

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

"(...) I also propose to refocus the current game engine (...) I propose to make the GE to become a real part of Blender code – to make it not separated anymore. This would make it more supported, more stable and (I'm sure) much more fun to work on as well. (...) What should then be dropped is the idea to make Blender have an embedded "true" game engine. (...) We never managed to make something with the portability and quality of Unreal or Crysis... or even Unity3D. And Blender's GPL license is not helping here much either. (...) The main cool feature of our GE is that it was integrated with a 3D tool, to allow people to make 3D interaction for walkthroughs, for scientific sims, or game prototypes. If we bring back this (original) design focus for a GE, I think we still get something unique and cool, with seamless integration of realtime and "offline" 3D".

-Ton Roosendaal, creator of 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

Componenti

