# Guida dell’XML per VRGame

# Struttura base

<vrgame>

<settings>

<resources>

*... risorse del gioco ...*

</resources>

<objects>

*... instanziazione delle risorse del gioco ...*

</objects>

<states>

*... stati presenti nel gioco ...*

</states>

<camera>

*... informazioni per la camera ...*

</camera>

<user />

<connections>

*... informazioni di accesso a un server remoto ...*

</connections>

</settings>

<game>

<actions>

*... azioni del gioco ...*

</actions>

</game>

</vrgame>

## Risorse <resources>

<resource id=*""* src=*""* type=*""* />

Si definisce un parametro univoco **id**, il path della risorsa nel parametro **src** e il tipo della risorsa **type**. Quest’ultimo può essere *Image|Billboard|Particle |Mesh|Bot |Sound*, e sono caratterizzati da:

* **Image**: una texture su un piano;
* **Billoboard**: una texture su un piano che rivolge sempre la normale alla sua superficie verso la camera;
* **Particle**: un insieme di billboard con un comportamento definito in fase di instanziazione;
* **Mesh**: un modello tridimensionale;
* **Bot**: un modello tridimensionale con un comportamento definito in fase di istanziazione. I bot hanno anche i parametri **walk**, **wait** e **speak** che prendono una terna o una quaterna divisa da ‘:’ (due punti) col seguente formato:
  + walk=*"<int>:<int>:<float>(:<string>)"*
  + wait=*"<int>:<int>:<float>(:<string>)"*
  + speak=*"<int>:<int>:<float>(:<string>)"*

I valori espressi indicano rispettivamente il frame di inizio dell’animazione in questione (walk, wait o speak), il frame di fine, la velocità di scansione dei frame e un valore stringa che, se uguale a ‘r’, indica che la scansione segue l’algoritmo dell’ascensore rispetto a quello meramente crescente (quindi da inizio a fine e viceversa ripetuto, rispetto a inizio a fine ripetuto).

* **Sound**: un effetto sonoro di tipo wave.

## Oggetti <objects>

<obj id=*""* ref=*""* ... />

Gli oggetti sono istanze di risorse. Ogni oggetto alcuni parametri standard, altri hanno dei parametri dedicati. **id** indica l’identificatore univoco dell’oggetto, mentre **ref** è un riferimento alla risorsa che viene istanziata. È possibile creare più oggetti (istanze) di una singola risorsa. La tabella seguente indica tutti i parametri.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Tipo*  *Parametro* | *Image* | *Billboard* | *Particle* | *Mesh* | *Bot* | *Sound* | *Tipo di dato* |
| id | X | X | X | X | X | X | String |
| ref | X | X | X | X | X | X | String |
| position | X | X | X | X | X |  | Vector(3) |
| size | X | X | X | X | X |  | Vector(3) |
| scale | X | X | X | X | X |  | Vector(3) |
| normalize | X | X | X | X | X |  | Float |
| rotate | X | X | X | X | X |  | Float:Vector(3) |
| visible | X | X | X | X | X |  | Bool |
| locked | X | X | X | X | X |  | Bool |
| caption | X | X | X | X | X |  | String |
| num |  |  | X |  |  |  | Int |
| room\_checking |  |  |  | X |  |  | Bool |
| direction |  |  |  |  | X |  | Vector(3) |
| dispose |  |  | X |  |  |  | Vector(3) |

## Stati <states>

Gli stati definiscono l’elenco degli stati presenti nella macchina a stati interna al gioco. Il valore dello stato consentirà l’attivazione o la disattivazione delle azioni.

<state id=*""* enabled=*""* />

I parametri sono **id**, ossia l’indicatore univoco e **enabled**, ossia il valore dello stato all’avvio dell’applicazione.

## Camera <camera>

Definendo le impostazioni della camera è necessario inserire vari parametri, come mostrato di seguito:

|  |  |
| --- | --- |
| <position x=*""* y=*""* z=*""* /> | Indica la posizione iniziale della camera nelle coordinate spaziali |
| <direction x=*""* y=*""* z=*""* /> | Indica il vettore di direzione della camera nelle coordinate spaziali |
| <movements>  <keyboard/>  <mouse/>  </movements> | Indica i parametri di spostamento come tipi di input.  Keyboard, mouse e joystick prevedono informazioni su come usare queste periferiche e come interpretare gli input. |

Il nodo <movements> prevede quattro parametri: **walk** di tipo boolean, che se impostato a *true* indica se è possibile spostarsi solo sul piano XZ, **collision**, sempre di tipo boolean, che se impostato a *true* indica che verranno eseguite le collision detection con gli altri oggetti presenti nella scena tridimensionale e non sarà possibile attraversarli. Infine, i parametri facoltativi **step\_move**, di tipo float, che indica la velocità di un singolo passo nello spostamento della camera e **step\_angle**, sempre di tipo float, che indica l’angolo per cui la camera viene ruotata.

Nel sottoalbero contenuto in <movements> si indicano i vari input che arrivano da keyboard, mouse o joystick. I parametri sono indicati nella tabella sottostante:

|  |  |  |  |
| --- | --- | --- | --- |
| *Nodo* | *Parametro* | *Tipo* | *Significato* |
| keyboard | mouse | bool | Indica se è previsto l’uso del mouse per direzionale la camera |
| keyboard | forward | Virtual Key|Keyboard button | Input per spostamento in avanti |
| keyboard | back | Virtual Key|Keyboard button | Input per spostamento in indietro |
| keyboard | turn\_left | Virtual Key|Keyboard button | Input per girare la camera a sinistra |
| keyboard | turn\_right | Virtual Key|Keyboard button | Input per girare la camera a destra |
| keyboard | Left | Virtual Key|Keyboard button | Input per spostamento a sinistra |
| keyboard | Right | Virtual Key|Keyboard button | Input per spostamento a destra |
| keyboard | Run | Virtual Key|Keyboard button | Input per abilitare la modalità corsa (spostamento più veloce) in combinazione con i tasti spostamento |
| keyboard | Freeze | Virtual Key|Keyboard button | Attiva/disattiva lo spostamento della camera |
| keyboard | Updown | Virtual Key|Keyboard button | Input per abilitare la modalità spostamento verso l’alto e verso il basso in combinazione, rispettivamente, con i tasti spostamento forward e back |
| mouse | Mouse | Int (1 o 2) | Imposta lo spostamento della camera secondo i metodi predefiniti CAMERA\_INPUT\_MOVE\_MOUSE\_MBL\_BS se impostato a 1 o CAMERA\_INPUT\_MOVE\_MOUSE\_MBL\_LC se impostato a 2 |
| none |  |  | La camera rimane ferma |
| joystick |  | TODO |  |

## User <user>

<user required=*""* password=*""* role=*""* background=*""*/>

Il campo user prevede l’inserimento di una user-login. Per default, se questo campo non viene dichiarato, come parametri prevede quelli indicati nella tabella sottostante

|  |  |  |
| --- | --- | --- |
| *Parametro* | *Tipo* | *Significato* |
| Required | Bool | Se true, indica che è richiesta una user-login all’inizio del gioco, false altrimenti |
| Password | Bool | Se true, indica che viene richiesta una password durante la user-login, false altrimenti |
| Role | Bool | Se true, indica che viene richiesta una differenziazione tra ruoli nel gioco, che viene richiesta dopo l’operazione di user-login |
| Background | String | L’id della risorsa di tipo Image da utilizzare come sfondo nelle operazioni di user-login e di scelta del ruolo |

## **Connessioni remote** <connections>

Questo campo serve per stabilire le connessioni con il server, e prevede i seguenti campi e parametri:

<connections>

<connection id=*""* url=*""* ip\_address=*""* server=*""*>

<login path=*""* password=*""* msg=*""* error=*""* param=*""*/>

<roles path=*""* msg=*""*>

<role name=*""* genre=*""* default\_icon=*""* checked\_icon=*""* avatar=*""*>

<state ref=*""* enabled=*""*/>

</role>

...

</roles>

<chat path=*""* callback=*""* />

<logout path=*""* />

</connection>

</connections>

|  |  |  |  |
| --- | --- | --- | --- |
| *Nodo* | *Parametro* | *Tipo* | *Significato* |
| connection | Id | String | Identificatore univoco della connessione |
| connection | url | String | Server URL per la connessione |
| connection | ip\_address | String | Indirizzo IP del server per la connessione |
| connection | Server | String | Nome del server |
| login | Path | String | Path della pagina per eseguire la login |
| login | Password | Bool | True se la password è richiesta, false altrimenti |
| login | Msg | String | Messaggio di richiesta di login |
| login | Error | String | Messaggio di errore in caso di login scorretta |
| login | Param | String | Parametro aggiuntivo che viene inviato al server |
| roles | Path | String | Path della pagina per impostare un ruolo |
| roles | Msg | String | Messaggio di richiesta di scelta di un ruolo |
| role | Name | String | Nome del ruolo |
| role | Genre | string (m o f) | Genere del ruolo |
| role | default\_icon | String | Riferimento alla risorsa di tipo Image per l’icona non ancora scelta |
| role | checked\_icon | String | Riferimento alla risorsa di tipo image per l’icona una volta che è stata scelta |
| role | Avatar | String | Riferimento alla risorsa di tipo Bot legata al ruolo |
| chat | Path | String | Path della pagina per inviare messaggi chat |
| chat | Callback | String | Path della pagina per scaricare i messaggi chat, richiamata da callback |
| logout | Path | String | Path della pagina per eseguire una user-logout |

I ruoli prevedono degli stati: quando viene scelto un ruolo rispetto che un altro, viene impostato il valore definito, analogamente alla reaction statechange (vedi in seguito).

# Il gioco

<game>

<actions>

...

</actions>

</game>

Il gioco vero e propri è racchiuso nei tag game, ed è composto da un insieme di azioni che possono essere eseguite anche simultaneamente. Ogni azione viene attivata se le sue condizioni sono verificare. Le condizioni sono riferite all’abilitazione o meno di stati del gioco.

Azione <action>

Una singola azione può avere o meno un effetto visivo, ma in ogni caso genera una o più reazioni. Alcune azioni che prevedono una “verifica” possono prevedere la generazione di reazioni se non vengono completate correttamente.

<action id=*""* type=*""* subject=*""* objects=*""*>

<conditions>

<state ref=*""* enabled=*""* />

...

</conditions>

<reactions object=*""*>

...

</reactions>

...

<reactions object=*""*>

...

</reactions>

<unsuccess>

...

</unsuccess>

<guide ref=*""* position=*""*/>

</action>

Un’azione prevede i parametri **id** (identificatore univoco dell’azione), **type** (il tipo dell’azione), **subject** (-opzionale- l’id dell’oggetto scatenante l’azione) e **objects** (-opzionale- una lista di oggetti su cui l’azione può avere effetto, separati da “;”) e altri parametri.

I tipi di azione previsti sono indicati nella tabella sottostante

|  |  |  |
| --- | --- | --- |
| *Tipo* | *Parametri aggiuntivi* | *Descrizione* |
| Hover | subject | Riconosce la presenza del cursore sopra all’oggetto **subject**. |
| Click | subject | Riconosce l’azione di click sull’oggetto **subject**. |
| pick\_and\_release | subject, objects | Riconosce l’azione di pick e trascinamento dell’oggetto **subject** su un altro oggetto **object**, e quindi rilasciamento. Prevede azioni in caso di non rilascio sull’**object**. |
| clock | seconds, post\_verify | Prevede l’azione di inizio di un timer, di durata (in secondi) specificata dal parametro **seconds**. A fine timer se **post\_verify** è impostato a true, vengono verificate nuovamente le condizioni di inizio del timer per attivare le reazioni.  Di default, **post\_verify** è impostato a true. |
| sleep | seconds, post\_verify | Prevede l’azione di inizio di un timer bloccante per quanto riguarda gli input da utente, di durata (in secondi) specificata dal parametro **seconds**. A fine timer se **post\_verify** è impostato a true, vengono verificate nuovamente le condizioni di inizio del timer per attivare le reazioni.  Di default, **post\_verify** è impostato a true. |
| enter\_room | room | Prevede l’azione di ingresso in un’area dello spazio tridimensionale con etichetta **room**.  Questo parametro è la descrizione interna alla mesh istanziata come oggetto col parametro room\_checking impostato a true. |
| exit\_room | room | Prevede l’azione di uscita da un’area dello spazio tridimensionale con etichetta **room**.  Questo parametro è la descrizione interna alla mesh istanziata come oggetto col parametro room\_checking impostato a true. |
| dialogue | objects | Prevede l’azione di dialogo con elementi nello spazio tridimensionale, specificati in **objects**. |
| chat |  | Attiva una chat (vedi paragrafo connessioni Remote). |
| move\_bot | subject, destination, speed | Prevede l’azione di spostamento di un oggetto **subject** di tibo “Bot” in una destinazione **destination** alla velocità **speed**. |
| navigate | url | Prevede l’azione di apertura di un browser interno al gioco alla pagina web **url**. |
| read | resource\_ref | Prevede l’azione di apertura di un immagine di tipo “Billboard”.  Il parametro **resorce\_ref**, a differenza dei parametri objects e subject, necessita del riferimento all’id della RISORSA billboard che si desidera leggere. |

I sottoalberi **conditions**, **reactions** e **unsuccess** indicano rispettivamente:

* le condizioni con le quali l’azione viene attivata, e contengono riferimenti a stati predefiniti,
* le reazioni che vengono scatenate su ogni oggetto (contiene il parametro object, che indica l’id dell’oggetto su cui la reazione ha effetto)
* le reazioni che vengono scatenate in cado di insuccesso nell’eseguire l’azione.

La guida **guide** indica, per l’azione attiva, una billboard che può essere attivata in caso di aiuto premendo il tasto “h”. I parametri ref e position indicano rispettivamente il riferimento alla RISORSA (non all’oggetto!) Billboard che descrive l’azione.

## Le reazioni

# Parametri e messaggi http-GET

Sono previsti diversi tipi di messaggi al server, tutti di tipo get, per gestire in tracking delle chat tra più utenti. In particolare, al momento si possono inviare i seguenti messaggi con i seguenti parametri:

|  |  |  |  |
| --- | --- | --- | --- |
| *Messaggio* | *Parametro 1* | *Parametro 2* | *Parametro 3* |
| Login – esegue la login su server all’inizio del gioco, se richiesto | login | [password] | [param] \* |
| Scelta del ruolo – esegue la scelta del ruolo all’inizio del gioco, se necessario | login | genre=m/f \*\* | role |
| Callback su server – esegue le callback sul server per aggiornare la chat a livello locale e aggiorna la posizione dell’avatar | login | position |  |
| Chat – avvia la chat e invia un messaggio nella chat | login | msg | [wlc=1/0]\*\*\* |
| Exit – esce dalla chat | login |  |  |

\* indica se deve essere inviato un altro parametro generico restrittivo per la login  
\*\* indica se il genere del ruolo è maschile (‘m’) o femminile (‘f’)  
\*\*\* indica se è un messaggio di avvio della chat (‘1’) o un semplice messaggio (‘0’). wlc sta ad indicare *welcome*









