

Progetto

PROVA FINALE  
(INGEGNERIA DEL SOFTWARE)

Anno accademico 2010/2011

# **L'isola dei dinosauri**

Versione 1.3

# Premessa

L'isola dei dinosauri è un adattamento, appositamente creato per il laboratorio del corso PROVA FINALE (INGEGNERIA DEL SOFTWARE), dell'engineering puzzle di Facebook "Dinosaur Island"<sup>1</sup>. Questo significa che il progetto presentato in questo documento prende spunto dal puzzle di Facebook per quanto riguarda l'idea generale e parte delle meccaniche di gioco, ma allo stesso tempo si discosta da esso sotto vari aspetti.

---

<sup>1</sup>[http://www.facebook.com/careers/puzzles.php?puzzle\\_id=19](http://www.facebook.com/careers/puzzles.php?puzzle_id=19)

# Introduzione

L'idea alla base dell'Isola dei dinosauri è piuttosto semplice: ogni utente deve gestire una specie di dinosauri, cercando di farla sopravvivere (evitando così un'estinzione prematura) e di farla crescere più possibile. Ciò permette all'utente di guadagnare punti e quindi di ambire a un ottimo piazzamento nella classifica.

L'ambiente di gioco (l'isola) è composto da un certo numero di caselle in cui:

- i dinosauri possono essere mossi o possono sostare;
- ci può essere della vegetazione;
- ci può essere una carogna;
- ci può essere acqua, che all'atto pratico rappresenta un terreno invalicabile.

L'isola, per definizione, è un terreno circondato da acqua.

Premessa: il gioco è a turni.

## Specifiche

In questa sezione del documento sono raccolte tutte le specifiche che devono essere rispettate nello svolgimento del progetto. Le specifiche sono suddivise in due categorie: quelle di gioco, che riguardano lo svolgimento della partita, l'ambiente di gioco e tutto ciò che lo popola, e quelle implementative, che invece stabiliscono l'architettura e un protocollo di comunicazione.

### Specifiche di gioco

Le specifiche di gioco sono fornite partendo dai dinosauri, dalla vegetazione e dalle carogne, cioè le componenti che possono essere trovate nell'ambiente di gioco. Seguono le specifiche riguardanti l'ambiente di gioco stesso e infine sono presentate quelle inerenti la partita.

#### I dinosauri

Esistono due tipologie di dinosauri:

- erbivori, che si nutrono di vegetazione;
- carnivori, che si nutrono di altri dinosauri oppure di carogne.

Una specie di dinosauri è interamente composta da dinosauri erbivori oppure carnivori, com'è ovvio che sia. Nell'isola possono coesistere più specie di dinosauri, ognuna controllata da un giocatore diverso. Ogni specie ha una durata massima di 120 turni, alla fine dei quali viene colpita da estinzione e il punteggio finale viene registrato.

**Caratteristiche.** Ogni dinosauro ha a disposizione una certa quantità di energia, che può incrementare cibandosi e può utilizzare per muoversi, crescere o deporre un uovo.

La dimensione ( $D$ ) di un dinosauro è rappresentata da un numero intero che va da 1 a 5 e la crescita del dinosauro corrisponde all'incremento di una unità di questo numero.

La massima energia ( $E_{\max}$ ) che un dinosauro può avere a disposizione è un valore intero dipendente dalla sua dimensione, che può essere calcolato semplicemente come  $1000 \cdot D$ . Un dinosauro di dimensione 1 ha quindi un'energia massima pari a 1000, per un dinosauro di

dimensione 5 invece è pari a 5000. L'energia di partenza di un dinosauro appena creato, sia per gli erbivori sia per i carnivori, è pari a 750.

Ogni dinosauro ha una vita di lunghezza limitata, compresa tra 24 e 36 ( $30 \pm 20\%$ ) turni, che viene casualmente scelta alla creazione del dinosauro stesso. L'età del dinosauro aumenta con il trascorrere del tempo di gioco, in particolare ogni volta che è il turno del giocatore che gestisce il dinosauro.

**Azioni.** Le azioni che un dinosauro può compiere, come anticipato sopra, sono 3. Sono elencate in tabella, con specificati i casi particolari e il costo in termini di energia.

Azione	Casi particolari	Energia necessaria
Movimento	I dinosauri erbivori possono spostarsi fino a 2 caselle di distanza (d), i carnivori fino a 3.	$E_m = 10 \cdot 2^d$
Crescita	Nessun dinosauro può crescere oltre il livello 5.	$E_c = E_{\max}/2 = 1000 \cdot D/2$
Deposizione uovo	Nessun dinosauro può deporre un uovo se è stato raggiunto il massimo numero di dinosauri per la specie (vedi in seguito).	$E_u = 1500$

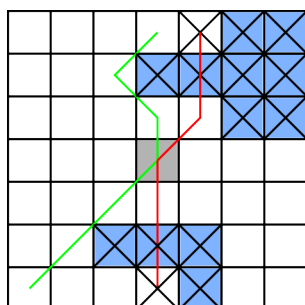
Si noti che la deposizione di un uovo non crea effettivamente un uovo in una casella; più semplicemente al turno successivo del giocatore viene creato un nuovo dinosauro in una casella libera scelta a caso più vicina possibile al luogo di deposizione dell'uovo.

Qualora un dinosauro tenti di compiere un'azione per cui non ha sufficiente energia a disposizione, non riesce completare l'azione e muore di fame.

**Visuale.** Ogni dinosauro vede una porzione di mappa attorno a se, di forma quadrata (eventualmente rettangolare ai bordi della mappa). La dimensione di quest'area dipende dalla dimensione del dinosauro stesso. Come mostrato nella figura a destra, ogni dinosauro di dimensione 1 vede fino a 2 caselle di distanza, ogni dinosauro di dimensione 2 o 3 vede fino a 3 caselle di distanza e infine ogni dinosauro di dimensione 4 o 5 vede fino a 4 caselle di distanza.

All'interno della propria visuale ogni dinosauro può sapere, per ogni casella, se essa è normale terra, acqua, se contiene vegetazione o una carogna oppure se occupata da un altro dinosauro. Automaticamente gli è nota la dimensione della vegetazione (e quindi la massima energia ricavabile se il dinosauro è erbivoro), l'energia ricavabile dalla carogna e la dimensione dell'altro dinosauro che occupa la casella, ma non l'energia che ha a disposizione.

4-5	4-5	4-5	4-5	4-5	4-5	4-5	4-5	4-5
4-5	2-3	2-3	2-3	2-3	2-3	2-3	2-3	4-5
4-5	2-3	1	1	1	1	1	2-3	4-5
4-5	2-3	1	1	1	1	1	2-3	4-5
4-5	2-3	1	1		1	1	2-3	4-5
4-5	2-3	1	1	1	1	1	2-3	4-5
4-5	2-3	1	1	1	1	1	2-3	4-5
4-5	2-3	2-3	2-3	2-3	2-3	2-3	2-3	4-5
4-5	4-5	4-5	4-5	4-5	4-5	4-5	4-5	4-5



**Movimento.** Un dinosauro, come specificato nella tabella soprastante, può muoversi fino a un certo numero di caselle di distanza. Il movimento non deve essere necessariamente rettilineo, cioè il dinosauro può muoversi in qualsiasi casella facente parte del quadrato di lato 5 caselle (se erbivoro) o 7 caselle (se carnivoro) che lo circonda. L'unica limitazione è che la casella di destinazione deve essere raggiungibile attraverso un percorso di massimo 2 caselle (per gli erbivori) o 3 caselle (per i carnivori) di cui non fanno parte caselle con

acqua. Nella figura, in cui le caselle blu sono quelle con l'acqua, sono mostrati due movimenti permessi (in verde) e due vietati (in rosso); le caselle barrate sono tutte quelle non raggiungibili da un dinosauro carnivoro. Si noti che il risultato di uno spostamento consentito è "la comparsa" del dinosauro nella casella desiderata, ma non c'è alcuna interazione con le caselle che si trovano tra quella di partenza e quella di arrivo. Un dinosauro erbivoro non può muoversi nella stessa casella in cui è già presente un altro dinosauro erbivoro, in quanto su ogni casella può essere presente un singolo dinosauro. Qualora sia coinvolto un dinosauro carnivoro il problema non si pone poiché avviene un combattimento, a cui sopravvive un singolo dinosauro.

**Combattimento.** Un dinosauro può muoversi in una casella già occupata da un altro dinosauro (a meno che siano entrambi erbivori). Quando ciò accade i due dinosauri combattono. Sopravvive un unico dinosauro, quello vincitore, sulla base della forza (F) di cui ognuno dei due dinosauri dispone. Tale forza, come mostrato nella tabella sottostante, varia a seconda della tipologia del dinosauro, dell'energia (E) che ha a disposizione e della sua dimensione (D).

Tipologia dinosauro	Forza per il combattimento
Carnivoro	$F=2 \cdot E \cdot D$
Erbivoro	$F=E \cdot D$

Se la forza dei due dinosauri che combattono è uguale, vince il dinosauro che ha attaccato, ovvero quello che ha compiuto il movimento sulla casella occupata dall'altro dinosauro.

Qualora vinca un dinosauro carnivoro, esso somma alla sua energia un quantitativo pari al 75% dell'energia della sua vittima. Nel caso vinca il dinosauro erbivoro, la sua energia rimane invariata.

## La vegetazione

Esiste un solo tipo di vegetazione, che funge da sostentamento per i dinosauri erbivori. In una casella dell'ambiente di gioco la vegetazione può essere presente oppure assente. Le caselle con vegetazione hanno un tetto massimo di crescita della vegetazione e quindi di energia che può guadagnare un dinosauro erbivoro che mangia tale vegetazione. Questo valore ( $E_{\max}$ ) è compreso tra 150 e 350 ( $250 \pm 40\%$ ) ed è scelto casualmente all'atto di creazione dell'ambiente di gioco per ogni singola casella con vegetazione. Quando un dinosauro erbivoro arriva su una casella con la vegetazione la mangia e somma tutta l'energia della vegetazione alla sua. Può rimanere una parte della vegetazione se il dinosauro raggiunge la massima energia possibile. La vegetazione ricresce aumentando la propria energia di un valore pari a  $E_{\max}/10$  per ogni turno di gioco (non riferito al singolo giocatore, ma alla partita in generale).

## Le carogne

Alcune caselle dell'ambiente di gioco possono contenere una carogna, in alternativa alla vegetazione. Le carogne forniscono un facile pasto ai dinosauri carnivori, che possono cibarsene spostandosi sulla casella contenente la carogna. L'energia massima fornita da una carogna ( $E_{\max}$ ) è compresa tra 350 e 650 ( $500 \pm 30\%$ ) e viene scelta al momento della sua creazione nell'ambiente di gioco. L'energia però decresce di  $E_{\max}/10$  per ogni turno di gioco (non riferito al singolo giocatore, ma alla partita in generale). Quando un dinosauro carnivoro arriva su una

casella con una carogna la mangia e somma tutta l'energia della carogna alla sua. Può rimanere una parte della carogna se il dinosauro raggiunge la massima energia possibile. Ogni volta che una carogna viene mangiata completamente oppure la sua energia arriva a zero viene creata un'altra carogna e posizionata in un punto casuale della mappa.

## L'ambiente di gioco

L'ambiente di gioco è rappresentato da una griglia di 40\*40 caselle. Le caselle, come già anticipato, possono essere di quattro tipi:

- terra semplice;
- terra con vegetazione;
- terra con carogna;
- acqua.

L'origine delle coordinate (x,y) corrisponde all'angolo in basso a sinistra della mappa.

La mappa può essere caricata da un apposito file oppure può essere creata in maniera casuale all'inizio della partita, seguendo questi criteri:

- la terra occupa l'80% dell'intero ambiente di gioco, l'acqua il restante 20%;
- l'isola non è partizionata, ovvero ogni casella di terra deve essere raggiungibile da tutte le altre;
- l'isola deve essere circondata da acqua su tutti i lati;
- le caselle di acqua all'interno dell'isola devono essere a gruppi da minimo 5 a massimo 15 elementi;
- le caselle con vegetazione devono essere il 40% di tutte le caselle di terra;
- le caselle con carogna devono essere 20.

Le specifiche qui fornite sono pensate con l'obiettivo di permettere una partita sufficientemente vivace ed equilibrata. Nulla vieta di implementare un ambiente di gioco parametrico, in cui le dimensioni e le altre opzioni possono essere scelte.

## La partita

La partita inizia con la creazione/caricamento dell'ambiente di gioco. La partita è persistente, ovvero rimane attiva anche in assenza di giocatori collegati, ovviamente congelando lo stato dell'ambiente di gioco e dei dinosauri in assenza di giocatori. Quando uno o più giocatori sono collegati, la partita procede per turni, il che significa che i giocatori possono effettuare le mosse desiderate a rotazione. La partita deve gestire correttamente il collegamento di un nuovo giocatore o la sua uscita dal gioco:

- se il giocatore entra per la prima volta in una partita, deve prima creare una specie assegnandole un nome e una tipologia; dopo che la specie è stata creata il suo primo dinosauro viene posizionato casualmente sulla mappa al momento del suo primo turno;
- se il giocatore lascia la partita viene salvata la posizione dei suoi dinosauri nell'ambiente di gioco e quindi i suoi dinosauri vengono rimossi;
- se il giocatore rientra in partita con una specie già avviata, i suoi dinosauri vengono riposizionati al momento del suo primo turno nelle posizioni salvate qualora queste non siano occupate da altri dinosauri, altrimenti in una casella libera scelta a caso tra quelle più vicine.

La partita è unica e deve poter essere salvata/ripristinata in caso di spegnimento/riavvio del server, così come il resto dello stato del server (es. classifica, utenti...).

**Giocatori.** Un giocatore può gestire una sola specie di dinosauri alla volta all'interno della partita. Creata una specie l'utente la gestisce fino alla sua estinzione e solo in seguito ha la possibilità di crearne una nuova. Il numero massimo di giocatori che possono contemporaneamente essere connessi a una partita è 8, mentre non è previsto un limite al numero totale di giocatori che partecipano alla stessa partita. Per partecipare alla partita ogni utente deve registrarsi con username e password.

**Turni.** In ogni turno un giocatore può compiere una sola azione di movimento, della lunghezza desiderata, per ognuno dei dinosauri a sua disposizione; inoltre, nello stesso turno, può anche compiere un'azione di crescita o alternativamente di deposizione uovo per ognuno dei suoi dinosauri. Il giocatore deve anche avere la possibilità di passare il turno senza aver compiuto tutte le azioni a sua disposizione.

Per evitare che la partita rimanga bloccata in quanto un giocatore, durante il suo turno, non fa alcuna azione e non passa il turno, è previsto un time-out di 30 secondi entro cui bisogna confermare la volontà di utilizzare il proprio turno e quindi un tempo di 2 minuti per poter eseguire le proprie azioni.

A ogni turno (proprio o degli altri) tutti i giocatori possono essere aggiornati sugli eventuali cambiamenti avvenuti nell'area di visibilità dei propri dinosauri.

**Classifica e punteggi.** Alla partita è associata una classifica, in cui sono presenti i punteggi relativi a tutte le specie di dinosauri create all'interno della partita, anche se già estinte. Il punteggio relativo a una specie viene aggiornato alla fine del suo turno di gioco, incrementandolo in base alla seguente regola: ogni dinosauro vale  $1+D$  punti, dove  $D$  è la sua dimensione.

## Specifiche implementative

Il progetto deve essere realizzato tramite un sistema client/server. Il server si occupa della gestione della partita e della logica di gioco. Il client deve permettere ai giocatori di partecipare alla partita, interagendo con il server e mettendo a disposizione un'interfaccia grazie a cui svolgere le diverse azioni possibili durante il gioco.

La logica applicativa deve risiedere sul server, i client devono solo mostrare agli utenti lo stato della partita e permettere loro di compiere le azioni di gioco.

Nella fase iniziale è consigliabile sviluppare un'interfaccia testuale con cui testare il funzionamento del sistema. L'interfaccia grafica potrà essere sviluppata in seguito, quando il resto del sistema avrà raggiunto un buon livello di stabilità.

Per il progetto è prevista l'implementazione della comunicazione tra client e server in tre modi differenti:

- locale;
- socket;
- RMI.

Nell'implementazione locale, client e server possono comunicare chiamando direttamente i rispettivi metodi. Nell'implementazione con i socket ogni chiamata a metodo viene sostituita dall'invio di un messaggio tramite socket secondo il protocollo definito nel seguito di questo documento. Infine nell'implementazione RMI le chiamate tra client e server sono remote e utilizzano RMI (Remote Method Invocation).

## Protocollo standard di comunicazione

Il protocollo tra client e server consiste nello scambio di messaggi. La sintassi generale per i comandi, che vanno da client a server è la seguente:

@comando,parametro1=x1,parametro2=x2,...,parametroN=xN

Si noti che la virgola viene utilizzata come separatore e che è presente un solo comando, mentre possono esserci zero o più parametri.

In senso opposto, cioè da server a client, viaggiano le risposte ai comandi oppure eventuali notifiche del server. In questo caso non è definibile una sintassi generale, ma si può dire che il messaggio inizia sempre con una parola chiave (identificata dalla @) seguita da eventuali altre parole chiave o dai dati che il server restituisce al client in risposta al comando ricevuto.

Di seguito sono presentati tutti e soli i messaggi del protocollo standard di comunicazione, divisi per categoria e con una breve spiegazione. Si noti che i parametri o i dati indicati con una lettera maiuscola devono essere sostituiti da un opportuno valore (es. lo username desiderato, il token restituito dal server...), mentre le lettere minuscole fanno parte del protocollo e quindi non possono essere modificate. Si noti inoltre che il carattere “|” è utilizzato come “oppure”, per segnalare una scelta (es. c|e vuol dire che ci deve essere c oppure e); ciò che è racchiuso tra parentesi quadre invece è opzionale, cioè può essere o meno presente, tranne nel caso delle mappe, in cui le parentesi quadre fanno parte del protocollo e servono a confinare il contenuto di una cella.

In ogni caso, qualora il server riceva un comando non riconosciuto, risponde con @comandoNonValido.

### Comandi fuori partita

**Creazione utente**, per creare un nuovo utente, con username (U) e password (P), con cui poi accedere al server; lo username viene utilizzato come identificativo e deve essere unico. Il server può confermare l'avvenuta creazione dell'utente oppure rispondere negativamente, specificando che lo username è già occupato se è il caso. Username e password devono essere alfanumerici.

client	@creaUtente,user=U,pass=P
server	@ok @no[,@usernameOccupato]

**Login**, per effettuare il login, identificandosi e autenticandosi. Il messaggio deve contenere username e password. Il server può confermare il login e inviare un token (T) univoco da utilizzare per tutte le successive comunicazioni, oppure rispondere negativamente, eventualmente specificando che l'autenticazione è fallita. Il token è una stringa e può contenere caratteri alfanumerici.

client	@login,user=U,pass=P
server	@ok,T @no[,@autenticazioneFallita]



**Creazione razza dinosauri**, per chiedere la creazione di una nuova razza di dinosauri e bisogna fornire il proprio token (T), il nome della razza (N) e il tipo, cioè se carnivora (c) oppure erbivora (e). Il server può confermare oppure rispondere negativamente, eventualmente segnalando che il nome è già stato usato oppure che è già presente una razza di dinosauri per il giocatore o ancora che il token inviato non è valido.

client	@creaRazza,token=T,nome=N,tipo=c e
server	@ok @no[,@nomeRazzaOccupato ,@razzaGiaCreata ,@tokenNonValido]

**Accesso partita**, per entrare nella partita. Il server può eventualmente segnalare che il limite massimo dei giocatori è stato raggiunto oppure che non è stata ancora creata una razza di dinosauri.

client	@accessoPartita,token=T
server	@ok @no[,@troppiGiocatori ,@razzaNonCreata ,@tokenNonValido]

**Uscita partita**, per abbandonare la partita.

client	@uscitaPartita,token=T
server	@ok @no[,@tokenNonValido]

**Lista giocatori**, per ottenere la lista degli utenti attualmente in partita. Il server restituisce una semplice lista, eventualmente vuota. L'asterisco indica che possono essere restituite zero o più tuple; le parentesi tonde non fanno parte del messaggio. Un esempio di risposta è @listaGiocatori oppure @listaGiocatori,pippo,pluto

client	@listaGiocatori,token=T
server	@listaGiocatori(,USERNAME) * @no[,@tokenNonValido]

**Classifica**, per ottenere la classifica generale. La classifica viene restituita per tuple, racchiuse tra parentesi graffe, il cui contenuto è, in ordine, lo username dell'utente, il nome della razza dei dinosauri, il punteggio e se tale razza è attualmente in partita (se sta ancora giocando oppure se si è estinta). Un esempio di risposta è @classifica oppure @classifica,{pippo,Trex,100,s},{pluto,stego,80,n}

client	@classifica,token=T
server	@classifica(,{USERNAME,RAZZA,PUNTEGGIO,s n}) * @no[,@tokenNonValido]

**Logout**, per effettuare il logout.

client	@logout,token=T
server	@ok @no[,@tokenNonValido]

### Comandi in partita (informazioni)

**Mappa generale**, per chiedere la mappa generale aggiornata. La mappa generale viene restituita fornendo una tupla contenente il numero di righe e colonne e poi le celle, divise per righe da un punto e virgola. La prima riga che deve essere restituita è quella più in basso (coerentemente con l'origine delle coordinate), l'ultima è quella più in alto. All'interno di una cella ci può essere acqua (a), terra (t), terra con vegetazione (v), buio (b), cioè mappa non ancora esplorata. Il server può anche rispondere negativamente, eventualmente segnalando che l'utente non è in partita. Un esempio di risposta, per una mappa 3x3, è:

@mappaGenerale,{3,3},[a][t][t];[a][v][t];[v][b][b];

In termini di coordinate (x,y) la mappa restituita dall'esempio è [0,0][1,0][2,0];[0,1][1,1][2,1];[0,2][1,2][2,2]; cioè si ottiene la mappa a destra.

v	b	b
a	v	t
a	t	t

client	@mappaGenerale,token=T
server	@mappaGenerale,{RIGHE,COLONNE},(( [a t v b] ) ^COLONNE; ) ^RIGHE @no[,@tokenNonValido ,@nonInPartita]

**Lista dinosauri**, per ottenere la lista degli identificatori dei propri dinosauri. Il server restituisce una semplice lista contenente gli identificatori dei propri dinosauri, che contiene almeno un dinosauro, altrimenti non si è in partita, e ciò è definito dal simbolo +.

client	@listaDinosauri,token=T
server	@listaDinosauri(,IDDINO)+ @no[,@tokenNonValido ,@nonInPartita]

**Vista locale**, per ottenere la vista locale dettagliata di uno dei propri dinosauri; bisogna passare l'id del dinosauro desiderato. Il server restituisce la vista locale partendo dalla tupla {X,Y} che fornisce la posizione dell'angolo in basso a sinistra della vista nella mappa (l'origine della mappa è sempre l'angolo in basso a sinistra), quindi c'è la tupla con il numero di righe e colonne della vista e poi le celle, divise per righe da un punto e virgola. In una cella ci può essere acqua (a), terra (t), terra con vegetazione (v,E), terra con carogna (c,E), un dinosauro (d,IDDINO), dove E è l'energia attuale di vegetazione e carogna, mentre IDDINO è l'id del dinosauro (sicuramente nella vista ci sarà l'id del proprio dinosauro). Anche in questo caso, così come nella mappa generale, l'ordine delle righe va dalla prima in basso all'ultima in alto. Il server può anche rispondere negativamente, eventualmente segnalando che l'id fornito non è valido (es. perché non esiste oppure perché fa riferimento a un dinosauro non nostro). Un esempio di risposta, per una vista 3x3, è:

@vistaLocale,{5,2},{3,3},[a][t][t];[a][d,12][t];[v,300][t][d,3];

client	@vistaLocale,token=T,idDino=I
server	@vistaLocale,{X,Y},{RIGHE,COLONNE},(( [a t v,E c,E d,IDDINO]) ^COLONNE;) ^RIGHE @no[,@tokenNonValido ,@idNonValido ,@nonInPartita]

**Stato dinosauro**, per ottenere lo stato di un proprio dinosauro oppure di un dinosauro altrui; bisogna passare l'id del dinosauro desiderato. Se l'id fa riferimento a un proprio dinosauro il server restituisce una lista di informazioni: lo username dell'utente, la razza, il tipo (c oppure e), la posizione del dinosauro sulla mappa (l'origine della mappa è l'angolo in basso a sinistra), la dimensione, l'energia e i turni vissuti; se il dinosauro è altrui vengono fornite tutte le informazioni tranne le ultime due solo se il dinosauro è nella visuale di uno dei propri dinosauri. Il server può rispondere negativamente, eventualmente segnalando che l'id non è valido (es. perché non esiste oppure perché fa riferimento a un dinosauro altrui fuori visuale). Due esempi di risposta sono:

```
@statoDinosauro,pippo,Trex,c,{5,2},3,2700,12
```

```
@statoDinosauro,pluto,stego,e,{4,4},4
```

client	@statoDinosauro,token=T,idDino=I
server	@statoDinosauro,USERNAME,RAZZA,c e,{X,Y},DIMENSIONE [,ENERGIA,TURNIVISSUTI] @no[,@tokenNonValido ,@idNonValido ,@nonInPartita]

## Comandi in partita (azioni)

**Movimento dinosauro**, per spostare un proprio dinosauro in una nuova posizione sulla mappa. Bisogna passare l'id dinosauro desiderato e le coordinate assolute della posizione in cui si vuole spostare il dinosauro. Il server può confermare il movimento, eventualmente segnalando un combattimento con l'esito (vinto o perso). Il server può anche rispondere negativamente, eventualmente segnalando che la destinazione non è raggiungibile oppure che non è possibile compiere altri movimenti con il dinosauro scelto in questo turno oppure la morte del dinosauro in quanto non si aveva sufficiente energia per l'azione oppure che non è il proprio turno.

client	@muoviDinosauro,token=T,idDino=I,dest={X,Y}
server	@ok[,@combattimento,v p] @no[,@tokenNonValido ,@idNonValido ,@destinazioneNonValida  ,@raggiuntoLimiteMosseDinosauro ,@mortePerInedia  ,@nonIlTuoTurno ,@nonInPartita]

**Crescita dinosauro**, per far crescere un proprio dinosauro. Bisogna passare l'id del dinosauro desiderato. Il server può confermare la crescita. Il server può anche rispondere negativamente, eventualmente segnalando che non è possibile compiere un'altra azione azione di crescita o deposizione uovo con il dinosauro scelto in questo turno oppure la morte per del dinosauro in quanto non si aveva sufficiente energia per l'azione.

client	@cresciDinosauro,token=T,idDino=I
server	@ok @no[,@tokenNonValido ,@idNonValido  ,@raggiuntoLimiteMosseDinosauro ,@mortePerInedia  ,@nonIlTuoTurno ,@nonInPartita]

**Deposizione uovo**, per creare un nuovo dinosauro nei pressi di un proprio dinosauro. Bisogna passare l'id del dinosauro desiderato. Il server può confermare la deposizione dell'uovo fornendo anche l'id del nuovo dinosauro, che sarà disponibile al turno successivo. Il server può anche rispondere negativamente, eventualmente segnalando che non è possibile compiere un'altra azione di crescita o deposizione uovo con il dinosauro scelto in questo turno oppure la morte per del dinosauro in quanto non si aveva sufficiente energia per l'azione o ancora che è stato raggiunto il limite massimo di dinosauri che si possono avere contemporaneamente.

client	@deponiUovo,token=T,idDino=I
server	@ok,IDDINO @no[,@tokenNonValido ,@idNonValido ,@mortePerInedia  ,@raggiuntoLimiteMosseDinosauro ,@nonIlTuoTurno  ,@raggiuntoNumeroMaxDinosauri ,@nonInPartita]

## Comandi in partita (turni)

**Conferma turno**, per confermare la volontà di utilizzare il turno (entro 30 secondi dalla notifica che è il proprio turno).

client	@confermaTurno,token=T
server	@ok @no[,@tokenNonValido ,@nonInPartita ,@nonIlTuoTurno]

**Passa turno**, per passare il turno prima della scadenza del tempo a disposizione (2 minuti).

client	@passaTurno,token=T
server	@ok @no[,@tokenNonValido ,@nonInPartita ,@nonIlTuoTurno]

## Notifiche in partita

**Cambio turno**, con cui il server segnala a tutti i giocatori (broadcast) che è cambiato turno e fornisce lo username del giocatore che è abilitato a fare le proprie mosse.

server	@cambioTurno,USERNAME
--------	-----------------------