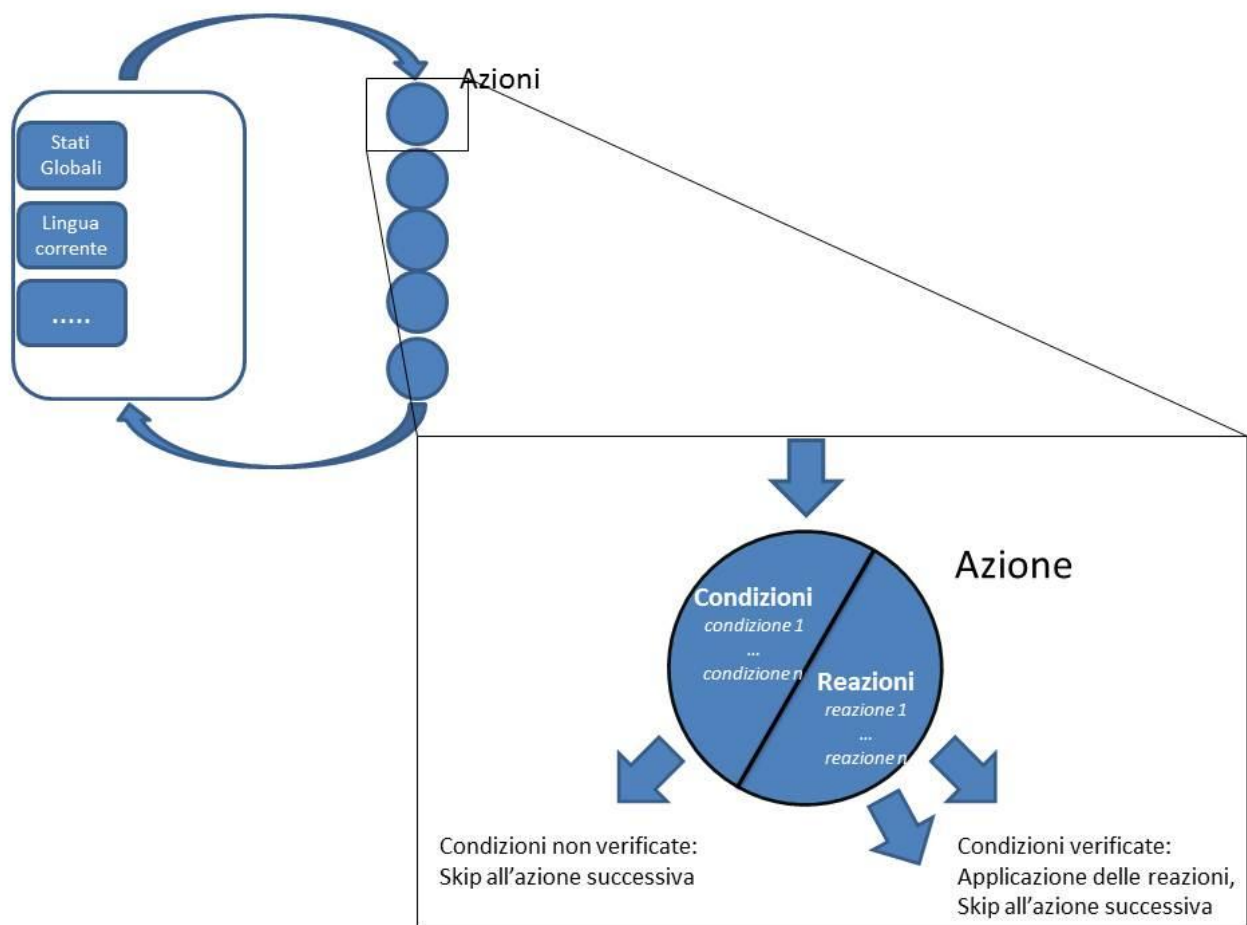


Schema generale

```
<vrgame>
  <settings>
  </settings>

  <game>
    <actions>
    </actions>
  </game>
</vrgame>
```

Schema di Funzionamento



Settings

Scorciatoie

```
* <shortcuts>
  <shortcut action="" key="" />
</shortcuts>
```

shortcut

definisce le azioni per un singolo shortcut

action = tipo di azione: **help** | **split_language** | **reset**
key = un qualche tasto della tastiera (anche **VK** di XVR)

Risorse

```
<resources>
  <resource id="" src="" type="" centerbb="" speak="" wait="" walk=""/>
  ...
  <resource id="" src="" type="" centerbb="" speak="" wait="" walk=""/>
</resources>
```

resource

definisce i parametri di ogni singola risorsa

id = identificatore univoco della risorsa
src = url del file risorsa (oppure tipo della mesh procedurale di XVR: sono previsti **VRP_BOX**, **VRP_SPHERE**, **VRP_COLIMN**, **VRP_CONE**, **VRP_PYRAMID**, **VRP_CYLINDER**, **VRP_CONEFRUSTUM**, **VRP_CROSSVAULT**, **VRP_PRISM**, **VRP_POLYGONAL_PYRAMID**, **VRP_POLYGONAL_FRUSTUM**)
type = tipo di risorsa: **Image** | **Billboard** | **Particle** | **Mesh** | **Bot** | **Sound** | **Text**
centerbb = indica se la bounding box della risorsa deve essere centrata sulla risorsa stessa (solo per risorse di tipo **Mesh** e **Bot**): **true** | **false** (default: **false**)
speak = terna indicante il range di frame per l'animazione **speak** (da **n** a **m**), la velocità di alterazione (**v**), eventuale quarto elemento che indica se l'animazione deve funzionare in modo solo in modo crescente o crescente e decrescente ecc... (solo per risorse di tipo **Bot**): **n:m:v:(r)** (es: 0:2:0,5 oppure 0:2:0,5:r).
wait = terna indicante il range di frame per l'animazione **wait** (da **n** a **m**), la velocità di alterazione (**v**), eventuale quarto elemento che indica se l'animazione deve funzionare in modo solo in modo crescente o crescente e decrescente ecc... (solo per risorse di tipo **Bot**): **n:m:v:(r)** (es: 0:2:0,5 oppure 0:2:0,5:r).
walk = terna indicante il range di frame per l'animazione **walk** (da **n** a **m**), la velocità di alterazione (**v**), eventuale quarto elemento che indica se l'animazione deve funzionare in modo solo in modo crescente o crescente e decrescente ecc... (solo per risorse di tipo **Bot**): **n:m:v:(r)** (es: 0:2:0,5 oppure 0:2:0,5:r).

Introduzione

```
* <intro src="" skip=""/>
```

src = .avi file source
skip = id della risorsa immagine per il bottone skip (tipo risorsa: **Billboard** o **Image**)

Supporto multilingua

```
* <languages background="">
  <language name="" alt="" icon="" caption="">message</language>
  ...
  <language name="" alt="" icon="" caption="">message</language>
</languages>
```

background = id della risorsa immagine per il background di inizio (tipo risorsa: Billboard o Image)

language

Informazioni per la definizione di una lingua abilitata per il gioco

name = nome della lingua
alt = nome alternativo della lingua
icon = id della risorsa immagine per il bottone della lingua (tipo risorsa: Billboard o Image)
caption = breve testo di descrizione della lingua (default: stesso valore di name)
message = messaggio che viene mostrato quando viene cambiata la lingua attraverso le shortcuts

Stati

```
<states>
  <state id="" value="" />
  ...
  <state id="" value="" />
</states>
```

state

definizione di ogni singolo stato

id = id univoco dello stato
value = valore intero iniziale dello stato (default: 0)

Camera

```
<camera>
  <position x="" y="" z="" />
  <direction x="" y="" z="" />
  <movements walk="" collision="" height="" depth="" step_move="" step_angle="">
  * <keyboard mouse="" forward="" back="" turn_up-down="" turn_left="" turn_right="" />
  * <mouse mouse="" />
  </movements>
</camera>
```

position

indica le coordinate della camera all'inizio del gioco

x = coordinata X
y = coordinata Y
z = coordinata Z

direction

indica la direzione camera sulle coordinate spaziali all'inizio del gioco

x = direzione sulla coordinata X

y = direzione sulla coordinata Y
z = direzione sulla coordinata Z

movements

indica i tipi di movimenti che si possono fare nel gioco

collision= se deve essere attiva la collision detection: **true** | **false** (default: **false**)
walk = se è possibile spostarsi solo a una certa altezza: **true** | **false** (default: **false**)
height = la coordinata Y supplementare in a cui si verifica la collision detection
depth = la distanza per verificare la collision detection
step_move= il delta di movimento in una direzione
step_angle= il delta dell'angolo di rotazione

keyboard

Definisce tutti i comandi da keyboard per lo spostamento

mouse = abilita lo spostamento di tipo mouse e tastiera invece di solo tastiera:
true | **false** (default: **false**)
forward = pulsante della tastiera per far avanzare la camera (anche **VK** di XVR)
back = pulsante della tastiera per far indietreggiare la camera (anche **VK** di XVR)
left = pulsante della tastiera per far spostare la camera a sinistra (anche **VK** di XVR)
right = pulsante della tastiera per far spostare la camera a destra (anche **VK** di XVR)
turn_left= pulsante della tastiera per ruotare la camera a sinistra (anche **VK** di XVR)
turn_right= pulsante della tastiera per ruotare la camera a destra (anche **VK** di XVR)
updown = pulsante della tastiera per far spostare in alto o in basso la camera in combinazione con i tasti forward e back rispettivamente (anche **VK** di XVR)
turn_up-down= pulsante della tastiera per far ruotare in alto o in basso la camera in combinazione con i tasti forward e back rispettivamente (anche **VK** di XVR)
run = pulsante della tastiera per far spostare più velocemente (con un moltiplicatore di 2 volte e mezzo) la camera in combinazione con i tasti forward e back rispettivamente (anche **VK** di XVR)
freeze = solo per spostamento di tipo mouse e tastiera, pulsante che blocca lo spostamento da congela gli input da tastiera, permettendo di far ruotare la camera usando il mouse (anche **VK** di XVR)

mouse

attiva lo spostamento attraverso la combinazione mouse e tasto del mouse

mouse = indica il tipo di funzione da usare per lo spostamento via mouse: 1 | 2 (rispettivamente **CAMERA INPUT MOVE MOUSE MBL BS** o **CAMERA INPUT MOVE MOUSE MBL LC** degli spostamenti standard di camera di XVR)

Opzioni video

```
* <video stereo="" eye_separation="" screen_size="" >  
  <tracker head="" position="" />  
  <scene fov="" near="" far="" />  
</video>
```

stereo = opzione che indica il tipo di stereo da usare, attivo, passivo o disabilitato: **active** | **passive** | **disable** (default: **disable**)
eye_separation= separazione degli occhi per la visualizzazione stereo (default: 0.08)
screen_size= vettore che indica la dimensione dello schermo (default: [2,1.5])

tracker

impostazioni del tracker

head = valore della testa del tracker (default: 1)
position= vettore di posizione del tracker (default: [0,0,2])

scene

impostazioni della scena

fov = fov della scena (default: 75)
near = near della scena (default: 0.05)
far = far della scena (default: 100)

User information

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

required= indica se la login dell'utente è richiesta: true | false (default: false)

password= indica se la login necessita della richiesta di password: true | false
(default false)

role = indica se è richiesta una divisione per ruoli: true | false (default: false)

background= id della risorsa immagine per il background di inizio (tipo risorsa: Billboard o Image)

Gestione delle connessioni

```
* <connections>
  <connection id="" url="" ip_address="" server="">
    <login path="" msg="" error="" param=""/>
    <roles path="" msg="">
      <role name="" genre="" default_icon="" checked_icon="">
        <avatar ref="" normalize="" />
        <state ref="" value=""/>
        ...
        <state ref="" value=""/>
      </role>
      ...
      <role name="" genre="" default_icon="" checked_icon="" >
        ...
      </role>
    </roles>
    <chat path="" callback="" />
    <logout path="" />
  </connection>
</connections>
```

connection

impostazioni di connessione

id = identificatore univoco della connessione
url = url del server
ip_address= indirizzo IP del server
server = il nome del server

login

informazioni per le operazioni di login

path = il path di accesso per l'operazione login (verrà concatenato automaticamente all'url del server)
msg = il messaggio per l'utente di richiesta di dati
error = il messaggio per l'utente di risposta in caso di connessione errata
param = un parametro aggiuntivo, se necessario

roles

insieme di informazioni per i ruoli che possono essere scelti nel gioco, per la sessione in corso

path = il path di accesso per l'operazione di impostazione del ruolo (verrà concatenato automaticamente all'url del server)
msg = il messaggio per l'utente di richiesta di dati (usare #name# per ottenere lo username, es: "Ciao #name#, scegli il ruolo con cui vuoi giocare.")

role

informazioni per la definizione di un ruolo disponibile

name = il nome del ruolo
genre = il genere del ruolo: m | f rispettivamente maschile e femminile. La definizione del genere è utile per la traduzione automatica delle azioni
default_icon= id della risorsa immagine per l'icona del ruolo non selezionata (tipo risorsa: Billboard o Image)
checked_icon= id della risorsa immagine per l'icona del ruolo selezionata (tipo risorsa: Billboard o Image)

avatar

informazioni in merito a un avatar tridimensionale per un ruolo

ref = id della risorsa immagine per l'avatar del ruolo (tipo risorsa: Bot)
normalize= valore per la normalizzazione dell'avatar, se necessaria

state

informazioni per l'inizializzazione degli stati quando viene scelto un ruolo

ref = id dello stato da modificare nel caso in cui venga scelto questo ruolo
value = valore da assegnare allo stato per questo ruolo (default: 0)

chat

informazioni per le operazioni di chat

path = il path di accesso per l'operazione di invio della chat (verrà concatenato automaticamente all'url del server)
callback= il path di accesso per l'operazione di callback (verrà concatenato automaticamente all'url del server)

logout

informazioni per le operazioni di logout

path = il path di accesso per l'operazione di Logout (verrà concatenato automaticamente all'url del server)

Oggetti

```
<objects>
  <obj id="" ref="" ... />
  ...
```

```

<obj id="" ref="" ...>
  <description lang="">message</description>
  ...
  <description lang="">message</description>
</obj>
<obj id="" ref="" ... />
</objects>

```

Gli oggetti sono istanze di risorse, e vengono mostrati nella scena. Raramente vengono visualizzate le risorse non istanziate. Le istanze delle risorse possono essere di tipo Sound, Text, Image, Billboard, Particle, Mesh o Bot, analogamente alle risorse, anche se non deve essere specificato.

obj

Definizione dei singoli oggetti

id = identificatore univoco dell'istanza
ref = id della risorsa da istanziare
lang = il nome (o il nome alternativo) della lingua in cui questo oggetto è abilitato (solo oggetti di tipo Sound), opzionale. Se indicato, l'audio sarà abilitato solo quando è attiva la lingua
visible = indica se l'oggetto deve essere visibile al momento del caricamento della scena: **true** | **false** (default: **false**). (solo per oggetti che vengono posizionati sulla scena, quindi tutti eccetto quelli di tipo Sound)
position = vettore che indica la posizione sulla scena al momento del caricamento (solo per oggetti che vengono posizionati sulla scena, quindi tutti eccetto quelli di tipo Sound)
size = vettore di 2 o 3 elementi che indica la dimensione sulla scena al momento del caricamento (solo per oggetti che vengono posizionati sulla scena, quindi tutti eccetto quelli di tipo Sound. Per oggetti di tipo Image o Billboard si ha un vettore di 2 elementi)
scale = (vedi size)
normalize = valore per la normalizzazione dell'oggetto (solo per oggetti che vengono posizionati sulla scena, quindi tutti eccetto quelli di tipo Sound)
locked = indica se l'oggetto può essere spostato: **true** | **false**, rispettivamente non può essere spostato oppure può esserlo (default: **false**)
num = numero di particelle da istanziare (solo per oggetti di tipo Particle)
dispose = vettore di 3 elementi indicante l'eventuale range di disposizione delle particelle nella posizione iniziale (solo per oggetti di tipo Particle) (default: [0,0,0])
room_checking = parametro che indica se l'oggetto può essere utilizzato per geolocalizzare la camera: **true** | **false** (default: **false**). La geolocalizzazione utilizza i nomi dei subset dell'oggetto (solo per oggetti di tipo Mesh)
rotate = parametro che indica la rotazione dell'oggetto (solo per oggetti di tipo Image, Mesh o Bot) nella forma angolo:[vettore degli assi di rotazione]. Esempio: 90:[0,1,0] ruota di 90 gradi attorno all'asse Y.
direction = parametro che indica il vettore di direzione dell'oggetto (solo per oggetti di tipo Bot)
ico = id della risorsa per l'icona dell'oggetto (solo per oggetti di tipo Bot)
caption = testo etichetta dell'oggetto (solo per oggetti di tipo Bot)

description

Un insieme di descrizioni per ogni lingua abilitata. Di default le descrizioni sono vuote

lang = il nome (o il nome alternativo) della lingua a cui si riferisce questa descrizione
message = il testo della descrizione

Game

Il gioco è definito da un insieme di azioni, che vengono abilitate in base a determinate condizioni. Ogni azione, quando viene eseguita, ha un numero definito di reazioni.

```
<game>
  <actions>
    <action id="" type="" subject="" objects="" ...>
      <conditions>
        ...
      </conditions>
      <reactions object="">
        ...
      </reactions>
      * <unsuccess>
        ...
      </unsuccess>
      * <dialogue>
        ...
      </dialogue>
    </action>
    ...
    <action>
      ...
    </action>
  </actions>
</game>
```

Azioni

Il gioco è basato su un sistema azione-reazione. L'azione definisce l'operazione base del gioco, può essere generata in caso di determinate condizioni e può scatenare una o più reazioni.

id = identificatore dell'azione
type = indica il tipo dell'azione. I tipi possono essere: **hover** (rileva il passaggio del mouse su un oggetto) | **click** (rileva il click sinistro del mouse su un oggetto) | **click_left** (rileva il click sinistro del mouse su un oggetto) | **click_right** (rileva il click destro del mouse su un oggetto) | **pick_and_release** (rileva il doppio click sinistro del mouse su un oggetto, che viene preso. Al successivo doppio click viene rilasciato) | **clock** (attiva un timer) | **sleep** (attiva un timer e blocca l'applicazione) | **enter_room** (rileva l'ingresso in un'area - azione di geolocalizzazione) | **exit_room** (rileva l'uscita da un'area - azione di geolocalizzazione) | **move_bot** (muove un bot in una posizione) | **read** (apre un'immagine e la porta in primo piano) | **read_text** (legge un file di testo) | **dialogue** (attiva un dialogo con un bot) | **chat** (attiva una chat - vedi anche Gestione delle connessioni) | **navigate** (apre un browser all'interno del gioco)
subject = indica l'id dell'oggetto che scatena l'azione
objects = indica l'insieme di id degli oggetti che possono subire l'azione (separati da ':')
color = un vettore indicante colore in formato RGB o RGBA, con valori da 0 a 1 (solo per azioni di tipo **pick_and_release**). Indica il colore che assume un oggetto su cui un oggetto può essere rilasciato. Si attiva soltanto se per quell'oggetto è definita un'azione di tipo **hover**
seconds = un numero di secondi (solo per azioni di tipo **clock** o **sleep**)

post_verify= *verifica le condizioni sia all'inizio dell'intervallo di tempo che alla fine: true | false (default: true) (solo per azioni di tipo clock)*
room = *il nome della stanza per la geolocalizzazione della camera (solo per azioni di tipo enter_room o exit_room)*
destination= *un vettore di 3 elementi che indica la posizione da raggiungere nello spazio tridimensionale (solo per azioni di tipo move_bot)*
speed = *la velocità con cui un bot deve raggiungere una destinazione (solo per azioni di tipo move_bot)*
url = *una URL web da aprire con il browser interno al gioco (solo per azioni di tipo navigate)*
resource_ref= *l'id della risorsa che viene letta (solo per azioni di tipo read)*
lang = *il nome (o il nome alternativo) della lingua per la quale è attiva l'immagine dal leggere (solo per azioni di tipo read)*

Condizioni

Le condizioni sono un insieme di controlli che attivano o meno i controlli per il successo delle azioni. Le condizioni vengono verificate sugli stati o sulla lingua attiva. Se le condizioni per un'azione non sono definite, l'azione è sempre attiva. Perché un'azione si attivi, tutte le condizioni devono essere verificate (in altre parole, sono verificate in AND)

```
* <conditions>
  * <state ref="" value="" op="" />
  * <language name="" />
</conditions>
```

state

verifica la condizione su uno stato

ref = l'id dello stato da verificare
value = il valore con cui eseguire il controllo
op = l'operazione con cui viene fatto il controllo: **equals** | **minor** | **major** | **different** | **minorequals** | **majorequals** (default: equals)

language

verifica la condizione sulla lingua

name = il nome (o il nome alternativo) della lingua per la quale è attiva l'azione

Dialoghi

Le azioni di dialogo consentono di intraprendere un dialogo con un bot. Il dialogo è formato da frasi singole e da domande: le frasi singole possono essere dette dal bot oppure dal personaggio, mentre le domande vengono poste dal bot al personaggio. Le risposte alle domande possono prevedere una risposta singola e, eventualmente, un nuovo dialogo, che viene generato ricorsivamente. Le risposte possono prevedere anche delle reazioni (vedi il paragrafo delle Reazioni).

```
<dialogue>
  * <sentence from="">message</sentence>
  ...
  * <sentence>message</sentence>
  ...
  * <question from="">
    <ask>question text</ask>
    <answer>
      <text>answer text</text>
    <dialogue>
      ...
```

```

    </dialogue>
    <reactions>
        ...
    </reactions>
</answer>
<answer>
    ...
</answer>
</question>
    ...
</dialogue>

```

Sentence

Una sentence è una frase singola.

from = l'id del bot che dice la frase. Se non è indicato, è il personaggio usato dall'utente a parlare

message = il testo della frase. Il sistema riesce a tradurre automaticamente una frase per il genere dell'utente. Ad esempio userà il termine "entrata" invece di "entrato" se il genere dell'utente è femminile. Per usare questa funzionalità si può usare il sistema veloce o quello per sostituzione. Il sistema veloce prevede l'inserimento dell'asterisco '*' al posto del carattere da sostituire con 'a' o con 'o': per continuare l'esempio precedente si userà il termine "entrat*". Il sistema per sostituzione è utile per composizioni più complesse, e pone l'alternativa racchiusa dai caratteri '##' e separati dalla barra '/': sempre con lo stesso esempio si può avere "#entrato/entrata#", o, per termini più complessi come "#studente/studentessa# o con parti di frasi. Le due costruzioni di seguito sono identiche: "#lo studente e' entrato/la studentessa e' entrata#", "l* #studente/studentessa# e' entrat*". È sconsigliato usare caratteri accentati come "ò,à,è,è,ì" mentre è preferibile usare l'apice.

Question

Una question è una domanda a cui sono previste varie alcune risposte chiuse.

from = l'id del bot che dice la frase. Se non è indicato, è il personaggio usato dall'utente a parlare

ask

Una domanda.

question text= il testo della domanda. Valgono le regole di traduzione per genere come per le sentence (vedi il paragrafo dedicato)

answer

Una risposta alla domanda. Per ogni domanda in genere ce ne sono più di una.

text

answer text= il testo della risposta. Valgono le regole di traduzione per genere come per le sentence (vedi il paragrafo dedicato)

dialogue

Un nuovo dialogo che inizia con la risposta.

reactions

Vedi paragrafo sulle reazioni.

Reazioni

Le reazioni sono il fulcro del gioco. Possono essere racchiusi in due tag differenti: `reactions` e `unsuccess`. Le reazioni racchiuse in `unsuccess` valgono solo per azioni di tipi `pick_and_release`, e vengono scatenate quando non si rilascia un oggetto sul target designato. Le reazioni reatrions vengono scatenate quando l'azione ha successo. Le reazioni standard possiedono un parametro `object` che indica l'oggetto su cui verificare l'azione (in genere è un oggetto `target` per le reazioni).

```
<reactions object="">
  ...
</reactions>
<unsuccess>
  ...
</unsuccess>
```

object = *id dell'oggetto su cui verificare il successo dell'azione*

Qua di seguito dividiamo le reazioni per tipo. Ciascuna reazione ha un tag personalizzato, ad esempio:

```
<fix ref="" position="" teleport=""/>
```

È una reazione di tipo `fix`.

Show

Mostra l'oggetto `ref`.

```
<show ref="" />
```

ref = *l'id dell'oggetto da mostrare sulla scena (per tutti gli oggetti che possono essere mostrati)*

Hide

Nasconde l'oggetto `ref`.

```
<hide ref="" />
```

ref = *l'id dell'oggetto da nascondere (per tutti gli oggetti che possono essere mostrati)*

Modulate

Modifica il colore dell'oggetto `ref`.

```
<modulate ref="" color="" speed=""/>
```

ref = *l'id dell'oggetto da modulare (per tutti gli oggetti di tipo `Image`, `Mesh` o `Bot`)*

color = *vettore RGB o RGBA che indica il colore che si vuol ottenere nella modulazione dell'oggetto*

speed = *la velocità con cui si vuol modulare l'oggetto*

Play

Attiva un suono.

```
<play ref="" volume="" loop="" bill="" speed=""/>
```

ref = *l'id dell'oggetto da attivare (per tutti gli oggetti di tipo `Sound`)*

volume = un valore da 0 a 1 indicante il volume (default: 1)
replay = un valore che indica se l'audio deve suonare in loop: **true** | **false** (default: **false**)
bill = (opzionale) L'id di un oggetto che deve essere visualizzato mentre viene attivato il suono (oggetto di tipo Billboard)
speed = (opzionale) la velocità di comparsa/scomparsa dell'oggetto bill

Stop

Ferma un suono. È possibile anche indicare replay e volume, ma sono inutili.

```
<stop ref="" volume="" loop="" />
```

ref = L'id dell'oggetto da fermare (per tutti gli oggetti di tipo Sound)
volume = (opzionale) un valore da 0 a 1 indicante il volume
replay = (opzionale) un valore che indica se l'audio deve suonare in loop: **true** | **false** (default: **false**)

State

Modifica il valore di uno stato.

```
<state ref="" value="" change="" />
```

ref = L'id di uno stato (valido solo per gli oggetti di tipo State)
value = il valore da assegnare allo stato
change = parametro alternativo al parametro 'value', funzionante soltanto se 'value' non viene definito. Incrementa o decrementa di una unità il valore dello stato:
increase | **decrease** (default: **increase**)

Scale

Scala progressivamente le dimensioni di un oggetto.

```
<scale ref="" operation="" speed="" limit="" />
```

ref = L'id dell'oggetto da scalare (valido per gli oggetti che vengono visualizzati nella scena come Mesh, Bot e Image)
operation= indica se l'operazione deve iniziare o finire: **start** | **stop** (default: **start**)
speed = la velocità con cui si altera l'oggetto
limit = (opzionale) un vettore indicante il limite da raggiungere in tutte le dimensioni (X, Y e Z). L'animazione termina quando il limite è raggiunto in tutte le direzioni

Fix

Posiziona un oggetto sulla scena in una posizione.

```
<fix ref="" position="" teleport="" mode="" />
```

ref = L'id dell'oggetto da posizionare (valido per gli oggetti che vengono visualizzati nella scena come Mesh, Bot, Billboard e Image)
position= un vettore di 3 elementi indicante la posizione da raggiungere per l'oggetto
teleport= indica se l'oggetto deve spostarsi immediatamente nella posizione o tramite animazione: **true** | **false** (default: **false**)
mode = indica come spostare l'oggetto in base a relazioni con altri oggetti: **free** | **from** | **to** (default: **free**). Se è impostato a 'free', l'oggetto viene spostato nella posizione **position**. Se è impostato a 'from', l'oggetto viene spostato di una distanza pari a **position** (posizione iniziale + **position**). Se è impostato a 'to', l'oggetto **ref** viene spostato alla posizione data dalla posizione dell'object della reazione spostato di **position** (posizione dell'oggetto **target** + **position**).

Lock

Posiziona un oggetto sulla scena in una posizione e lo blocca non rendendolo più spostabile.

```
<lock ref="" position="" teleport="" mode=""/>
```

ref = l'id dell'oggetto da posizionare e bloccare (valido per gli oggetti che vengono visualizzati nella scena come Mesh, Bot, Billboard e Image)

position= un vettore di 3 elementi indicante la posizione da raggiungere per l'oggetto

teleport= indica se l'oggetto deve spostarsi immediatamente nella posizione o tramite animazione: **true** | **false** (default: **false**)

mode = indica come spostare l'oggetto in base a relazioni con altri oggetti: **free** | **from** | **to** (default: **free**). Se è impostato a 'free', l'oggetto viene spostato nella posizione **position**. Se è impostato a 'from', l'oggetto viene spostato di una distanza pari a **position** (posizione iniziale + **position**). Se è impostato a 'to', l'oggetto **ref** viene spostato alla posizione data dalla posizione dell'object della reazione spostato di **position** (posizione dell'oggetto **target** + **position**).

Unlock

Libera un oggetto e lo rende spostabile.

```
<unlock ref="" position="" teleport="" mode=""/>
```

ref = l'id dell'oggetto da sbloccare (valido per gli oggetti che vengono visualizzati nella scena come Mesh, Bot, Billboard e Image)

Move camera

Sposta la camera in una posizione.

```
<move_camera ref="" position="" target=""/>
```

ref = l'id di un oggetto

position= un vettore di 3 elementi indicante la posizione da raggiungere

target = un vettore di 3 elementi indicante il punto verso cui la camera è puntata

Run particle

Avvia l'animazione delle particelle.

```
<run_particle ref="" operation="" lifetime="" deviation="" direction="" dissolve="" enlarge=""/>
```

ref = l'id dell'oggetto da avviare (solo di tipo Particle)

operation= indica se l'azione viene avviata o fermata: **start** | **stop** (default: **start**)

lifetime= indica il tempo di vita di una particella

deviation= un valore indicante l'eventuale range di deviazione di ogni particella in un singolo spostamento sull'asse delle X (default: 0). L'uso di una 'deviazione' rende lo spostamento della particella più casuale

direction= un vettore di 3 elementi indicante la direzione e la velocità verso ogni direzione che una particella ha per ogni singolo spostamento (esempio: [0,2,-1] sposta la particella di 0 su X, 2 su Y e -1 su Z per ogni step).

lifetime= indica il tempo di vita di una particella

Link

Collega un oggetto a un altro oggetto. Un oggetto X collegato a un altro oggetto Y verrà spostato automaticamente quando verrà spostato l'oggetto Y.

```
<link ref="" to="" />
```

ref = l'id dell'oggetto da collegare (valido per gli oggetti che vengono visualizzati nella scena come Mesh, Bot, Billboard e Image)
to = l'id dell'oggetto a cui deve essere collegato l'oggetto ref (valido per gli oggetti che vengono visualizzati nella scena come Mesh, Bot, Billboard e Image)

Unlink

Scollega due oggetti collegati.

```
<unlink ref="" to="" />
```

ref = l'id dell'oggetto da scollegare (valido per gli oggetti che vengono visualizzati nella scena come Mesh, Bot, Billboard e Image)
to = l'id dell'oggetto da cui scollegare l'oggetto ref (valido per gli oggetti che vengono visualizzati nella scena come Mesh, Bot, Billboard e Image)

Run

Avvia o ferma l'animazione propria dell'oggetto.

```
<run ref="" operation="" speed="" first="" final="" />
```

ref = l'id dell'oggetto da animare (solo di tipo Mesh o Bot)
operation= indica se l'animazione viene avviata o fermata: **start** | **stop** (default: start)
speed = indica la velocità dell'animazione per ogni frame
first = (opzionale) indica l'eventuale primo frame dell'animazione
final = (opzionale) indica l'eventuale ultimo frame dell'animazione. Quando viene raggiunto questo frame, l'animazione si ferma automaticamente

Animate

Avvia o ferma l'animazione di una texture di un oggetto.

```
<animate ref="" operation="" params="" />
```

ref = l'id dell'oggetto da animare (solo di tipo Mesh o Bot)
operation= indica se l'animazione viene avviata o fermata: **start** | **stop** (default: start)
params = parametro composto da vari parametri che indicano la modalità con cui si animano le texture dell'oggetto, dove ogni animazione è divisa da un ';' analogamente a: 'texture_name_1:[vect_1];...;texture_name_n:[vect_n]'. I valori texture_name_i indicano il nome della texture da animare, mentre i vettori vect_i indicano vettori di 4 elementi che corrispondono a [VR_U_OFFSET, VR_V_OFFSET, VR_U_TILING, VR_V_TILING] secondo le specifiche del metodo ChangeTextureParams di XVR.

Change description

Cambia la descrizione di un oggetto.

```
<change_description ref="" lang="">message</change_description>
```

ref = l'id dell'oggetto di cui cambiare la descrizione
lang = il nome (o il nome alternativo) della lingua per cui cambiare la descrizione
message = la nuova descrizione (può essere anche una stringa vuota)

Show description

Mostra la descrizione dell'oggetto, corrispondente alla lingua attiva come tooltip sul cursore del mouse.

```
<show_description />
```

(Non necessita di parametri, agisce direttamente sull'oggetto "soggetto" dell'azione).

Show text

Mostra un testo qualsiasi come tooltip sul cursore del mouse in base alla lingua specificata.

```
<show_text lang="">message</show_text>
```

lang = il nome (o il nome alternativo) della lingua per cui visualizzare il testo
message = il testo da visualizzare (può essere anche una stringa vuota, ma la reazione sarebbe inutile)

Show caption

Mostra la caption dell'oggetto.

```
<show_caption ref="" position="" dinstance="" caption="" lang="" />
```

ref = l'id dell'oggetto di cui mostrare la caption
position= vettore di 3 elementi che indica la posizione che deve raggiungere la caption
dinstance= (opzionale) vettore di 3 elementi che indica la distanza dall'oggetto ref che deve essere raggiunta dalla caption (da usare in alternativa a position)
caption = l'id dell'oggetto caption da mostrare (di tipo billboard)
lang = il nome (o il nome alternativo) della lingua associato alla caption

If

Una reazione che pone alternative tra azioni in base a determinate condizioni. Utile ottimizzazione e semplificazione del documento).

```
<if>  
  <conditions>  
    ...  
  </conditions>  
  <then>  
    ...  
  </then>  
  <else>  
    ...  
  </else>  
</if>
```

La costruzione è analoga a quella dei linguaggi di programmazione: nel il sottoalbero **conditions** possono essere indicate una o più condizioni verificabili in AND (vedi il capitolo dedicato alle Condizioni) e nei sottoalberi **then** e **else** possono essere indicate rispettivamente una o più reactions, tra cui la stessa reazione 'If' (vedi il capitolo dedicato alle Reazioni). Le reazioni nel ramo **then** vengono attivate nel caso in cui le condizioni siano verificate, viceversa vengono attivate quelle nel ramo **else**.