

* Unity 3D

- Unity 3D → Unity è un framework x lo sviluppo di videogiochi.

Primo rilascio: 2005.

- Ci sono due versioni → Free
→ Professional.

- Si utilizza unity3D con C# in Visual Studio.

- Un progetto Unity si basa sui pannelli: Un pannello è un componente fondamentale dell'interfaccia grafica. Essi possono essere posizionati a piacimento.

- L'interfaccia base di Unity è composta da 6 pannelli

- 1) Hierarchy → dove vengono elencati tutti gli oggetti presenti nella scena
- 2) Game → dove viene mostrato il nostro gioco effettivo.
- 3) Scene → più importante → viene mostrata la scena + oggetti
- 4) Inspector → contiene dettagli dell'oggetto selezionato.
- 5) Console → x msg errori. Viene usato in fase di debug.
- 6) Project → dove si trovano i nostri assets.

- Gli oggetti sono gli elementi di un videogioco. La scena è il contenitore di tali oggetti. Un progetto può essere composto da una o più scene. Di default nella scena "vagina" sono presenti due oggetti di default: → Camera
→ Directional Light

- Nel pannello Hierarchy è possibile stabilire relazioni di parentela

tra due o più oggetti.

- Gli Asset sono tutti gli elementi presenti in un gioco: scene, texture, eccetera.

- Unity 3D ha tre strumenti per trasformare gli oggetti in una scena

1) Spostamento

2) Rotazione

3) Scalatura

- Unity è un game engine. Tutti i poligoni in Unity vengono ridotti a triangoli.

Esi:



- I triangoli sono i poligoni basilari (e semplici). Dal punto di vista computazionale, sono i più semplici da gestire.

- Una scheda video riesce a lavorare solo con triangoli.

- Le normali:



La normale → servono per capire orientamento del poligono nello spazio.

- Le luci sono molto importanti. Di default si usa la Directional Light.

- Directional light → illumina l'intera area del gioco.

- Point light → x oggetti che emettono luce attorno a loro.

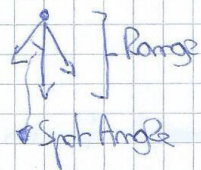
- Spot light → x fascio di luce di forma conica.

- Area light → tipo luci dei cartelloni pubblicitari.

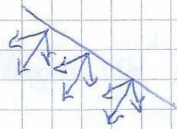
- La Directional light viene usata per simulare la luce solare che cade. Illumina uniformemente l'intera scena.

- Point light → es: piuma della candela.

• Spot light →



• Area light →



Dispendiosa in termini di risorse.

• Gestione delle informazioni sulle luci in tempo reale è dispendiosa in termini di risorse hardware. Con il BAKUG è possibile precalcolare le informazioni sulle luci e salvarle. Per usare il BAKUG bisogna salvare come statici tutti gli oggetti della scena.

• Una texture è una immagine che va a ricoprire un'altra immagine. Es: cicatrizza sul volto di un personaggio.

• I tre canali: R = Red

G = Green

B = Blue

α = quanto canale che gestisce la trasparenza

Non tutti i formati di immagine gestiscono la trasparenza.

Per esempio JPEG non lo supporta. PNG o TIFF lo supportano.

• Per usare una texture bisogna usare una immagine e quindi usare un MATERIALI x quella immagine.

• Uno SHADER è un frammento di codice che si occupa di gestire il comportamento del materiale in funzione della luce.

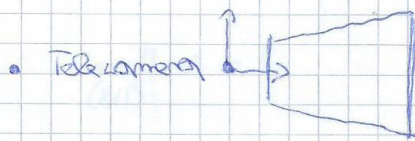
In Unity 5 si usa spesso lo STANDARD SHADER.

• Con lo STANDARD SHADER si potrà dare il materiale di trasparenza, opaco, ...

• In Unity una videocamera renderizza solo quello che è intorno ad un FRUSTUM



Frustum di visione.

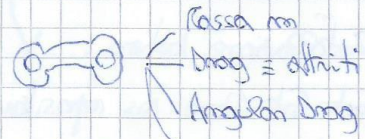


- **ALIASING** → scalatura nelle linee oblique
con un filtro anti-aliasing si risolve il problema.

- Gli oggetti in una scena devono interagire fra loro. Ci sono dei componenti 2D o 3D che ci permettono di fare interagire questi oggetti.

- **Componenti:**

→ **RIGID BODY** → componente base.



→ **COLLIDERS** → per gestire le collisioni fra i vari oggetti.

- In Unity gli script di programmazione si possono sviluppare in:

→ C#

→ JAVASCRIPT

→ Boo (linguaggio di programmazione object-oriented con sintassi Python-like)

- Negli script ci sono varie event functions:

- **START** → avviato all'avvio del gioco o quando entra in scena ^{di}

- **UPDATE** → eseguita ad ogni frame del nostro gioco.

- È nel metodo **START** che si inizializzano tutte le variabili.

* Unity:

- È un framework per lo sviluppo di videogiochi. È primo rilascio nel 2005.
- All'interno del framework ci sono tutti i tool necessari per creare

• un gioco: - gestione dei materiali

- Animation tool

- Gestione texture,...

- Ci sono due versioni di Unity:

- Free

- Professional

- Ogni progetto creato con Unity deve contenere tutte le risorse di cui l'applicazione ha bisogno.

- Una scena è un contenitore di oggetti, e quando viene caricata una scena l'engine si occupa di allocare in memoria tutti gli oggetti e di deallocarli una volta chiusa la scena.

- Tre oggetti fondamentali in Unity:

- Game object → primitiva di base

- Component → script,...

- Asset → oggetti creati in Unity e usati dove occorre.

- La parte di scripting → si possono usare i tre seguenti linguaggi di programmazione:

- JavaScript

- C#

- Boo

La parte di scripting viene definita in MonoBehaviour

- MonoBehaviour → implementazione Open-Source del framework .NET.

- Unity possiede una versione proprietaria di MonoBehaviour.

- Ogni script è assegnato ad un Game Object.

- La classe dello script è derivata da MonoBehaviour.

- I files di progetto (asset) si spostano dentro la finestra Project.

• In Unity ci sono due tipi di viste:

- isometrica
- prospettiva.

• In unity ci sono tre possibili trasformazioni su un oggetto:

- ROTAZIONE
- TRASLAZIONE
- SCALATURA.

• Le trasformazioni possono essere:

- Locali → trasformazione rispetto al sistema di riferimento relativo
- Globali → trasformazione rispetto al sistema di riferimento assoluto

• Pivot → punto immaginario usato per la rotazione della posizione di un oggetto nella scena 3D.

NB: Decidere se fare le rotazioni su pivot o su centro geometrico può ridurre le lag.

• Gli oggetti possono avere una parentela.

• Ogni gameobject ha un nome, delle proprietà di base, o può essere posizionato in 3D.

• In unity, per poter creare interattività bisogna creare componenti personalizzati. Si usano gli script.

GameObject — chiama —> Script

I componenti script risiedono in file esterni (.cs, ...).

• Gli script in unity ereditano dalla classe MonoBehaviour.

• Creare uno script associato ad un oggetto, editando → apre IDE sotto di default, per esempio Visual Studio.

• Le funzioni che servono a gestire gli eventi si chiamano event handler.

• Ad ogni frame, unity scansa tutti gli oggetti in scena e se trova associate funzioni eventi le esegue.