

RagnarokListener

INHERITS FROM

Listener : Object

DECLARED IN

OpponentApp/InvitationListener.h

CLASS DESCRIPTION

The RagnarokListener class receives messages pertinent to a game being currently played with a Ragnarok application.

INSTANCE VARIABLES

Inherited from Object

Class

isa;

Inherited from Listener

char
port_t
port_t
id delegate;
int timeout;

*portName;
listenPort;
signaturePort;

int priority;

Declared in RagnarokListener (none)

METHOD TYPES

- | | |
|---------------------------------|---|
| Receiving changes of state | - submitMoveFrom:To:
- submitUndos:
- submitResetGame |
| Receiving requests | - pleaseAllow:Undo:
- pleaseAllow:StartOver:
- pleaseHurryUp: |
| Receiving responses to requests | - allow:
- refuse:
- ok: |
| Miscellaneous communication | - submitTentativeMoveFrom:To: |

- submitNoTentativeMove
- submitMessage:

Receiving a notification of termination - goodbye

INSTANCE METHODS

allow:

-(int)**allow** : (int) *tag*

Receives permission in response to a previous request *tag*.

See also: - **pleaseAllow:Undo:**, **pleaseAllow:StartOver:**

goodbye:

-(int)**goodbye**

Receives notification that the sending application has terminated the connection.

ok:

-(int)**ok** : (int) *tag*

Receives acknowledgment of a previous **pleaseHurryUp:** message *tag*.

See also: - **pleaseHurryUp:**

pleaseAllow:StartOver:

-(int)**pleaseAllow** : (int) *tag*

StartOver : (int) *ignored*

Receives a request to start the game over. Either an **allow:** or a **refuse:** message should be sent in reply, with the number *tag*. Whatever you do, be sure to not pay any attention to *ignored*.

See also: - **allow:**, **refuse:**

pleaseAllow:Undos:

-(int)**pleaseAllow** : (int) *tag*

Undos : (int) *howmany*

Receives a request to undo *howmany* moves. Either an **allow:** or a **refuse:** message should be sent in reply, with the number *tag*.

See also: - **allow:**, **refuse:**

pleaseHurryUp:

-(int)**pleaseHurryUp** : (int) *tag*

Receives a request to hurry up. An **ok:** message should be sent in response, with the number *tag*.

See also: - **ok:**

refuse:

-(int)**refuse** : (int) *tag*

Receives refusal of permission in response to a previous request *tag*.

See also: - **pleaseAllow:Undo:**, **pleaseAllow:StartOver:**

submitMessage:

-(int)submitMessage : (char *) *aString*

Receives message *aString*. No reply or action is expected.

submitMoveFrom:To:

-(int)submitMoveFrom : (int) *from*
To : (int) *to*

Receives notification that the sender has made a move from *from* to *to*. The sender has already updated its state; the receiver should update its state to match (i.e., make the move).

submitNoTentativeMove

-(int)submitNoTentativeMove

Receives notification that the sender is no longer considering any moves. No reply or action is expected.

See also: - **submitTentativeMoveFrom:To:**

submitResetGame

-(int)submitResetGame

Receives notification that the sender has reset the game. The sender has already updated its state; the receiver should update its state to match (i.e., reset the game).

submitTentativeMoveFrom:To:

-(int)submitNoTentativeMoveFrom : (int) *from*
To : (int) *to*

Receives notification that the sender is considering moving from *from* to *to*. No reply or action is expected.

See also: - **submitNoTentativeMove**

submitUndos:

-(int)submitUndos: (int) *howMany*

Receives notification that the sender has undone *howMany* moves. The sender has already updated its state; the receiver should update its state to match (i.e., undo the moves).