

Client State Machine

	registering	waiting for marco	responding	waiting for location	finishing
<b>registerSuccess</b>	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
<b>registerFail</b>	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
<b>marcoMsg</b>	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
<b>gameOverMsg</b>	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
<b>unknownMsg</b>	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
<b>respondSuccess</b>	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
<b>respondFail</b>	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
<b>getLocSuccess</b>	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
<b>getLocFail</b>	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
<b>gameOver</b>	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
<b>unknown</b>	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

## Server State Machine

	<b>wait for players</b>	<b>resister player</b>	<b>playing</b>	<b>wait for player turn</b>	<b>move players</b>	<b>take turn</b>	<b>finishing</b>
<b>init</b>	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
<b>registerSuccess</b>	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
<b>cntAtMax</b>	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
<b>cntLessThanMax</b>	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
<b>sendLocSuccess</b>	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
<b>sendLocFail</b>	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
<b>registerFail</b>	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
<b>sendMarcoSuccess</b>	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
<b>sendMarcoFail</b>	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
<b>getResponseSuccess</b>	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
<b>getResponseFail</b>	a: log error exit s: done	a: log error exit s: done	a: log error exit s: done	a: receive player locs log error	a: log error exit s: done	a: log error exit s: done	a: log error exit s: done

[illegible]

## 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 location
    U8    name
endPDU
```

```
PDU registerFail
    loc_t location
    U8    name
    U8    errorMsg
endPDU
```

```
PDU marcoMsg
    msg_t msg
    loc_t location
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
```