sd 1_ClientConnection

sd 2_SearchGame_2

Client | Server | OtherClient

alt

[No Game currently available]

1: PlayGame(idUser, idRequest, gameRules)

GameLobby is not another actor, but is represented to explain what happens when the clients sends certain messages

<<create>>
1.1: CreateGame(numPlayers, GameRules) → GameLobby

1.2: AddPlayer(idUser)

Added(emptySeats)

AddedToLobby(Host)

[There's an available lobby]

2: PlayGame(idUser, idRequest, gameRules)

2.1: AddPlayer(idUser)

Added(emptySeats)

AddedToLobby(Guest)

loop : until the game starts [Guard]

opt

3: ReadyToStart(idUser, idRequest)

3.1: UserReady(idUser)

opt

4: NotReady(idUser, idRequest)

4.1: UserNotReady(idUser)

opt : a lobby change occurs

5: AnyLobbyModifyingMessage()

6: ServerLobbyStatus(beanType, bean)

6.1: PrintLobby()

opt

7: LeaveLobby()

7.1: RemovePlayer(idUser)

alt [If this user is the host]

alt [All Users ready]

8: StartGame(idUser, idRequest)

8.1: CheckAllPlayersReady()

OK

OK

break

ref
GameStart

ref
GameStart

[Users not ready]

9: StartGame(idUser, idRequest)

9.1: CheckAllPlayersReady()

ERR

ERR

opt : if this user is NOT the host

10: StartGame()

11: ServerLobbyStart()

ref
GameStart

opt

courtesy of the client

12: Quit(idUser, idRequest)

12.1: RemovePlayer(idUser)

opt : emptySeats == numPlayers

<<destroy>>
13: DestroyLobby()

2 / 7

sd 3_GameStart

Client   Server   OtherClient

loop [Guard]

  alt

  [wizard_available]

    1: SelectWizard(idUser, idRequest, idWizard)

    break

    OK

  [wizard_taken]

    2: SelectWizard(idUser, idRequest, idWizard)

    ERR

loop [Guard]

  alt

  [color_available]

    3: SelectTowerColor(idUser, idRequest, idTowerColor)

    break

    OK

  [color_taken]

    4: SelectTowerColor(idUser, idRequest, idTowerColor)

    ERR

loop : until the game starts [Guard]

  alt

    5: SelectWizard()

  [Guard]

    6: SelectTowerColor()

  7: ServerGameInitializationStatus(beanType, bean)

  7.1: updateAvailableChoices()

opt : If the selections have been made

  alt

  [Guard]

    8: SelectWizard()

  [Guard]

    9: SelectTowerColor()

  10: CheckEveryoneReady()

  loop : All players in the lobby [Guard]

    11: ServerGameStart()

sd 4_Turn

Client   **Server**

Planning Phase

alt : phase == PLANNING

1: ServerYourTurn(phase)

2: showGameOptions(PLANNING)

3: ChooseAssistant(idUser, idRequest, idAssistant)

ACK

Action Phase

alt : phase == ACTION

4: ServerYourTurn(phas)

5: showGameOptions(ACTION)

loop : 3-4 times [Guard]

activation of character card available at the red arrows but only shown at the end

winning conditions checked at the blue arrows but only shown at the end

6: SelectStudent(idUser, idRequest, position)

ACK

moving of students

alt

[hall_chosen]

7: PutInHall(idUser, idRequest)

ACK

[island_chosen]

8: PutInIsland(idUser, idRequest, IdIslandGroup)

ACK

[deselect]

9: DeselectStudent(idUser, idRequest)

ACK

10: MoveMotherNature(idUser, idRequest, steps)

ACK

moving of mother nature

11: ChooseCloud(idUser, idRequest, cloud)

ACK

choice of cloud

12: EndTurn(idUser, idRequest)

ACK

opt [character_activation]

ref

character_card_played

opt [win checking]

13: isGameOver()

opt [game is over]

loop : for all users in the game

14: ServerGameWon(asyncWinner)

sd 5_ViewUpdate

Client	Server	OtherClient

1: AnyModelUpdate()

1.1: UpdateModel()

2: ServerGameUpdate()

2.1: UpdateView()

sd 6_Character_card_played

Client   Server

alt [Enough coins]

1: SelectCharacter(idUser, idRequest, characterCardPos)

CardRequirements

[Insufficient coins]

2: SelectCharacter(idUser, idRequest, characterCardPos)

break : Exits the routine

ERR

loop : as many times as needed [Guard]

alt [no_more_inputs_required]

3: PlayCharacter(idUser, idRequest, characterCardPos)

3.1: activateEffect()

break

[effect_activated]

ACK

[input_required]

opt [student_color_required]

4: SelectedStudentColors(idUser, idRequest, selectedColors)

ACK

opt [entrance_student_required]

5: SelectEntranceStudents(idUser, idRequest, atEntraceSelected)

ACK

opt [island_group_required]

6: SelectIslandGroup(IdUser, idRequest, idIslandGroup)

ACK

opt [student_on_card_required]

7: SelectStudentsOnCard(idUser, idRequest, onCardSelected)

ACK

opt

8: PlayCharacter()

8.1: deselectEverything()

break : Exits the routine

ERR

sd 7_UserDisconnection

Client | Server | DisconnectedClient

1: Quit()

2: handleDisconnection()

alt : Players are in a game with the disconnected user

loop : For each player in the game

3: ServerUserDisconnected(asyncIdUser, asyncUserNickname)

3.1: notifyUser()

4: destroyGame()