

# 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; Sono modificabili, combinabili e:

- ❖ 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;

Se un Sensore è positivo invia un impulso al Controllore che valuta la condizione ed innesca l'Attuatore.

- ❖ sono 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, i sensori **innestano i Controller** ad essi collegati ad ogni cambio di stato;
  - ❖ *"True level triggering"* ([img])  
innesta i controllers solo se lo stato del sensore è positivo
  - ❖ *"False level triggering"* ([img])  
innesta i controllers solo se lo stato del sensore è negativo
  - ❖ *"Freq"*  
imposta il ritardo tra i trigger ripetuti, in frame (tick logici)

img

I principali sensori, aggiungibili con il pulsante "Add Sensor":

***Nota:** Un sensore invia un impulso positivo (TRUE) a tutti i suoi Controllers quando si attiva; quando si disattiva 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

I principali sensori, aggiungibili con il pulsante "Add Sensor":

***Nota:** Un sensore invia un impulso positivo (TRUE) a tutti i suoi Controllers quando si attiva; quando si disattiva 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/L/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** sull'output del Sensore ed, in base a determinate condizioni, **attivano gli Actuators** ad esso connessi.
- 1. **States**: mostra la griglia degli stati, per spostare il Controller (e tutti i suoi blocchi collegati) in un altro State dell'oggetto.
- 2. Esegue il Controller prima degli altri

[img]

I principali controllori, aggiungibili con il pulsante "Add Controller":

***Nota:** un impulso positivo (TRUE) inviato dal Controller, attiva i suoi Actuators; uno negativo (FALSE) li disattiva.*

- ❖ **And:** TRUE se riceve **tutti** impulsi positivi, FALSE altrimenti;
- ❖ **Or:** TRUE se riceve **almeno** un impulso positivo, FALSE altrimenti;
- ❖ **Nand:** TRUE se riceve **almeno** un impulso negativo, FALSE se riceve **tutti** impulsi positivi;
- ❖ **Nor:** TRUE se non riceve **nessun** impulso positivo, FALSE altrimenti;
- ❖ **Xor:** TRUE se riceve **un solo** impulso positivo, FALSE altrimenti;
- ❖ **Xnor:** TRUE se riceve **un solo** impulso negativo, FALSE altrimenti;
- ❖ **Expression:** valuta un'espressione booleana scritta dall'utente;
- ❖ **Python:** scripting in python, v. slides successive.



- ✚ Blocchi logici che interagiscono direttamente con il gioco





# Componenti





