

Software Distribuït

SESSIÓ 1: Bytes i Streams

Presentació Professorat



Núria Pujol Vilanova

Contacte:

- npujolvi@ub.edu
- Campus Virtual
- GitHub (ús moderat, ho reben la resta de professors)

Hores de consulta presencials:

- Hores a concertar a partir de les 15h.

Presentació Sessions de Pràctiques

Tota la informació es penjarà a la pàgina de l'assignatura:

<http://ub-gei-sd.github.io/>

L'ús del Campus Virtual queda limitat a dubtes i entregues de pràctiques.

Pràctiques que es realitzaran durant el quadrimestre:

Projecte 1: Sockets Client/Servidor (Java)

Projecte 2: Desenvolupament Web/API Rest (Python/Django)

Calendari Pràctica 1

14/02/2018	<ul style="list-style-type: none">- Creació de grups.- Preparació GitHub.- Extensió de la classe ComUtils.java (seguint el guió de pràctiques).
21/02/2018	Definició i implementació del protocol (RFC)
28/02/2018	<ul style="list-style-type: none">- Fi definició protocol (RFC)- Implementació Client.
07/03/2018	Implementació Servidor multithread
14/03/2018	Implementació Servidor amb Selector
21/03/2018	Sessió de Test.
28/03/2018	Entrega codi, memòria i .log d'execució al Campus Virtual - SETMANA SANTA

Grups de Pràctiques

Les pràctiques es realitzaran en grups de **2 persones**.

Tant bon punt tingueu clar el vostre company/a de pràctiques comuniquem-m'ho via correu o campus virtual indicant també els vostres usuaris de GitHub.

Comproveu que se us dóna accés a:

- 1 repositori per parella de pràctiques.
- 1 repositori de grup per definir el protocol del Projecte 1 (SESSIÓ 2 de pràctiques).

Qualsevol problema que tingueu contacteu el més aviat possible amb el professor/a.

Sessió 1

L'enunciat a seguir en aquesta sessió es troba en la pàgina de l'assignatura.

Se us demana fer una sèrie de modificacions per estendre la llibreria [ComUtils.java](#), que s'utilitzarà al llarg del Projecte 1.

En aquesta sessió emularem el **DataInputStream** i **DataOutputStream** que utilitzarem més endavant en la comunicació entre sockets amb els que s'utilitzen per escriure i llegir per fitxer. D'aquesta manera podrem visualitzar i quantificar els bytes que s'escriuen/llegeixen.

Sessió 1

Les funcions principals que componen **ComUtils.java** són:

- **String: read_string()**
- **int: read_int32()**
- **String: read_string_variable(int head_size)**

- **write_string(String s)**
- **write_int32(int)**
- **write_string_variable(int head_size, String s)**

Les funcions estan explicades amb més detall en l'enunciat corresponent.

Sessió 1

Els principals objectius d'aquesta sessió:

- Entendre i saber utilitzar les funcions que formen part de **ComUtils.java**
- Saber llegir i escriure correctament diferents tipus de dades
- Entendre la diferencia entre les funcions i en quin cas utilitzar cadascuna d'elles.
- Ser capaç de modificar aquestes funcions de base per aconseguir escriure i llegir un missatge amb unes característiques concretes.

Sessió 1

EXERCICI EXTRA:

Realitzar l'escriptura/lectura d'un fitxer amb el següent format:

String[20] <El vostre nom i cognom>

int32 <Edat>

String[] <Comentari>

- Que observeu? Quants bytes s'han escrit en el fitxer? Tots són llegibles?
- En quin ordre s'han de llegir? Quines funcions heu fet servir?

Sessió 1

Al final d'aquesta sessió hauríeu de ser capaços de respondre a les següents preguntes:

- Quina diferencia hi ha entre **read_string** i **read_string_variable**? Són intercanviables?
- Si volem que un int32 sigui llegible en un fitxer, que haurem de fer?
- Quines creus que podrien ser altres funcions útils?

Pràctica 1: Kuhn Poker

El [Kuhn Poker](#) és un joc clàssic de teoria de jocs. Es tracta d'una versió simplificada del pòquer convencional.

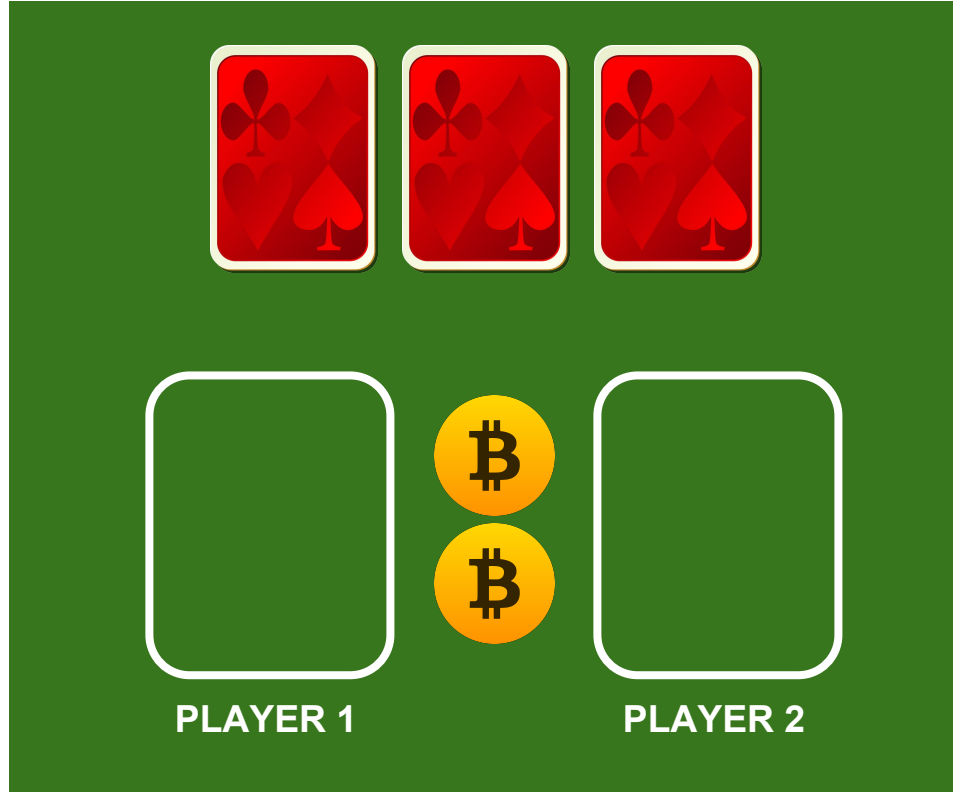
Disposem d'una baralla de 3 úniques cartes: **JQK** (també existeix la versió AQK).



Guanya la ronda el jugador que té **la carta més alta** (o no abandona).

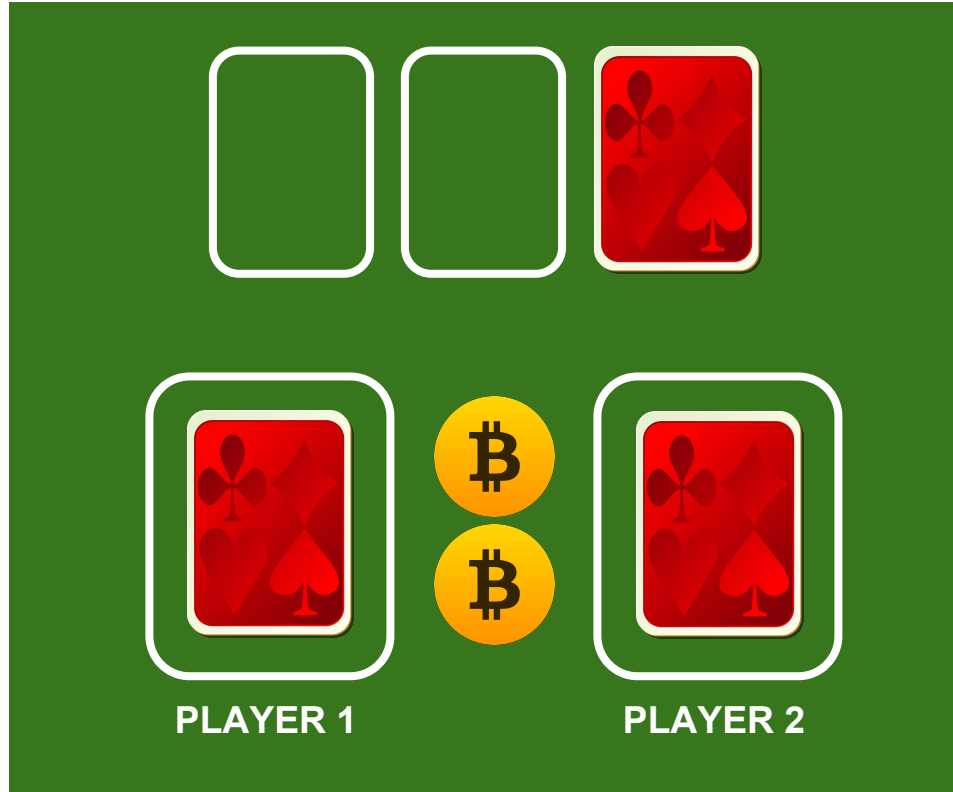
El guanyador del joc és el que acumula més tokens al llarg de les diferents rondes.

Pràctica 1: Kuhn Poker



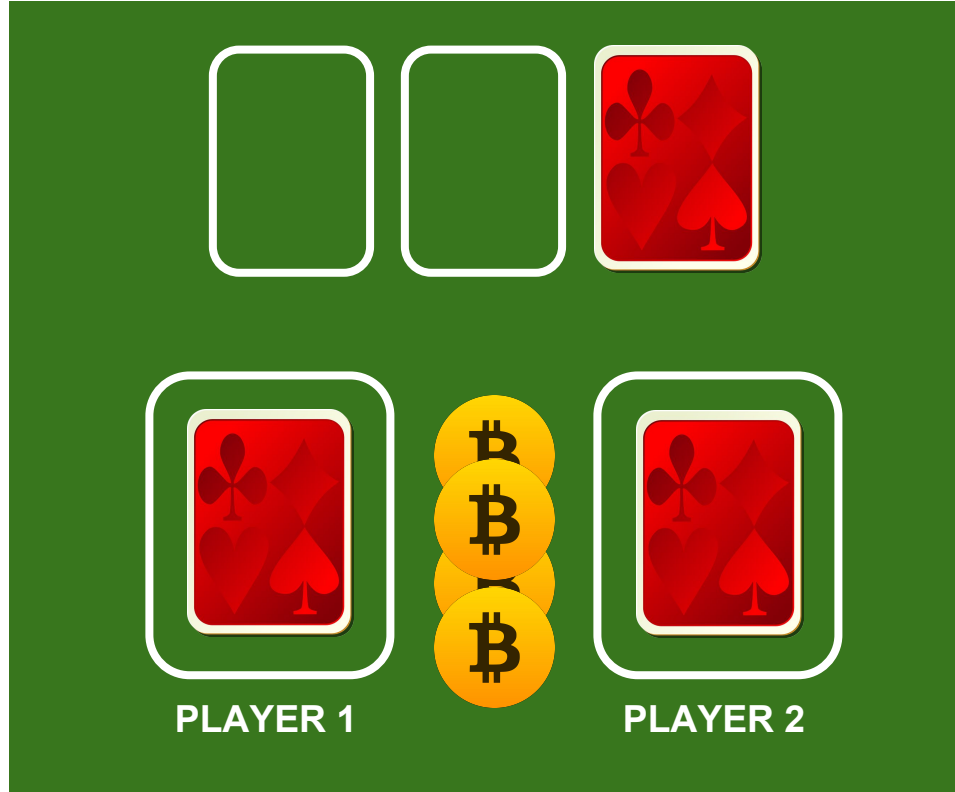
- Per obrir el joc i abans de fer cap moviment cada jugador farà una aposta inicial igual (**ANTES 1**).

Pràctica 1: Kuhn Poker



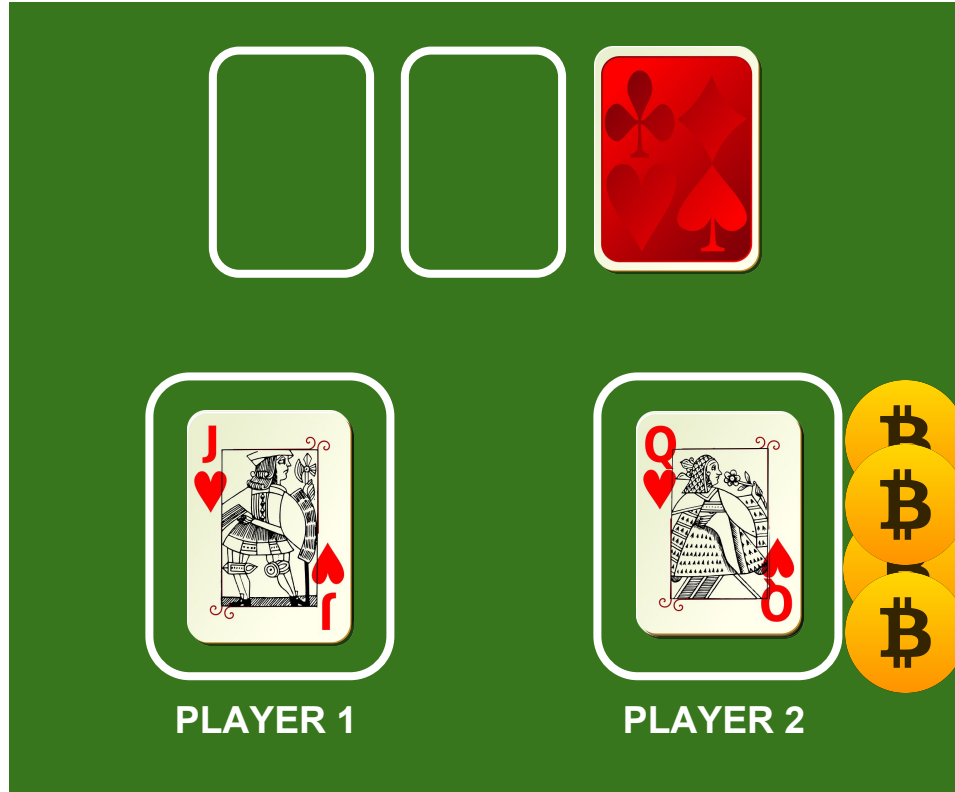
- El **jugador 1** (o el que tingui el torn) un cop s'han repartir les cartes pot decidir mirar (**CHECK**) o pujar l'aposta (**BET**).
- El **jugador 2**, espera l'acció del jugador 1 abans de fer res.

Pràctica 1: Kuhn Poker



- El **jugador 1** decideix pujar l'aposta.
- El **jugador 2** en aquest punt pot abandonar (**FOLD**) o jugar (**CALL**).
- El **jugador 2**, finalment, decideix jugar i igualar l'aposta.

Pràctica 1: Kuhn Poker



- Al destapar les cartes comprovem que el **jugador 2** té la carta més alta i per tant, és el guanyador.
- El **jugador 2** s'emporta el total de diners apostats:
 - **JUGADOR 1** (-2)
 - **JUGADOR 2** (+2)

Pràctica 1: Kuhn Poker

Entregables de la pràctica per la seva evaluació:

- **Client**
 - Mode manual
 - Mode automàtic
- **Servidor**
 - Multithread
 - Selector (no és obligatori però necessari per optar al 10 en aquesta part).
- **Logs d'exemple**
- **Memòria de pràctiques de 5-6 pàgines (PART IMPORTANT DE LA NOTA)**
 - Descripció del treball realitzar (estructura de classes, esquemes, tests, etc.).
 - Resum de la sessió de Test (resultats i comentaris de les proves creuades).

Projecte 1: Kuhn Poker

També es tindrà en compte:

- L'ús de GitHub.
- L'assistència **OBLIGATÒRIA** a la sessió de Test de com a mínim un membre del grup.
- **Correcta implementació del protocol** (pot suposar suspendre la pràctica).
- Participació en el disseny del protocol.
- Robustesa i control d'errors.
- Estructuració del codi, etc.

Pròxima Sessió

En la pròxima sessió de pràctiques tota la classe haurà de definir un protocol a seguir per tots els clients i servidors per tal “d’entendre’s” entre les diferents parelles de pràctiques.

Es recomana fer abans la pròxima sessió:

- Acabar de llegir la documentació referent al Kuhn Poker i entendre el funcionament complet del joc
- Diagrama d’estats corresponent al Kuhn Poker.
- Acabar els apartats de la sessió 1.
- Preguntar els dubtes que puguin sorgir