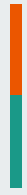
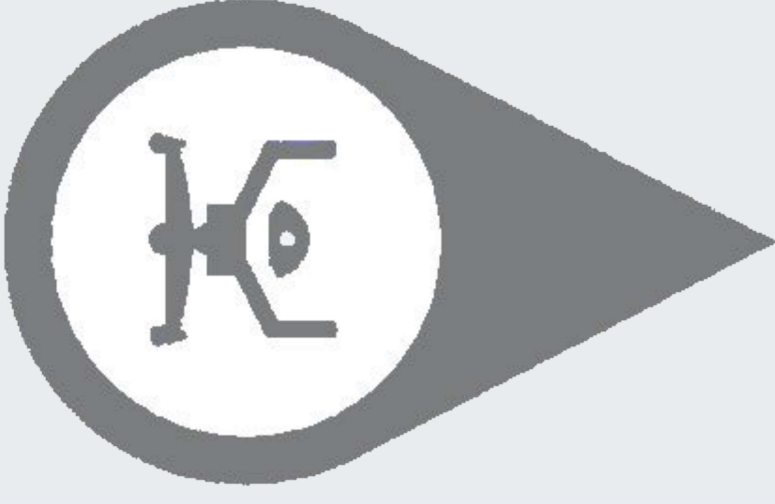


Soluzioni (step: 1-2):
<https://bit.ly/2FjFF30>



Esercitazione - DronET

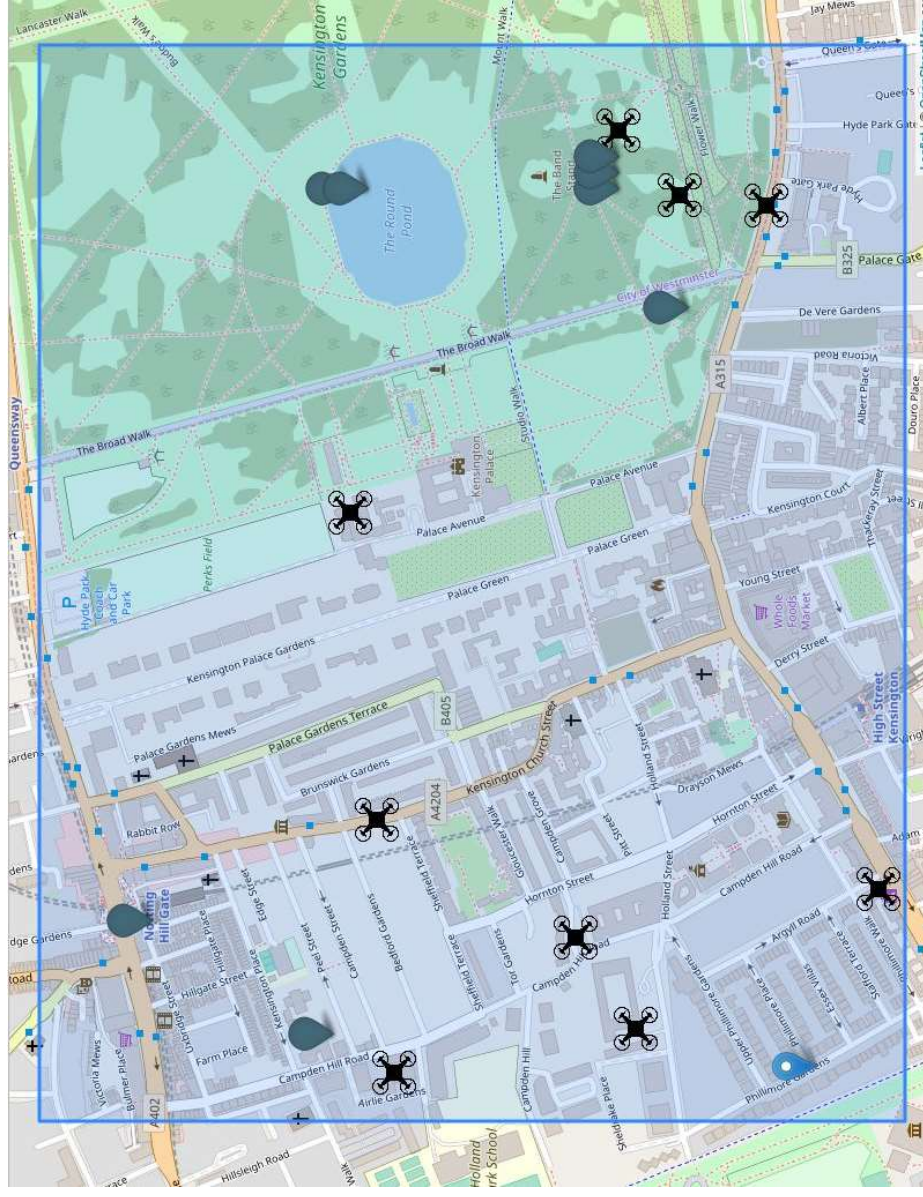
Reti di Elaboratori - 18/19
Corso di Laurea in Informatica
Università degli Studi di Roma “La Sapienza”



Mauro Piva - Andrea Coletta
{coletta | piva}@di.uniroma1.it

DronET

- Mappa con punti di interesse
- Una serie di droni che si muovono
- Un drone per ogni utente
- Ogni drone può muoversi in 8 direzioni:
 - Nord - Sud
 - Est - Ovest
 - NordEst - NordOvest
 - SudEst - SudOvest
- A POI (Point-Of-Interest)

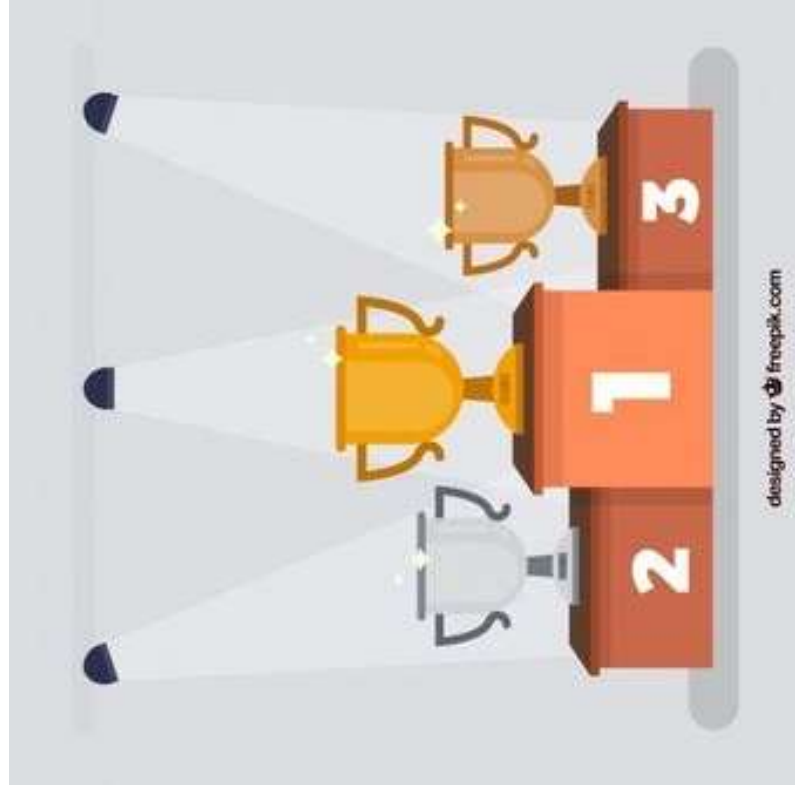


Obiettivo finale

Creare un drone e visitare più punti possibile sulla mappa!



Non far schiantare il drone

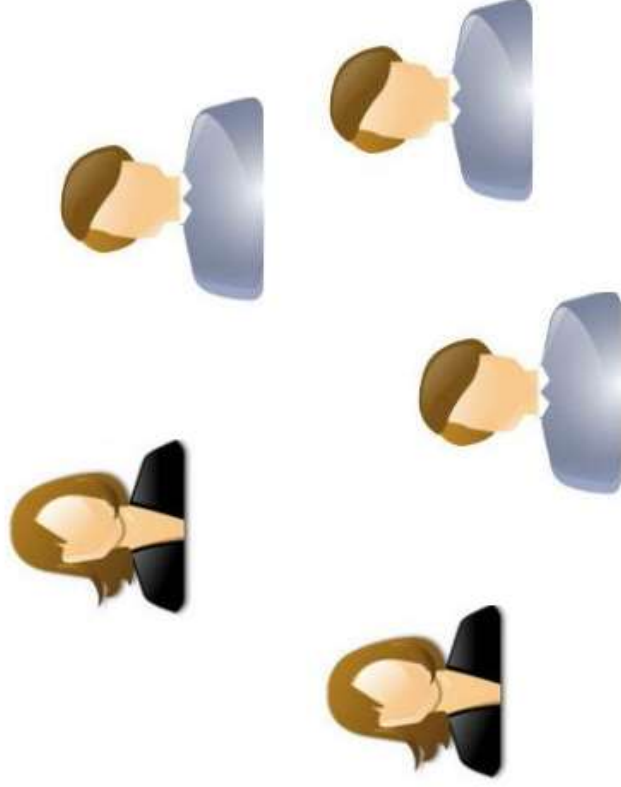


DroNET - Come funziona



Server

35.228.1.72



Clients

DroNET - Server



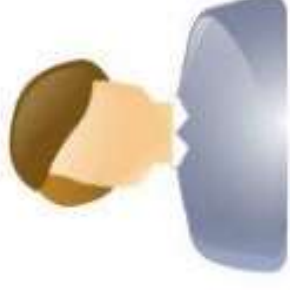
Server

35.228.1.72



Accetterà solamente
richieste HTTP BEN
FORMATTE!

DroNET - Client




- Il client dovrà permettere di:
 - Scaricare la mappa e i dati di gioco
 - Elaborare la scoreboard
 - Creare un nuovo drone
 - Pilotare/gestire il drone

DroNET - Rules

- 5 livelli
- Python (preferibilmente) / Java
- Non è consentito:
 - Utilizzare librerie come `urllib2`, `requests`, `HTTPURLConnection`, `HttpComponents`, etc...
- E' consentito invece l'utilizzo di:
 - `socket` (python - Java), `json`, `sys`

DroNET - Details



- IP Server: 35.228.1.72
- Mappa online con i droni: <http://35.228.1.72/dronet-ms-core/v1/getmap>
 - Se aggiungiamo il parametro “droneUnique=identificativo_drone” alla richiesta il drone che ha quel identificativo verrà messo in evidenza.

Trick:

- <http://35.228.1.72/swagger-ui.html#/>

DroNET - Details



- Problemi di linea?
- \$ git clone <https://github.com/mauropv/dronet-ms>
- cd dronet-ms
- mvn clean install
- sudo java -jar dronet-ms-core/target/dronet-ms-core-1.0.0-SNAPSHOT.jar
- Nuovo endpoint! (localhost)

DroNET - client details

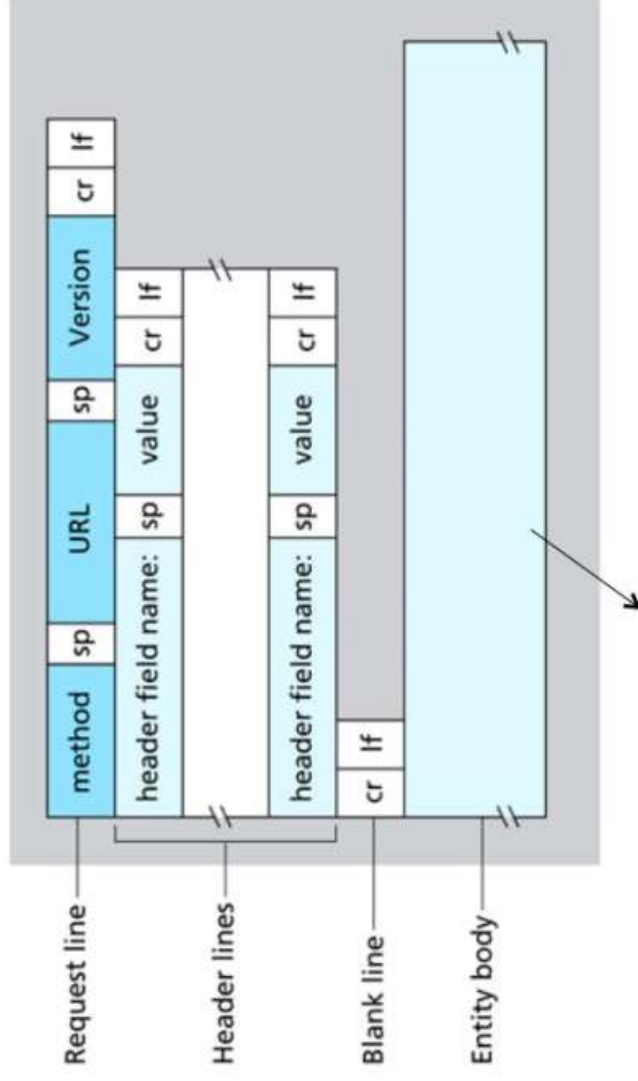
Formato generale dei messaggi di richiesta HTTP:

Come fare?

Scrivere le chiamate HTTP.

In particolare:

- GET
- POST
- PUT



Campo vuoto per il GET, utilizzato per il POST

Remember:

- <http://35.228.1.72/dronet-ms-core/v1/getmap>

DroNET - Step 1

- Scaricare dati di gioco (20min)

URL: GET /dronet-ms-core/v1/mapStatus

Descrizione: Restituisce un oggetto JSON contenente una serie di informazioni sulla mappa

Parametri: Header: user-agent deve essere chat-client

Body: ?

Response: un oggetto JSON contenente le info sul gioco/mappa

Pssw step 1: #reti_\$MZ_!K39Y

Obiettivo: Stampare a schermo il numero di droni presenti, i loro nomi, il numero di POI e i dettagli della mappa (dimensioni e posizione della “base station”)

Remember:

- <http://35.228.1.72/dronet-ms-core/v1/getmap?droneUnique=uniqueID>

DroNET - Step 2

- Creare un nuovo drone (15min)

URL: PUT /dronet-ms-core/v1/createDrone

Descrizione: Permette di creare un nuovo drone

Parametri: Header: user-agent deve essere **drone-client**

Body: droneName (scegliete voi il nome del drone)

Response: un oggetto JSON contenente uniqueId e secret. Dovrete riutilizzare questi parametri nelle successive comunicazioni.

Obiettivo: Creare un nuovo drone con il vostro nome e trovarlo sulla mappa