

Software Distribuït

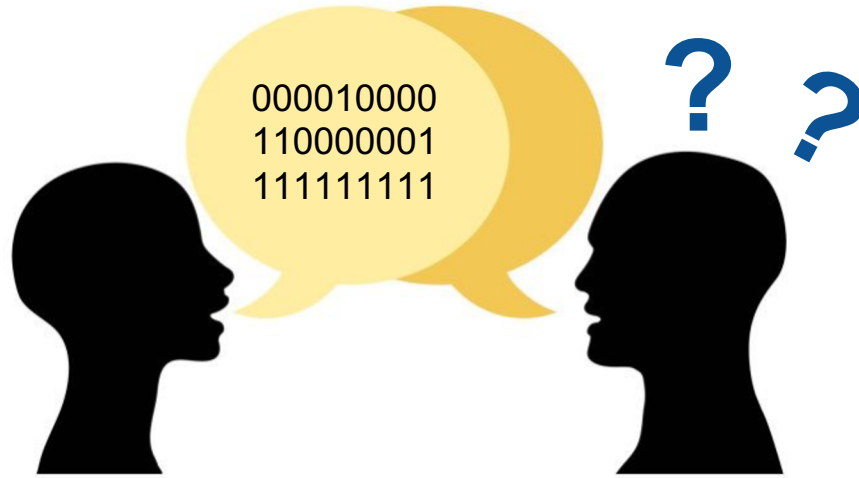
SESSIÓ 2: Pràctica 1 - Protocol

Calendari Pràctica 1

Data Sessió	Tasca	Data límit Peer Review	Puntuació del Review
17/02/2021	a) Creació de grups. b) Preparació del GitHub. c) Realització de la prac0	24/02	1
24/02/2021	Disseny del sistema distribuït. Implementació Protocol.	03/03	1
03/03/2021	Implementació Client.	10/03	1
10/03/2021	Implementació Servidor amb Multithread 1 Player.	17/03	1
17/03/2021	Implementació Servidor amb Multithread 2 Players.	24/03	1
24/03/2021	Sessió de Test creuat. Tasca Taller.	30/03	1
30/03/2021	Entrega codi, memòria i log execució al CV. Tasca Fitxer		

Que és un protocol?

Seguit de normes escrites que regeixen la comunicació entre dispositius



En el nostre cas ens permetrà que client i servidor puguin “entendre’s” per tal de jugar a “la batalla d’insults”.

Idees bàsiques

- El protocol ha de ser conegut pels diferents interlocutors i l'han de seguir estrictament
- Les característiques i el contingut de les trama han d'estar ben definides:
 - Longitud de dades a llegir (tot i que no sempre cal llegir tota la trama sencera)
 - Tipus de dades a llegir
 - Significat dels paràmetres
- A l'hora de definir els datagrames sempre triar la opció més senzilla (més fàcil d'entendre, més fàcil d'implementar).
 - Tots els missatges segueixin una mateixa estructura però amb diferenciadors clars
 - Tenir que els diferents tipus de trames siguin fàcilment distingibles

Exemples

Trames GPS (tot i que no és ben bé un protocol client/servidor):

\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47
\$GPGLL,4916.45,N,12311.12,W,225444,A,*1D

Altres exemples de protocols de correu que trobareu a la pàgina web de l'assignatura:

MAIL FROM:<Smith@Alpha.ARPA>
RCPT TO:<Brown@Beta.ARPA>

Com definir un protocol?

- Identificar les accions que duen a terme el client i servidor
- Definir una trama “inconfusible” per cada tipus d’acció (també els paràmetres associats i el seu tipus)
- Definir una seqüència o seqüències vàlides (tenir en compte els moments en que s’espera una resposta)
- Definir un datagrama d’error (tenint en compte els errors més comuns)

Se us dóna el protocol fet i el podeu consultar en el README de la tasca Practica 1 del githubClassroom

Avaluació

Avaluació de les pràctiques :

En cas de que la pràctica funcioni de forma correcta la **nota individual** de cada alumne es ponderarà de la següent forma:

- 80% Codi:
 - 50% Review del Codi
 - 50% Codi (Sense bugs, 1 client i servidor en els modes) + Tests + Memòria i diagrames.
- 20% Peer Testing sessió de test

Es recomana llegir atentament l'enunciat de pràctiques per tenir clars tots els requisits del projecte en la tasca del Github Classroom

Entorn de treball per a les pràctiques

- Per la sessió de test de la Pràctica 1, farem servir Hamachi by LogMeIn per crear xarxes virtuals VNP que pogueu fer les proves creuades.
(<https://www.vpn.net/>)
- Per executar el codi de les Pràctiques :
 - A través de IntelliJ IDE.
 - Indicar que ho faci com a projecte de MAVEN, si ho demana
 - Especificar la JDK : java-8-openjdk-amd64 o superior fins JDK15
 - Executar les comandes MAVEN en la consola del propi IntelliJ.
- L'estructura del fitxer MAVEN del projecte és vàlida des de JDK 8 fins JDK15

Format dels datagrames del Protocol

- Llegir detingudament el protocol i detectar possibles dubtes, errors o ambigüitats.
- Fer un esquema de les trames que fan avançar en el joc (diagrama de bloc, diagrama d'estats).
- Intentar incrementar els exemples que apareixen en el protocol.
- Pensar i anar ampliant les funcions de ComUtils per poder implementar el protocol (tenir en compte l'herència i encapsulament de les funcions de nivell a bits a nivell de trames).

Format dels datagrames del Protocol

Les trames estan codificades seguint la sintaxis dels missatges:

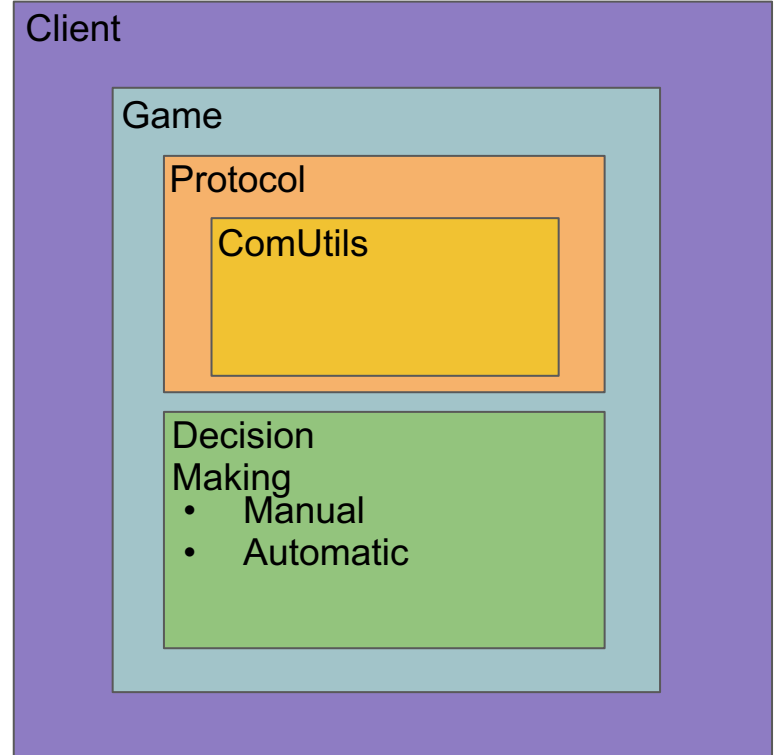
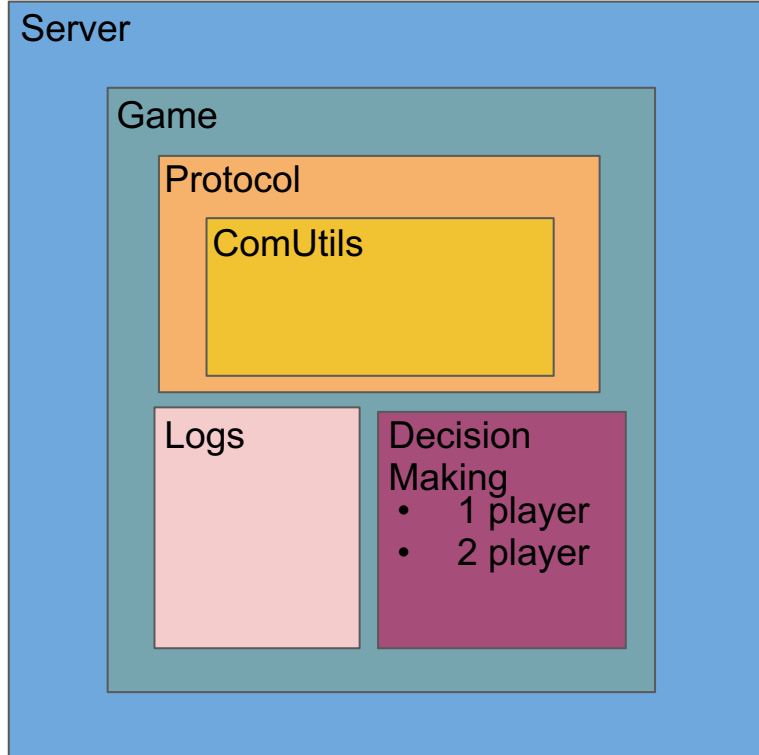
1 byte | string | 1 byte

| Opcode | Name | 0 |

En el protocol es posen en format “Humà”. Podeu transformar de trama màquina a humana per fer els logs:

HELLO JOE → 1'JOE'0 (5 bytes)

Estructura de Classes

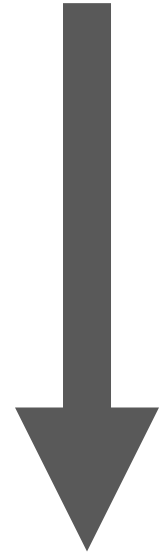


* Estructura básica propuesta

Estructura de Classes

- Ampliar funcionalitats ComUtils \Leftrightarrow TESTS
 - Lectura/Escriptura de dades
 - Conversió de dades
 - ...
- Crear classe a nivell de Trames \Leftrightarrow TESTS
 - Lectura/Escriptura de Trames
 - Control d'errors de les trames
 - ...
- Classe/classes auxiliars \Leftrightarrow TESTS
 - Control de paràmetres d'entrada
 - Logs
 - Menús i missatges per pantalla
 - ...

BAIX NIVELL



ALT NIVELL

Estructura de Classes

En properes classes parlarem sobre el disseny de client i servidor:

- Classe amb la lògica del Joc \Leftrightarrow TESTS
 - Màquina d'estats
 - Control d'errors
- Classes principals de Client i Servidor \Leftrightarrow TESTS Creuats
 - Creació de sockets
 - Threads

Per a la pròxima setmana ...

- Llegir detingudament el protocol i detectar possibles dubtes, errors o ambigüitats.
- Ampliar les funcions de ComUtils i crear noves classes per poder implementar el protocol (tenir en compte l'herència i encapsulament de les funcions de nivell a bits a nivell de trames).
 - Podeu executar el codi emulant els sockets sobre fitxers.
- Realitzar els tests corresponents (conjuntament amb les funcions).
- Fer un esquema de les trames que fan avançar en el joc (diagrama de bloc, diagrama d'estats).