

Software Distribuït

Pràctica 0: Bytes i Streams

Presentació Professorat



Núria Pujol Vilanova

Contacte:

- npujolvi@ub.edu
- Campus Virtual
- GitHub (ús moderat, ho reben la resta de professors)
- Telegram: <https://t.me/nuriapujol>
- **Altres propostes? (Slack, etc.)**

Hores de consulta: 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 (Flask i Vue.js)

Calendari Pràctica 1

12/02/2020	<ul style="list-style-type: none">- Creació de grups.- Preparació GitHub.- Pràctica sobre la llibreria ComUtils
19/02/2020	<ul style="list-style-type: none">- Definició i implementació del protocol
26/02/2020	<ul style="list-style-type: none">- Disseny Client/Servidor i Tests
04/03/2020	<ul style="list-style-type: none">- Implementació Client/Servidor
11/03/2020	<ul style="list-style-type: none">- Implementació Client/Servidor i primeres proves entre diferents grups
18/03/2020	Sessió de Test.
25/03/2020	Entrega al Campus Virtual

Grups de Pràctiques

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

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

L'activitat en el GitHub de cada membre de la parella es podrà tenir en compte a l'hora d'avaluar individualment a cada membre.

Qualsevol problema que tingueu amb el funcionament de la parella de pràctiques contacteu el més aviat possible amb el professor/a.

Pràctica 0

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 diferència 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.
- Comprendre el test d'exemple i realitzar-ne de nous.
- Saber executar el codi i els tests desde l'editor i/o consola.

IMPRESINDIBLE PER PODER FER LA PRÀCTICA 1

Pràctica 0

Podeu trobar l'enunciat de la Pràctica 0 i el codi corresponent a :
<https://ub-gei-sd.github.io/Practica0/>

El podeu obrir en qualsevol editor de text però està preparat per ser obert com a un Projecte de IntelliJ amb versió JDK 8.

Pràctica 0

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 les classes de la llibreria **utils** (**ComUtils.java** i **ComUtilsService.java**) , que s'utilitzarà i ampliarà 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.

Pràctica 0

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.

Pràctica 0

EXERCICIS EXTRES PROPOSATS:

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?

Pràctica 0

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

- Quina diferència 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 poden ser altres funcions útils?

Pràctica 0

```
public class ComUtilsTest {  
    @Test  
    public void example_test() {  
        File file = new File("test");  
        try {  
            file.createNewFile();  
            ComUtils comUtils = new ComUtils(new FileInputStream(file),  
            new FileOutputStream(file));  
  
            comUtils.write_int32(2);  
            int readedInt = comUtils.read_int32();  
            assertEquals(2, readedInt);  
  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Pràctica 0

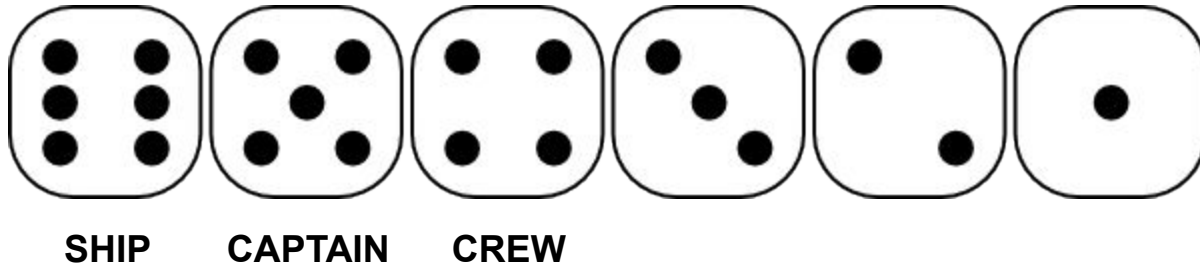
Serieu capaços de realitzar les funcions **write_char** i **read_char** i realitzar-ne els tests corresponents?

Aquí un link que us pot ser útil: <https://www.guru99.com/junit-assert.html>

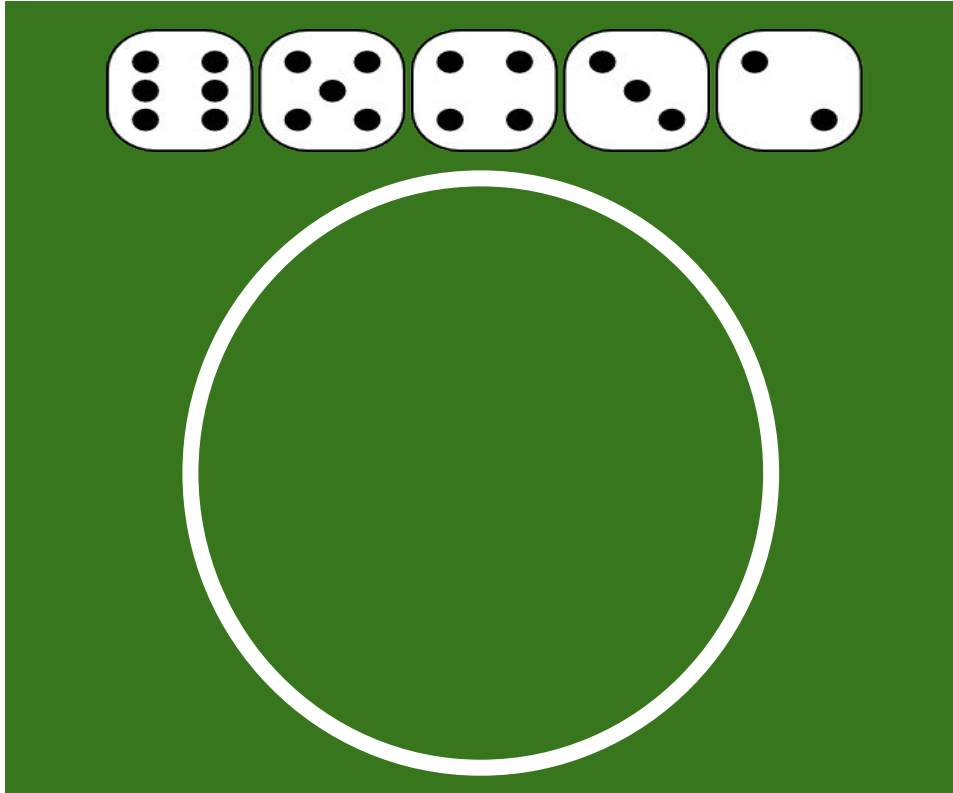
Pràctica 1: Ship, Captain and Crew

“Ship, Captain and Crew” o “Batalla naval” és un joc de daus popular entre les tavernes de mariners. Es juga per torns amb 5 daus de 6 cares.

L'objectiu principal del joc és obtenir la seqüència de daus 6 (“Ship”), 5 (“Captain”) i 4 (“Crew”) i amb els 2 daus restants obtenir la màxima puntuació.

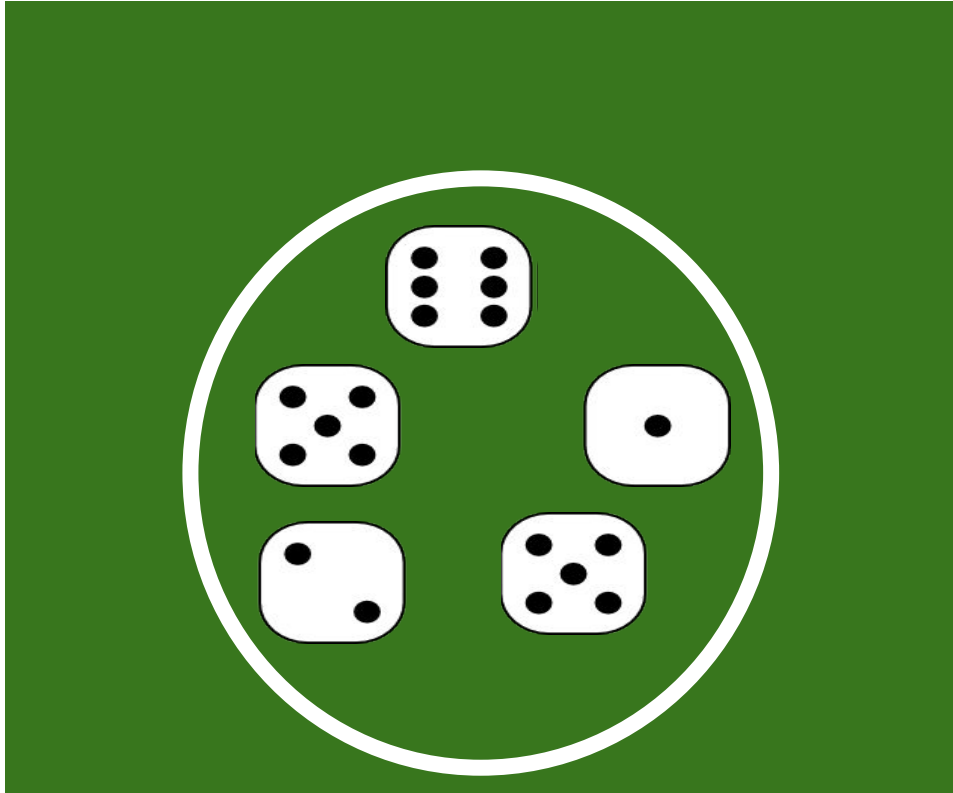


Pràctica 1: Ship, Captain and Crew



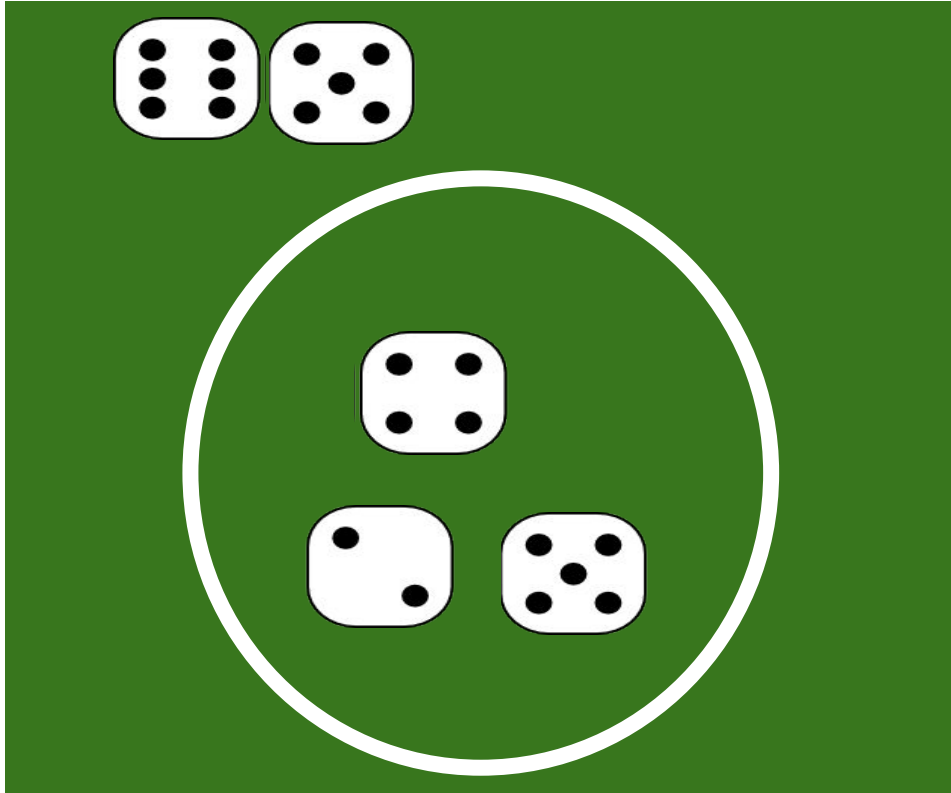
- El jugador que tingui el torn comença tirant els 5 daus.
- És el torn del JUGADOR 1

Pràctica 1: Ship, Captain and Crew



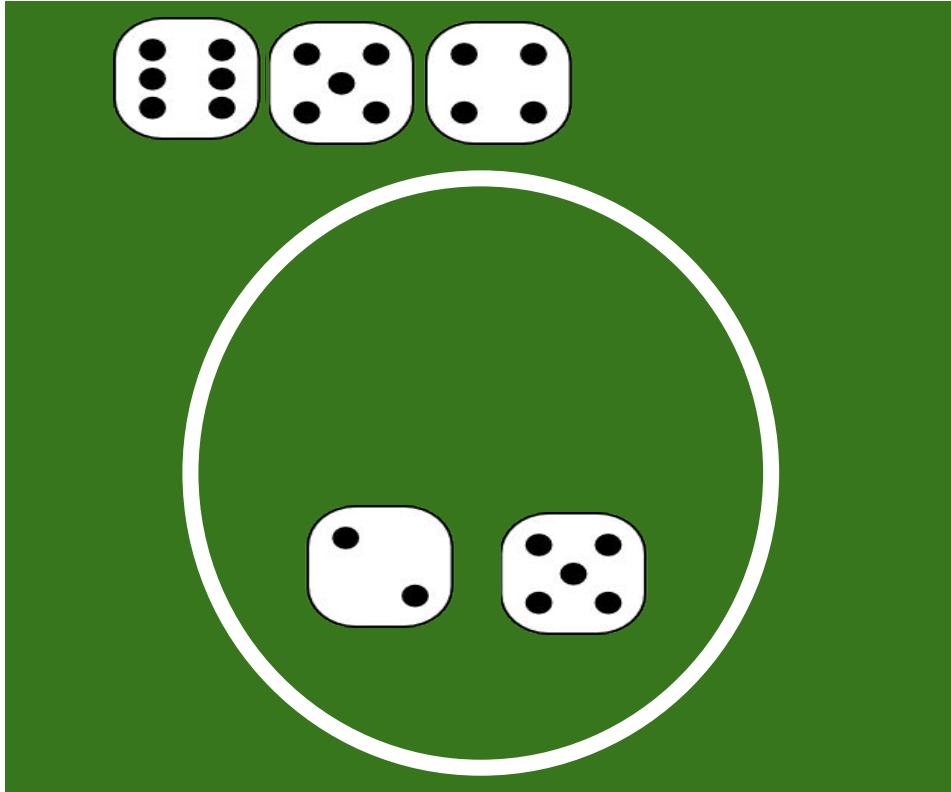
- ROLL 1
- Obté un 6 i un 5 i els aparta i torna a tirar la resta

Pràctica 1: Ship, Captain and Crew



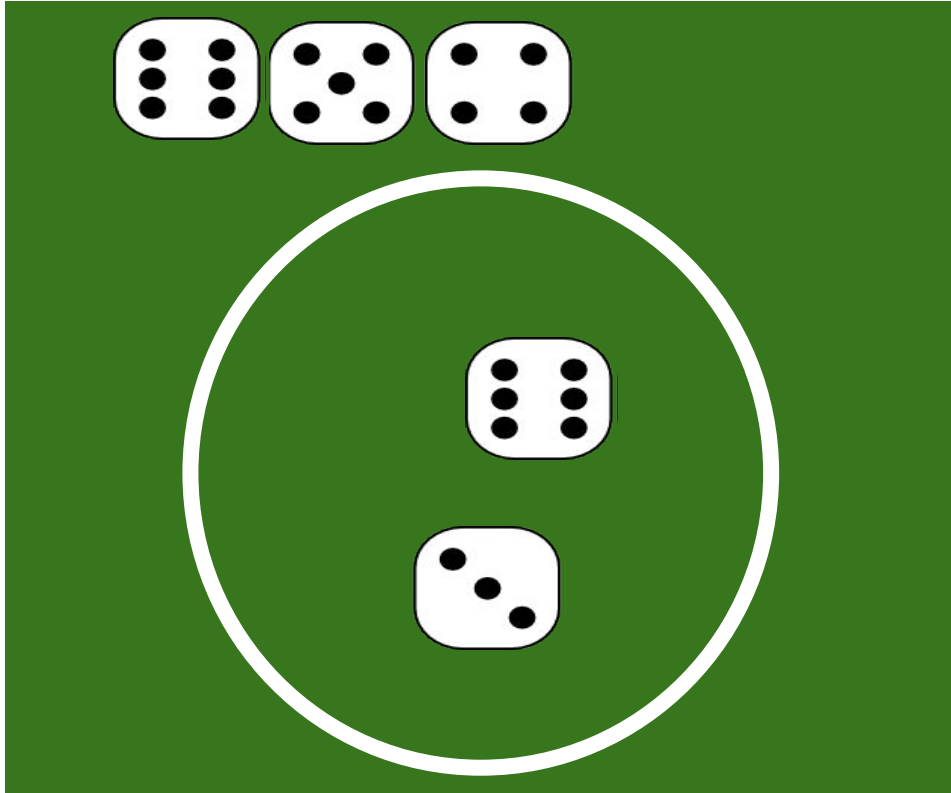
- ROLL 2
- Obté un 6 i un 5 i els aparta i torna a tirar la resta

Pràctica 1: Ship, Captain and Crew



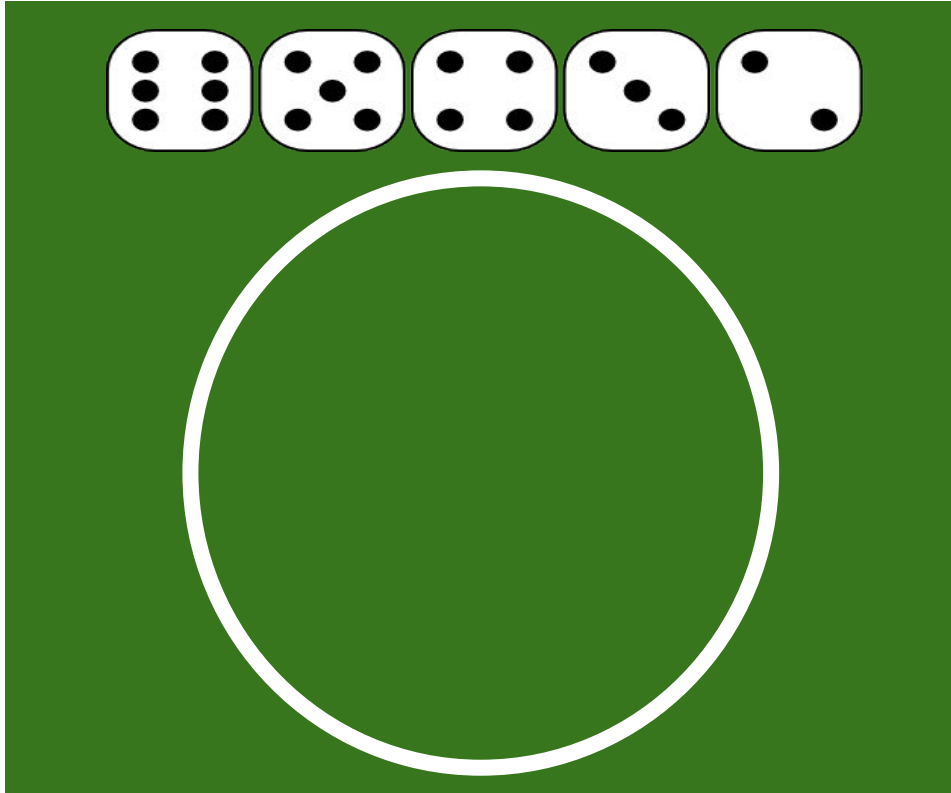
- Ja té el Ship, Captain and Crew i amb la resta de daus obtindria 7 PUNTS (5+2).
- Com que el jugador té un màxim de 3 intents arrisca per obtenir una millor puntuació i tira de nou.

Pràctica 1: Ship, Captain and Crew



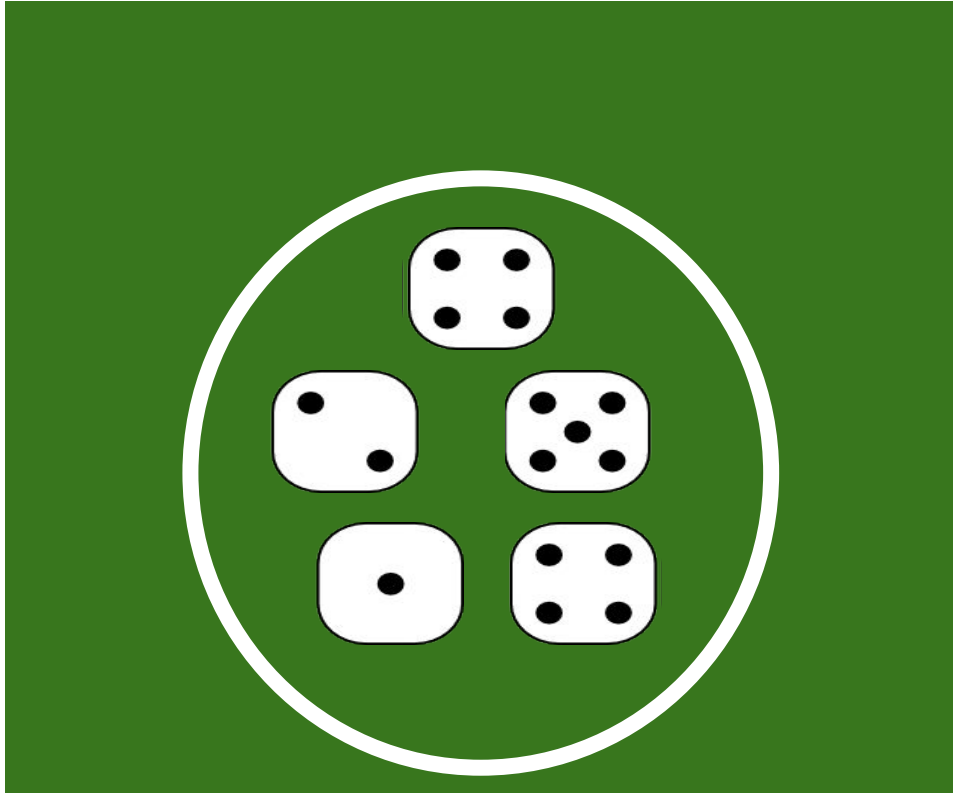
- ROLL 3
- Amb aquesta nova tirada el JUGADOR 1 obté 9 PUNTS.
- És ara el torn del JUGADOR 2.

Pràctica 1: Ship, Captain and Crew



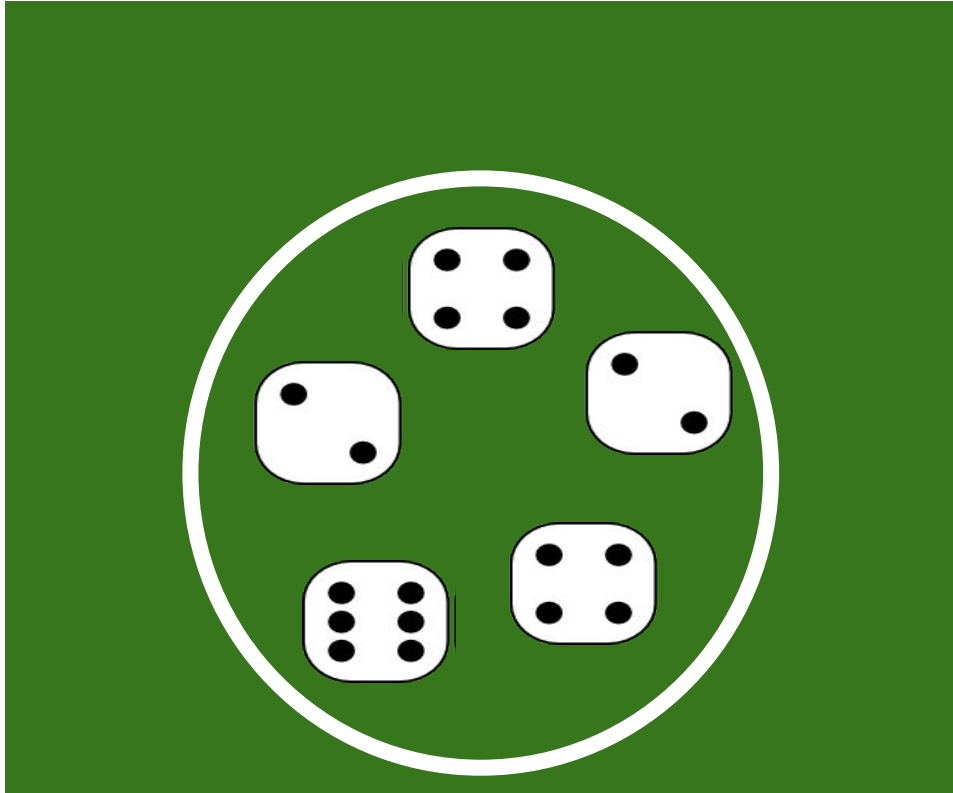
- Torn del JUGADOR 2

Pràctica 1: Ship, Captain and Crew



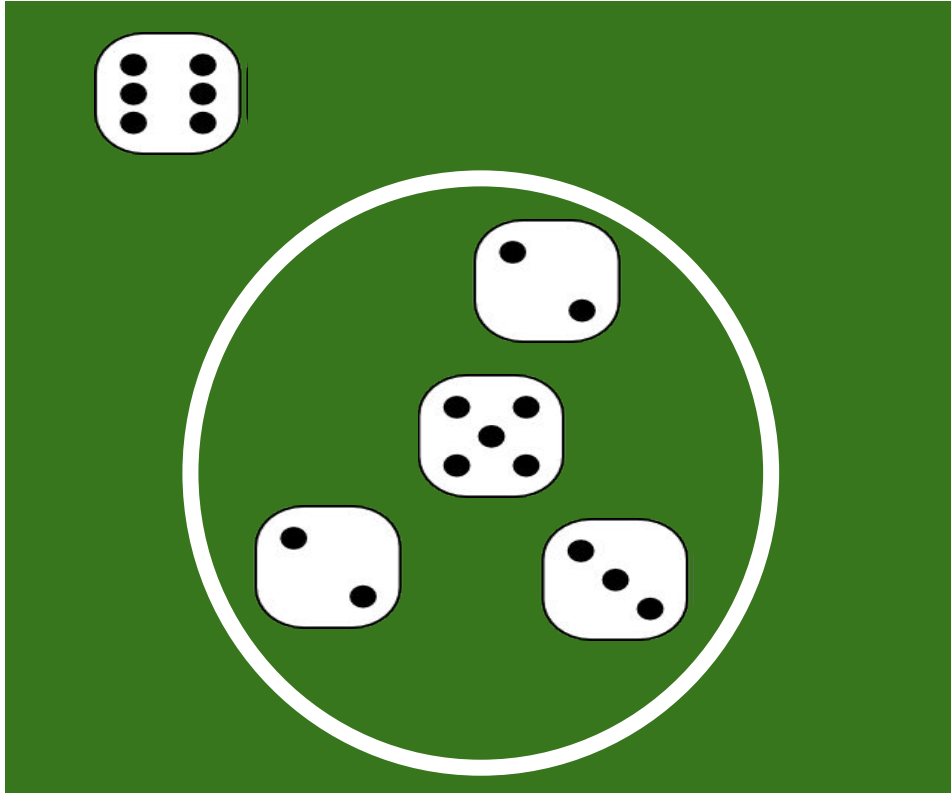
- ROLL 1
- No ha obtingut cap 6, necessari per poder agafar el 5, a la vegada necessari per poder agafar el 4.
- Torna a tirar tots els daus.

Pràctica 1: Ship, Captain and Crew



- ROLL 2
- Aquest cop pot apartar el 6 i tira la resta de daus.

Pràctica 1: Ship, Captain and Crew



- ROLL 3
- Aquest cop pot apartar el 5, però li faltaria el 4 per poder puntuar.
- El JUGADOR 2 obté 0 PUNTS.
- El guanyador és el JUGADOR 1.

Pràctica 1: Ship, Captain and Crew

Entregables de la pràctica per la seva avaluació:

- **Client**
- **Servidor (Multithread)**

- **Logs d'exemple**
- **Tests**
- **Javadocs**
- **Esquemes** (diagrama de classes, etc.)
- **Resum sessió de test** (resultats i comentaris de les proves creuades)

Projecte 1: Ship, Captain and Crew

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).
- Robustesa i control d'errors.
- Estructuració del codi, etc.
- Qualitat i adequació dels tests.