1. Acceso al gioco di un giocatore

Client — **Server**

Connection()

Loop [while nickname is not valid]
- NicknameRequest(enter nickname)
- NicknameReply(nickname)

Opt
- LoginFailure(nickname not valid)

LoginSuccess()

Opt [If is a new game]
- SetGame(numberOfPlayers)

[If is an existed game]
- RequestGames()
- AvailableGames(availableGames)
- JoinAvailableGame(gameId)
- JoinGameSuccess(gameInfo)

Case [Player connection]
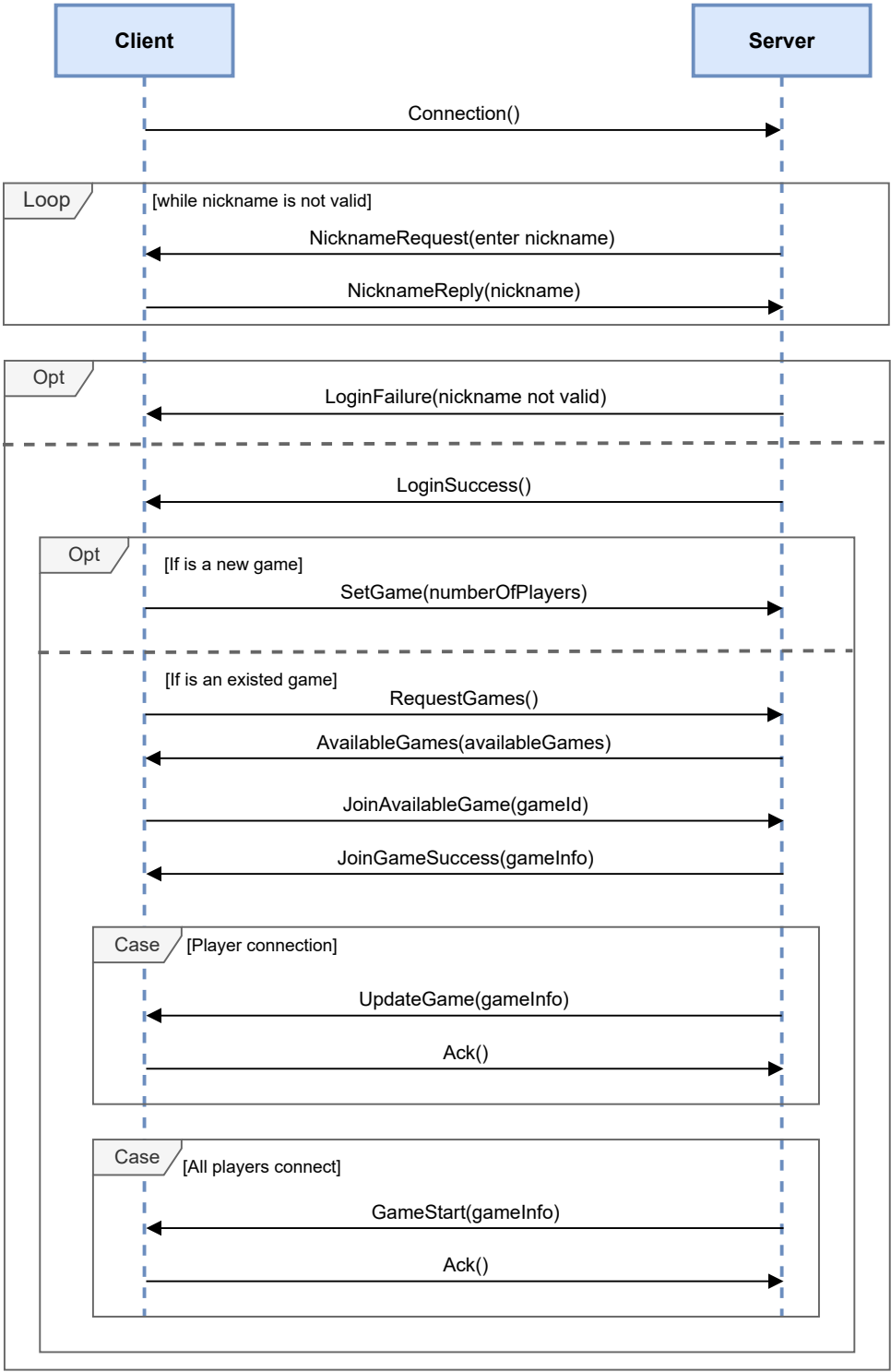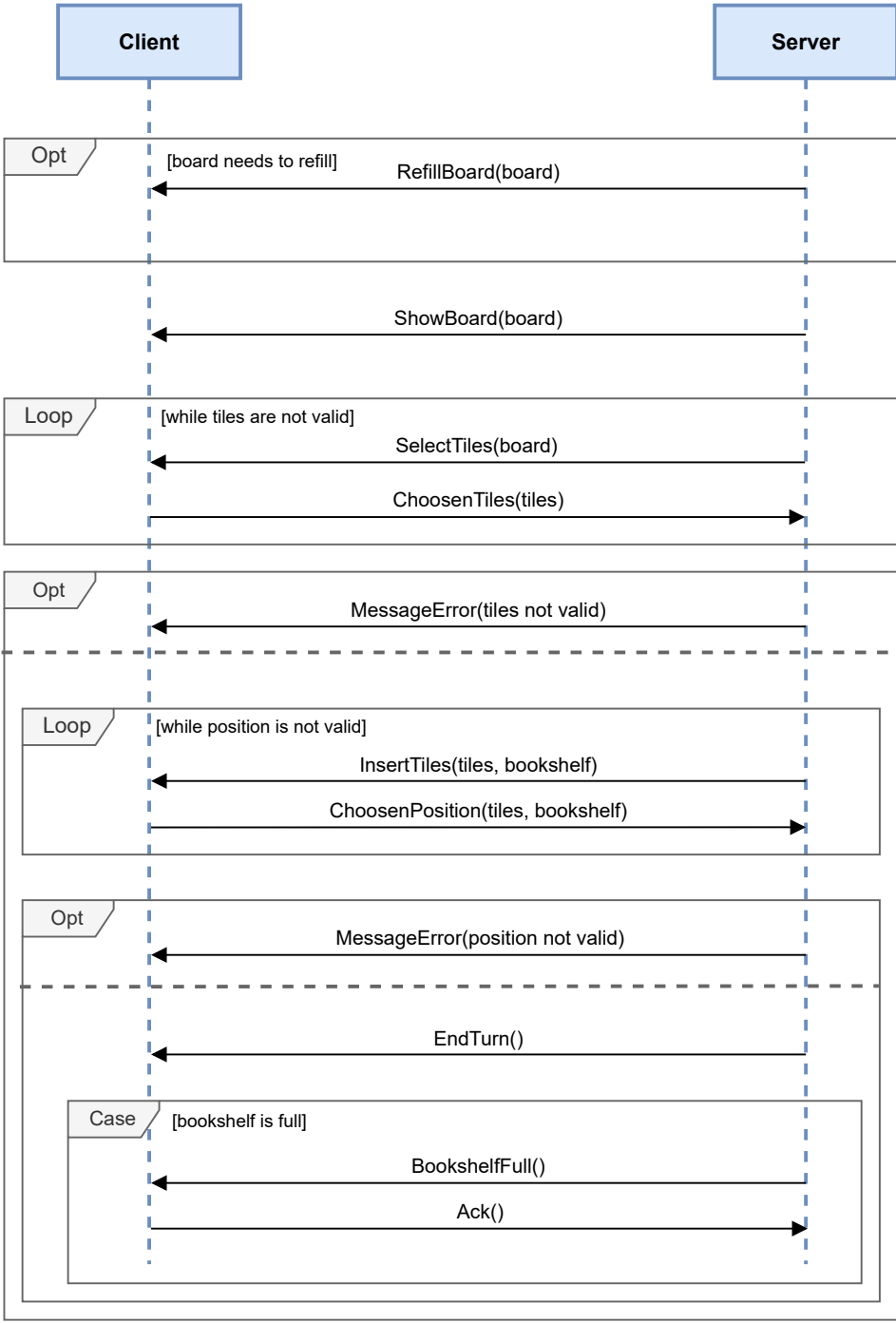- UpdateGame(gameInfo)
- Ack()

Case [All players connect]
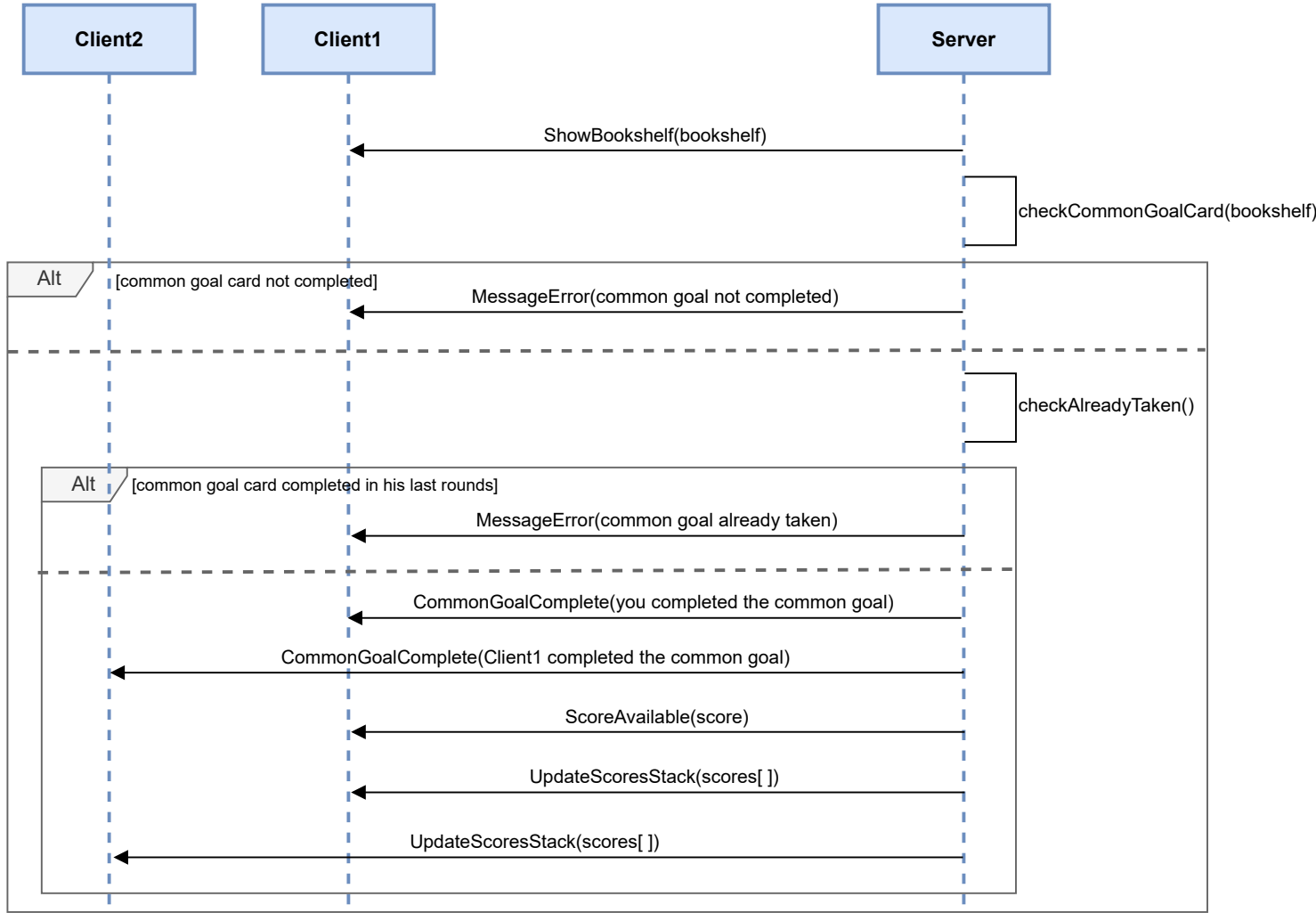- GameStart(gameInfo)
- Ack()

- Connection message establishes the connection between client and server
- Server sends a NicknameRequest to the client asking him to enter his desired nickname
- Client enters his nickname in the NicknameReply
- If nickname has already been taken by another user, the Server sends a LoginFailure message to notify the Client, in order to choose a new nickname, otherwise the Server saves the nickname and sends a LoginSuccess message to the Client
- Client has two possibilities: he can create a new game and enter his desired number of players in SetGame message or he can enter in an already existing game
- In case of joining a game, Client sends a RequestGames message and waits the AvailableGames message from the Server, containing all existed games
- Client chooses a game to join and sends JoinAvailableGame message with the game id of the selected game
- Server replies with a JoinGameSuccess message containing the informations about the game (number of players, current players waiting)
- When a new player joins to the game, Server sends an UpdateGame message to the Client with the updated information of the game
- When all players are connected to the game, Server sends a GameStart message to the Client in order to inform the beginning of the game with the informations about the game (number of player, list of players)

2. Selezione delle tessere e inserimento nella libreria

Client — **Server**

Opt [board needs to refill]
- RefillBoard(board)

ShowBoard(board)

Loop [while tiles are not valid]
- SelectTiles(board)
- ChoosenTiles(tiles)

Opt
- MessageError(tiles not valid)

Loop [while position is not valid]
- InsertTiles(tiles, bookshelf)
- ChoosenPosition(tiles, bookshelf)

Opt
- MessageError(position not valid)

EndTurn()

Case [bookshelf is full]
- BookshelfFull()
- Ack()

- If all Tiles in the Board are isolated or the Board is empty, the Server sends a RefillBoard message to the Client, otherwise the Server shows the Board to the Client
- Server sends a SelectTiles message with the Board to the Client
- Client chooses the Tiles that he wants to insert in his bookshelf and sends a ChoosenTiles message to the Server
- If the selected Tiles are not valid, the Server sends a MessageError to notify the Client, in order to choose other Tiles from the board
- Client selects the column in his bookshelf where he wants to insert the selected Tiles and sends a ChoosenPosition message
- If the selected column is not valid, the server sends a MessageError to notify the Client, in order to choose another column
- At the end of the turn, the Server checks if the Client's bookshelf is full and if this happens notifies the Client and sends him a BookshelfFull message

3. Completamento di un obiettivo comune

Client2 — **Client1** — **Server**

ShowBookshelf(bookshelf)

checkCommonGoalCard(bookshelf)

Alt [common goal card not completed]
- MessageError(common goal not completed)

checkAlreadyTaken()

Alt [common goal card completed in his last rounds]
- MessageError(common goal already taken)

CommonGoalComplete(you completed the common goal)

CommonGoalComplete(Client1 completed the common goal)

ScoreAvailable(score)

UpdateScoresStack(scores[ ])

UpdateScoresStack(scores[ ])

- Server checks if the Client1 has completed the Common Goal Card
- If the Common Goal Card is not completed, the Server sends a Message to notify the Client1
- If the Client1 has already completed the Common Goal Card in his last rounds, the Server sends a Message to notify the Client1
- If the Client1 has completed the Common Goal Card, Server sends a message to the Client1 and a ScoreAvailable message with the relative score he has obtained
- Also Server sends a message to Client2 to advise him about the Client1's completion of the Common Goal Card
- Server updates the stack of the scores available for that Common Goal Card and sends it to all the Clients