



# Astro Allies

Ingegneria del Software a.a. 24/25

Marco Anghienti – Andrea Cagnolati

# Panoramica del Progetto

## Obiettivo

- Gioco ispirato ai giochi arcade tipo Space Invader
  - Ispirato ad Asteroids come gameplay base
- Gioco multigiocatore cooperativo
  - Un giocatore gestisce il movimento della navicella
  - Altri giocatori muovono e sparano con i cannoni
- Applicativo Web
  - PHP per il Back end
  - HTML-CSS-JavaScript per Client

▶ **Progettazione** Requisiti, SRS, Story cards e Testing

▶ **Sistema** Container Docker

▶ **Logica di Gioco** Classi e Logica del Gioco

▶ **Database** Account e Punteggi

▶ **Interfaccia Utente** Pagine web e Grafica di Gioco

▶ **Server “Real Time”** Server con sockets per comunicazione realtime

# Progettazione

Requisiti, SRS, Story cards e Testing

# Requisiti Utente

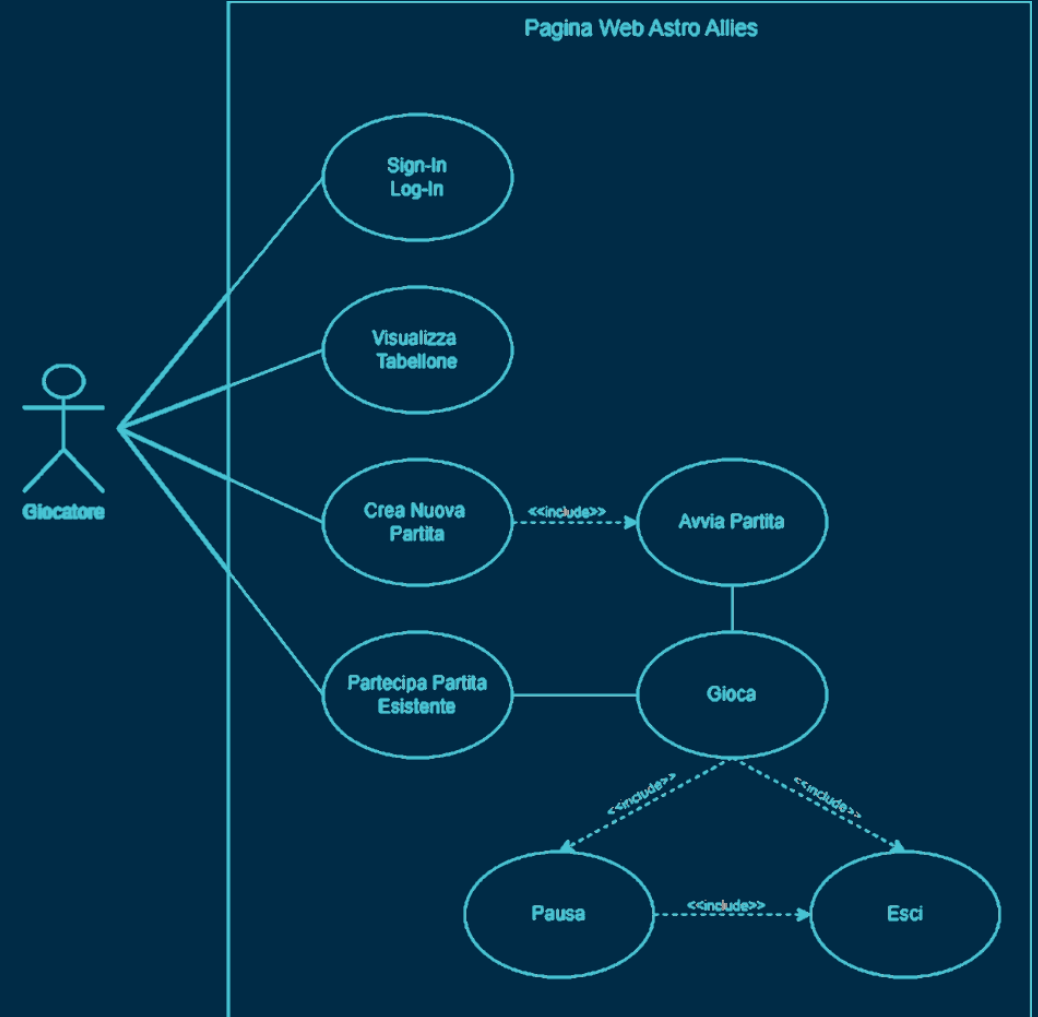
Requisiti principali e vincoli

- Applicativo Web
- Multigiocatore
- Tempo Reale
- Controllo Navicella Cooperativo
- Comunicazioni
- Registrazione ed Accesso
- Punteggio Cumulativo
- Classifica Globale
- Creazione e Unione a Partite
- Utilizzo di PHP per il lato Server
- Utilizzo di JavaScript, HTML e CSS per il lato Client

# Requisiti Utente

## Use Case

- L'utente (nel diagramma "Giocatore") che entra nell'applicativo potrà:
  - Registrarsi o Accedere
  - Visualizzare la Classifica Globale
  - Creare una Partita
  - Unirsi ad una Partita
- Se creata la partita
  - Condividere il codice
  - Avviare la partita
- Se avviata la partita tutti coloro all'interno della stanza passeranno alla fase di gioco



# Analisi dei Requisiti

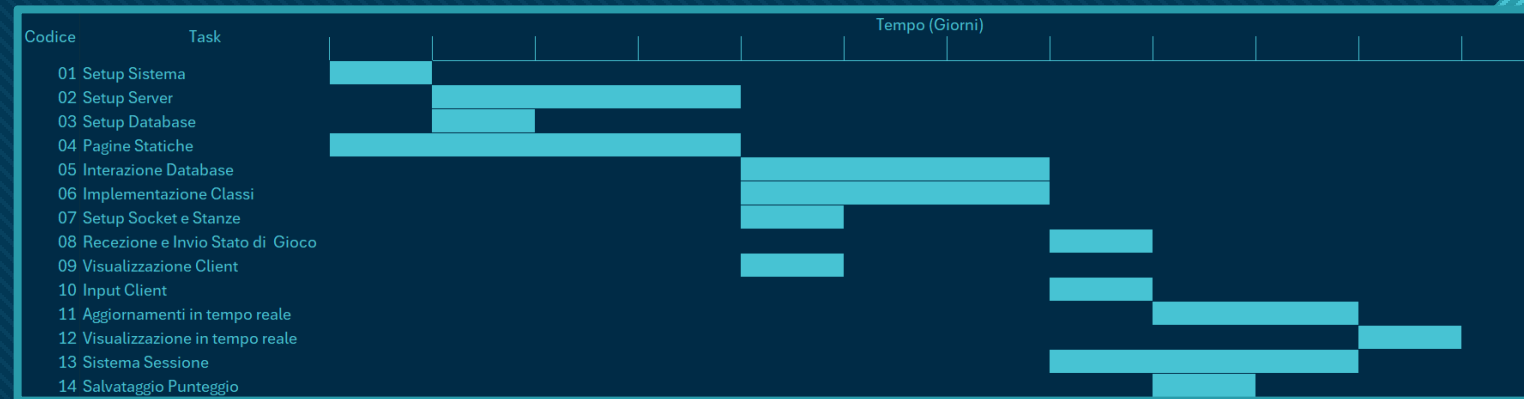
## Funzionalità

- Must
  - Navicella controllata da un giocatore
  - Cannoni controllati da gli altri giocatori
  - Asteroidi
  - Creazione della Partita
  - Stanza
  - Unione a Partita
  - Registrazione ed Accesso
  - Classifica Globale
- Should
  - Abbandono della partita
  - Zona di uscita dalla partita
  - Boost per Navicella
  - Items
- Could
  - Rank Asteroidi
  - Rank Proiettili
  - Comunicazioni

# Suddivisione in Task

Task, Dipendenze e Grantt

- Suddivisione dei compiti in task
- Segnate le dipendenze dei task
- Stima del tempo necessario per ogni task
- Diagramma di Grantt
- Assegnata ad ogni task una priorità







# Sistema

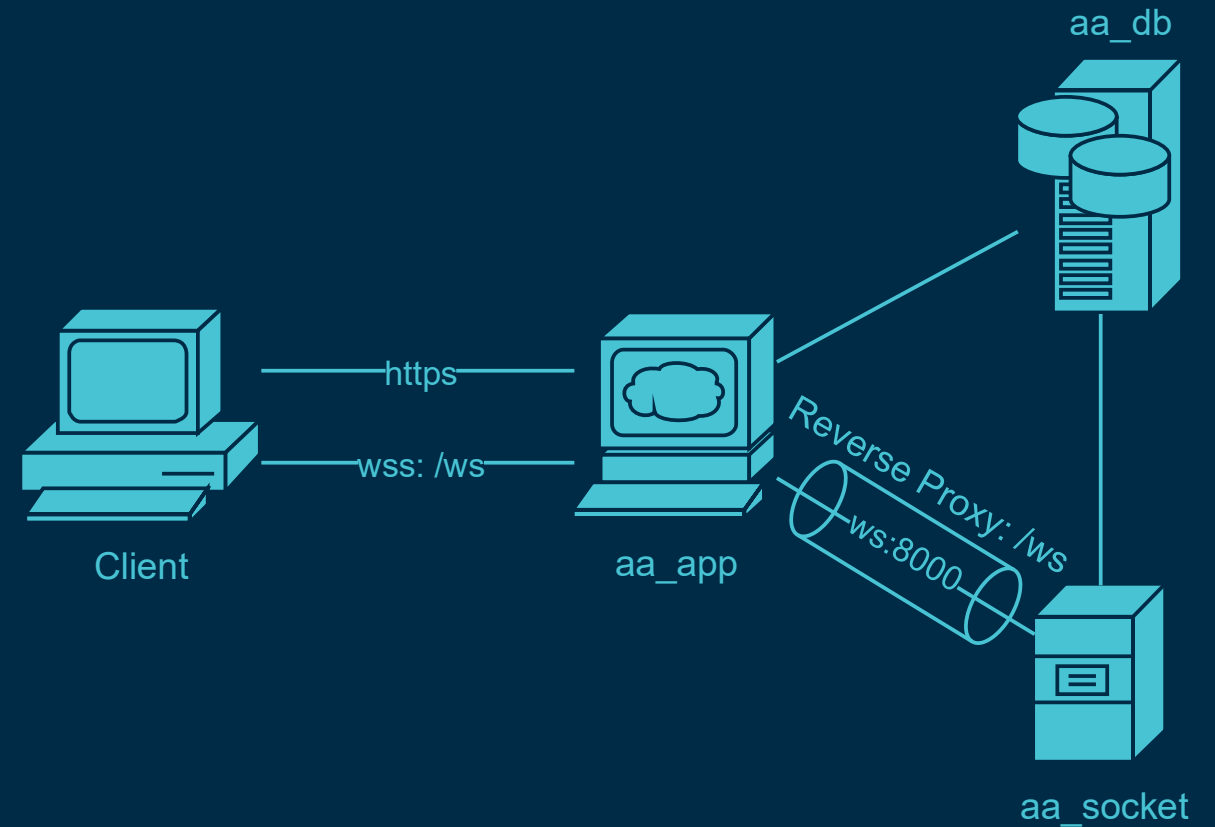
Container Docker

# Sistema

## Container Docker

### Container Docker:

- aa\_db
  - Database
  - MariaDB
- aa\_app
  - Web Server principale
  - Server Apache
  - Connessione https
- aa\_socket
  - Server dedicato alle connessioni socket
  - Accessibile tramite reverse Proxy da aa\_app



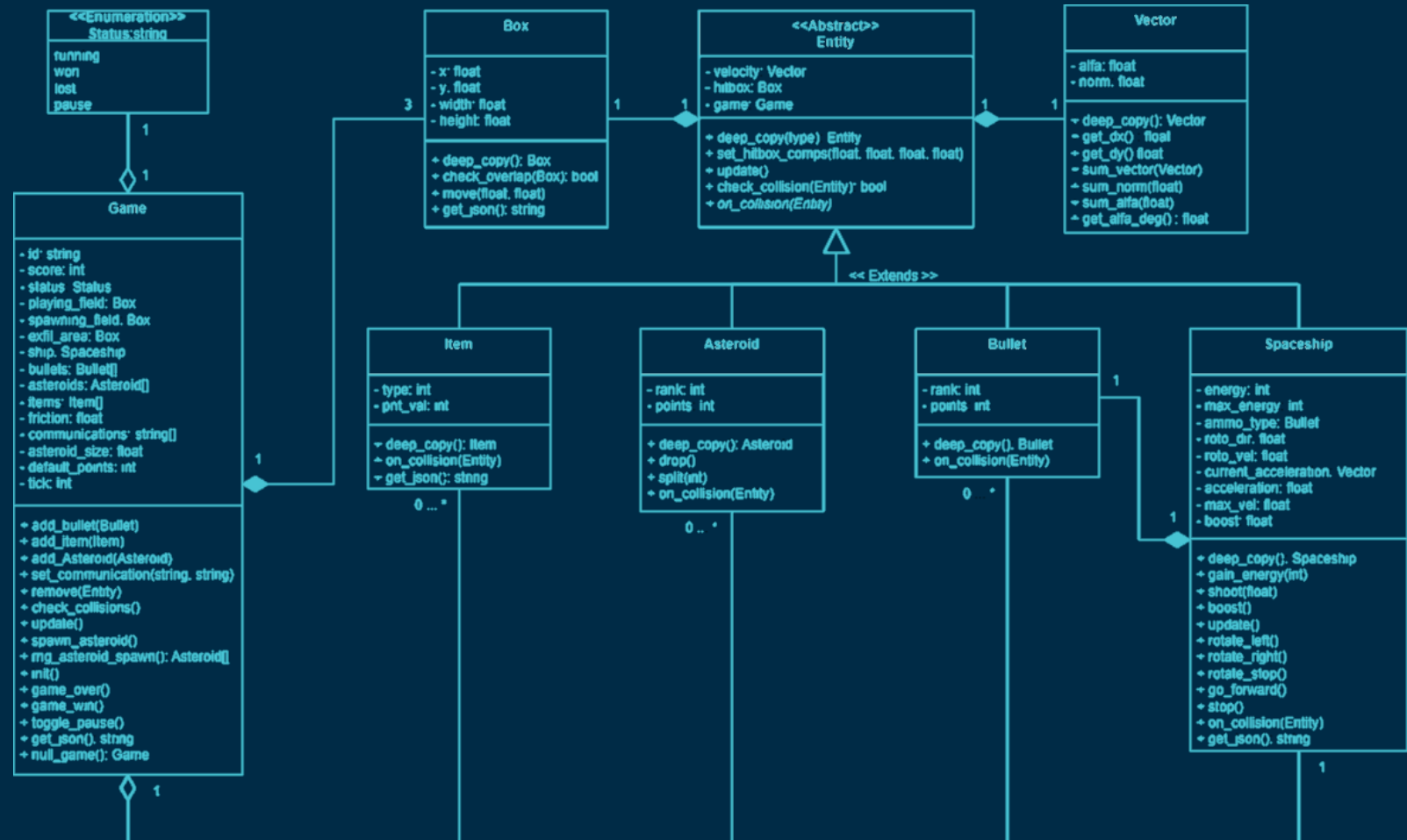
# Logica di Gioco

Classi e Logica del Gioco

# Logica di Gioco

## Schema delle classi

- La classe Entity per tutto ciò che interagisce con altri elementi del gioco
  - Ogni sotto classe gestisce cosa fare in caso di collisione con un'altra Entity
- La classe Game contiene tutte le logiche di gioco esterne alle entità, come ad esempio lo spawn degli asteroidi.



# Database

Account e Punteggi

# Database

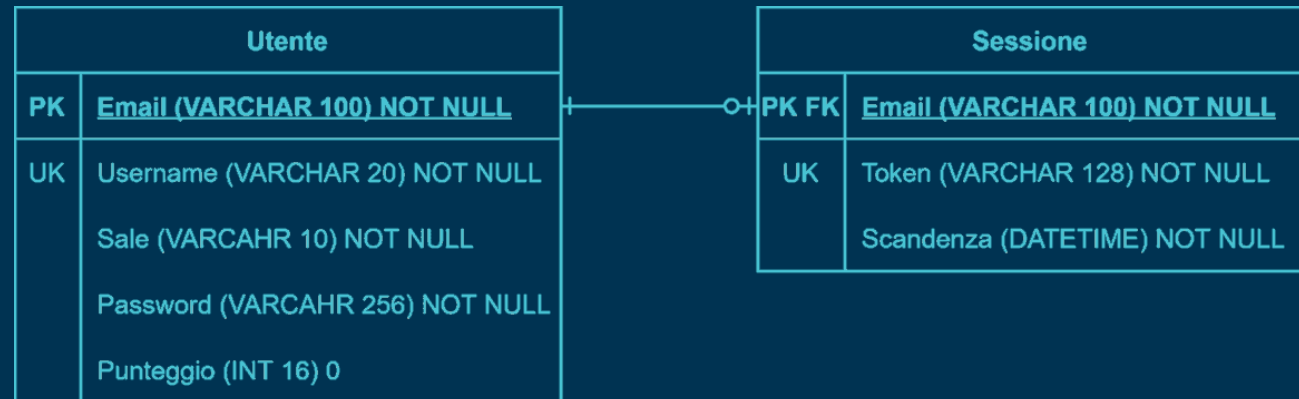
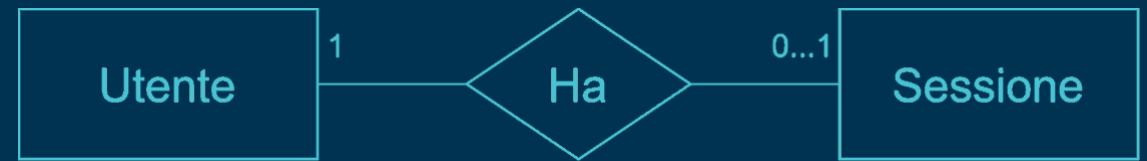
## Struttura

Entità:

- Utente (email, username, sale, password, punteggio)
- Sessione (email, token, scadenza)

Modifiche:

- Vista la già presente implementazione di php delle sessioni, si è rimossa l'entità Sessione
- Visto che la funzione password\_hash di php genera e tiene conto del sale è stato rimosso il campo dalla tabella Utente



# Interfaccia Utente

Pagine web e Grafica di Gioco



# Interfaccia Utente

## Pagine Web

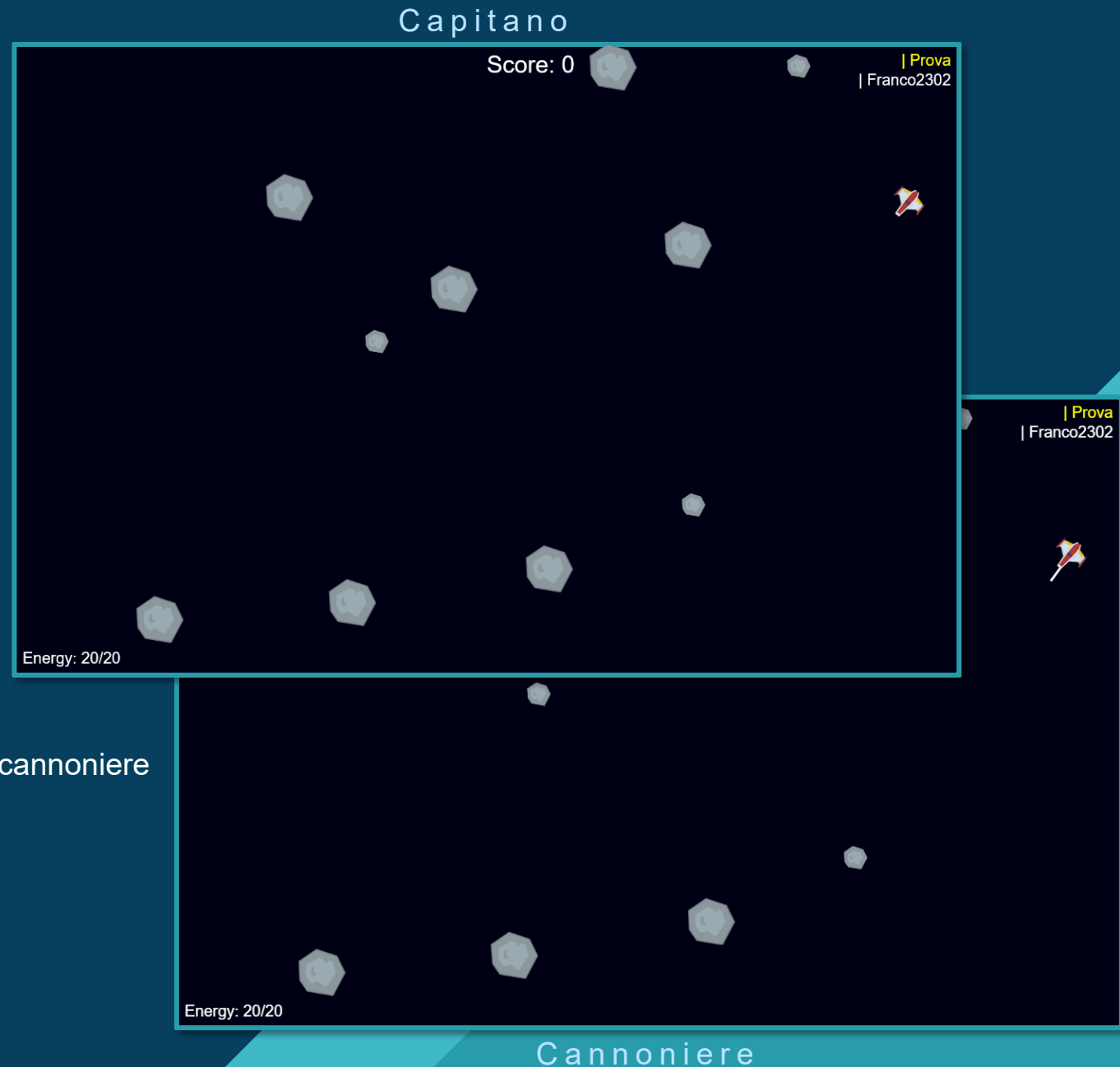
- Pagina Iniziale
  - Pop-up per creare e unirsi ad una partita
  - Link alle altre pagine utili
  - PHP mostra username in caso sia stato eseguito l'accesso
- Pagina di Accesso
  - PHP controlla che la password coincida per quella email nel database
- Pagina di Registrazione
  - PHP controlla che non esista già un account con quell'email o username e se la password sia ben formattata per poi aggiungerlo al database
- Classifica Globale
  - PHP mostra 50 username in ordine per punteggio e il punteggio dell'utente che ha effettuato l'accesso
- Stanza di Attesa e Gioco
  - Connessione in tempo reale, mostra i giocatori nella stanza e il capitano (creatore della stanza) può avviare la partita



# Interfaccia Utente

## Grafica di Gioco

- Utilizzo del Canvas
  - Non utilizzo di librerie grafiche
- Disegno delle entità del gioco tramite sprite
  - Sprite per Navicella, Asteroidi e i diversi Item
- Disegno del cannone per mirare (solo i cannonieri)
  - Cannone gestito dal client e quindi visibile dal singolo cannoniere
- Disegno delle HUD (Heads-Up Display)
  - Energia della Navicella, Punteggio e comunicazioni



# Server “Real Time”

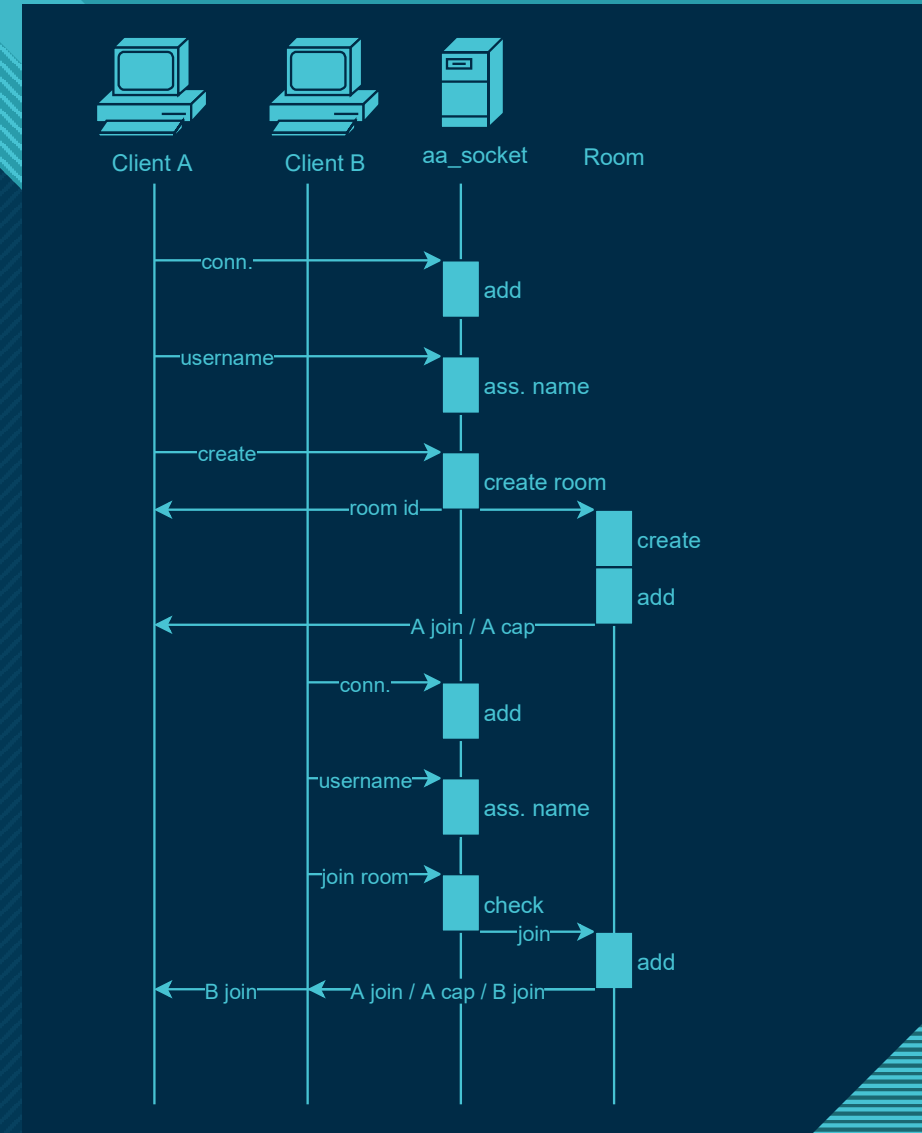
Server con sockets per comunicazione realtime

# Server “Real Time”

Connessione al server, creazione e unione alla stanza

- Per aggiornamenti Real Time è necessario l'utilizzo delle connessioni socket
- Quando un utente si connette tramite socket manda il suo username che viene salvato all'interno del server
- Alla richiesta di creazione della partita viene creata la room lato server con un codice univoco che viene poi mandato al client richiedente della creazione
- Alla connessione di altri utenti viene mandato un messaggio a tutti i presenti nella stanza per notificarli del nuovo utente
- A chiunque si unisca viene mandato un messaggio con il nome del capitano
- Il capitano può iniziare la partita

Nel diagramma la stanza è già vista come un elemento parallelo al server per chiarezza, ma il parallelismo dei programmi avviene in realtà solo una volta la partita è iniziata



# Server “Real Time”

## Update della partita

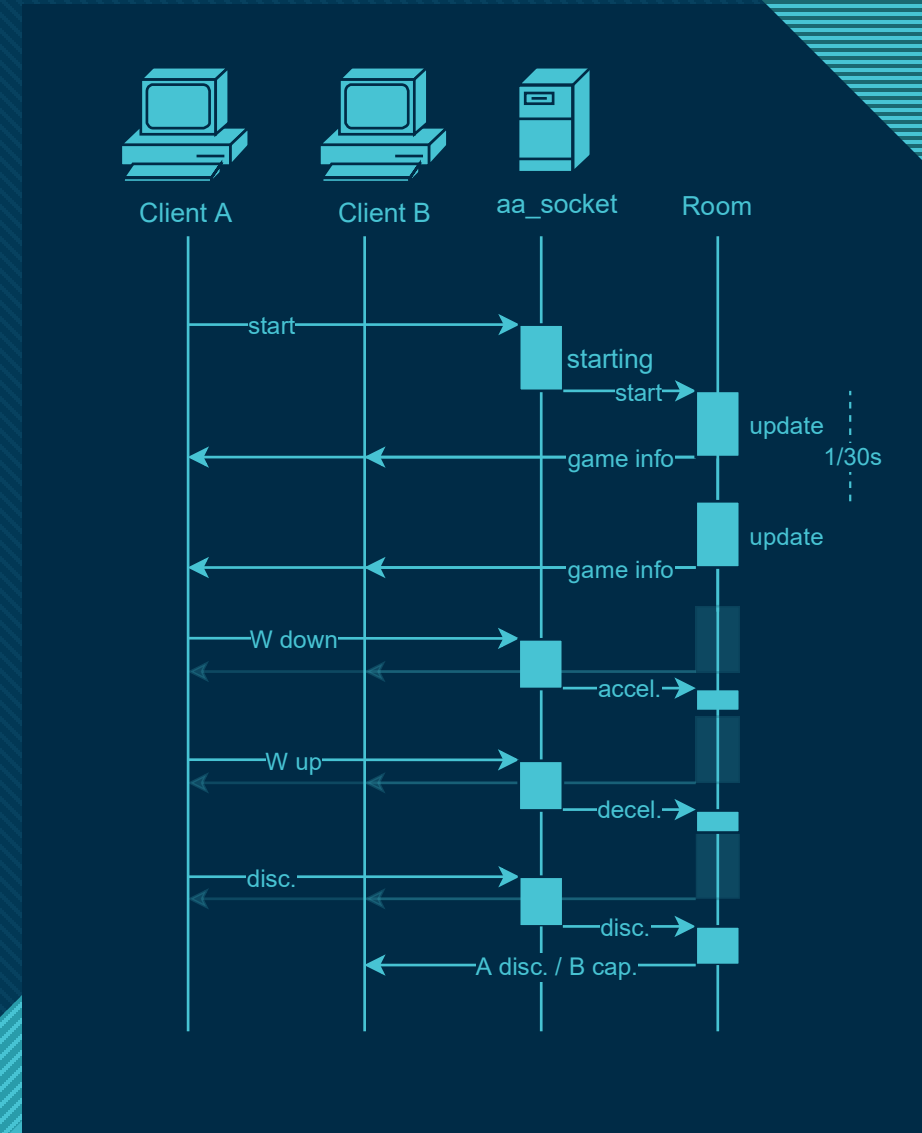
Quando una partita è avviata viene periodicamente (30 volte al secondo) chiamato l'update del gioco

Dal Server al Client:

- Alla stessa frequenza degli update, informazioni sulle entità e le condizioni del gioco (posizioni, energia, punteggio, comunicazioni, ...)
- Disconnessione di giocatori ed eventuale cambio di capitano

Dal Client al Server

- Pressione dei tasti del controllo della navicella (Esclusa la rotazione del cannone)
- Capitano: Fine pressione dei tasti di input del movimento della navicella
- Cannoniere: Sparo con direzione in cui si sta mirando



The background features a dark blue field with a diagonal design. A light blue area in the top-left corner is separated from the dark blue by a diagonal line. This line is flanked by a thin hatched pattern and a thin solid dark blue line.

**Fine  
Presentazione**