Super Socket

Eccoci a lavorare ad un altro bel progetto in Java con i socket! :)

Potete scegliere uno dei progetti indicati sotto o pensarne uno voi. In ogni caso ogni progetto avrà questa parte in comune:

Condizioni generali:

1. Ogni progetto ha un **capo progetto**. Il capo progetto e il team, prima di cominciare, si trovano per definire i requisiti/funzionalità di progetto da cui partire e le annotano nel [documento di progetto](https://docs.google.com/spreadsheets/d/1KITEbMsQdYt-x-Mr4QcuePk3Aq5wsBggsW-dNFEV5hs/edit?usp=sharing). Il capo progetto è responsabile del progetto e del fatto che il documento sia compilato. Può comunque delegare ad un’altra persona le attività che preferisce.
2. Ogni progetto utilizzerà un **protocollo** da descrivere nel [documento di progetto](https://docs.google.com/spreadsheets/d/1KITEbMsQdYt-x-Mr4QcuePk3Aq5wsBggsW-dNFEV5hs/edit?usp=sharing). La descrizione del protocollo è simile a quella del [gelato](https://docs.google.com/presentation/d/1Df-BB1LD_9_uNKZQpncJ7ri5jO8KSJjdq-dLrRcEoUc/edit?usp=sharing)
3. Ogni progetto viene realizzato da un **gruppo** di persone e ogni persona annota man mano le proprie attività nel [documento di progetto](https://docs.google.com/spreadsheets/d/1KITEbMsQdYt-x-Mr4QcuePk3Aq5wsBggsW-dNFEV5hs/edit?usp=sharing). Il capo progetto controlla ogni giorno (in cui si lavora) che il documento sia aggiornato.
4. Ogni gruppo farà un **progetto diverso** a scelta tra quelli proposti o inventando un proprio progetto (in tal caso i docenti potranno avallare o meno il progetto proposto).
5. Il progetto deve essere pubblicato su **github**, il capo progetto sceglierà dove salvarlo e verificherà le autorizzazioni
6. Al termine del progetto ogni gruppo **presenterà** il proprio lavoro. Su questo maggiori dettagli in arrivo. :)

Per la valutazione consideriamo che siano verificati questi punti e poi le funzionalità del progetto. La presentazione, da realizzare a conclusione del progetto, avrà qualche altra istruzione e un altro voto.. prossimamente!

## Valutazione

* Codice su github (1)
* Documento con i requisiti (1)
* Documento con il protocollo (1)
* Il gioco utilizza i socket multi o datagram (1)
* Il gioco funziona come previsto (4)
* Il gioco ha qualcosa di originale rispetto a quanto descritto (2)

I membri del team devono essere in grado di capire come funziona

# 

# 

# Elenco progetti

Di fianco ad ogni progetto il numero di persone previste. Da verificare comunque se le attività identificate sono sufficienti per il numero di persone.

## Tris (2/3)

Il gioco del tris online! Più giocatori si sfidano tra loro connettendosi al server. Quando vogliono giocare il server individua due giocatori connessi disponibili e li fa giocare. A turno muoveranno le pedine nella loro scacchiera. Nel server si potranno vedere gli utenti connessi e lo stato delle partite. Server e client sono grafici. Il client mostra anche il nome dell’avversario.

## Sasso/Carta/Forbice (2/3)

Sasso, carta e forbice è un gioco ben conosciuto. Due persone scelgono un simbolo e poi giocano. Comunicano al server il simbolo e il server decreta la vittoria di uno dei due partecipanti. Il client mostra il numero di vittorie, con chi si sta giocando e altre informazioni a scelta.

## Quizzone (2/3)

Come un quiz televisivo. Più persone possono giocare online al quizzone in cui si devono indovinare le domande nel più breve tempo possibile. Quando il gioco comincia il server trasmette ai client una domanda a risposta multipla e poi annota i tempi di risposta. In base ai tempi di risposta e alla correttezza delle domande calcola un punteggio e dopo dieci domande decreta un vincitore. Server e client sono grafici.

## Roulette (2/3)

Si gioca alla roulette! Vari vari utenti possono giocare alla roulette e segnalano al croupier la loro giocata. Il croupier ad un certo punto avvisa che tutti non possono più puntare, estrae un numero a caso e lo comunica ai giocatori, indicandogli anche la vincita eventuale. Poi permette loro di giocare nuovamente. Sia server che client sono grafici.

## Sette e mezzo (2/3)

Vari giocatori (client) si giocano una partita di sette e mezzo dialogando con un mazziere (server) che distribuisce le carte ai giocatori. Quando un certo numero di giocatori sono pronti il mazziere decide di cominciare la partita e il gioco ha inizio. Il gioco prosegue con le regole del [sette e mezzo](http://www.cavallore.it/index.php/giochi/33-sette-e-mezzo). Sia server che client sono grafici. Si preveda la possibilità che il client e il server possano scegliere il tipo di illustrazione delle carte in base ad un menu di scelta rispetto le diverse [illustrazioni proposte](https://it.wikipedia.org/wiki/Carte_da_gioco_italiane). Esempi di [carte da utilizzare](https://github.com/hayeah/playing-cards-assets/tree/master/png).

## Bla bla (2/3)

Una chat in cui ogni utente può scegliere un nickname e può inviare messaggi a tutti i partecipanti. I messaggi possono essere colorati, si può fare suonare la chat degli altri e ci saranno altre opzioni a scelta. Il server gestisce la chat e comunica con i client. Sia il server che i client sono grafici. La chat potrebbe anche avere dei gruppi privati o delle stanze di discussione.

## Tappo (2/3)

Tappo è un gioco di carte molto conosciuto anche se sotto un nome un po’ diverso! Ogni giocatore ha un determinato numero di carte fornite dal mazziere e tutti contemporaneamente devono scambiarsi una carta. Il primo che fa quattro carte uguali deve gridare m...tappo! [Su wikipedia le istruzioni](https://it.wikipedia.org/wiki/Tappo_(gioco)). Si preveda la possibilità che il client e il server possano scegliere il tipo di illustrazione delle carte in base ad un menu di scelta rispetto le diverse [illustrazioni proposte](https://it.wikipedia.org/wiki/Carte_da_gioco_italiane).

## Il pingatore (2/3)

Realizza un programma che effettua il ping usando il protocollo [ICMP](https://it.wikipedia.org/wiki/Internet_Control_Message_Protocol). Deve prevedere un’interfaccia grafica e poter pingare i server esistenti. L’ouput può essere simile al comando “ping [www.google.it”](about:blank) che puoi eseguire nella shell di Windows. Nel documento del protocollo spiegate con vostre parole come funziona il protocollo.

Se siete in 3 potete realizzare anche una versione del server che risponde al vostro ping e magari anche al ping standard.

## Che tempo fa? (3/4)

Un **server** raccoglie informazioni dalle stazioni meteorologiche (i client) che gli inviano dei dati periodicamente al server (via Datagram). I dati possono essere di vario tipo: velocità vento, quantità pioggia, temperatura, ecc.... Le **stazioni metereologiche** hanno un pannello grafico e sono impostate in una determinata zona. I dati saranno generati in qualche modo. Il server memorizza tutti i dati. Vi sarà poi un altro tipo di client, **il cruscotto**, che chiede al server serie di dati e mostra i risultati con dei grafici. Il cruscotto chiede i dati con TCP. Si può utilizzare JavaFx o altro per i grafici.