

# Blender Game Engine

Università degli Studi di Perugia

a cura di

Arianna Masciolini e Giorgio Mazza

## ❖ Introduzione

Casi d'uso e caratteristiche generali per il Blender Game Engine

## ❖ Strumenti

Editor grafico e scripting con Python

## ❖ Componenti

Sensori, controller e attuatori

# Introduzione

- ❖ Casi d'uso: sviluppo di giochi e simulazioni interattive in 3D
- ❖ C++
- ❖ Rendering in tempo reale
- ❖ Interfaccia: editor logico con interfaccia grafica
- ❖ Supportato scripting in Python
- ❖ Possibilità di esportare l'applicazione per ambienti Linux, macOS, e MS-Windows

- ✦ Audaspace

Controllo dell'audio

- ✦ Bullet

Rilevamento delle collisioni e simulazione di fisica

- ✦ Recast

Costruzione di percorsi

- ✦ Detour

Pathfinding e ragionamento spaziale

Il BGE non sarà più supportato a partire dalla versione 2.8 di Blender ...

1. Creazione degli elementi visuali (modelli 3d o immagini)
2. Uso dei blocchi logici per permettere l'interazione con la scena (programmazione a eventi)
3. Creazione di una o più "macchine da presa" per il rendering e modifica dei parametri di visualizzazione
4. Esportazione dell'applicazione

# Strumenti



- ❖ L'editor Logico è il metodo principale di impostare e modificare la logica del gioco per i vari attori che lo compongono
- ❖ La logica dei singoli oggetti selezionati nella 3D View è rappresentata dai Blocchi Logici (**Logic Bricks**)

img

1. **Logic Bricks:** funzioni programmate; Modificabili e combinabili, sono:

- ❖ Suddivisi in tre tipi (colonne):
  - ▶ **Sensors:** si attivano al verificarsi di un evento,
  - ▶ **Controllers:** operano sull'input che ricevono dai sensori
  - ▶ **Actuators:** interagiscono con il gioco;

Quando un Sensore è positivo invia un impulso al Controllore che valuta la condizione e se è il caso innesca l'Attuatore.

- ❖ Collegati tra loro mediante **link**, che conducono il flusso logico tra sensor->controller e controller->actuator

img

2. **Properties:** variabili dei linguaggi di programmazione; Usate per salvare ed accedere dati associati ad un oggetto o del gioco stesso
3. **States:** stati in cui può trovarsi un oggetto; usati per definire comportamenti diversi in situazioni diverse

- ❖ Blocchi logici che si mettono in ascolto per un determinato evento (*event listeners*) e cambiano stato quando esso si verifica.
- ❖ Di default, ad ogni cambio di stato, i sensori innestano i Controller ad essi collegati;
  - ❖ *"True level triggering"* ([img]) li innesta solo se lo stato del sensore è positivo
  - ❖ *"False level triggering"* ([img]) li innesta solo se lo stato del sensore è negativo
  - ❖ *"Freq"* imposta il ritardo tra i trigger ripetuti, in frame (tick logici)

img

Di seguito, i sensori principali, aggiungibili con il pulsante "Add Sensor":

**Nota:** *Un sensore invia un impulso positivo (TRUE) a tutti i suoi Controllers quando l'evento si verifica; quando l'evento finisce invia un impulso negativo (FALSE)*

- ❖ **Actuator:** rileva quando un particolare Actuator riceve un impulso di attivazione o di disattivazione
- ❖ **Always:** innesta sempre i relativi Controllers (ad ogni tick logico)
- ❖ **Collision:** rileva quando l'oggetto proprietario collide (tocca) un'altro oggetto
- ❖ **Delay:** ritarda l'invio dell'impulso positivo

Di seguito, i sensori principali, aggiungibili con il pulsante "Add Sensor":

***Nota:** Un sensore invia un impulso positivo (TRUE) a tutti i suoi Controllers quando l'evento si verifica; quando l'evento finisce invia un impulso negativo (FALSE)*

- ❖ **Keyboard:** rileva la pressione di un tasto da noi assegnato
- ❖ **Mouse:** rileva gli eventi del mouse. Eventi: *Over any, Over, Movement, Wheel down/up, R button, L button, M button.*
- ❖ **Near:** si attiva quando rileva degli oggetti ad una distanza  $x$  dall'oggetto proprietario.
- ❖ **Properties:** rileva cambiamenti nelle Proprietà (variabili) del suo oggetto.

- ❖ Blocchi logici che eseguono operazioni logiche sull'output del Sensore ed attivano gli Actuators ad esso connessi

- ✚ Blocchi logici che interagiscono direttamente con il gioco



# Scripting con Python





# Componenti





