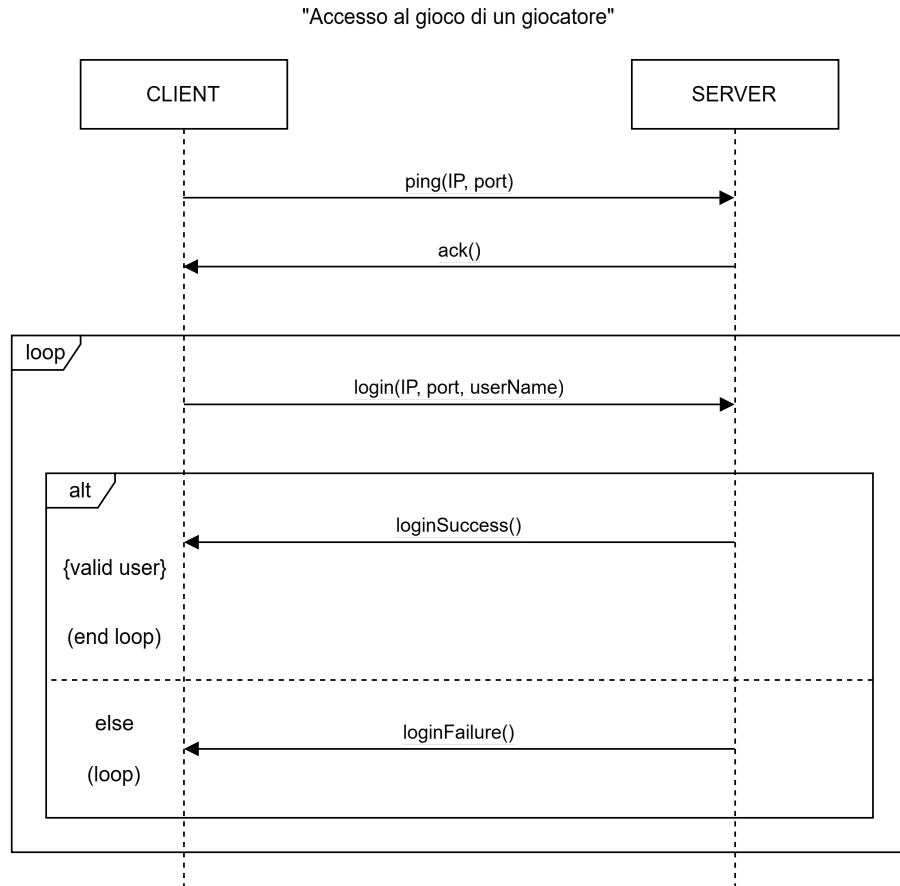# Sequence Diagrams
Software Engineering 2025

Elisa Garsetti, Nicolò Pallavicini, Luca Papiro, Dmitrii Meshcheriakov

25th March 2025 Delivery - Sequence Diagrams

1   Accessing the game as a player

"Accesso al gioco di un giocatore"

```
        ┌─────────────┐                          ┌─────────────┐
        │   CLIENT    │                          │   SERVER    │
        └─────────────┘                          └─────────────┘
               │                                         │
               │            ping(IP, port)               │
               │────────────────────────────────────────▶│
               │                                         │
               │                 ack()                   │
               │◀────────────────────────────────────────│
               │                                         │
  ┌loop╱───────┼─────────────────────────────────────────┼───────┐
  │            │      login(IP, port, userName)          │       │
  │            │────────────────────────────────────────▶│       │
  │            │                                         │       │
  │  ┌alt╱─────┼─────────────────────────────────────────┼─────┐ │
  │  │         │             loginSuccess()              │     │ │
  │  │         │◀────────────────────────────────────────│     │ │
  │  │{valid user}                                       │     │ │
  │  │         │                                         │     │ │
  │  │(end loop)│                                        │     │ │
  │  ├─────────┼─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─┼─────┤ │
  │  │  else   │             loginFailure()              │     │ │
  │  │         │◀────────────────────────────────────────│     │ │
  │  │ (loop)  │                                         │     │ │
  │  └─────────┼─────────────────────────────────────────┼─────┘ │
  └────────────┼─────────────────────────────────────────┼───────┘
               │                                         │
```

1.1   Behaviour

The *ping(IP, port)* message is used to search for the IP and port of the desired server.

The server sends an *ack()* for all the ping messages that it receives.

A loop then begins, in which the client asks to *login(IP, port, userName)* into a game, with the unique userName checked by the server.

If the check is successful: *loginSuccess()*, the loop ends with the client accessing the game.

Otherwise, the server notifies that it was not: *loginFailure()*, and the client has to retry logging in, continuing the loop until it gets it right.

2 Placing a component tile

"Agganciare una carta componente"
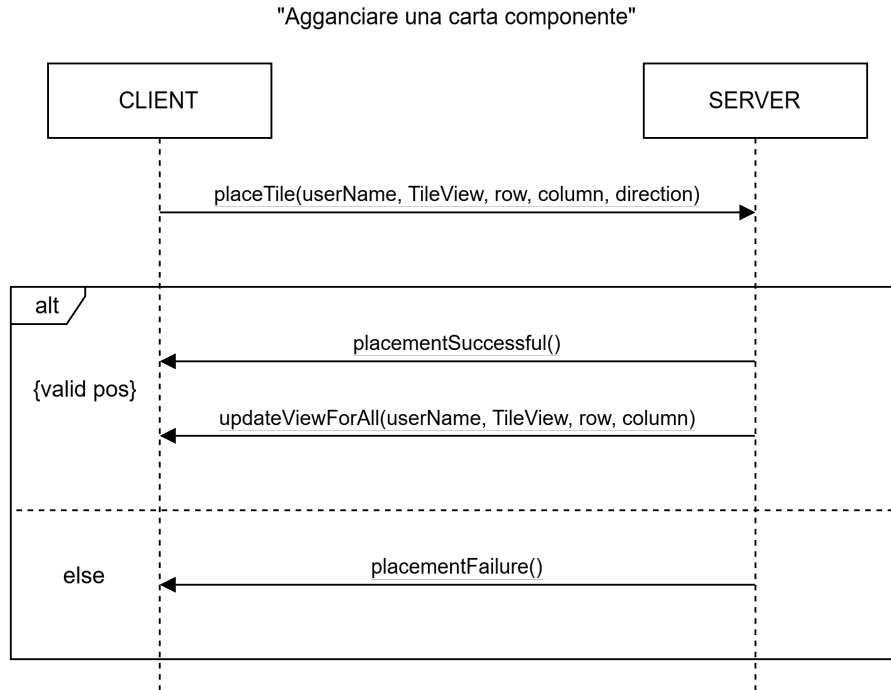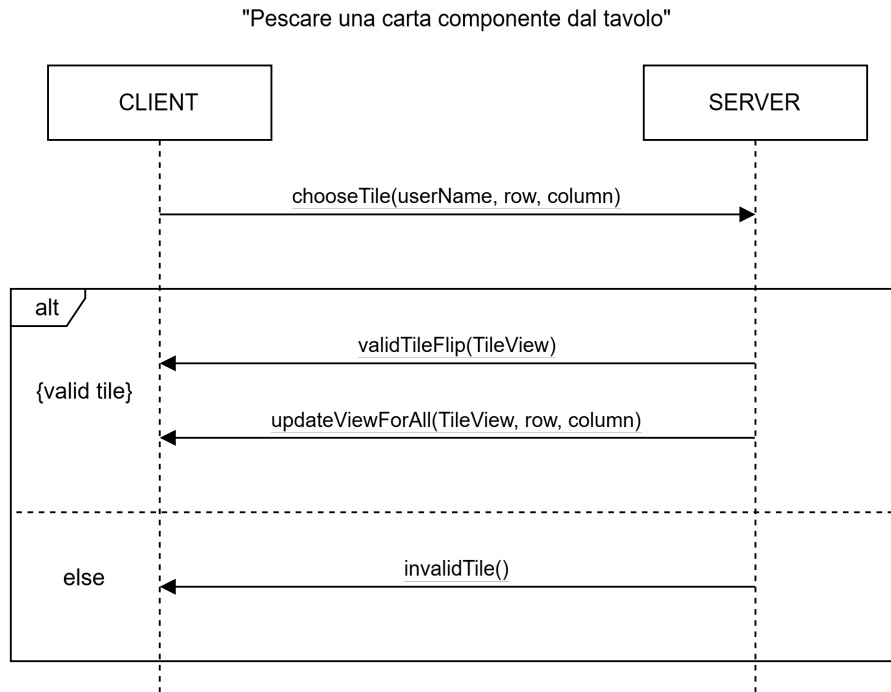

2.1 Behaviour

The client asks the server to place a tile on its floorplan: *placeTile(userName, TileView, row, column, direction)*, in a specific row and column, with a decided orientation.

The server then checks for adjacency issues but avoids connector-related ones, and if it finds none, then the tile gets placed: *placementSuccessful()*.

The server then updates every player's view: *updateViewForAll(userName, TileView, row, column)*.

If instead the client tries to place a tile in an inaccessible spot the server notifies it: *placementFailure()*.

3   Drawing a component tile from the table

"Pescare una carta componente dal tavolo"

3.1   Behaviour

The client sends the server a request to select a tile from the table:
chooseTile(userName, row, column).
The server checks for validity of the tile: if it is currently being viewed by
another player, or already placed on a ship, it cannot be picked up. If none of
these are the case, then it gets picked up, and only the player gets to see it:
validTileFlip(TileView).
The server updates everyone else's view: *updateViewForAll(TileView, row,
column)*, and that tile is currently unavailable to be picked up by others.
If instead the tile is invalid: *invalidTile()*, the server notifies the client.