ScARtch

Ambiente didattico in realtà aumentata per l’avviamento alla programmazione

# Descrizione del sistema

ScARtch è costituito da un **linguaggio di programmazione visuale** e da un **ambiente di sviluppo** VR. Realizzando semplici **script** visuali diviene possibile controllare gli **attori** presenti in un **playground**.

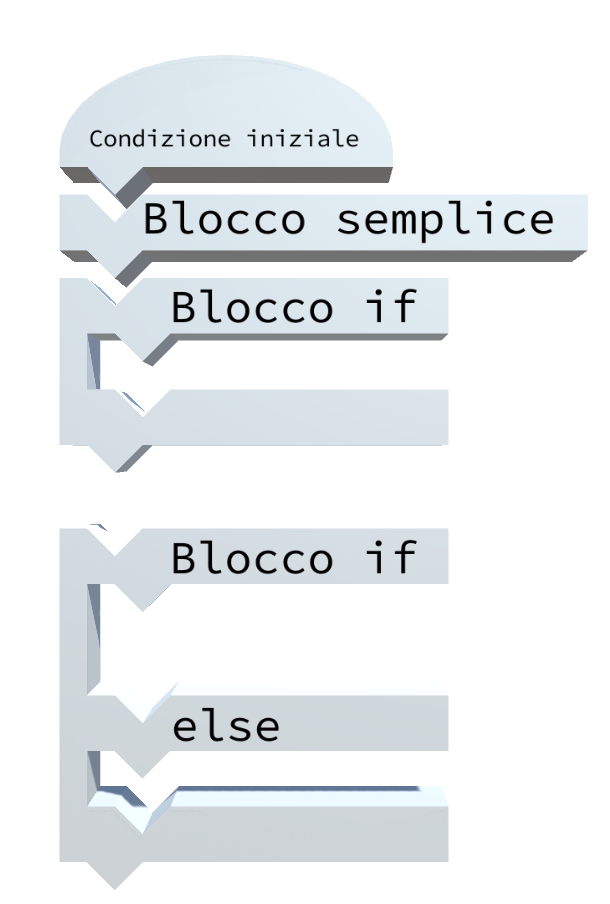
Il **playground** è costituito dai seguenti elementi:

* Una **scena** (che assumiamo statica per le finalità del progetto).
* **Attori**: entità tridimensionali che possono muoversi nello sfondo.
* Un **archivio di modelli** (un elenco dato a priori di modelli tridimensionali da usare per gli attori).
* Un **archivio di suoni** (un elenco di file audio che possiamo usare nella scena).
* Controlli per **avviare** la scena e **arrestarla**. L’avvio della scena porta il sistema in **play mode**, l’arresto lo riporta in **edit mode**.
* Un **timer**, che viene avviato all’ingresso in play mode.

Ad ogni **attore** sono associati

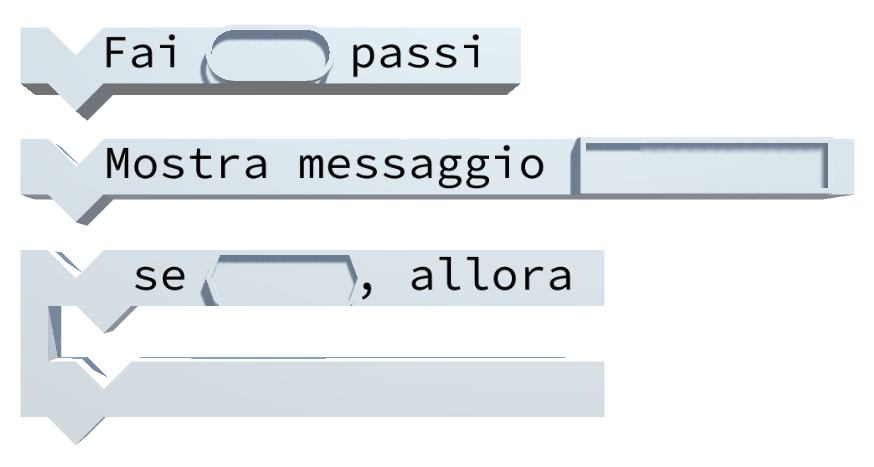
* **Posizione** nell’ambito della scena (limiti fissi), **Rotazione** (espressa in angoli attorno ai 3 assi cartesiani), **Dimensione** (un solo valore), **Volume** di emissione dei suoni.
* **Script**: una serie di programmi che possono essere composti usando l’apposita interfaccia a **blocchi**.
* **Modello** tridimensionale che lo raffigura.
* Un **messaggio**, che può essere visualizzato in una nuvoletta (es. per finalità di debug).

Gli **script** che sono associati agli attori consentono loro di spostarsi sulla scena e compiere diverse tipologie di azioni. In play mode non è possibile alterare gli script: occore che il sistema sia in edit mode. Gli script sono interpretati ed eseguiti nel momento in cui viene soddisfatta la loro **condizione di inizio** (specificata da un particolare blocco) oppure quando si sceglie di testare lo script (rimanendo in edit mode). Uno script è costituito da 1 o più **blocchi** di programmazione.

I **blocchi** di programmazione possono essere organizzati in sequenza per dar luogo ad uno script. I blocchi sono presenti nelle 4 forme mostrate a lato:

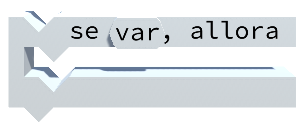
* **Cappelli**: blocchi che contengono la condizione iniziale di uno script. Quando questa si verifica in *play mode*, lo script viene eseguito.
* **Blocchi semplici**: blocchi che contengono una singola istruzione del linguaggio.
* **Blocchi di controllo**: blocchi che hanno una “bocca” in cui è possibile inserire una sequenza di blocchi e che sono usati per il controllo del programma (espressioni *if*, *while*, ecc…).
* **Blocchi di controllo doppi**: blocchi che contengono due “bocche” in cui possono essere alloggiate sequenze di blocchi e che si usano per il controllo del programma per esprimere condizioni *if/else*.

I blocchi vengono combinati dall’utente congiungendoli in corrispondenza dei loro “denti” presenti sul lato sinistro. Vi sono alcuni blocchi specifici che concludono l’esecuzione di uno script (es. blocco di arresto del *play mode*) e che, quindi, non presentano un dente per la prosecuzione della sequenza.

All’interno dei blocchi possono essere presenti **caselle** da riempire con **operandi** opportuni, che possono essere **immediati**, **variabili** oppure risultati di **espressioni**. I tipi di operandi supportati dal linguaggio sono:

* **Numeri** reali, che possono essere inseriti nelle caselle con i bordi arrotondati (primo blocco nella figura a fianco).
* **Stringhe** costituite di caratteri di qualsiasi tipo, che possono essere inserite in caselle con i bordi rettangolari (secondo blocco mostrato in figura).
* **Booleani**, che possono essere inseriti in caselle con i bordi spigolosi (terzo blocco in figura).

In alcuni blocchi particolari sono presenti **caselle** **con menù a tendina** per la selezione del valore all’interno di un elenco prestabilito.

L’inserimento di valori immediati avviene attraverso elementi di interfaccia VR (rispettivamente, una tastiera fluttuante per le stringhe, un tastierino numerico per i numeri, un tastierino con i due valori ammessi per booleani). L’inserimento di variabili o espressioni, invece, avviene trascinando nella casella il corrispondente **riferimento** (un blocco che non può essere inserito nel normale flusso di programmazione). La forma del riferimento consente di individuare a colpo d’occhio il tipo della variabile o del risultato dell’espressione secondo le corrispondenze elencate per le caselle (sopra viene mostrato un riferimento di una variabile booleana). È possibile inserire in caselle di forma rettangolare anche riferimenti con bordi arrotondati o spigolosi: il loro contenuto verrà automaticamente convertito a stringa.

Le variabili sono definite scegliendo un *nome* *univoco*, un *tipo* e uno *scope* (**locale** per l’Attore o **globale** per tutto il sistema).

# Blocchi proposti

I blocchi sono classificati per comodità dell’utente in categorie (a partire dalle quali è possibile scorrerne la lista con maggior agio). Nell’elenco seguente, si indicano con le *parentesi tonde* caselle per numeri, con le *parentesi quadre* caselle per stringhe, con le *parentesi angolari* caselle per booleani e con le *parentesi graffe* caselle a scelta multipla (con menù a tendina).

|  |  |  |  |
| --- | --- | --- | --- |
| **Blocco** | **Categoria** | **Forma** | **Descrizione** |
| Fai ( ) passi | Movimento | Sempl. | Sposta l’attore in avanti o indietro (rispetto alla direzione corrente). |
| Ruota di ( ) gradi attorno all’asse {x|y|z} | Movimento | Sempl. | Ruota l’attore in senso orario o antiorario rispetto all’asse specificato. |
| ~~Rivolgiti verso {altro Attore|testa utente|mano dx utente| mano sx utente}~~ | ~~Movimento~~ | ~~Sempl.~~ | ~~Ruota l’attore in modo da rivolgerlo verso l’attore specificato o verso la posizione dell’utente in VR.~~ |
| Vai alla posizione x: ( ), y: ( ), z: ( ). | Movimento | Sempl. | Porta l’attore nella posizione indicata. |
| ~~Vai alla posizione di {altro Attore|testa utente|mano dx utente| mano sx utente}~~ | ~~Movimento~~ | ~~Sempl.~~ | ~~Porta l’attore nella posizione di un altro attore o dell’utente.~~ |
| Cambia {x|y|z} di ( ) | Movimento | Sempl. | Modifica una delle coordinate della posizione dell’attore di un valore specificato. |
| Assegna ( ) a {x|y|z} | Movimento | Sempl. | Modifica una delle coordinate della posizione dell’attore con il valore specificato. |
| Rimbalza quando tocchi il bordo. | Movimento | Sempl. | Quando l’attore raggiunge il bordo della scena, ne inverte la rotazione intorno all’asse y. |
| Posizione x | Movimento | Riferimento a numero | Usato per accedere al valore della posizione x. |
| Posizione y | Movimento | Riferimento a numero | Usato per accedere al valore della posizione y. |
| Posizione z | Movimento | Riferimento a numero | Usato per accedere al valore della posizione z. |
| Cambia modello con {lista di modelli} | Aspetto | Sempl. | Cambia il modello dell’attore con quello specificato. |
| Di’ [ ] | Aspetto | Sempl. | Fa comparire una nuvoletta sull’attore contenente il messaggio indicato. |
| Di’ [ ] per ( ) secondi | Aspetto | Sempl. | Fa comparire una nuvoletta sull’attore contenente il messaggio indicato, che viene nascosta dopo il tempo indicato. |
| Rimuovi fumetto | Aspetto | Sempl. | Rimuove fumetti mostrati da istruzioni precedenti. |
| Imposta dimensioni a ( ) | Aspetto | Sempl. | Ridimensiona l’attore come specificato. |
| Imposta dimensioni al ( )% | Aspetto | Sempl. | Ridimensiona l’attore in relazione alla sua dimensione corrente come specificato. |
| Dimensione | Aspetto | Riferimento a numero | Usato per accedere al valore della dimensione |
| ~~Mostra~~ | ~~Aspetto~~ | ~~Sempl.~~ | ~~Rende l’attore visibile~~ |
| ~~Nascondi~~ | ~~Aspetto~~ | ~~Sempl.~~ | ~~Rende l’attore invisibile~~ |
| Riproduci suono {lista suoni} | Suono | Sempl. | Riproduce uno dei suoni dell’archivio. |
| ~~Riproduci suono {lista suoni} fino alla fine~~ | ~~Suono~~ | ~~Sempl.~~ | ~~Riproduce uno dei suoni dell’archivio ed attende il suo completamento prima di proseguire nel programma.~~ |
| Arresta tutti i suoni | Suono | Sempl. | Interrompe i suoni |
| ~~Cambia volume di ( )~~ | ~~Suono~~ | ~~Sempl.~~ | ~~Altera il volume dell’attore della quantità specificata.~~ |
| Imposta volume al ( )% | Suono | Sempl. | Imposta il volume dell’attore alla quantità specificata. |
| Volume | Suono | Riferimento a numero | Usato per accedere al valore del volume dell’attore |
| All’entrata in play mode | Controllo | Cappello | Dà il via ad uno script appena si entra in play mode |
| Alla pressione di {tasti del controller} | Sensori | Cappello | Dà il via ad uno script quando viene toccato uno specifico tasto del controller. |
| ~~Al tocco dell’attore~~ | ~~Controllo~~ | ~~Cappello~~ | ~~Dà il via ad uno script quando l’utente tocca l’attore.~~ |
| Aspetta ( ) secondi | Controllo | Sempl. | Mette in pausa l’esecuzione dello script per il numero di secondi specificati. |
| Per sempre | Controllo | Controllo | Riproduce la porzione di script inserita per sempre. |
| Ripeti ( ) volte | Controllo | Controllo | Riproduce la porzione di script inserita il numero di volte indicato. |
| Trasmetti [] | Controllo | Sempl. | Trasmette un messaggio in broadcast a tutti gli attori. |
| ~~Trasmetti [] e attendi~~ | ~~Controllo~~ | ~~Sempl.~~ | ~~Trasmette un messaggio in broadcast a tutti gli attori e attende il completamento di script che siano innescati di conseguenza.~~ |
| Quando ricevi [] | Controllo | Cappello | Dà il via ad uno script quando viene trasmesso un messaggio. |
| Se < > | Controllo | Controllo | Esegue la sequenza di blocchi inclusi se è soddisfatta la condizione. |
| Se < > … altrimenti | Controllo | Controllo doppio | Esegue la prima sequenza di blocchi inclusi se è soddisfatta la condizione, altrimenti esegue la seconda sequenza. |
| Aspetta finché < > | Controllo | Sempl. | Mette in pausa lo script fino al soddisfacimento della condizione. |
| Finché < > | Controllo | Controllo | Ripete la sequenza di blocchi inclusi fino a che rimane verificata la condizione. |
| ~~Ferma lo script~~ | ~~Controllo~~ | ~~Sempl.~~ | ~~Arresta lo script.~~ |
| Termina play mode | Controllo | Sempl. | Arresta tutti gli script e termina il play mode. |
| ~~Tocca {altro attore o controller utente}~~ | ~~Sensori~~ | ~~Riferimento a booleano~~ | ~~Verificato se l’attore tocca l’attore indicato o la parte del controller indicata.~~ |
| Posizione {x, y, z} del controller {sinistro, destro} | Sensori | Riferimento a numero | Usato per accedere alle posizioni dei controller. |
| Rotazione {x, y, z} del controller {sinistro, destro} | Sensori | Riferimento a numero | Usato per accedere alle rotazioni dei controller. |
| Posizione {x, y, z} del visore | Sensori | Riferimento a numero | Usato per accedere alle posizioni del visore. |
| Rotazione {x, y, z} del visore | Sensori | Riferimento a numero | Usato per accedere alle rotazioni del visore. |
| Il bottone {lista di bottoni} del controller {sinistro, destro} è schiacciato | ~~Sensori~~ | ~~Riferimento a booleano~~ | ~~Usato per accedere ai bottoni dei controller~~ |
| ~~Distanza da {altro attore o controller}~~ | ~~Sensori~~ | ~~Riferimento a numero~~ | ~~Usato per accedere alla distanza dell’attore corrente dall’elemento indicato.~~ |
| Resetta il timer | Sensori | Sempl. | Imposta il timer del playground a 0. |
| Timer | Sensori | Riferimento a numero | Usato per ottenere il valore del timer in secondi. |
| ~~{posizione x, y, z, rotazione x, y, z, dimensione, volume} di {lista attori}~~ | ~~Sensori~~ | ~~Riferimento a numero~~ | ~~Usato per ottenere il valore di una proprietà di altri attori.~~ |
| ( ) + ( ) | Operatori | Riferimento a numero | Somma due numeri |
| ( ) - ( ) | Operatori | Riferimento a numero | Effettua una sottrazione tra due numeri |
| ( ) × ( ) | Operatori | Riferimento a numero | Moltiplica due numeri |
| ( ) ÷ ( ) | Operatori | Riferimento a numero | Effettua una divisione tra due numeri |
| ( ) > ( ) | Operatori | Riferimento a booleano | Esegue la disequazione mostrata |
| ( ) < ( ) | Operatori | Riferimento a booleano | Esegue la disequazione mostrata |
| ( ) ≥ ( ) | Operatori | Riferimento a booleano | Esegue la disequazione mostrata |
| ( ) ≤ ( ) | Operatori | Riferimento a booleano | Esegue la disequazione mostrata |
| ( ) = ( ) | Operatori | Riferimento a booleano | Confronta due numeri |
| Numero casuale tra ( ) e ( ) | Operatori | Riferimento a numero | Estrae un numero casuale compreso tra gli estremi specificati |
| < > e < > | Operatori | Riferimento a booleano | Esegue l’operazione booleana AND |
| < > o < > | Operatori | Riferimento a booleano | Esegue l’operazione booleana OR |
| Non < > | Operatori | Riferimento a booleano | Esegue l’operazione booleana NOT |
| [ ] è uguale a [ ] | Operatori | Riferimento a booleano | Confronta due stringhe |
| Unione di [ ] e [ ] | Operatori | Riferimento a stringa | Concatena due stringhe |
| Lunghezza di [ ] | Operatori | Riferimento a numero | Restituisce la lunghezza di una stringa |
| Lettera ( ) di [ ] | Operatori | Riferimento a stringa | Restituisce l’elemento di posizione specificata nella stringa |
| {abs, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e^, 10^} ( ) | Operatori | Riferimento a numero | Esegue una delle funzioni indicate sull’operatore numerico. |
| ( ) mod ( ) | Operatori | Riferimento a numero | Restituisce il resto della divisione. |
| Arrotonda ( ) | Operatori | Riferimento a numero | Arrotonda un numero reale all’intero più vicino. |
| Nome\_variabile | Variabili | Riferimento a numero, stringa o booleano | Fornisce accesso al valore della variabile |
| ~~Mostra la variabile [ ]~~ | ~~Variabili~~ | ~~Sempl.~~ | ~~Mostra nel playground il valore della variabile.~~ |
| ~~Nascondi la variabile [ ]~~ | ~~Variabili~~ | ~~Sempl.~~ | ~~Nasconde nel playground il valore della variabile.~~ |
| Imposta <> a < > | Variabili | Sempl. | Assegna il valore indicato alla variabile. |
| Imposta () a ( ) | Variabili | Sempl. | Assegna il valore indicato alla variabile. |
| Imposta []a [ ] | Variabili | Sempl. | Assegna il valore indicato alla variabile. |
| Incrementa () di ( ) | Variabili | Sempl. | Incrementa il valore della variabile della quantità indicata. |