

Use Case 3.2: Join Game

Author: Yasin Garip

Characteristic Information

Goal in Context: The user can join a game to play over network with human remote players

Scope: Network

Level: Subfunction

Primary Actor: User

Stakeholders & Interests:

- ❖ User: Wants to join a network game of Scrabble

Preconditions: User navigated through the application and clicked 'Join Game Lobby'

Success Guarantee (Postconditions): User joined the lobby successfully and all other participants were notified about the event.

Trigger: click on Button 'Join Game Lobby'

Main Success Scenario

1. User clicks on "Join Game"-Button
2. GUI shows prompt for user to enter connection details
3. User types in the connection details and clicks 'Join game lobby'- Button
4. Server gets JoinRequest of User-Client
5. Server accepts JoinRequest of User-Client
6. User gets connected to the Game Lobby
7. Server notifies all other connections about event

Extensions

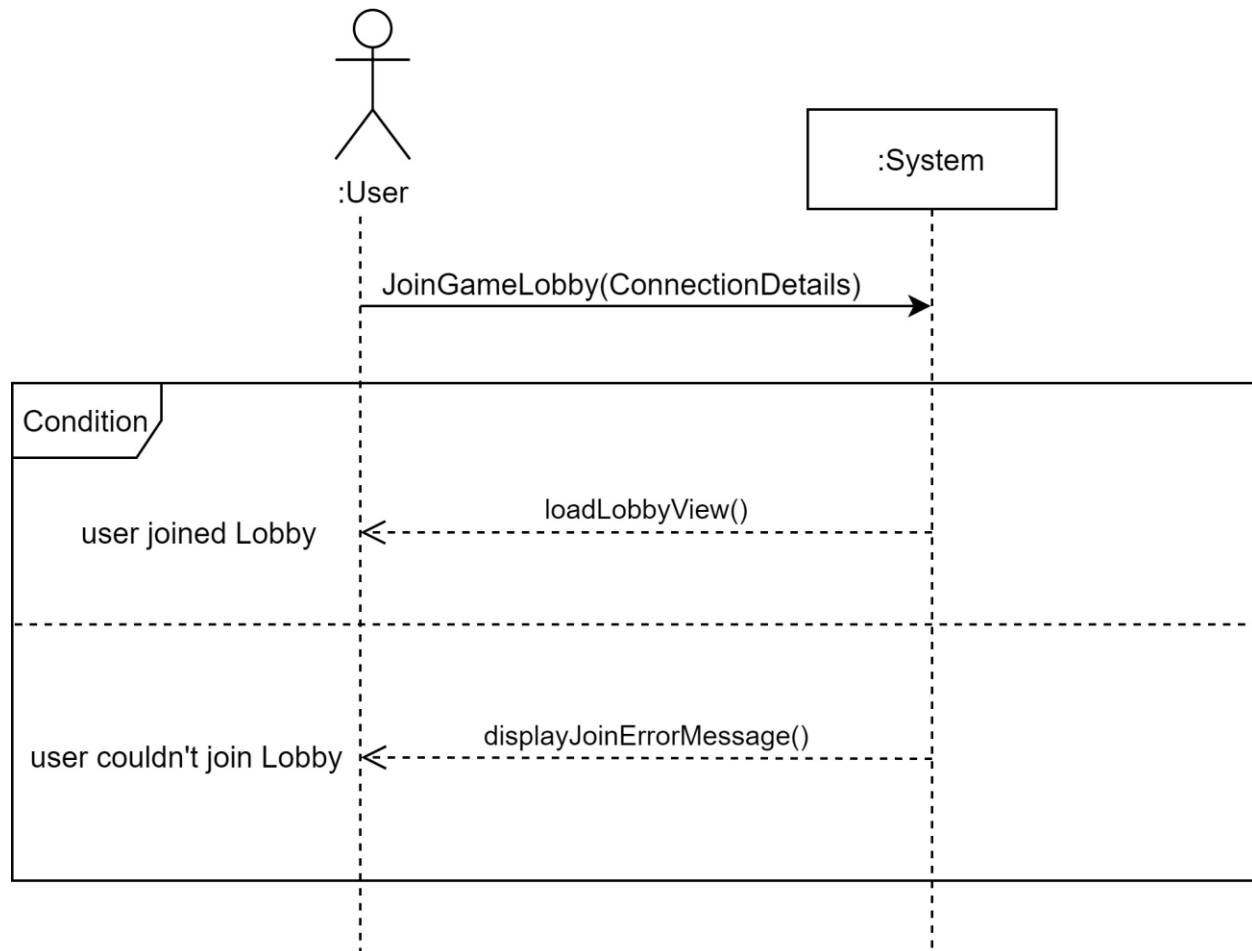
- *a. At any time, System fails:
 - 1. System reopens system, logs in, and requests recovery of prior state.
 - 2. System reconstructs prior state.
 - 2a. System detects anomalies preventing recovery: System signals error and records the error. System starts new tutorial.
 - 3. Warning sign that the game crashed appears.
- 4. Connection cannot be established:
 - 1. User gets informed that connection details do not lead to a running server.
- 5. Server declines JoinRequest:
 - 1. Lobby is filled with enough players.
 - 2. System notifies user about filled Lobby via an appropriate error message.

Due Date

10.05.2021

System Sequence Diagram

Author: Yasin Garip



Operation Contracts

Author: Yasin Garip

Contract 3.2.1: joinLobby

Operation: joinLobby() : void

Cross References: UC 3.2: Join Game

Preconditions:

- User u1 created a lobby and runs a server s on his machine
- User u2 knows the connection details of s

Postconditions:

- Association Player p of u2 was associated with s