

Contratti

Precondizione globale: Il *Manager M* esiste dall'inizio del caso d'uso e resta lo stesso per tutto l'UC.

Iniziare partita

iniziaCreazionePartita(porta, parolaChiave)

Precondizioni: *porta* è un numero, *parolaChiave* è un testo. Non esistono oggetti *Server*

Postcondizioni: viene creato un oggetto *Server* con attributi pari ai parametri dell'operazione.

creaPartita(robotdromo, nMaxGiocatori, nRobotPerGiocatore, condizioneFinePartita, upgradeInizialiSiONo)

Precondizioni: esiste un oggetto *Server S*, *robotdromo* è un testo che identifica un tipo di robotdromo, *nMaxGiocatori* e *nRobotPerGiocatore* sono numeri, *condizioneFinePartita* è uno fra **primo**, **primi3**, **tutti**, *upgradelInizialiSiONo* è **sì** oppure **no**.

Postcondizioni: viene creato un oggetto *Partita P* con attributi *nMaxGiocatori*, *nRobotPerGiocatore*, *condizioneFinePartita* e *upgradelInizialiSiONo* pari ai parametri omonimi dell'operazione, e *avviataSiONo* = **no**.

P viene associata al *Server S*.

Il *Manager M* crea *P*.

Viene creato un *Robodromo R* con *tipoRobodromo* pari al parametro *robotdromo*, e *P* ha *R*.

Vengono creati *n* (vedi **regole di business**) *Robot disponibili in P* con *nome* e *colore* come da **regole di business**.

avviaPartita()

Precondizioni: Esiste una partita *P* con *avviataSiONo* = **no**. i *partecipanti* di *P* sono almeno 2 e non più di *nMaxGiocatori* di *P*. Sul *Robodromo* di *P* *gareggiano* almeno 2 *Robot*. Ciascun *Giocatore* che *partecipa* a *P* *controlla* al massimo *nRobotPerGiocatore* di *P*.

Postcondizioni:

L'attributo *avviataSiONo* di *P* viene impostato a **sì**.

Se *upgradelInizialiSiONo* in *P* è pari a **sì**, l'attributo *upgrade* di ciascun *RobotInPartita* viene assegnato come da **regole di business**.

annullaPartita()

Precondizioni: Esiste una partita *P* con *avviataSiONo* = **no**.

Postcondizioni: Vengono eliminati: l'oggetto *P*, il suo *Robodromo* (con eventuali *Robot* che *gareggiano* su di esso), il suo *Server*, i *Robot disponibili su P*, le *Richieste di Partecipazione relative a P*, e tutte le altre associazioni di *P*.

Gestire richieste di partecipazione

accettaRichiesta(richiesta, robot)

Precondizioni: Esiste una *Partita P* con *partecipanti* in numero inferiore a $nMaxGiocatori$.

richiesta è una *Richiesta di Partecipazione* relativa a **P**, effettuata da un *Giocatore G*, con *parolaChiave* uguale alla *parolaChiave* del *Server* associato a **P**, e con *accettataSiONo* non assegnato.

robot è una lista di *Robot disponibili in P*.

Postcondizioni:

Ciascun *Robot Rb* nella lista *robot* gareggia sul *Robodromo R* associato a **P** con *dockDiPartenza* assegnato come da **regole di business** e **Rb** non è più *disponibile in P*.

Il *Giocatore G* *partecipa* a **P** e *controlla* ciascun *Robot* nella lista *robot*.

L'attributo *accettataSiONo* di *richiesta* viene impostato a **sì**.

rifiutaRichiesta(richiesta)

Precondizioni: Esiste una *Partita P* con *partecipanti* in numero inferiore a $nMaxGiocatori$.

richiesta è una *Richiesta di Partecipazione* relativa a **P**, effettuata da un *Giocatore G*, con *parolaChiave* uguale alla *parolaChiave* del *Server* associato a **P**, e con *accettataSiONo* non assegnato.

Postcondizioni:

L'attributo *accettataSiONo* di *richiesta* viene impostato a **no**.