INFO0010: Socket Programming Project Part 1 - Report

Arthur LOUIS - s191230

November 2, 2021

1 Software architecture :

For this project, I used only two class to solve the problem :

- Subscriber.java : The main class that creates enables the connection between the server and the client (I use localhost when connected to Montefiore's machines but the IP can easely be modified). It's in this class that are the two most important methods : sendStream and readStream. These two take in charge the translation of inputs and outputs into protocol messages understandable by the Broker running on the server.
- MHG.java : In this class, the game grid is implemented, simply using a matrix of integers to compute the number of times a square is in a sensor's radius of confidence. There's a method (computeSensor) that computes the input sended by the server to correctly set the values in the game matrix. The class takes also the printing of matrix (through method printMatrix) in charge.

2 Message-oriented communication using a stream :

The recovery of a message was quite easy, the number of bytes you were expected to receive being in the message, loosing data was almost impossible. The protocol relies on two channels from server to client (position and victory) where client was listening to the Broker running on the server. The position channel was the one where the server told where the monster was (with confidence radius) and the victory channel was only used to check if the game was won by the player. In the other way, the client communicates with the server with position data and acknoledgement of the received data.

The data was broken down by the method sendStream from client to server and was collected by the method readStream when receiving from server to client.

3 Limits & Possible Improvements :

Although my implementation is quite compact, it could be a lot better. The error handling could be a lot better, I could implement a way to keep the program running even if there's unexpected packets received or sent.On the one hand, I also have the feeling that my implementation of the protocol could be stronger (extending the error handling) because it seems that if one byte is unexpected, the all program crashes. On the other hand, the implementation feels efficient being so compact.