Report

Implemented functionality

Function	
Client establishes a connection with the server	yes
Client is assigned a unique ID when joining the game	yes
Client displays up-to-date information about the game state	All information about state of the game, including who has the ball are displayed, but list of currently active players is displayed only for the player who has to choose to whom pass the ball, and only at this exact moment.
Client allows passing the ball to another player	yes
Server manages multiple client connections	yes
Server accepts connections during the game	yes
Server correctly handles clients leaving the game	Yes. When player leaves the ball is passed to another still playing player. If the last player leaves value of the variable storing which player has ball is

	changed to 1, because if there are no players playing, then the next player who joins will have a ball and get assigned first free ID, which will be 1.

Protocol

My project uses use TCP/IP protocol, which is default use by socket in package java.net(). At the beginning of the game player, who is the client joins the game. Server immediately sends him message with first free ID from the list of players stored on a server, and assigns it to him. Then every player constantly sends a query to server asking “who has the ball”, and the server answers by sending message to client. Player who has the ball gets form server list of other active players, all other players are getting only information who currently has the ball. After that player with the ball sends to server ID of the player he wants to pass the ball to, and right after finishing this cycle starts again – every client is asking ‘who has the ball” and the server responds as appropriate.

Client Threads

In the client process there are is one thread for every player (client) to allow ClientProgram to run. They are created when ClientProgram starts running, so player joins the game and terminated when players leave the game.

Server Threads

In the server process there is one thread for ServerProgram, which is needed to run the server, and one thread for each single new connection with joining player. Server thread is created when server starts running and terminated when it stops. The other threads are needed and created when new client want to connect with the server, and are terminated when client disconnects.

Project review

This project was definitely challenging for me. At the beginning nothing wanted to work, and unfortunately I wasted too much time because of logical mistakes I made in code, but after some time it started to finally working. At the point when I was able to allow many players to join the game and each of them was automatically assigned a unique, next free ID number I knew I was probably able to complete this assignment pretty well. It was also quite

demanding to allow the players to pass the ball to each other, because even if I knew quickly enough that server passes the ball to other players correctly I still had problems with sending this information to all players. Finally making this work properly was really satisfying for me, and since that moment everything else was much easier to achieve. It is hard for me to say anything about my project management, because my plan was changing many times throughout the project, depending on what I was able to do correctly and what not. I think it was maybe a little too chaotical, and if I had to do similar project once again I would spend at the beginning much more time on planning the whole architecture, instead of starting working immediately after a simple idea what I wanted to do, thinking I can come up with a plan as I progress. To conclude, I am satisfied with final product of my work, even if I know, that I could probably do it better, and less chaotically.