The dummy way for Data communication between OpenBCI and Unity,

TCP/IP communication is usually handy but having trouble with multi-server/client running with processing. Since the I’m using a wifi version of OpenBCI, running another client/server becomes a challenging task. I do believe there is some way to solve this problem, but the case is: a new thread of OSC communication and TCP communication are all incompatible with the existing OpenBCI GUI.

So, the alternative method is streaming the value into a binary file(.txt), and read it from Unity. The trick is, we have to avoid exceptions using try{} catch{} on both side. The Unity could report exception for a failure in reading the file, the Java could report exception for a failure in writing into a being read file.

For java, we should construct two writers:

try{

fftPlot=createWriter("C:/Git/ml-agents/unity-environment/Assets/ML-Agents/Alphawave/AlphaRecord.txt");

} catch(Exception e){

fftPlot=createWriter("C:/Git/ml-agents/unity-environment/Assets/ML-Agents/Alphawave/AlphaRecordbackup.txt");

}

fftPlot.print(tempAlpha);

println(tempAlpha);

fftPlot.close();

Since in Java, the close would automatically flush the steam, so there is no need to add a line

fftPlot.flush();

For Unity, the idea is the same:

public float TextReadToFloat() {

float readvalue;

try {

string lines = System.IO.File.ReadAllText (@filepath);

//string[] sArray = lines.Split ('/');

//int lineNumber = System.Text.RegularExpressions.Regex.Matches (lines, "/").Count;

readvalue = float.Parse (sArray[lineNumber]);

return readvalue;

} catch (IOException) {

Debug.Log ("Read failed");

return oldalphaValue;

}

}