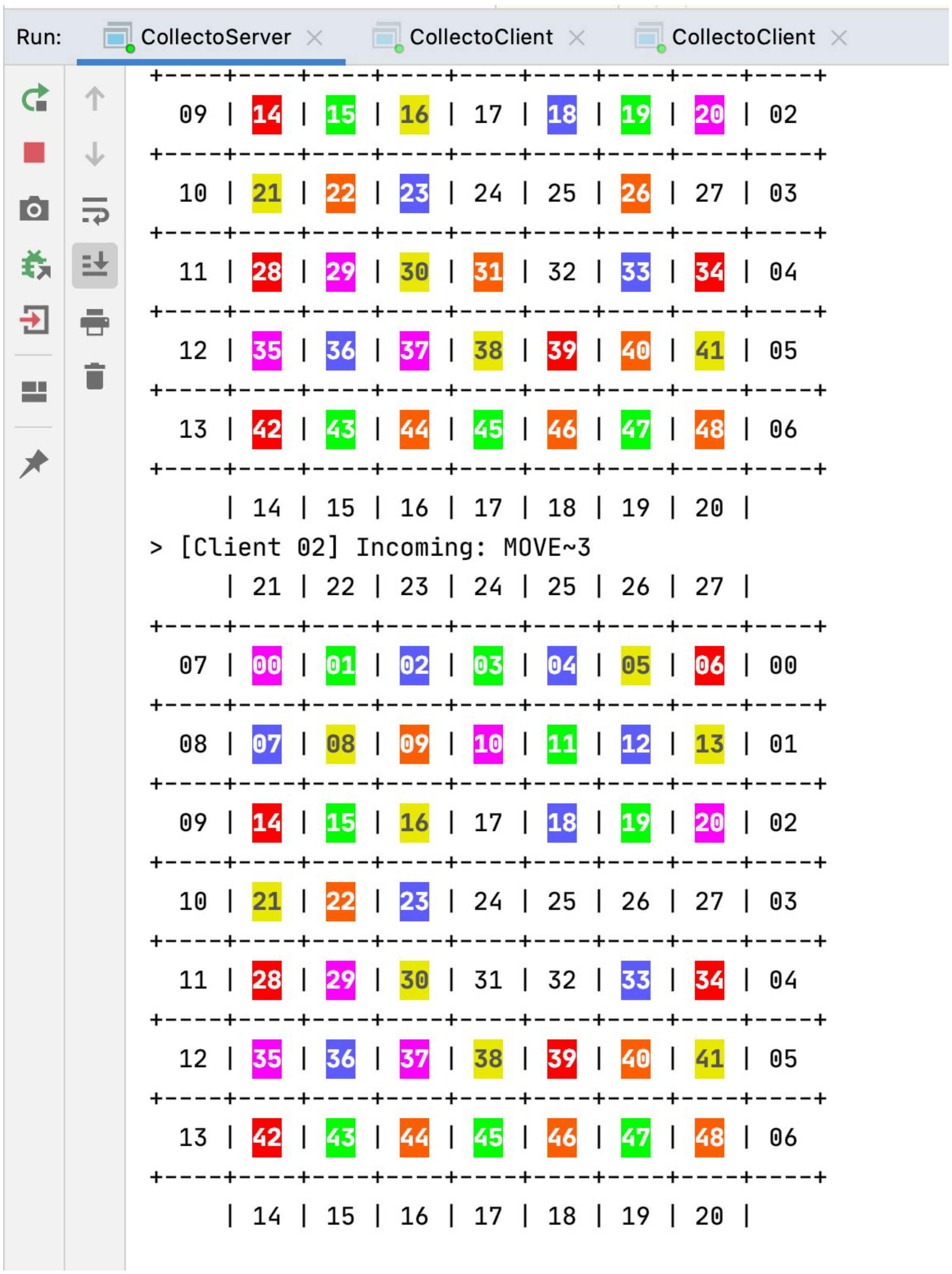


```
> [Client 01] Incoming: MOVE~3
     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
MOVE~3

     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
```



Run: CollectoServer × CollectoClient × CollectoClient ×

```
   09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
   10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
   11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
   12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
   13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
      | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
MOVE~3
MOVE~3
      | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
+----+----+----+----+----+----+----+----+----+
   07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
   08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
   09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
   10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
   11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
   12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
   13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
      | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
```


Run: CollectoServer CollectoClient CollectoClient

```
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
> [Client 02] Incoming: MOVE~3
     | 21 | 22 | 23 | 24 | 25 | 26 | 27 |

+----+----+----+----+----+----+----+----+----+
  07 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 00
+----+----+----+----+----+----+----+----+----+
  08 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 01
+----+----+----+----+----+----+----+----+----+
  09 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 02
+----+----+----+----+----+----+----+----+----+
  10 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 03
+----+----+----+----+----+----+----+----+----+
  11 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 04
+----+----+----+----+----+----+----+----+----+
  12 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 05
+----+----+----+----+----+----+----+----+----+
  13 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 06
+----+----+----+----+----+----+----+----+----+
     | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
```

Unfortunately, we did not find enough time to test every possible scenario for the board and so our product is not perfect. Our testing covered the most important aspects of the application - connection establishment and board updating when making moves.

Reflection On Process

Yordan's reflection:
The initial planning for the project was well thought off and provided us with plenty of time to complete every aspect of the project. However, due to partly procrastinating and lack of communication we started just "going with the flow" which resulted in not following the plan for the project. This disrupted our flow of work and led us to have to work overnight the last week of the project deadline. Furthermore, because of my lack of communication with my partner and my belief that I can do everything alone I made the project times harder for both my partner and myself.
Fortunately, because of this, I realised my faults and now know that I should take my project more seriously and rely on my group mates more. Also because of this problem, I ended up improving my communication with my partner and both of us sharing the rest of the workload and helping each other.
Because of the project, I believe that both I and my partner have learnt the importance of following a plan and helping each other and relying on each other.

Yunchen's reflection:
As Yordan said at the beginning of the project we did not have good communication. We did not discuss how to separate work and we haven't even analyzed the overall process of this project. Unfortunately, we discovered the problem too late, so we need to work on this project overnight. Since there was no good plan from the very beginning, many of our existing codes have to be modified so that the local game code can be connected to the server. This project taught me the importance of time and planning. With good planning, we can work more efficiently. We don't need to modify the written code. If we determine the projects that should be completed each week in the beginning, it will not cause us to rush to work in the last week.
Our ultimate goal is to create a Collecto game that can connect to the server and provide two-player battles. We can now successfully run the game locally and implement all the functions of the game.