|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **registering** | **waiting for marco** | **responding** | **waiting for location** | **finishing** |
| **registerSuccess** | **a: string name**  **loc\_t loc**  **send name & loc to server**  **s: waiting for marco** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **registerFail** | **a: send name & loc to server**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **marcoMsg** | a: log error  exit  s: done | **a: receive msg from server**  **s: responding** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **gameOverMsg** | a: log error  exit  s: done | **a: receive msg from server**  **s: finishing** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **unknownMsg** | a: log error  exit  s: done | **a: receive msg from server**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **respondSuccess** | a: log error  exit  s: done | a: log error  exit  s: done | **a: send dir\_t to server**  **s: waiting for location** | a: log error  exit  s: done | a: log error  exit  s: done |
| **respondFail** | a: log error  exit  s: done | a: log error  exit  s: done | **a: send dir\_t to server**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done |
| **getLocSuccess** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: receive new loc from server**  **reset loc**  **s: waiting for marco** | a: log error  exit  s: done |
| **getLocFail** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: receive new loc from server**  **log error**  **exit**  **s: done** | a: log error  exit  s: done |
| **gameOver** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: log loser**  **exit**  **s: done** |
| **unknown** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |

**Client State Machine**

Cohen Adair

**Server State Machine**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **wait for players** | **resister player** | **playing** | **wait for player turn** | **move players** | **take turn** | **finishing** |
| **init** | **a: plrs[max]**  **cnt = 0**  **s: register player** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **registerSuccess** | a: log error  exit  s: done | **a: set plrs[cnt]**  **cnt++**  **s: wait for players** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **cntAtMax** | **a: check cnt**  **s: playing** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **cntLessThanMax** | **a: check cnt**  **s: wait for players** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **sendLocSuccess** | a: log error  exit  s: done | **a: send player loc**  **s: wait for players** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **sendLocFail** | a: log error  exit  s: done | **a: send player loc**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **registerFail** | a: log error  exit  s: done | **a: set plrs[cnt]**  **cnt++**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **sendMarcoSuccess** | a: log error  exit  s: done | a: log error  exit  s: done | **a: send msg & loc to players**  **s: wait for player turn** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **sendMarcoFail** | a: log error  exit  s: done | a: log error  exit  s: done | **a: send msg & loc to players**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **getResponseSuccess** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: receive player locs**  **s: move players** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **getResponseFail** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: receive player locs**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |
| **movePlayersSuccess** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: update player locs**  **send locs**  **s: take turn** | a: log error  exit  s: done | a: log error  exit  s: done |
| **sendNewLocsSuccess** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: send new player locs**  **s: move players** | a: log error  exit  s: done | a: log error  exit  s: done |
| **sendNewLocsFail** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: send new player locs**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done |
| **movePlayersFail** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: update player locs**  **send locs**  **log error**  **exit**  **s: done** | a: log error  exit  s: done | a: log error  exit  s: done |
| **moveSuccess** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: move to new loc**  **check loc**  **s: finishing** | a: log error  exit  s: done |
| **moveFail** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: move to new loc**  **check loc**  **s: playing** | a: log error  exit  s: done |
| **gameOver** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | **a: send GAMEOVER msg and loser**  **s: done** |
| **unknown** | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done | a: log error  exit  s: done |

**PDU Names**

**Global**

dir\_t

client\_response\_t

loc\_t

msg\_t

**Client**

registerSuccess

registerFail

marcoMsg

gameOverMsg

unknownMsg

respondSuccess

respondFail

getLocSuccess

getLocSuccess

getLocFail

gameOver

unknown

**Server**

Init

registerSuccess

cntAtMax

cntLessThanMax

sendLocSuccess

sendLocFail

registerFail

sendMarcoSuccess

sendMarcoFail

getResponseSuccess

getResponseFail

movePlayerSuccess

movePlayerFail

sendNewLocsSuccess

sendNewLocsFail

moveSuccess

moveFail

gameOver

unknown

**Global PDUs**

ENUM dir\_t

UINT32 N = 1

UINT32 NE = 2

UINT32 E = 3

UINT32 SE = 4

UINT32 S = 5

UINT32 SW = 6

UINT32 W = 7

UINT32 NW = 8

UINT32 STAY = 9

endENUM

STRUCT client\_response\_t

dir\_t dirToServer

dir\_t dirToMove

UINT32 distToMove

endSTRUCT

STRUCT loc\_t

UINT32 x

UINT32 y

endSTRUCT

STRUCT msg\_t

U8 msg

loc\_t loc

endSTRUCT

**Client PDUs**

PDU registerSuccess

loc\_t locaction

U8 name

endPDU

PDU registerFail

loc\_t location

U8 name

U8 errorMsg

endPDU

PDU marcoMsg

msg\_t msg

loc\_t locaction

endPDU

PDU gameOverMsg

msg\_t msg

U8 loser

endPDU

PDU unknownMsg

msg\_t msg

U8 errorMsg

endPDU

PDU respondSuccess

dir\_t direction

endPDU

PDU respondFail

dir\_t direction

U8 errorMsg

endPDU

PDU getLocSuccess

loc\_t newLocation

loc\_t currentLocation

endPDU

PDU getLocFail

loc\_t newLocation

U8 errorMsg

endPDU

PDU gameOver

U8 loser

endPDU

PDU unknown

U8 errorMsg

endPDU

**Server PDUs**

PDU init

loc\_t loc

UINT32 cnt = 0

SEQUENCE OF (cnt)

U8 playerName

loc\_t playerLocation

endPDU

PDU registerSuccess

U8 playerName

loc\_t playerLocation

UINT32 cnt++

endPDU

PDU cntAtMax

UINT32 cnt

U8 gameIsFullMsg

endPDU

PDU cntLessThanMax

UINT32 cnt

endPDU

PDU sendLocSuccess

loc\_t playerLocation

endPDU

PDU sendLocFail

loc\_t playerLocation

U8 errorMsg

endPDU

PDU registerFail

U8 playerName

loc\_t playerLocation

UINT32 cnt++

U8 errorMsg

endPDU

PDU sendMarcoSuccess

U8 marcoMsg

loc\_t playerLocation

endPDU

PDU sendMarcoFail

U8 marcoMsg

loc\_t playerLocation

U8 errorMsg

endPDU

PDU getResponseSuccess

loc\_t playerLocation

endPDU

PDU getResponseFail

loc\_t playerLocation

U8 errorMsg

endPDU

PDU movePlayerSuccess

loc\_t oldPlayerLocation

loc\_t newPlayerLocation

endPDU

PDU sendNewLocsSuccess

loc\_t playerLocation

endPDU

PDU sendNewLocsFail

loc\_t playerLocation

U8 errorMsg

endPDU

PDU movePlayersFail

loc\_t oldPlayerLocation

loc\_t newPlayerLocation

U8 errorMessage

endPDU

PDU moveSuccess

loc\_t oldLoc

loc\_t newLoc

endPDU

PDU moveFail

loc\_t oldLoc

loc\_t newLoc

U8 errorMsg

endPDU

PDU gameOver

U8 gameOverMsg

U8 loser

endPDU

PDU unknown

U8 errorMsg

endPDU